Kronecker Compressed Sensing With Structured Sparsity

Algorithms, Guarantees, and Applications

KRONECKER COMPRESSED SENSING WITH STRUCTURED SPARSITY

ALGORITHMS, GUARANTEES, AND APPLICATIONS

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology
by the authority of the Rector Magnificus, Prof.dr.ir. H. Bijl,
chair of the Board for Doctorates
to be defended publicly on
Thursday 30, April 2026 at 12:30 o'clock

by

Yanbin HE

This dissertation has been approved by the (co)promotors.

Composition of the doctoral committee:

Rector Magnificus,	chairperson
Prof. dr. ir. A.J. van der Veen,	Delft University of Technology, *promotor*
Dr. G. Joseph,	Delft University of Technology, *copromotor*

Independent Members:

Prof. dr. ir. R. Heusdens,	Delft University of Technology/Netherlands Defence Academy
Prof. dr. ir. G.J.T. Leus,	Delft University of Technology
Prof. dr. E. Björnson,	KTH Royal Institute of Technology, Sweden
Prof. dr. C.R. Murthy,	Indian Institute of Science, India
Prof. dr. E. Ollila,	Aalto University, Finland
Prof. dr. A. Yarovoy,	Delft University of Technology, reserve member

Keywords: Compressed sensing, structured sparsity, Kronecker product, sparse Bayesian learning, tensor, convergence, restricted isometry property

Cover by: Gemini AI.

An electronic version of this dissertation is available at
https://repository.tudelft.nl/.

“I have looked into her eyes. I had my proof there. Take yours how you will.”

Contents

Summary

This dissertation focuses on Kronecker compressed sensing, recovering multidimensional sparse signals from their linear projections on Kronecker product measurement matrices. Multidimensional signals are functions of different dimensions, each conveying a specific physical quantity and they arise in applications such as wireless communications and image processing. Kronecker product matrices naturally capture the multidimensional nature, making Kronecker compressed sensing a powerful framework for the recovery. Beyond the standard sparsity, practical signals typically have additional structures. We examine three structured sparsity models: hierarchical, Kronecker-supported, and Kronecker-structured. We start with algorithms and guarantees for the Kronecker-supported and Kronecker-structured patterns, and then proceed to a unified algorithmic and theoretical framework, showing how leveraging structure in measurement matrices and sparsity patterns yields gains in accuracy and efficiency.

First, for *Kronecker-supported* signals, we introduce *Kronecker sparse Bayesian learning*. To promote the Kronecker-supported sparsity, we adopt a Kronecker-structured prior in the sparse Bayesian learning framework. Such a prior leads to a nonconvex optimization problem, which is solved using an alternating-minimization method with convergence guarantee and a singular value decomposition-based variant that attains competitive accuracy with lower computation time. The analysis further shows that encoding the Kronecker support reduces undesirable local minima, contributing to the improved performance. We test our algorithms on the channel estimation problem for intelligent reflecting surface-aided systems, showing better performance and efficiency.

Second, for *Kronecker-structured* signals where the sparse vector itself can be factorized across multiple dimensions, we develop a decomposition-based recovery that breaks the high-dimensional problem into multiple lower-dimensional subproblems. Furthermore, in the context of basis expansion model for unknown parameter estimation, we design an off-grid sparse Bayesian learning to mitigate grid mismatch. Theoretical results guarantee the error bound of the decomposition step, explain the denoising effect which we attribute to improved performance, and provide convergence guarantees for our off-grid method. We test our algorithms on the channel estimation problem for intelligent reflecting surface-aided systems in the off-grid setting, showing not only significant performance gains but also improved accuracy for off-grid angle estimation.

Third, moving beyond structure-specific designs, we further present a *hierarchical multi-stage framework* that reframes the Kronecker compressed sensing problem as a sequence of dimension-wise subproblems. This yields a unified algorithmic framework that can be combined with any sparse recovery algorithm, and is capable of handling standard, hierarchical, and Kronecker-supported sparsity patterns. We demonstrate its practical utility in both wireless communications and video sequence reconstruction, showing competitive performance with reduced computational cost.

Overall, the work offers practical tools and insights for Kronecker compressed sensing by coupling structure-aware algorithms with rigorous guarantees and realistic applications.

Samenvatting

Dit proefschrift richt zich op *Kronecker compressed sensing*, het reconstrueren van meerdimensionale spaarse signalen uit hun lineaire projecties op Kronecker productmeetmatrices. Meerdimensionale signalen zijn functies van verschillende dimensies, die elk een specifieke fysieke grootheid overbrengen en voorkomen in toepassingen zoals draadloze communicatie en beeldverwerking. De Kronecker-productmatrix vat de meerdimensionale aard op natuurlijke wijze, waardoor *Kronecker compressed sensing* een krachtig kader vormt voor de reconstructie. Naast de standaard spaarzaamheid bezitten praktische signalen doorgaans aanvullende structuren. We onderzoeken drie gestructureerde spaarzaamheidspatronen: *hierarchical*, *Kronecker-supported* en *Kronecker-structured*. We beginnen met algoritmen en garanties voor de *Kronecker-supported* en *Kronecker-structured* patronen, en gaan vervolgens over op een verenigd algoritmisch en theoretisch kader, waarbij we laten zien hoe het benutten van structuur in meetmatrices en spaarzaamheidspatronen winst oplevert in nauwkeurigheid en efficiëntie.

Ten eerste introduceren we voor *Kronecker-supported* signalen *Kronecker sparse Bayesian learning*. Om de *Kronecker-supported* spaarzaamheid te bevorderen, passen we een *Kronecker-structured prior* toe binnen het *sparse Bayesian learning*-kader. Een dergelijke *prior* leidt tot een niet-convex optimalisatieprobleem, dat wordt opgelost met behulp van een alternerende minimalisatiemethode met convergentiegarantie en een op singuliere waardenontbinding gebaseerde variant die competitieve nauwkeurigheid behaalt met minder rekentijd. De analyse toont verder aan dat het coderen van de Kronecker-support ongewenste lokale minima vermindert, wat bijdraagt aan de verbeterde prestaties. We testen onze algoritmen op het kanaalschattingsprobleem voor systemen die worden ondersteund door *intelligent reflecting surfaces*, waarbij betere prestaties en efficiëntie worden aangetoond.

Ten tweede ontwikkelen we voor *Kronecker-structured* signalen, waarbij de spaarse vector zelf over meerdere dimensies kan worden gefactoriseerd, een op decompositie gebaseerde reconstructie die het hoogdimensionale probleem opbreekt in meerdere lagerdimensionale deelproblemen. Verder ontwerpen we, in de context van het *basis expansion model* voor onbekende parameterschatting, een *off-grid sparse Bayesian learning*-methode om roostermismatch te verminderen. Theoretische resultaten garanderen de foutgrens van de decompositiestap, verklaren het ruisonderdrukkende effect dat we toeschrijven aan verbeterde prestaties, en bieden convergentiegaranties voor onze *off-grid*-methode. We testen onze algoritmen op het kanaalschattingsprobleem voor *intelligent reflecting surfaces*-ondersteunde systemen in de *off-grid*-setting, wat niet alleen aanzienlijke prestatiewinst laat zien, maar ook verbeterde nauwkeurigheid voor *off-grid* hoekschatting.

Ten derde presenteren we, verdergaand dan structuurspecifieke ontwerpen, een *hiërarchisch meertrapskader* dat het *Kronecker compressed sensing*-probleem herformuleert als

een reeks dimensiegewijze deelproblemen. Dit levert een verenigd algoritmisch raamwerk op dat kan worden gecombineerd met elk algoritme voor spaarzaam herstel, en in staat is om *standard, hierarchical* en *Kronecker-supported* spaarzaamheidspatronen te verwerken. We demonstreren het praktische nut ervan in zowel draadloze communicatie als de reconstructie van videosequenties, waarbij we competitieve prestaties laten zien met lagere rekenkosten.

Samenvattend biedt dit werk praktische instrumenten en inzichten voor *Kronecker compressed sensing* door structuurbewuste algoritmen te koppelen aan rigoureuze garanties en realistische toepassingen.

Nomenclature

symbol	description
a	Scalar
$\boldsymbol{a}$	Vector
$\boldsymbol{A}$	Matrix
A	Tensor
$[\boldsymbol{a}]_i$	The i-th entry of vector $\boldsymbol{a}$
$[\boldsymbol{A}]_i$	The i-th column of matrix $\boldsymbol{A}$
$[\boldsymbol{A}]_{ij}$	The entry on the i-th row and j-th column of matrix $\boldsymbol{A}$
$(\cdot)^{\mathsf{T}}$	Transpose
$(\cdot)^*$	Conjugate
$(\cdot)^{\mathsf{H}}$	Conjugate transpose
$\lvert\cdot\rvert$	Determinant
$(\cdot)^\dagger$	Pseudo-inverse
$\boldsymbol{a}^{-1}$	Element-wise inverse
$\boldsymbol{A}^{-1}$	Matrix inverse
$\mathbf{1}_N$	All-one vector
$\boldsymbol{I}_N$	Identity matrix of size $N \times N$
$\boldsymbol{a} > 0$	(Element-wise comparison) All-positive vector
$\boldsymbol{a} \geq 0$	(Element-wise comparison) All-non-negative vector
$\mathrm{diag}(\boldsymbol{a})$	Diagonal matrix with vector $\boldsymbol{a}$ on the diagonal
$\mathrm{supp}(\boldsymbol{a})$	Support of vector $\boldsymbol{a}$, i.e., the indices of its non-zero elements
$\mathrm{diag}(\boldsymbol{A})$	Vector containing the diagonal entries of square matrix $\boldsymbol{A}$
$\mathrm{tr}(\boldsymbol{A})$	Trace of square matrix $\boldsymbol{A}$
$\mathrm{vec}(\cdot)$	Vectorization
$\lVert\cdot\rVert_{\ell_p}$	ℓ_p norm
$\lVert\cdot\rVert_{\mathrm{F}}$	Matrix Frobenius norm
$\mathbb{E}(\cdot)$	Expectation operator
$\mathbb{P}(\cdot)$	Probability
$\otimes$	Kronecker product
$\odot$	Khatri-Rao product
$\circ$	Vector outer product
$\times_i$	Mode-i tensor-matrix product
$\binom{N}{M}$	The number of M-combinations of N elements
$\mathrm{N}(\boldsymbol{a}, \boldsymbol{A})$	Gaussian distribution with mean $\boldsymbol{a}$ and covariance $\boldsymbol{A}$
$\mathrm{CN}(\boldsymbol{a}, \boldsymbol{A})$	Complex Gaussian distribution with mean $\boldsymbol{a}$ and covariance $\boldsymbol{A}$
$[I]$	Set $\{1, 2, \cdots, I\}$
$\mathbb{R}^{M \times N}$	Set of real matrices of size $M \times N$
$\mathbb{C}^{M \times N}$	Set of complex matrices of size $M \times N$
$\mathbb{R}^{M_1 \times M_2 \times \cdots \times M_I}$	Set of real tensors of size $M_1 \times M_2 \times \cdots \times M_I$

1

Introduction

Our physical world is inherently structured. Understanding and exploiting such structures is crucial for advancing our knowledge and technology. In the realm of signal processing, many real-world signals from audio, image signal processing, or communication settings also exhibit structures. These structures are often expressed as patterns, correlations, or redundancies within the signal. Through such structures, we can unveil the essence of the signals by representing them using mathematical models and understanding their properties, which further enables us to process them more efficiently. In this dissertation, we specifically focus on the *sparsity* structure of the signal.

1.1. Background on Sparsity and Compressed Sensing

Signals in this dissertation are represented using vectors, which can be, for example, the samples of a continuous-time signal, or the measurements from a sensor array. A signal $\boldsymbol{x}$ is called *sparse* if most of its entries are zero. More precisely, if $\boldsymbol{x}$ has only s nonzero entries where s is much smaller than its length $\bar{N}$, i.e., $s \ll \bar{N}$, then we say $\boldsymbol{x}$ is s *sparse* and s is the *sparsity level*. For illustration, we show a sparse $\boldsymbol{x}$ and in contrast a dense signal in Figure 1.1.

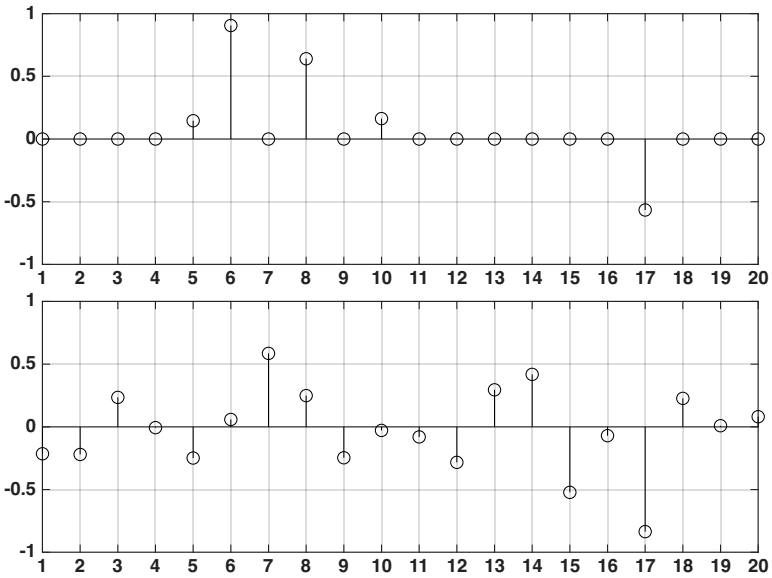


Figure 1.1.: (Top) A sparse signal $\boldsymbol{x}$ with sparsity level $s = 5$. (Bottom) A dense signal.

Sparsity, as a property of a signal, can serve as prior information for many tasks, such as signal reconstruction, denoising, image inpainting, etc. Reconstructing signals from fewer measurements than what is required by traditional Nyquist sampling by exploiting the *sparsity* of the signal is formally framed as a paradigm called *compressed sensing* [1–3]. Compressed sensing adopts a measurement process, typically linear, with random or structured basis functions, resulting in a set of linear measurements as

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{n}, \tag{1.1}$$

where $\boldsymbol{H} \in \mathbb{R}^{\bar{M}\times\bar{N}}$ is called the measurement matrix, $\boldsymbol{y} \in \mathbb{R}^{\bar{M}}$ is the measurement, and $\boldsymbol{n} \in \mathbb{R}^{\bar{M}}$ is the measurement noise. The number of measurements $\bar{M}$ is significantly smaller than the signal dimension $\bar{N}$.

Compressed sensing has found wide applications in various fields, including wireless communications, medical imaging, and radar systems. For example, in wireless communications, millimetre wave and terahertz communication channels, which are key enablers for fifth-generation communication, inherently exhibit sparsity [4, 5]. Due to significant path loss and the directionality of transmissions, only a limited number of paths dominate the channel between the transmitter and receiver [6]. This indicates that the channel is intrinsically sparse [7], and framing channel estimation as a sparse recovery problem can greatly reduce the pilot signal overhead [8]. Another example is magnetic resonance imaging, which suffers from a significant acquisition time due to high demand for spatial and spectral resolutions. Magnetic resonance images, similar to most natural images, are sparse when being represented using proper basis such as wavelet [9]. Formulating the magnetic resonance image reconstruction as a compressed sensing problem recovers magnetic resonance images from significantly fewer samples than traditional methods, leading to a substantial reduction in scan time [10–12]. A third example is radar systems [13]. In the scenario of autonomous driving, perceiving the surrounding environment accurately is essential for navigation and safe decision-making. Automotive radar systems play a crucial role in this context by providing reliable positions of nearby objects [14]. The radar signals reflected from the environment are often sparse, as only a few objects, e.g., vehicles, pedestrians, are present in the view at a given time instant. By leveraging this sparsity, advanced signal processing techniques can be employed to enhance target detection, e.g., through occupancy grid mapping [15]. Sparse recovery algorithms enable the extraction of meaningful information from limited and noisy radar measurements, improving the accuracy and reliability of object detection and tracking in driving scenarios.

The typical compressed sensing framework adopts a dense and unstructured measurement matrix $\boldsymbol{H}$. However, in certain contexts, it is beneficial to design measurement matrices that are more aligned with the signal properties. In the next section, we introduce the compressed sensing framework with Kronecker-structured, or Kronecker product matrix as measurement matrix for a specific type of signal, i.e., the multidimensional signal.

1.1.1. Multidimensional Signals and Kronecker Compressed Sensing

Many real-world signals are inherently multidimensional, arising in several engineering applications such as image processing [16–20] and wireless communications [21–25]. In these contexts, the signal is represented as a function of different dimensions, each conveying a specific physical quantity. For example, hyperspectral images, which can be represented by two spatial dimensions and one spectral dimension, follow a multidimensional structure [26]. Another notable example is in the uplink narrowband intelligent reflecting surface (IRS)-aided system. The received signal at the base station (BS) from the mobile station (MS) is a function of angle-of-departure (AoD) at MS, the difference of angle-of-arrival (AoA) and AoD at IRS, and AoA at BS [27]. Considering the angular domains of arrays at BS, MS, and IRS as separate dimensions, this signal is multidimen-

sional [23, 27].

Traditional compressed sensing requires the measurement matrix to simultaneously operate along all signal dimensions [16]. This is inefficient since it results in a large measurement matrix that inherently fails to capture the multidimensional structure of the signal. It is also infeasible in practice since such global measurements set high practical hardware constraints for the acquisition system, requiring it to work on different physical domains, e.g., time and frequency, at the same time [16].

It is beneficial to consider designing measurement schemes that are more aligned with the multidimensional nature of the signals. Employing the Kronecker product of multiple factor matrices as a measurement matrix helps to overcome these shortcomings. This is because Kronecker product measurement matrices naturally capture the multidimensional signal structure and further help to effectively reduce measurement complexity since the measurement is performed along each dimension separately [16, 28–30]. Compressed sensing for multidimensional signals with Kronecker product measurement matrices is formally called *Kronecker compressed sensing*. It appears in many acquisition systems, such as sensor arrays in wireless communication systems [23, 24, 31] or separable filters in imaging [32]. The canonical form for Kronecker compressed sensing problem follows the linear model

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{n} = (\boldsymbol{H}_1 \otimes \boldsymbol{H}_2 \cdots \otimes \boldsymbol{H}_I)\,\boldsymbol{x} + \boldsymbol{n} = \left(\otimes_{i=1}^{I} \boldsymbol{H}_i\right)\boldsymbol{x} + \boldsymbol{n}. \tag{1.2}$$

Here, $\boldsymbol{y} \in \mathbb{R}^{\tilde{M}}$ is the measurement vector, $\boldsymbol{H}_i \in \mathbb{R}^{M_i \times N_i}$ is the ith factor matrix with full row rank, $\boldsymbol{H} \in \mathbb{R}^{\tilde{M} \times \tilde{N}}$ is the known Kronecker product measurement matrix with $\prod_{i=1}^{I} M_i = \tilde{M}$ and $\prod_{i=1}^{I} N_i = \tilde{N}$, $\boldsymbol{n}$ is the measurement noise, and $\boldsymbol{x}$ is the unknown multidimensional sparse vector.

Furthermore, the measurement model we discussed so far in (1.1) and (1.2) is a *linear* model, while in many applications, the measurement process can be *nonlinear*. However, through the technique of *basis expansion model* (BEM), linear models can be readily leveraged to handle a nonlinear measurement process. Next, we introduce the BEM and how it handles nonlinear measurement processes, and then we proceed to its multidimensional extension.

BEM and its Multidimensional Extension

BEM is a key technique to handle a nonlinear measurement process, especially to estimate underlying parameters with nonlinear measurements. The idea is to discretize the unknown parameter to construct known functions, usually named *basis functions*. Then, the nonlinear measurements can be approximated by the linear combinations of such basis functions with different weights [33]. This transforms a nonlinear estimation problem into a linear one. Specifically, in (1.1), matrix $\boldsymbol{H}$ is constructed using basis functions as columns, and vector $\boldsymbol{x}$ collects the weights associated with different basis functions. Moreover, only a few basis functions corresponding to the true parameters are contributing, resulting in a sparse $\boldsymbol{x}$. Thus, this paradigm, united with compressed sensing techniques, can be leveraged to estimate the unknown parameters through reconstructing the sparse coefficients using nonlinear measurement. BEM is widely adopted in applications such as image processing [34–37] and wireless communications [8, 38, 39].

In the context of multidimensional signals, the BEM framework can be naturally extended. When estimating multiple parameters simultaneously, each parameter can be associated with a specific dimension and its own dictionary of basis functions. Multidimensional BEMs are thus typically associated with Kronecker product dictionaries. Each factor dictionary represents a different dimension and acts as the overcomplete basis for the respective dimension. The inherent nature of the Kronecker product calculates all possible combinations of the parameter values across the factor dictionaries. As a result, the combined dictionary captures every conceivable combination of basis vectors (values of the unknown parameters) across multiple dimensions. This leads to a highly expressive representation that can accurately model complex multidimensional signals. Furthermore, the multidimensional BEM coefficient vector is often sparse since only a few combinations of parameter values correspond to the true values. Therefore, estimating the unknown parameters embedded in multidimensional signals can be formulated as a Kronecker compressed sensing problem, where the goal is to recover the sparse model coefficient vector from the linear measurements obtained through a Kronecker product measurement matrix.

1.1.2. Structured Sparsity

Given the multidimensional signals, traditional compressed sensing is equipped with a structured measurement process with Kronecker product matrices, effectively capturing the multidimensional structure and reducing the measurement complexity. However, beyond the measurement structure, the sparse signals themselves often exhibit additional structures that can be exploited to enhance recovery performance [40, 41]. The standard sparsity, where the nonzero entries can be positioned arbitrarily in the sparse vector, is the most fundamental and general model, tied to various applications, such as image processing [26, 28, 42], system identification [43, 44], regression [45], and communications [46–48]. The sparsity in $\boldsymbol{x}$ can further follow specific, regulated patterns. Exploiting this *structured sparsity* is critical for developing highly effective recovery algorithms with improved performance. In this section, we present the sparsity structures that will be considered in this dissertation and motivate why such sparsity models should be considered.

(1) *Hierarchical Sparsity:* In this model, the vector $\boldsymbol{x}$ is partitioned into blocks at multiple levels, and sparsity is structured across these levels. For example, the two-level hierarchical sparsity means that only a few blocks are active, i.e., containing non-zero entries, at the first level, and within each active block, only a few entries are non-zero at the second level. This model is particularly relevant in scenarios where signals exhibit groupings or clusters of activity. An example arises in massive machine-type communications, where device-level sparsity is combined with signal-level sparsity, creating a two-level hierarchical pattern [49–51] as

- *Level 1 (Device Sparsity)*: At any given moment, only a small fraction of the devices are active and transmitting data.
- *Level 2 (Signal Sparsity)*: The signal from each active device is sparse. For instance, the data of an environmental sensor only change significantly at a few points within a certain period of time.

Such sparsity patterns enable us to jointly detect active users by identifying the support of the sparse activity vector and to recover the transmitted signal [52].

(2) *Kronecker-Supported Sparsity:* This sparsity model assumes that the support of $\boldsymbol{x}$ is a Kronecker product of several lower-dimensional binary support vectors as

$$\text{supp}(\boldsymbol{x}) = \otimes_{i=1}^{I} \text{supp}(\boldsymbol{x}_i),$$

where $\text{supp}(\boldsymbol{x})$ is the (binary) support vector indicating the nonzero entries of $\boldsymbol{x}$. Some motivating applications are in wireless communications [27, 31, 53] and image processing [17–19, 42, 54]. As an example, consider the channel estimation problem for the IRS-aided wireless communication systems [27]. The BEM for the unknown channel matrix is constructed by sampling pre-defined spatial angle grids and forming a dictionary using the corresponding steering vectors. Then, the channel matrix can be represented using a three-dimensional sparse BEM coefficient vector where the three dimensions are the AoD, AoA, and difference of the AoA and AoD at the IRS. Further, different combinations of AoDs and AoAs naturally elicit the Kronecker product, leading to Kronecker-supported sparse coefficients. We note that Kronecker-supported sparsity is equivalent to the tensor block sparsity considered in [17–19].

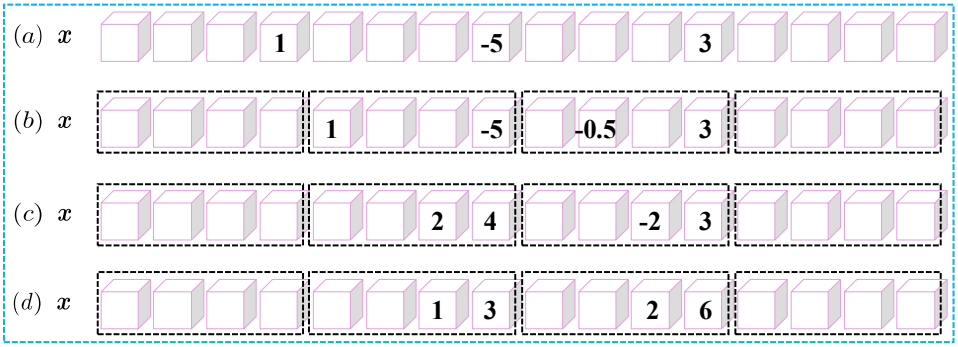


Figure 1.2.: Different sparsity patterns. Cubes with numbers and without numbers represent nonzero and zero entries, respectively. (*a*) Standard sparsity. (*b*) Hierarchical sparsity. (*c*) Kronecker-supported sparsity. (*d*) Kronecker-structured sparsity.

(3) *Kronecker-structured sparsity:* Compared to the Kronecker-supported sparsity, the Kronecker-structured sparsity adopts one more constraint on the amplitude of $\boldsymbol{x}$ by assuming that $\boldsymbol{x}$ itself is a Kronecker product of several lower-dimensional sparse vectors as

$$\boldsymbol{x} = \otimes_{i=1}^{I} \boldsymbol{x}_i. \tag{1.3}$$

This structure is motivated by problems where a signal depends on multiple *independent* parameters in *separable* dimensions. Consider the narrowband IRS-aided wireless communication system as an example in Figure 1.3. The received pilot signal in this setting is a function of *independent* variables such as the AoA at the receiver and the AoD from the transmitter. Considering the angular domain of each array as a separate dimension, the high-dimensional pilot signal can thus be modeled as a Kronecker

product of individual and independent signal components, each encapsulating the signal in the corresponding and separable dimension (e.g., AoA and AoD) and being the nonlinear function of AoAs, or AoDs, and the path gain corresponding to each AoA or AoD. In the BEM to estimate the unknown parameters (e.g., the *true* angles), due to the Kronecker structure of the received pilot signal, its sparse coefficient vector is also Kronecker-structured, following (1.3), where $\boldsymbol{x}_i$ is the sparse coefficient vector for ith dimension. The structure (1.3) directly mirrors the *separable* nature of the underlying problem.

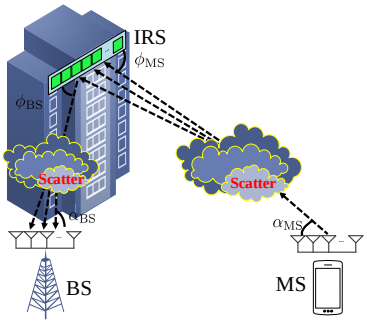


Figure 1.3.: An illustration of AoAs and AoDs in an IRS-aided channel.

To illustrate, we provide simple examples of different sparsity patterns in Figure 1.2 with $I = 2$. Figure 1.2(a) illustrates the standard sparsity with $s = 3$. Figure 1.2(b) depicts the hierarchical sparsity where there are two nonzero blocks out of four blocks. And each nonzero block is 2-sparse. Figure 1.2(c) shows a Kronecker-supported sparse vector, where the supports across different nonzero blocks are the same. Figure 1.2(d) is the Kronecker-structured sparsity, where not only the supports remain identical, but also the amplitudes of nonzero entries are constrained. The sparse vector shown in Figure 1.2(d) is obtained by $\boldsymbol{x} = [0,1,2,0]^{\mathsf{T}} \otimes [0,0,1,3]^{\mathsf{T}}$.

1.2. Key Challenges, Existing Arts, Research Gaps, and Contributions

In this dissertation, we focus on the Kronecker compressed sensing problem with different sparsity patterns. In this section, we first present the key challenges in solving the Kronecker compressed sensing problem. Then, we review existing works, identify research gaps, and outline the questions that this dissertation aims to answer.

1.2.1. Challenges

Solving the Kronecker compressed sensing problem in (1.2) presents several challenges, spanning from computational complexity, the exploitation of signal structure to theoretical guarantees.

1. *High Dimensionality*: The first challenge in solving (1.2), regardless of the sparsity pattern, is the high dimensionality of the measurement matrix $\boldsymbol{H}$ and multidimensional signal $\boldsymbol{x}$. The dimension of the measurement matrix $\boldsymbol{H}$ is $\bar{M} \times \bar{N}$ with $\bar{M} = \prod_{i=1}^{I} M_i$ and $\bar{N} = \prod_{i=1}^{I} N_i$. Directly handling such a large-scale matrix can be computationally demanding, especially when the number of dimensions I increases. The computational complexity of traditional compressed sensing algorithms that do not exploit the Kronecker structure can be prohibitive, even for the low complexity of classic sparse recovery algorithms, such as orthogonal matching pursuit (OMP) and hard thresholding pursuit (HTP).

2. *Structural Prior Exploitation*: The second challenge stems from the proper exploitation of sparsity patterns as prior knowledge. As we have motivated before, beyond simple sparsity, the nonzero elements in $\boldsymbol{x}$ can often exhibit more complex but regulated patterns. Exploiting these structured sparsity patterns can significantly enhance recovery performance. However, designing algorithms that can properly and effectively leverage these structures while also handling the Kronecker structure in $\boldsymbol{H}$ is nontrivial. Many existing algorithms either ignore the Kronecker structure or are tailored to specific sparsity patterns, limiting their applicability.

3. *Theoretical Guarantees*: The third challenge lies in establishing theoretical guarantees for recovery performance. While the restricted isometry property (RIP) has been a cornerstone in analyzing classic sparse recovery algorithms such as iterative hard thresholding (IHT) and HTP, extending these analyses to the Kronecker compressed sensing framework with various sparsity patterns is complicated. For more advanced compressed sensing algorithms such as sparse Bayesian learning (SBL), theoretical guarantees such as convergence guarantees are absent. The interplay between the Kronecker structure of $\boldsymbol{H}$ and the structured sparsity of $\boldsymbol{x}$ introduces additional complication that must be carefully addressed to provide meaningful theoretical insights.

Given all the challenges mentioned above, various approaches have been proposed in the literature to tackle the Kronecker compressed sensing problem. In the next section, we review existing works and investigate how these research challenges are handled, and identify research gaps and questions that this dissertation aims to answer.

1.2.2. Existing Arts and Limitations

The Kronecker product measurement matrix is introduced for compressed imaging in [20]. Kronecker compressed sensing is formally defined in [16] with a RIP analysis for Kronecker compressed sensing with standard sparsity [16, 55]. It bounds the restricted isometry constant (RIC) of the Kronecker product using the RICs of factor matrices $\boldsymbol{H}_i$. However, the recovery algorithm fails to leverage the Kronecker structure in $\boldsymbol{H}$ but relies on a general compressed sensing solver. To leverage this structure, Kronecker orthogonal matching pursuit (KroOMP) in [18] adopts tensor operations. Nonetheless, it still incurs a high computation burden because in each iteration, the newly added column has length $\bar{N}$, which requires a complexity of $\mathcal{O}(\bar{N})$, and lacks theoretical analysis. In parallel, two algorithms are presented in [32, 56]: one uses tensor unfolding for sequential recovery

in dimension, and the other uses approximate Tucker decomposition to recover along each dimension in parallel. Although both approaches consider the Kronecker product structure of the measurement matrix $\boldsymbol{H}$, they are both limited to standard sparsity and extending them to other sparsity structures is nontrivial. Later, the unfolding-based approach is transformed into multiple subproblems through decomposition in [26], which are then solved *independently*. However, it fails to exploit that these subproblems can actually be solved jointly due to the shared sparsity patterns and is not immediately extendable to other sparsity patterns either.

Kronecker compressed sensing with structured sparse recovery is also investigated in the literature. For *hierarchically sparse* vectors, the hierarchical hard thresholding pursuit (HiHTP) is discussed in [57], adapting the traditional HTP with a designed thresholding operator. The RIP and coherence analysis are also tailored with the algorithm. However, it treats $\boldsymbol{H}$ as a standalone matrix and fails to incorporate the Kronecker structure in $\boldsymbol{H}$, leading to higher computational costs. For *Kronecker-supported sparsity*, both greedy and Bayesian methods have been explored. An OMP-based algorithm offers reduced complexity [17, 18] but performs poorly in noisy settings [23]. Bayesian algorithms, designed for applications such as hyperspectral image processing [42] and wireless communication [31, 53], use a structured prior distribution. Although they achieve good empirical performance, they have the following three drawbacks. First, they involve iterative updates of large covariance matrices, leading to high computational complexity. Second, they rely on approximations for tractability of their underlying optimization problems, resulting in a lack of theoretical guarantees on convergence and performance [31, 53]. Third, due to the usage of a structured prior distribution, they suffer from poor generalization to other sparsity patterns.

To summarize, existing approaches fail to fully exploit the computational benefits of the Kronecker structure in $\boldsymbol{H}$ and are generally restricted to single specific sparsity patterns rather than being adaptable to multiple structures. Furthermore, these methods lack a unified theoretical framework, leaving significant gaps in convergence and performance guarantees for structured sparsity contexts. Based on these limitations, we identify the following research questions that this dissertation aims to address.

1.2.3. Research Objectives and Contributions

In this dissertation, our main objective is to develop algorithms for solving the Kronecker compressed sensing problem in (1.2) that effectively exploit both the Kronecker structure of the measurement matrix $\boldsymbol{H}$ and various sparsity patterns in $\boldsymbol{x}$, including standard, hierarchical, Kronecker-supported, and Kronecker-structured sparsity. Alongside the algorithms, we also aim to establish theoretical guarantees for these algorithms. We break down our main research objectives into the following research questions, starting with the algorithms and guarantees for *Kronecker-supported* and *Kronecker-structured* sparsity and then moving towards *a unified algorithmic and theoretical framework*. The first research question is summarized as follows.

Q1 *How can we design an efficient algorithm that leverages the Kronecker structure of the measurement matrix $\boldsymbol{H}$ and enforces the Kronecker-supported sparsity pattern in $\boldsymbol{x}$? Apart from the algorithm design, how can we provide theoretical guarantees on these*

algorithms?

We present the answer to this question in Chapter 2. We design novel Kronecker SBL (KroSBL) algorithms for recovering vectors with Kronecker-structured support. We first design a prior distribution parameterized by a Kronecker-structured covariance, which enforces the Kronecker-supported sparsity pattern in $\boldsymbol{x}$. This prior, however, leads to a non-convex optimization problem. We further develop two approaches for a solution. Our alternating minimization (AM)-based approach comes with proven convergence guarantees, addressing a major gap in prior Bayesian works. Our singular value decomposition (SVD)-based approach enjoys a faster convergence speed, serving as a more efficient alternative. We also design a complexity reduction technique, leveraging the Kronecker structure of the measurement matrix $\boldsymbol{H}$, which significantly reduces the computation time compared to existing Bayesian methods.

The next research question further considers a more constrained Kronecker-structured sparsity. Although KroSBL can be directly applied to this sparsity pattern, they do not sufficiently enforce the additional Kronecker structure in $\boldsymbol{x}$. Thus, we next aim to design specialized algorithms that leverage both the Kronecker structure of $\boldsymbol{H}$ and the Kronecker-structured sparsity in $\boldsymbol{x}$, leading to the second research question.

Q2 *How can we develop an efficient algorithm that exploits both the Kronecker structure of $\boldsymbol{H}$ and the Kronecker-structured sparsity in $\boldsymbol{x}$? Can we also provide theoretical guarantees on these algorithms?*

We address this question in Chapter 3. We design a decomposition-based approach, leveraging the mixture property of the Kronecker product. The original high-dimensional sparse vector recovery problem reduces to multiple lower-dimensional sparse recovery subproblems, allowing us to recover the sparse vector in each dimension separately and to construct a Kronecker-structured estimation directly. Furthermore, in the context of parameter estimation through multidimensional BEM, to handle the grid mismatch issue, we design a novel off-grid SBL algorithm. We provide statistical guarantees for the decomposition step and convergence analysis for off-grid SBL. We also explore the reason for a better performance of decomposition-based solution. We attribute such performance enhancement to the denoising effect of decomposition, which we also quantitatively characterize.

The answers to Q1 and Q2 pertain to Challenges 1 and 3. But a generic algorithmic and theoretical framework to address Challenge 2 still remains open. In the last part, we shift our focus from *structure-specific* solutions to a *unified framework* targeting the following research question.

Q3 *How can we develop a unified algorithmic framework that can efficiently handle standard, hierarchical, and Kronecker-supported sparsity? Can we also establish a unified theoretical analysis that provides recovery guarantees for all three sparsity patterns under this framework?*

Addressing this question leads to a *hierarchical view* of Kronecker compressed sensing in Chapter 4. This view, rooted in the dimension-wise nature of the Kronecker product measurement process, provides the foundation for a unified approach to algorithm design

and theoretical analysis. It allows us to reinterpret the high-dimensional Kronecker compressed sensing problem as a sequence of related, lower-dimensional problems, providing significant gains in efficiency and enabling a cohesive theoretical understanding. We further present how to leverage this hierarchical view to design an algorithmic framework and establish recovery guarantees for all three sparsity patterns. A natural follow-up question can be raised regarding the applicability of this framework to the Kronecker-structured signal. We note that the unified approach only exploits the Kronecker structure in the measurement matrix and the Kronecker support of the Kronecker-structured signal, but it does not utilize the prior knowledge regarding its amplitude. This amplitude property is closely related to a rank constraint, which cannot be effectively exploited by the unified approach. Therefore, while the unified framework offers greater versatility, the decomposition-based approach serves as a better option for signals exhibiting strict Kronecker structure. The answers to Q3, presented in Chapter 4, comprehensively address Challenge 2 and Challenge 3.

In Chapter 5, we conclude this dissertation by summarizing the key findings and discussing potential directions for future research in Kronecker compressed sensing and structured sparsity. Appendix A, B, and C contain detailed proofs of our theoretical results and additional discussions that support the main content of this dissertation.

To conclude, we map the connections between the research output of this Ph.D. and the structure of this dissertation.

Chapter 2 is based on:

- Y. He and G. Joseph. 'Structure-aware sparse Bayesian learning-based channel estimation for intelligent reflecting surface-aided MIMO'. In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*. 2023, pp. 1–5
- Y. He and G. Joseph. 'Bayesian algorithms for Kronecker-structured sparse vector recovery with application to IRS-MIMO channel estimation'. In: *IEEE Transactions on Signal Processing* 73 (2025), pp. 142–157

Chapter 3 is based on:

- Y. He and G. Joseph. 'Kronecker-structured sparse vector recovery with application to IRS-MIMO channel estimation'. In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*. 2025, pp. 1–5
- Y. He and G. Joseph. 'Efficient off-grid Bayesian parameter estimation for Kronecker structured signals'. In: *IEEE Transactions on Signal Processing* 73 (2025), pp. 2616–2630

Chapter 4 is based on:

- Y. He and G. Joseph. 'A hierarchical view of structured sparsity in Kronecker compressive sensing'. In: *Proceedings of the European Signal Processing Conference*. 2025, pp. 1–5
- Y. He and G. Joseph. 'Hierarchical multi-stage recovery framework for Kronecker compressed sensing'. In: *Proceedings of the International Conference on Learning Representations*. 2026

2

Bayesian Algorithms for Kronecker-Supported Sparse Vector Recovery

In this chapter, we study the first research question, which concerns a sparse recovery problem with an underdetermined linear system characterized by a Kronecker product dictionary and a Kronecker-supported sparse vector. We cast this problem into the SBL framework and rely on the expectation-maximization (EM) method for a solution. To this end, we model the Kronecker-structured support with a hierarchical Gaussian prior distribution parameterized by a Kronecker-structured hyperparameter, leading to a non-convex optimization problem. The optimization problem is solved using an AM method and a SVD-based method, resulting in two algorithms. Further, we analytically guarantee that the AM-based method converges to the stationary point of the KroSBL cost function. The SVD-based method, though it adopts approximations, is empirically shown to be more efficient and accurate. We then apply our algorithm to estimate the uplink wireless channel in an IRS-aided multiple-input multiple-output (MIMO) system and extend the AM-based algorithm to address block sparsity in the channel. We also study the KroSBL cost to show that the minima of the cost function are achieved at sparse solutions and that incorporating the Kronecker structure leads to a lower number of local minima of the KroSBL cost function than those of SBL cost function. Our numerical results demonstrate the effectiveness of our algorithms compared to other benchmarks.

2.1. Introduction

In this chapter, we consider the Kronecker-structured linear inversion problem

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{n} = \left(\otimes_{i=1}^{I} \boldsymbol{H}_i\right)\boldsymbol{x} + \boldsymbol{n},$$

as in (1.2). Here, $\boldsymbol{y} \in \mathbb{R}^{\bar{M}}$ is the measurement vector, $\boldsymbol{H}_i \in \mathbb{R}^{M_i \times N_i}$ is the ith factor matrix, $\boldsymbol{H} \in \mathbb{R}^{\bar{M} \times \bar{N}}$ is the known Kronecker product measurement matrix with $\prod_{i=1}^{I} M_i = \bar{M}$ and $\prod_{i=1}^{I} N_i = \bar{N}$, and $\boldsymbol{n}$ is the measurement noise. We aim to reconstruct the Kronecker-supported sparse vector $\boldsymbol{x}$ given the Kronecker product measurement matrix $\boldsymbol{H}$ and noisy measurement $\boldsymbol{y}$ by exploiting its Kronecker-structured support. We start this section by first reviewing the related works on recovery of Kronecker-supported sparse vectors in Kronecker compressed sensing, and then summarizing our main contributions.

2.1.1. Related Works

Various signal processing techniques have been proposed to reconstruct the sparse vector $\boldsymbol{x}$ from the linear measurement $\boldsymbol{y}$ and Kronecker product dictionary $\boldsymbol{H}$. A greedy method called Kronecker-orthogonal matching pursuit (KOMP) is designed in [18] to generalize the traditional OMP for multidimensional signals. Like OMP, the KOMP algorithm has low complexity but requires hand-tuning of a sensitive stopping threshold. Recently, parameter tuning-free approaches based on SBL are studied [31]. SBL is known to have superior performance [62] and flexibility to incorporate several additional structures along with sparsity [63–66]. The Kronecker structure can also be incorporated into the SBL framework. An SBL approach for the Kronecker structure in (1.2) with $I = 3$ is introduced in [42], using tensor-wise hyperparameters. The linear inversion problem with $I = 2$ is considered in [67]. This framework is later generalized to the I-dimensional tensor and applied to wireless communications [31, 53]. However, the derivation of the SBL algorithm

in [31, 53] relies on several approximations, leading to a suboptimal recovery accuracy. Hence, we seek novel Bayesian algorithms exploiting the Kronecker and sparse structures to improve reconstruction accuracy and efficiency.

Furthermore, the existing Bayesian algorithms exploiting the Kronecker structure [31, 53] lack theoretical guarantees on their convergence and performance, and these properties are only demonstrated empirically. The study on convergence is limited because the convergence guarantee of the classic SBL using the EM algorithm cannot be trivially extended for these algorithms. It is claimed that the solution to the underlying optimization problem should be obtained at the stationary point of the cost function [31, 53]. However, it is unclear whether the stationary point can be reached due to its iterative nature and approximations. Similarly, the performance improvement due to incorporating the Kronecker structure into the SBL has been shown in [31] without any theoretical justification. Given a lack of theoretical analysis of the existing algorithms and analysis of the SBL cost function, we make progress on these problems by analyzing our new Bayesian algorithms.

2.1.2. Contributions

The main contributions of this chapter are as follows:

- *Algorithm development:* We present our new Bayesian recovery algorithms in Section 2.3. We first present two novel SBL algorithms to recover a sparse vector $\boldsymbol{x}$ with Kronecker-structured support, called KroSBL. The first KroSBL algorithm, based on AM, solves the underlying optimization problem of the SBL algorithm using the AM procedure. The second KroSBL algorithm, based on SVD, is faster and uses a simple approximation to obtain the SBL algorithm.

- *Application:* We apply our problem to a prototypical application of IRS-aided MIMO channel estimation in Section 2.6. Besides the sparsity with Kronecker-structured support, the BEM representation of IRS-cascaded channels also exhibits block sparsity where the nonzero entries occur in clusters. To handle this additional structure, we extend our AM-based algorithm for Kronecker-supported block sparsity based on non-negative least squares.

- *Convergence guarantee:* We derive convergence guarantees for the AM-based algorithm in Section 2.4.1. We establish that *i)* the AM procedure can attain the stationary point of the cost function in the M-step, and *ii)* the AM-based algorithm is guaranteed to converge to the stationary point of the KroSBL cost function in the noisy setting. These results address the gap in existing works by providing theoretical guarantees, which were previously lacking.

- *Cost function analysis:* We examine the local minima of the KroSBL cost function in Section 2.4.2. Beyond the structure that the sparse vector $\boldsymbol{x}$ has a Kronecker-structured support, we assume the sparse vector in (1.2) satisfies $\boldsymbol{x} = \otimes_{i=1}^{I} \boldsymbol{x}_i$. Then, we prove that all local minima are sparse in the noiseless case. Besides, we demonstrate that incorporating the Kronecker structure in the hyperparameter vector as the KroSBL cost function can significantly reduce the local minima under the unique representation property (URP) assumption for $\boldsymbol{H}$.

- *Numerical Results:* We assess our schemes in two scenarios in Section 2.5. Firstly, we study the sparse recovery performance of the presented schemes against algorithms in the literature to demonstrate their superior recovery accuracy and runtime. Secondly, we conduct a case study on channel estimation for IRS-aided systems with our approach and illustrate that incorporating block sparsity in our scheme can further enhance performance.

Overall, we present two algorithms for the sparse signal recovery problem in BEM with multiple unknown parameters. The first algorithm, AM-KroSBL, enjoys solid theoretical guarantees, while the other algorithm, SVD-KroSBL, is computationally light and practically more relevant.

2.2. Sparse Bayesian Learning

Before diving into the technical details, we first present the EM-based SBL algorithm, since SBL serves as the foundation of many of our approaches in the following chapters. Here, we consider a traditional, unstructured sparse recovery problem

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{n},$$

where $\boldsymbol{y} \in \mathbb{R}^M$ is the measurement, $\boldsymbol{H} \in \mathbb{R}^{M\times N}$ is the measurement matrix, $\boldsymbol{x} \in \mathbb{R}^N$ is the unknown sparse signal, and $\boldsymbol{n} \in \mathbb{R}^M$ is assumed to be the additive zero-mean Gaussian noise with unknown variance σ^2. The goal of SBL is to recover the sparse signal $\boldsymbol{x}$ from the measurement $\boldsymbol{y}$.

To promote sparsity, SBL assumes a Gaussian prior for the sparse signal $\boldsymbol{x}$ with a diagonal covariance matrix [62] as

$$p(\boldsymbol{x};\boldsymbol{\gamma}) = \mathrm{N}\left(\boldsymbol{0}, \mathrm{diag}(\boldsymbol{\gamma})\right),$$

where $\boldsymbol{\gamma} \in \mathbb{R}^N$ is a hyperparameter. Then, we turn to the type-II maximum likelihood (ML) estimation, i.e., we first estimate the hyperparameter $\boldsymbol{\gamma}$ and noise variance σ^2, based on which the maximum a posteriori (MAP) estimate of $\boldsymbol{x}$ is obtained as $\arg\max_{\boldsymbol{x}} p(\boldsymbol{x}|\boldsymbol{y};\boldsymbol{\gamma},\sigma^2)$ [68]. The ML estimates of $\boldsymbol{\gamma}$ and σ^2 are obtained by minimizing the negative log-likelihood, i.e., the SBL cost function, as

$$\min_{\boldsymbol{\gamma}\geq 0,\sigma^2} -\log p(\boldsymbol{y};\boldsymbol{\gamma},\sigma^2) = \min_{\boldsymbol{\gamma}\geq 0,\sigma^2} \log|\boldsymbol{\Sigma}_{\boldsymbol{y}}| + \boldsymbol{y}^{\mathsf{T}}(\boldsymbol{\Sigma}_{\boldsymbol{y}})^{-1}\boldsymbol{y}, \tag{2.1}$$

where $\boldsymbol{\Sigma}_{\boldsymbol{y}} = \sigma^2\boldsymbol{I} + \boldsymbol{H}\,\mathrm{diag}(\boldsymbol{\gamma})\boldsymbol{H}^{\mathsf{T}}$. The problem (2.1) does not admit a closed-form solution. Thus, we resort to the EM algorithm [62, 69] for a solution through an iterative procedure. Specifically, the rth iteration of EM is

$$\begin{aligned}
&\textbf{E-step: Compute } \mathbb{E}_{\boldsymbol{x}|\boldsymbol{y};\boldsymbol{\gamma}^{(r)},(\sigma^2)^{(r)}}\left[\log p(\boldsymbol{y},\boldsymbol{x};\boldsymbol{\gamma},\sigma^2)\right],\\
&\textbf{M-step: } \boldsymbol{\gamma}^{(r+1)},(\sigma^2)^{(r+1)} = \arg\max_{\boldsymbol{\gamma}\geq 0,\sigma^2} \mathbb{E}_{\boldsymbol{x}|\boldsymbol{y};\boldsymbol{\gamma}^{(r)},(\sigma^2)^{(r)}}\left[\log p(\boldsymbol{y},\boldsymbol{x};\boldsymbol{\gamma},\sigma^2)\right].
\end{aligned}$$

Further, we note that the M-step is separable in $\boldsymbol{\gamma}$ and σ^2 as

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{x}|\boldsymbol{y};\boldsymbol{\gamma}^{(r)},(\sigma^2)^{(r)}}\left[\log p(\boldsymbol{x},\boldsymbol{y};\boldsymbol{\gamma},\sigma^2)\right]=\mathbb{E}_{\boldsymbol{x}|\boldsymbol{y};\boldsymbol{\gamma}^{(r)},(\sigma^2)^{(r)}}\left[\log p(\boldsymbol{y}|\boldsymbol{x};\sigma^2)+\log p(\boldsymbol{x};\boldsymbol{\gamma})\right]\\ =&\mathbb{E}_{\boldsymbol{x}|\boldsymbol{y};\boldsymbol{\gamma}^{(r)},(\sigma^2)^{(r)}}\left[\log p(\boldsymbol{y}|\boldsymbol{x};\sigma^2)\right]+\mathbb{E}_{\boldsymbol{x}|\boldsymbol{y};\boldsymbol{\gamma}^{(r)},(\sigma^2)^{(r)}}\left[\log p(\boldsymbol{x};\boldsymbol{\gamma})\right].\end{aligned}$$

The optimization cost function regarding $\boldsymbol{\gamma}$ can be rewritten as

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{x}|\boldsymbol{y};\boldsymbol{\gamma}^{(r)},(\sigma^2)^{(r)}}\left[\log p(\boldsymbol{x};\boldsymbol{\gamma})\right]\\ =&-\frac{1}{2}\mathbb{E}_{\boldsymbol{x}|\boldsymbol{y};\boldsymbol{\gamma}^{(r)},(\sigma^2)^{(r)}}\left[\boldsymbol{x}^{\mathsf{T}}\operatorname{diag}(\boldsymbol{\gamma})^{-1}\boldsymbol{x}\right]-\frac{1}{2}\log((2\pi)^N|\operatorname{diag}(\boldsymbol{\gamma})|)\\ =&-\frac{1}{2}\log((2\pi)^N)-\frac{1}{2}\log|\operatorname{diag}(\boldsymbol{\gamma})|-\frac{1}{2}(\boldsymbol{d}^{(r)})^{\mathsf{T}}\boldsymbol{\gamma}^{-1},\end{aligned}\tag{2.2}$$

while the cost function of the optimization in M-step regarding σ^2 can be simplified as

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{x}|\boldsymbol{y};\boldsymbol{\gamma}^{(r)},(\sigma^2)^{(r)}}\left[\log p(\boldsymbol{y}|\boldsymbol{x};\sigma^2)\right]\\ =&-\frac{1}{2\sigma^2}\mathbb{E}_{\boldsymbol{x}|\boldsymbol{y};\boldsymbol{\gamma}^{(r)},(\sigma^2)^{(r)}}\left[\|\boldsymbol{y}-\boldsymbol{H}\boldsymbol{x}\|_2^2\right]-\frac{M}{2}\log(2\pi\sigma^2)\\ =&-\frac{1}{2\sigma^2}\left(\|\boldsymbol{y}-\boldsymbol{H}\boldsymbol{\mu}_{\boldsymbol{x}}\|_2^2+\operatorname{tr}(\boldsymbol{H}\boldsymbol{\Sigma}_{\boldsymbol{x}}\boldsymbol{H}^{\mathsf{T}})\right)-\frac{M}{2}\log(2\pi\sigma^2).\end{aligned}\tag{2.3}$$

We define

$$\boldsymbol{d}^{(r)}=\operatorname{diag}(\boldsymbol{\Sigma}_{\boldsymbol{x}}+\boldsymbol{\mu}_{\boldsymbol{x}}\boldsymbol{\mu}_{\boldsymbol{x}}^{\mathsf{T}}).\tag{2.4}$$

Here, $\boldsymbol{\mu}_{\boldsymbol{x}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{x}}$, which depend on $\boldsymbol{\gamma}^{(r)}$ and $(\sigma^2)^{(r)}$, are the mean and variance of conditional distribution $p(\boldsymbol{x}|\boldsymbol{y};\boldsymbol{\gamma}^{(r)},(\sigma^2)^{(r)})$, respectively, which can be computed as

$$\boldsymbol{\mu}_{\boldsymbol{x}}=\sigma^{-2}\boldsymbol{\Sigma}_{\boldsymbol{x}}\boldsymbol{H}^{\mathsf{T}}\boldsymbol{y},\ \boldsymbol{\Sigma}_{\boldsymbol{x}}=\left[\sigma^{-2}\boldsymbol{H}^{\mathsf{T}}\boldsymbol{H}+\operatorname{diag}(\boldsymbol{\gamma}^{(r)})^{-1}\right]^{-1}.\tag{2.5}$$

Thus, the M-step updates $\boldsymbol{\gamma}$ and σ^2 by solving (2.2) and (2.3), respectively, resulting in

$$\boldsymbol{\gamma}^{(r+1)}=\boldsymbol{d}^{(r)},\tag{2.6}$$

$$(\sigma^2)^{(r+1)}=\frac{\|\boldsymbol{y}-\boldsymbol{H}\boldsymbol{\mu}_{\boldsymbol{x}}\|_2^2+\operatorname{tr}(\boldsymbol{H}\boldsymbol{\Sigma}_{\boldsymbol{x}}\boldsymbol{H}^{\mathsf{T}})}{M}.\tag{2.7}$$

The EM algorithm iterates between the E-step and M-step until convergence, yielding the estimates of $\boldsymbol{\gamma}$ and σ^2. The sparse signal $\boldsymbol{x}$ can be estimated using $\boldsymbol{\mu}_{\boldsymbol{x}}$ through the final estimates of $\boldsymbol{\gamma}$ and σ^2. The EM-based SBL algorithm is summarized in Algorithm 1. In the next section, we present our algorithms on recovering a Kronecker-supported sparse vector from a Kronecker product underdetermined system, based on the SBL framework.

2.3. Kronecker-Structured Sparse Bayesian Learning

We study the model in (1.2) as

$$\boldsymbol{y}=\boldsymbol{H}\boldsymbol{x}+\boldsymbol{n}=\left(\otimes_{i=1}^{I}\boldsymbol{H}_i\right)\boldsymbol{x}+\boldsymbol{n}.$$

Algorithm 1 SBL

Input: Measurement $\boldsymbol{y}$, matrix $\boldsymbol{H}$, threshold ϵ
Initialization: set $r=1$, $\boldsymbol{\gamma}^{(0)}=\boldsymbol{0}$, $\boldsymbol{\gamma}^{(1)}=\boldsymbol{1}$, $(\sigma^2)^{(1)}=0.1$
while $\|\boldsymbol{\gamma}^{(r)}-\boldsymbol{\gamma}^{(r-1)}\|_2>\epsilon$ **do**
  Compute $\boldsymbol{d}^{(r)}$ using (2.4) and (2.5)
  Compute $\boldsymbol{\gamma}^{(r+1)}$ and $(\sigma^2)^{(r+1)}$ using (2.6) and (2.7)
  Update iteration index: $r \leftarrow r+1$
end while
Output: $\boldsymbol{x}=\boldsymbol{\mu}_{\boldsymbol{x}}$ using (2.5)

Here, $\boldsymbol{y}\in\mathbb{R}^{\bar{M}}$ is the measurement vector, $\boldsymbol{H}_i\in\mathbb{R}^{M_i\times N_i}$ is the ith factor matrix, $\boldsymbol{H}\in\mathbb{R}^{\bar{M}\times\bar{N}}$ is the known Kronecker product measurement matrix with $\prod_{i=1}^{I}M_i=\bar{M}$ and $\prod_{i=1}^{I}N_i=\bar{N}$, $\boldsymbol{n}$ is assumed to be additive Gaussian noise which follows $\mathrm{CN}(\boldsymbol{0},\sigma^2\boldsymbol{I}_{\bar{M}})$, and $\boldsymbol{x}$ is the unknown multidimensional sparse vector. For simplicity, we assume that the noise variance σ^2 is known and $N_i=N$ for $i=1,2,\ldots,I$. We relax the knowledge of noise variance in Section 2.3.3. This section presents new recovery algorithms to solve for $\boldsymbol{x}$ from (1.2), exploiting the Kronecker structure in the support.

Inspired by the SBL framework [62], we impose a fictitious sparsity promoting zero-mean Gaussian prior on $\boldsymbol{x}$ with an unknown covariance matrix $\boldsymbol{\Gamma}\in\mathbb{R}^{N^I\times N^I}$. We construct the covariance matrix as $\boldsymbol{\Gamma}=\mathrm{diag}(\boldsymbol{\gamma})$ with $\boldsymbol{\gamma}=\otimes_{i=1}^{I}\boldsymbol{\gamma}_i$ and $\boldsymbol{\gamma}_i\in\mathbb{R}^{N\times 1}$. We impose $\boldsymbol{\gamma}=\otimes_{i=1}^{I}\boldsymbol{\gamma}_i$ to mimic the Kronecker-structured support. Specifically,

$$p(\boldsymbol{x};\{\boldsymbol{\gamma}_i\})=\mathrm{CN}(\boldsymbol{0},\boldsymbol{\Gamma}),$$

where $\{\boldsymbol{\gamma}_i\}$ is the simplified notation for $\{\boldsymbol{\gamma}_i\}_{i=1}^{I}$ used henceforth. Then, we turn to the type-II maximum likelihood (ML) estimation, i.e., we first estimate the hyperparameters $\{\boldsymbol{\gamma}_i\}$, based on which the maximum a posteriori (MAP) estimate of $\boldsymbol{x}$ is obtained as $\arg\max_{\boldsymbol{x}}p(\boldsymbol{x}|\boldsymbol{y};\{\boldsymbol{\gamma}_i\})$ [68]. The ML estimates of $\{\boldsymbol{\gamma}_i\}$ are obtained by minimizing the negative log-likelihood, i.e., the KroSBL cost function

$$\mathscr{L}\left(\{\boldsymbol{\gamma}_i\}\right)=-\log p(\boldsymbol{y};\{\boldsymbol{\gamma}_i\})=\log|\boldsymbol{\Sigma}_{\boldsymbol{y}}|+\boldsymbol{y}^{\mathsf{H}}\boldsymbol{\Sigma}_{\boldsymbol{y}}^{-1}\boldsymbol{y}, \tag{2.8}$$

where $\boldsymbol{\Sigma}_{\boldsymbol{y}}=\sigma^2\boldsymbol{I}_{\bar{M}}+\boldsymbol{H}\boldsymbol{\Gamma}\boldsymbol{H}^{\mathsf{H}}$ [62]. We note that $\boldsymbol{\gamma}=\otimes_{i=1}^{I}\boldsymbol{\gamma}_i=\otimes_{i=1}^{I}\alpha_i\boldsymbol{\gamma}_i$ for any $\alpha_i>0$ when $\prod_{i=1}^{I}\alpha_i=1$. Thus, if $\{\boldsymbol{\gamma}_i\}$ maximizes (2.8), then $\{\alpha_i\boldsymbol{\gamma}_i\}$ also achieves the maximum for any $\alpha_i>0$ with $\prod_i\alpha_i=1$. To eliminate this scaling ambiguity, we normalize the hyperparameter vectors, i.e., we impose the constraint $\|\boldsymbol{\gamma}_i\|_2=1$ for $i=1,2,\ldots,I-1$. So the ML problem to estimate $\{\boldsymbol{\gamma}_i\}$ reduces to

$$\min_{\{\boldsymbol{\gamma}_i\}\in\mathscr{C}}\mathscr{L}\left(\{\boldsymbol{\gamma}_i\}\right), \tag{2.9}$$

where we define the constraint set

$$\mathscr{C}=\left\{\{\boldsymbol{\gamma}_i\}\middle|\boldsymbol{\gamma}_i\geq 0,i=1,2,\ldots,I,\|\boldsymbol{\gamma}_i\|_2=1,\forall i\neq I\right\}.$$

Algorithm 2 AM-KroSBL

Input: Measurements $\boldsymbol{y}$, matrix $\boldsymbol{H}$, noise power σ^2, threshold ϵ, ϵ_{AM}
Initialization: set $r=1$, $\{\boldsymbol{\gamma}_i\}^{(0)}=\mathbf{0}$, $\{\boldsymbol{\gamma}_i\}^{(1)}\in\mathscr{C}_+$
while $\|\otimes_{i=1}^{I}\boldsymbol{\gamma}_i^{(r)}-\otimes_{i=1}^{I}\boldsymbol{\gamma}_i^{(r-1)}\|_2>\epsilon$ **do**
  Compute $\boldsymbol{d}^{(r)}$ using (2.13) and (2.14)
  Set $t=1$, $\{\boldsymbol{\gamma}_i\}^{(r,1)}=\{\boldsymbol{\gamma}_i\}^{(r)}$, $\{\boldsymbol{\gamma}_i\}^{(r,0)}=\mathbf{0}$
  while $\|\otimes_{i=1}^{I}\boldsymbol{\gamma}_i^{(r,t)}-\otimes_{i=1}^{I}\boldsymbol{\gamma}_i^{(r,t-1)}\|_2>\epsilon_{\mathsf{AM}}$ **do**
    Compute $\{\boldsymbol{\gamma}_i\}^{(r,t+1)}$ using (2.16) and (2.17)
    Update AM iteration index $t\leftarrow t+1$
  end while
  $\{\boldsymbol{\gamma}_i\}^{(r+1)}=\{\boldsymbol{\gamma}_i\}^{(r,t)}$
  Update iteration index $r\leftarrow r+1$
end while
Output: $\boldsymbol{x}=\boldsymbol{\mu}_{\boldsymbol{x}}$ using (2.14)

The problem in (2.9) does not admit a closed-form solution. Thus, we resort to the standard EM algorithm [62, 69] for an iterative solution. Specifically, the rth iteration of EM is

$$\textbf{E-step: Compute } \mathbb{E}_{\boldsymbol{x}|\boldsymbol{y};\boldsymbol{\gamma}^{(r)}}\{\log[p(\boldsymbol{y},\boldsymbol{x};\{\boldsymbol{\gamma}_i\})]\}, \tag{2.10}$$

$$\textbf{M-step: } \{\boldsymbol{\gamma}_i\}^{(r+1)}=\underset{\{\boldsymbol{\gamma}_i\}\in\mathscr{C}_+}{\arg\max}\,\mathbb{E}_{\boldsymbol{x}|\boldsymbol{y};\boldsymbol{\gamma}^{(r)}}\{\log[p(\boldsymbol{y},\boldsymbol{x};\{\boldsymbol{\gamma}_i\})]\}, \tag{2.11}$$

where $\boldsymbol{\gamma}^{(r)}=\otimes_{i=1}^{I}\boldsymbol{\gamma}_i^{(r)}$ is the estimate in the rth iteration and

$$\mathscr{C}_+=\left\{\{\boldsymbol{\gamma}_i\}\in\mathscr{C}\,\middle|\,\boldsymbol{\gamma}_i>0,i=1,2,\ldots,I\right\}.$$

Here, we restrict the feasible set in (2.11) to $\mathscr{C}_+$ instead of $\mathscr{C}$ in (2.9) to avoid degenerate distributions. We also initialize the algorithm with $\{\boldsymbol{\gamma}_i\}^{(1)}\in\mathscr{C}_+$, for example, $\forall i=1,\cdots,I-1$, $\boldsymbol{\gamma}_i^{(1)}=\frac{1}{\sqrt{N}}\mathbf{1}$, and $\boldsymbol{\gamma}_I^{(1)}=\sqrt{N}^{I-1}\mathbf{1}$. Further, using straightforward algebraic simplifications [70], we can reduce (2.11) to

$$\{\boldsymbol{\gamma}_i\}^{(r+1)}=\underset{\{\boldsymbol{\gamma}_i\}\in\mathscr{C}_+}{\arg\min}\log|\mathrm{diag}(\boldsymbol{\gamma})|+(\boldsymbol{d}^{(r)})^{\mathsf{T}}\boldsymbol{\gamma}^{-1}, \tag{2.12}$$

where $(\cdot)^{-1}$ is the vector element-wise inversion, and we define

$$\boldsymbol{d}^{(r)}=\mathrm{diag}(\boldsymbol{\Sigma}_{\boldsymbol{x}}+\boldsymbol{\mu}_{\boldsymbol{x}}\boldsymbol{\mu}_{\boldsymbol{x}}^{\mathsf{H}}). \tag{2.13}$$

Here, $\boldsymbol{\mu}_{\boldsymbol{x}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{x}}$, which depend on $\boldsymbol{\gamma}^{(r)}$, are the mean and variance of conditional distribution $p(\boldsymbol{x}|\boldsymbol{y};\boldsymbol{\gamma}^{(r)})$, respectively,

$$\boldsymbol{\mu}_{\boldsymbol{x}}=\sigma^{-2}\boldsymbol{\Sigma}_{\boldsymbol{x}}\boldsymbol{H}^{\mathsf{H}}\boldsymbol{y},\ \boldsymbol{\Sigma}_{\boldsymbol{x}}=\left[\sigma^{-2}\boldsymbol{H}^{\mathsf{H}}\boldsymbol{H}+\mathrm{diag}(\boldsymbol{\gamma}^{(r)})^{-1}\right]^{-1}. \tag{2.14}$$

Here the dictionary matrix $\boldsymbol{H}$ is Kronecker-structured, i.e., $\boldsymbol{H} = \otimes_{i=1}^{I} \boldsymbol{H}_i$. This special structure can be used to reduce the computational complexity of computing $\boldsymbol{\Sigma}_{\boldsymbol{x}}$ in (2.14) [27]. Let $Q(\{\boldsymbol{\gamma}_i\}|\boldsymbol{\gamma}^{(r)}) = \log|\operatorname{diag}(\boldsymbol{\gamma})| + (\boldsymbol{d}^{(r)})^{\mathsf{T}}\boldsymbol{\gamma}^{-1}$, which depends on $\boldsymbol{\gamma}^{(r)}$ via $\boldsymbol{d}^{(r)}$. Then, the M-step is written as

$$\min_{\{\boldsymbol{\gamma}_i\}} Q(\{\boldsymbol{\gamma}_i\}|\boldsymbol{\gamma}^{(r)}) \quad \text{s.t. } \boldsymbol{\gamma} = \otimes_{i=1}^{I}\boldsymbol{\gamma}_i,\ \{\boldsymbol{\gamma}_i\} \in \mathscr{C}_+. \tag{2.15}$$

The solution to (2.15) without the constraint $\boldsymbol{\gamma} = \otimes_{i=1}^{I}\boldsymbol{\gamma}_i$ is straightforward [62]. However, since we adopt a Kronecker-structured prior (2.8), the Kronecker constraint poses a challenge to derive a closed-form solution. To solve (2.15) with the Kronecker constraint, two distinct ways are presented: AM-based and SVD-based approaches.

2.3.1. AM-Based KroSBL (AM-KroSBL)

AM-KroSBL solves (2.15) by alternatingly updating one hyperparameter vector while keeping the others fixed. We first compute the gradient of cost function[1] $Q(\{\boldsymbol{\gamma}_i\})$ with respect to $\{\boldsymbol{\gamma}_i\}$ and set it to zero, leading to

$$\tilde{\boldsymbol{\gamma}}_i = N^{-I+1}\big[(\otimes_{j=1}^{i-1}(\tilde{\boldsymbol{\gamma}}_j)^{-1}) \otimes \boldsymbol{I}_N \otimes (\otimes_{j=i+1}^{I}(\boldsymbol{\gamma}_j^{(r,t)})^{-1})\big]^{\mathsf{T}}\boldsymbol{d}^{(r)}, \tag{2.16}$$

for $i = 1,2,\ldots,I$, with $\boldsymbol{\gamma}_l^{(r,t)}$ is the estimate in the tth iteration of AM and the rth iteration of EM. To avoid the scaling ambiguity, in each iteration t, we project the hyperparameter vectors to $\mathscr{C}_+$ as

$$\boldsymbol{\gamma}_i^{(r,t+1)} = \begin{cases} \frac{\tilde{\boldsymbol{\gamma}}_i}{\|\tilde{\boldsymbol{\gamma}}_i\|_2} & \text{for } i = 1,2,\ldots,I-1 \\ \prod_{j=1}^{I-1}\|\tilde{\boldsymbol{\gamma}}_j\|_2\tilde{\boldsymbol{\gamma}}_I & \text{for } i = I. \end{cases} \tag{2.17}$$

The projection (2.17) does not change the cost function value Q as $\otimes_{i=1}^{I}\boldsymbol{\gamma}_i^{(r,t+1)} = \otimes_{i=1}^{I}\tilde{\boldsymbol{\gamma}}_i$. The steps are summarized in Algorithm 2. The AM-KroSBL is guaranteed to improve the cost function given by (2.12) after every iteration. However, due to its iterative nature, it is computationally inefficient. Thus, we next present a non-iterative method to solve (2.15) using SVD.

2.3.2. SVD-Based KroSBL (SVD-KroSBL)

This method solves (2.15) without the constraint $\boldsymbol{\gamma} = \otimes_{i=1}^{I}\boldsymbol{\gamma}_i$ first and then projects the solution to the constraint set. We note from [62] that

$$\operatorname*{argmin}_{\boldsymbol{\gamma}} \log|\operatorname{diag}(\boldsymbol{\gamma})| + (\boldsymbol{d}^{(r)})^{\mathsf{T}}\boldsymbol{\gamma}^{-1} = \boldsymbol{d}^{(r)}. \tag{2.18}$$

To project the above solution to the constraint set, we solve for $\{\boldsymbol{\gamma}_i\}$ that minimizes $\|\boldsymbol{d}^{(r)} - \otimes_{i=1}^{I}\boldsymbol{\gamma}_i\|_2$. We further approximate this optimization problem as $(I-1)$ rank-1 approximations:

$$\boldsymbol{\gamma}_i^{(r+1)} = \operatorname*{argmin}_{\boldsymbol{\gamma}_i:\|\boldsymbol{\gamma}_i\|_2=1, \bar{\boldsymbol{\gamma}}_i\in\mathbb{R}^{N(I-i)}} \|\bar{\boldsymbol{\gamma}}_{i-1} - \boldsymbol{\gamma}_i \otimes \bar{\boldsymbol{\gamma}}_i\|_2, \forall i = 1,2,\ldots,I-1, \tag{2.19}$$

[1] We interchangeably use $Q(\{\boldsymbol{\gamma}_i\}|\boldsymbol{\gamma}^{(r)})$, $Q(\{\boldsymbol{\gamma}_i\})$, and Q in this chapter.

Algorithm 3 SVD-KroSBL

Input: Measurements $\boldsymbol{y}$, matrix $\boldsymbol{H}$, noise power σ^2, threshold ϵ
Initialization: set $r=1$, $\{\boldsymbol{\gamma}_i\}^{(0)}=\mathbf{0}$, $\{\boldsymbol{\gamma}_i\}^{(1)}\in\mathscr{C}_+$
while $\|\otimes_{i=1}^{I}\boldsymbol{\gamma}_i^{(r)}-\otimes_{i=1}^{I}\boldsymbol{\gamma}_i^{(r-1)}\|_2>\epsilon$ **do**
  Compute $\boldsymbol{d}^{(r)}$ using (2.13) and (2.14)
  for $i=1,\ldots,I-1$ **do**
    Solve (2.19) for $\boldsymbol{\gamma}_i^{(r+1)}$
  end for
  Update iteration index: $r\leftarrow r+1$
end while
Output: $\boldsymbol{x}=\boldsymbol{\mu}_{\boldsymbol{x}}$ using (2.14)

where $\bar{\boldsymbol{\gamma}}_0=\boldsymbol{d}^{(r)}$ and $\bar{\boldsymbol{\gamma}}_{I-1}=\boldsymbol{\gamma}_I$. Since $\boldsymbol{\gamma}_i\otimes\bar{\boldsymbol{\gamma}}_i=\mathrm{vec}(\bar{\boldsymbol{\gamma}}_i\boldsymbol{\gamma}_i^{\mathsf{T}})$, we rearrange $\bar{\boldsymbol{\gamma}}_{i-1}\in\mathbb{R}^{N(I-i+1)\times 1}$ into a matrix $\boldsymbol{D}^{(i)}\in\mathbb{R}^{N(I-i)\times N}$ such that $\mathrm{vec}(\boldsymbol{D}^{(i)})=\bar{\boldsymbol{\gamma}}_{i-1}$. Thus, we rewrite (2.19) as

$$\boldsymbol{\gamma}_i^{(r+1)}=\underset{\boldsymbol{\gamma}_i:\|\boldsymbol{\gamma}_i\|_2=1,\bar{\boldsymbol{\gamma}}_i\in\mathbb{R}^{N(I-1)}}{\arg\min}\|\boldsymbol{D}^{(i)}-\bar{\boldsymbol{\gamma}}_i\boldsymbol{\gamma}_i^{\mathsf{T}}\|_{\mathrm{F}}. \tag{2.20}$$

As matrix $\bar{\boldsymbol{\gamma}}_i\boldsymbol{\gamma}_i^{\mathsf{T}}$ is a rank-1 matrix, solving problem (2.20) is equivalent to finding the rank-1 approximation of $\boldsymbol{D}^{(i)}$. The optimal solution to $\boldsymbol{\gamma}_i^{(r+1)}$ is the first right singular vector of $\boldsymbol{D}^{(i)}$, satisfying constraint $\|\boldsymbol{\gamma}_1\|_2=1$. So, the above problem can be solved using the SVD algorithm. This approach is summarized in Algorithm 3.

2.3.3. Noise Variance Estimation

The derivations so far assume the knowledge of the noise variance. However, in practical scenarios, the noise variance is not always provided and needs to be estimated. To compute the noise variance, we have added the noise variance as a hyperparameter of the EM algorithm and derived the corresponding update equations. The resulting M-step optimization problem is separable in $\boldsymbol{\gamma}$ and $\boldsymbol{\sigma}^2$, and it leads to an additional noise variance step in the EM iteration without changing the update equation for $\boldsymbol{\gamma}$. Specifically, in the rth EM iteration, we have

$$(\sigma^2)^{(r+1)}=\frac{\|\boldsymbol{y}-\boldsymbol{H}\boldsymbol{\mu}_{\boldsymbol{x}}\|_2^2+(\sigma^2)^{(r)}\left[N^I-\mathrm{tr}\left(\boldsymbol{\Sigma}_{\boldsymbol{x}}\,\mathrm{diag}(\boldsymbol{\gamma}^{(r)})^{-1}\right)\right]}{\bar{M}}, \tag{2.21}$$

and we replace the known noise variance σ^2 in (2.14) with the estimated value $(\sigma^2)^{(r)}$.

2.4. Theoretical Analysis of KroSBL

This section focuses on the theoretical analysis of KroSBL. We discuss the convergence guarantee for AM-KroSBL in Section 2.4.1. Then, we present results on the values to which the algorithm converges by studying the local minima of the KroSBL cost function (2.8).

2.4.1. Convergence Property of AM-KroSBL

The convergence of AM-KroSBL is established using the properties of the EM algorithm, which is well studied in [71]. It is known that under certain conditions, the EM algorithm guarantees convergence to stationary points of $\mathscr{L}$. Nonetheless, the guarantees of the EM algorithm in KroSBL depend on the convergence of the AM algorithm (inner loop). So, this section answers two questions: *What are the convergence properties of the AM algorithm? Do the properties of AM guarantee the convergence of AM-KroSBL?* The first question is answered by Proposition 1, serving as a cornerstone to the answer to the second question via Theorem 1. We first introduce a lemma that supports the main results.

Lemma 1. *Consider the AM algorithm that solves the M-step optimization problem of the r th EM iteration of AM-KroSBL (Algorithm 2) given by (2.15), for a fixed iteration index $r > 0$. If $\boldsymbol{d}^{(r)} > 0$ in (2.13), then the cost function sequence $Q(\{\boldsymbol{\gamma}_i\}^{(r,t)})|_{t=1}^{\infty}$ generated by the AM algorithm is non-increasing and convergent.*

Proof. The non-increasing nature of sequence $Q(\{\boldsymbol{\gamma}_i\}^{(r,t)})|_{t=1}^{\infty}$ is because the AM algorithm in every iteration optimizes one hyperparameter vector while keeping the others fixed

$$Q(\{\boldsymbol{\gamma}_i\}^{(r,t)}) \geq Q\left(\tilde{\boldsymbol{\gamma}}_1, \{\boldsymbol{\gamma}_i^{(r,t)}\}_{i=2}^{I}\right) \geq Q\left(\{\tilde{\boldsymbol{\gamma}}_i\}_{i=1}^{2}, \{\boldsymbol{\gamma}_i^{(r,t)}\}_{i=3}^{I}\right) \geq Q(\{\tilde{\boldsymbol{\gamma}}_i\}) = Q(\{\boldsymbol{\gamma}_i\}^{(r,t+1)}), \quad (2.22)$$

where the last step follows because the projection step in (2.17) does not change the cost function value. Further, when $\boldsymbol{d}^{(r)} > 0$, we have

$$Q(\{\boldsymbol{\gamma}_i\}) = \sum_{j=1}^{N^I} \log[\boldsymbol{\gamma}]_j + [\boldsymbol{d}^{(r)}]_j [\boldsymbol{\gamma}]_j^{-1} \geq \sum_{j=1}^{N^I} \log[\boldsymbol{d}^{(r)}]_j + N^I.$$

The non-increasing sequence $Q(\{\boldsymbol{\gamma}_i\}^{(r,t)})|_{t=1}^{\infty}$ is bounded from below and thus convergent. □

Although the sequence of the cost function $Q(\{\boldsymbol{\gamma}_i\}^{(r,t)})|_{t=1}^{\infty}$ converges, this does not indicate any convergence property of its iterates, i.e., $\{\boldsymbol{\gamma}_i\}^{(r,t)}|_{t=1}^{\infty}$ [72]. The following proposition uses the above lemma to show that the iterates of the AM algorithm converge.

Proposition 1. *[AM algorithm convergence] Consider the AM algorithm that solves the M-step optimization problem of the r th EM iteration of AM-KroSBL (Algorithm 2) given by (2.15), for a fixed iteration index $r > 0$. If $\boldsymbol{d}^{(r)} > 0$ in (2.13), then the sequence $\{\boldsymbol{\gamma}_i\}^{(r,t)}|_{t=1}^{\infty}$ converges to the set of stationary points of the M-step cost function $Q(\{\boldsymbol{\gamma}_i\})$ in $\mathscr{C}_+$.*

Proof. See Appendix A.1. □

It is interesting to note that the projection step (2.17) is coupled for $i = I$, because $\boldsymbol{\gamma}_I = \prod_{j=1}^{I-1} \|\tilde{\boldsymbol{\gamma}}_j\|_2 \tilde{\boldsymbol{\gamma}}_I$ involves computing the ℓ_2 norm of $\tilde{\boldsymbol{\gamma}}_i$ for $\forall\ i = 1, \cdots, I-1$. However, this coupled projection step does not affect the convergence of the sequence $\{\boldsymbol{\gamma}_i\}^{(r,t)}|_{t=1}^{\infty}$. The reason is that according to Zangwill's convergence theorem [73] used to prove our results, regardless of how to produce the sequence $\{\boldsymbol{\gamma}_i\}^{(r,t)}|_{t=1}^{\infty}$, if $\{\boldsymbol{\gamma}_i\}^{(r,t)}|_{t=1}^{\infty}$ satisfies the required properties (see Appendix A.1), the sequence converges to the stationary point. The AM procedure with projection is a way of producing such a sequence and possesses

the required properties. Nonetheless, we note that Proposition 1 only guarantees that the AM algorithm converges to a stationary point, which is not necessarily a global minimum. However, the following result establishes that the convergence of AM to a stationary point is sufficient to ensure the convergence of AM-KroSBL.

Theorem 1. *Consider the model in (1.2) with the assumptions i) the noise variance $\sigma^2 > 0$, ii) there exists $\epsilon > 0$ such that the dictionary satisfies $\|[\boldsymbol{H}]_i\|_2 > \epsilon$, for $i = 1, 2, \ldots, N^I$, and iii) the starting point of AM-KroSBL $\{\boldsymbol{\gamma}_i\}^{(1)} \in \mathscr{C}_+$. Then, the sequence $\{\boldsymbol{\gamma}_i\}^{(r)}|_{r=1}^{\infty}$ generated by AM-KroSBL (Algorithm 2) convergence to the set of the stationary points of its cost function $\mathscr{L}$ given by (2.8).*

Proof. See Appendix A.2. □

We note that the assumptions of Theorem 1 are realistic. Regarding assumption *i*), noise cannot be avoided in practice. It is also common to assume $\sigma^2 > 0$ in the SBL analysis since it is difficult to discuss the behavior of EM-based SBL in the noiseless case [72, 74]. However, the noiseless case can be seen as a limiting case of $\sigma^2 \to 0$. Intuitively, we can set σ^2 as a very small but nonzero value and drive this value arbitrarily to zero. Thus, asymptotically, the convergence property should also hold in the noiseless case. The assumption *ii*) on the dictionary holds when $\boldsymbol{H}$ has no zero columns. If the norm of a column in $\boldsymbol{H}$ is zero, indicating that all its elements are zero, then that column does not contribute to the measurement and can be removed.

Assumption *iii*) the starting point of AM-KroSBL $\{\boldsymbol{\gamma}_i\}^{(1)} \in \mathscr{C}_+$ is also easy to satisfy. It can be generated by projecting any $\{\boldsymbol{\gamma}_i\} > 0$ using (2.17). Furthermore, it is necessary to have $\{\boldsymbol{\gamma}_i\}^{(1)} \in \mathscr{C}_+$ since the starting point should be feasible, and more importantly, $\{\boldsymbol{\gamma}_i\}^{(1)} > 0$. Suppose that we initialize the n^*th entry of $\boldsymbol{\gamma}_{i^*}^{(1)}$ with zero for some i^* and n^*, then then all the N^{I-1} entries in $\boldsymbol{\gamma}^{(1)} = \otimes_{i=1}^{I} \boldsymbol{\gamma}_i^{(1)}$ involving $[\boldsymbol{\gamma}_{i^*}]_{n^*}$ are zeros. Let $\mathscr{M}$ be the set of indices in $\otimes_{i=1}^{I} \boldsymbol{\gamma}_i^{(1)}$ being zero. Then, the submatrix $[\boldsymbol{\Sigma}_{\boldsymbol{x}}]_{\mathscr{M}}$ is zero due to (2.14),

$$\boldsymbol{\Sigma}_{\boldsymbol{x}} = \boldsymbol{\Gamma}^{(1)} - \boldsymbol{\Gamma}^{(1)} \boldsymbol{H}^{\mathsf{H}} \left(\sigma^2 \boldsymbol{I}_{N^I} + \boldsymbol{H} \boldsymbol{\Gamma}^{(1)} \boldsymbol{H}^{\mathsf{H}} \right)^{-1} \boldsymbol{H} \boldsymbol{\Gamma}^{(1)}, \tag{2.23}$$

where the rows of $\boldsymbol{\Gamma}^{(1)} = \mathrm{diag}(\otimes_{i=1}^{I} \boldsymbol{\gamma}_i^{(1)})$ indexed by $\mathscr{M}$ are zero. Consequently, $[\boldsymbol{d}^{(1)}]_{\mathscr{M}}$ is also zero due to the following relation from (2.13) and (2.14),

$$\boldsymbol{d}^{(1)} = \mathrm{diag}\left(\boldsymbol{\Sigma}_{\boldsymbol{x}} \left[\boldsymbol{I}_{N^I} + \sigma^{-4} \boldsymbol{H}^{\mathsf{H}} \boldsymbol{y} \boldsymbol{y}^{\mathsf{H}} \boldsymbol{H} \boldsymbol{\Sigma}_{\boldsymbol{x}} \right] \right). \tag{2.24}$$

Further, if $[\boldsymbol{d}^{(1)}]_{\mathscr{M}}$ is zero, then the second term in Q of (2.12) involving $[\boldsymbol{\gamma}_{i^*}]_{n^*}$ vanishes, making the optimization problem (2.15) separable in $\boldsymbol{\gamma}_{i^*}$ and leading to $\arg\min \log[\boldsymbol{\gamma}_{i^*}]_{n^*}$. This optimization drives $[\boldsymbol{\gamma}_{i^*}^{(2)}]_{n^*} = 0$. This indicates that the entries of the hyperparameter $\{\boldsymbol{\gamma}_i\}^{(r)}$ within the index set $\mathscr{M}$ will remain zero throughout the EM iteration for $r = 1, 2, \cdots$. Therefore, initializing $\{\boldsymbol{\gamma}_i\}^{(1)}$ with zero intrinsically assumes that the subvector $[\boldsymbol{x}]_{\mathscr{M}}$ cannot be non-zero from the beginning since the associated hyperparameters $[\boldsymbol{\gamma}]_{\mathscr{M}}$ remain zero by initialization, which can be erroneous. Further, initialization with $\{\boldsymbol{\gamma}_i\}^{(1)} > 0$ is common in the EM-based SBL algorithm [72, 75], since the M-step naturally involves element-wise inversion of the hyperparameter $\boldsymbol{\gamma}$. Hence, we assume that $\{\boldsymbol{\gamma}_i\}^{(1)} \in \mathscr{C}_+$.

Furthermore, Proposition 1 suggests that $\{\boldsymbol{\gamma}_i\}^{(r)} \in \mathscr{C}_+$ and thus $\{\boldsymbol{\gamma}_i\}^{(r)} > 0$, which seems to contradict the expected sparsity of the estimates $\{\boldsymbol{\gamma}_i\}^{(r)}$. However, $\{\boldsymbol{\gamma}_i\}^{(r)} > 0$ only holds under the assumption $\boldsymbol{d}^{(r)} > 0$ and the sequence $\{\boldsymbol{d}^{(r)}\}_{r=1}^{\infty}$ belongs to an open set $\{\boldsymbol{d} | \boldsymbol{d} > 0\}$. From our experiments, we observe that the sequence converges to $\boldsymbol{d}^{(\infty)}$ that belongs to the boundary of the open set, leading to sparse $\{\boldsymbol{\gamma}_i\} \in \mathscr{C} \setminus \mathscr{C}_+$. Intuitively, this behavior can be viewed as follows. If the n^*th entry of $\boldsymbol{\gamma}_{i^*}^{(r)}$ goes to zero for some i^* and n^*, then, the submatrix $[\boldsymbol{\Sigma}_{\boldsymbol{x}}]_{\mathscr{M}}$ goes to zero because of (2.23) and (2.24) in the rth EM iteration. Conversely, suppose that $[\boldsymbol{d}^{(r)}]_{\mathscr{M}}$ goes to zero, then, the second term in Q of (2.12) involving $[\boldsymbol{\gamma}_{i^*}]_{n^*}$ becomes $\log[\boldsymbol{\gamma}_{i^*}]_{n^*}$ and drives $[\boldsymbol{\gamma}_{i^*}^{(r)}]_{n^*}$ to zero. Thus, a sparse $\boldsymbol{d}^{(r)}$ encourages a sparse $\boldsymbol{\gamma}^{(r)}$ and vice versa, leading to a sparse convergent point $\{\boldsymbol{\gamma}_i\}^{(\infty)} \in \mathscr{C} \setminus \mathscr{C}_+$.

2.4.2. Local Minima of KroSBL Cost Function

Having studied the convergence properties of the algorithm, we now look at the properties of the limit points. Unlike the previous section, the results of this section assume that the sparse vector is also Kronecker-structured, i.e., $\boldsymbol{x} = \otimes_{i=1}^{I} \boldsymbol{x}_i$. The first result of the section, Theorem 2, proves that all local minima $\{\boldsymbol{\gamma}_i\}$ of the KroSBL cost function $\mathscr{L}$ in (2.8) are sparse. Subsequently, in Theorem 3, we derive an upper bound on the number of local minima of the KroSBL cost function.

Theorem 2. *In the noiseless setting, every local minimum of $\mathscr{L}$ is achieved at a sparse solution $\{\boldsymbol{\gamma}_i\}$, that is, $\|\boldsymbol{\gamma}_i\|_0 \leq M_i$ for $i = 1, 2, \cdots, I$, if the sparse vector is Kronecker-structured, i.e., $\boldsymbol{x} = \otimes_{i=1}^{I} \boldsymbol{x}_i$, for some $\boldsymbol{x}_i \in \mathbb{C}^N$.*

Proof. See Appendix A.3. □

Theorem 2 indicates that the local minimum is sparse, not because some hyper-parameter vectors ($\boldsymbol{\gamma}_i$'s) are dense while others are sparse, leading to a sparse Kronecker product. Instead, it implies each $\boldsymbol{\gamma}_i$ generated by KroSBL is sparse, following our Kronecker-structured support model.

Now we discuss an upper bound for the number of local minima of $\mathscr{L}$. For this, we use the concept of URP. The matrix $\boldsymbol{H}$ is said to satisfy URP if any subset of $\bar{M}$ columns of $\boldsymbol{H}$ is linearly independent [62]. If the dictionary $\boldsymbol{H}$ satisfies the URP, the number of local minima of $\mathscr{L}$ in (2.8) without the Kronecker-structured support constraint $\mathscr{C}$ (the classic SBL algorithm) is [62]

$$\mathscr{N}_{\text{SBL}} \leq \binom{N^I}{\bar{M}} - \sum_{p=1}^{P} \binom{N^I - D_p}{\bar{M} - D_p} + P \leq \binom{N^I}{\bar{M}},$$

where D_p is ℓ_0-norm of the pth degenerate sparse solution of the SBL cost function, and P is the number of sparse solution. When we impose the Kronecker-structured support constraint, the upper bound of the number of local minima decreases, as discussed next.

Theorem 3. *Consider the model in (1.2) and assume that i) the noise variance $\sigma^2 = 0$, ii) $\boldsymbol{H}$ satisfies the URP, and iii) there exist P_i degenerate sparse solutions $\boldsymbol{x}_{i,p}$, $p = 1, 2, \ldots, P_i$ such*

that $\boldsymbol{y} = (\otimes_{i=1}^{I} \boldsymbol{H}_i)(\otimes_{i=1}^{I} \boldsymbol{x}_{i,p})$ *and* $\|\boldsymbol{x}_{i,p}\|_0 = D_{i,p} < M_i$*. Then, the number of distinct local minima of the KroSBL cost* $\mathscr{L}$ *in* $\mathscr{C}$*, denoted as* $\mathscr{N}$*, satisfies*

$$\mathscr{N} \leq \prod_{i=1}^{I} \left(\binom{N}{M_i} - \sum_{p=1}^{P_i} \binom{N - D_{i,p}}{M_i - D_{i,p}} + P_i \right) \leq \prod_{i=1}^{I} \binom{N}{M_i}. \tag{2.25}$$

Proof. See Appendix A.4. □

Theorem 3 and the result for classical SBL use the same assumption that $\boldsymbol{H}$ satisfies URP to derive an upper bound for the distinct number of local minima. However, our result shows that $\mathscr{N}$ is dominated by $\prod_{i=1}^{I} \binom{N}{M_i}$ while $\mathscr{N}_{\text{SBL}}$ is dominated by $\binom{N^I}{M} > \prod_{i=1}^{I} \binom{N}{M_i}$. Thus, incorporating the Kronecker structure can greatly diminish the solution space, explaining the better reconstruction performance of the KroSBL.

2.5. Performance Evaluation

We conduct numerical experiments to investigate the efficacy of our algorithms for general sparse vector recovery. We evaluate the recovery performance of AM-KroSBL and SVD-KroSBL by comparing them with three benchmarking algorithms: the classic SBL (cSBL) [62], KSBL [31], and KOMP [18].

We choose $I = 3$ i.e., $\boldsymbol{H} = \otimes_{i=1}^{3} \boldsymbol{H}_i$ and the sparse vector $\boldsymbol{x} = \otimes_{i=1}^{3} \boldsymbol{x}_i$ where $\boldsymbol{x}_i \in \mathbb{R}^{15\times 1}$. The entries of $\boldsymbol{x}_i \in \mathbb{R}^{15\times 1}$, $\boldsymbol{H}_1 \in \mathbb{R}^{M\times 15}$, $\boldsymbol{H}_2 \in \mathbb{R}^{12\times 15}$, and $\boldsymbol{H}_3 \in \mathbb{R}^{15\times 15}$ are drawn from $\mathrm{N}(0,1)$, where $M = \{6,8,10,12,14\}$, called measurement level, controls the total number of measurements $\bar{M} = 180M$ (or the under-sampling ratio $\bar{M}/N^I = 180M/3375$). The sparsity level for each $\boldsymbol{x}_i$ is $S = \{2,3,4,5,6\}$, and the support is generated uniformly at random. The zero-mean additive white Gaussian measurement noise level is decided by the signal-to-noise ratio SNR (dB) $= 10\log_{10} \mathbb{E}\{\|\boldsymbol{H}\boldsymbol{x}\|_2^2 / \|\boldsymbol{n}\|_2^2\}$ and takes values from $\{5,10,15,20,25,30\}$. We use three metrics for the assessment: NMSE, support recovery rate (SRR), and runtime. Here, we define

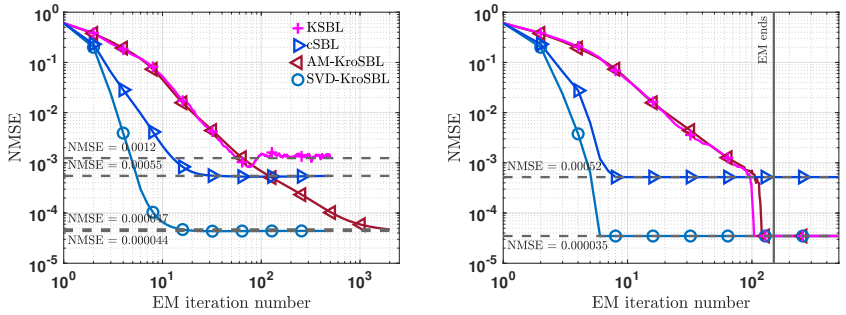
$$\text{NMSE} = \mathbb{E}\left\{ \frac{\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2^2}{\|\boldsymbol{x}\|_2^2} \right\}, \tag{2.26}$$

$$\text{SRR} = \frac{|\text{supp}(\hat{\boldsymbol{x}}) \cap \text{supp}(\boldsymbol{x})|}{|\text{supp}(\hat{\boldsymbol{x}}) \cup \text{supp}(\boldsymbol{x})|}, \tag{2.27}$$

where $\boldsymbol{x}$ is the ground truth and $\hat{\boldsymbol{x}}$ is the reconstructed vector. We cap EM iteration to 150 for the SBL-based methods. We also implement the complexity reduction technique for AM- and SVD-KroSBL and the technique in [31] for KSBL. The pruning is also included to prune small hyperparameters for all SBL-based methods [76]. We average over two hundred independent realizations. Our observations from the results, summarized in Figures 2.1 to 2.8 and Tables 2.1 and 2.3, are as follows.

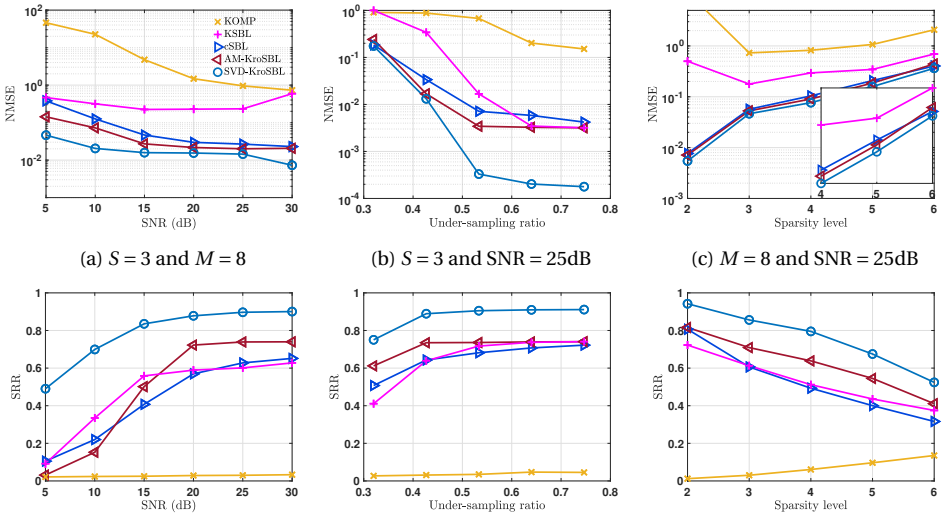
2.5.1. Convergence Illustration

Figure 2.1 demonstrates the convergence property of cSBL, KSBL, AM-, and SVD-KroSBL *with and without pruning*. We include the convergence without pruning because our



(a) NMSE without pruning step (b) NMSE with pruning step

Figure 2.1.: Convergence plots of AM-, SVD-KroSBL, KSBL, and cSBL with measurement level $M = 14$, sparsity $S = 4$, and SNR = 30dB.



(a) $S = 3$ and $M = 8$ (b) $S = 3$ and SNR = 25dB (c) $M = 8$ and SNR = 25dB

(d) $S = 3$ and $M = 8$ (e) $S = 3$ and SNR = 25dB (f) $M = 8$ and SNR = 25dB

Figure 2.2.: NMSE and SRR performance of different algorithms as functions of SNR, under-sampling ratio, and sparsity level S. Unless otherwise mentioned in the plots, measurement level $M = 8$, sparsity level $S = 3$, and SNR = 25 dB.

theoretical analysis does not account for pruning. First, we look at Figure 2.1. We note that our AM-KroSBL and SVD-KroSBL lead to lower NMSE because they incorporate the Kronecker structure, avoiding unwanted local minimum compared with cSBL. However, KSBL, despite incorporating the same prior knowledge, results in a higher NMSE than AM- and SVD-KroSBL. Interestingly, KSBL initially has a similar NMSE as AM-KroSBL, as both schemes solve (2.15) using the gradient-based iterative method. However, KSBL is trapped in a local minimum after a few iterations because it uses a loose approximation and does not constrain its hyperparameters $\{\boldsymbol{\gamma}_i\}$ in $\mathscr{C}$, i.e., does not normalize the hyperparameters. Our experiments show that the entries of some $\boldsymbol{\gamma}_i$'s of KSBL upon convergence become

very small while others become very large. Although the Kronecker products of $\{\boldsymbol{\gamma}_i\}$ in both algorithms are the same initially, small $\boldsymbol{\gamma}_i$ values destabilize KSBL. This numerical instability reduces the estimation accuracy and leads to a local minimum with high NMSE. To mitigate this issue, pruning small components is useful, helping KSBL to converge numerically, as shown in Figure 2.1(b). Pruning also accelerates other schemes. However, NMSE performance is sensitive to the pruning threshold, which is only empirically chosen. A larger threshold leads to faster convergence but is at the risk of eliminating true components.

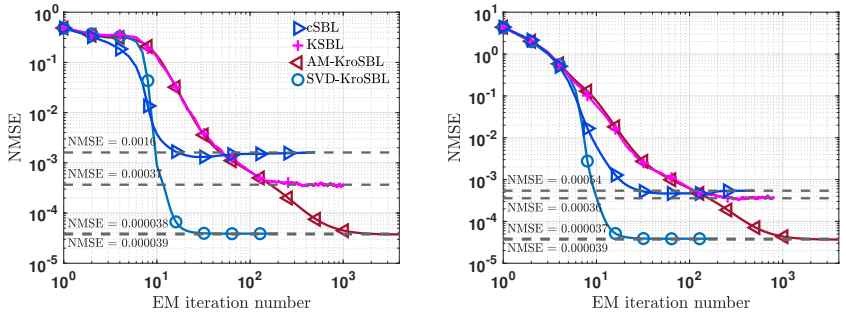
2.5.2. Comparison With the State-of-the-art

Figures 2.2(a) and 2.2(b) show that the performance of all the algorithms improves with SNR and the number of measurements $\bar{M} = 180M$ (or equivalently under-sampling ratio $\bar{M}/N^I = 180M/3375$), except for KSBL in Figure 2.2(a) due to over-pruning. Similarly, Figure 2.2(c) shows that increasing the sparsity level hinders the reconstruction of all schemes, as the number of measurements is unchanged. Further, our SVD-KroSBL outperforms all the other algorithms, both in terms of NMSE and SRR. The next best-performing algorithm is AM-KroSBL in most cases. The exception is in Figure 2.2(d), where KSBL and cSBL have higher SRR than the AM-based when the SNR is low. This behavior is because the pruning step of the AM-based fails to eliminate small components outside the true support. However, the nonzero entries are recovered accurately, indicated by low NMSEs. Finally, KSBL is expected to perform better than cSBL. However, this is not true in the low-measurement regime due to its approximations, seen in Figure 2.2(b).

Table 2.1 compares the runtime of different schemes. KOMP has the lowest runtime due to its greedy nature but suffers from high NMSE and low SRR, as shown in Figure 2.2. SVD-KroSBL has the lowest runtime among the remaining algorithms. It is faster as it is non-iterative and takes fewer EM iterations to converge (see Figure 2.1), and it also uses the complexity reduction technique [27] in the E-step. We also observe that AM-KroSBL outperforms the cSBL and has a runtime similar to KSBL. Although cSBL takes fewer EM iterations, each EM iteration of AM-KroSBL is faster than cSBL due to the complexity reduction [27]. Further, unlike KSBL and cSBL, AM-KroSBL has an iterative M-step, but the inner loop converges quickly, making it faster than KSBL.

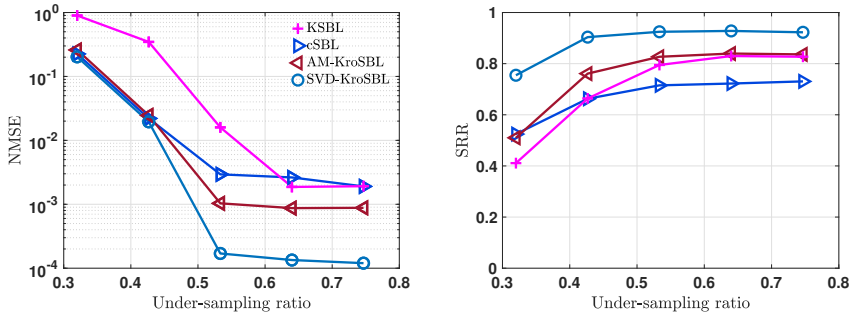
2.5.3. Recovery of Sparse Vector Without Kronecker-structured Non-zero Entries

In this section, we present simulation results on recovering sparse vectors with Kronecker-structured support but not Kronecker-structured nonzero values. To this end, we first construct $I = 3$ binary support vectors $\boldsymbol{b}_i$, $i = 1,2,3$, containing $S = 4$ ones, and obtain the support vector of $\boldsymbol{x}$ as $\boldsymbol{b} = \otimes_{i=1}^{I} \boldsymbol{b}_i$. The nonzero entries of $\boldsymbol{x}$ are drawn from a normal distribution $\mathrm{N}(0, 0.05)$. The results are summarized in Figure 2.3, showing the results for the cases of known and unknown noise variance. Comparing Figure 2.3 with Figure 2.1, we can see that both our AM- and SVD-KroSBL work well with lower reconstruction errors, indicating that our algorithms still converge to a better local minimum than the other competing algorithms when recovering sparse vector $\boldsymbol{x}$ with Kronecker-structured support.



(a) Unknown noise variance (b) Known noise variance

Figure 2.3.: Convergence performance of AM-, SVD-KroSBL, KSBL, and cSBL when the nonzero entries of sparse vector are not Kronecker-structured with measurement level $M = 14$, sparsity $S = 4$, and SNR = 30dB.



(a) NMSE performance (b) SRR performance

Figure 2.4.: NMSE and SRR performance of different algorithms as functions of under-sampling ratio when noise variance is unknown with sparsity level $S = 3$ and SNR = 25 dB.

2.5.4. Extension to Unknown Noise Variance Case

To evaluate our KroSBL algorithms when the noise variance is unknown, we incorporate the noise variance estimation technique (2.21) in the M-step. The simulation results are shown in Figure 2.4. We do not include KOMP since Figure 2.2 has already illustrated its poor performance. We observe that with more measurements, all algorithms can better estimate $\boldsymbol{x}$. Compared to Figs. 2.2(b) and 2.2(e), not providing the noise variance does not noticeably affect the overall NMSE and SRR performance of cSBL, AM-, and SVD-KroSBL. However, the performance of KSBL degrades both in NMSE and SRR, possibly due to the approximation adopted in its derivation.

Interestingly, we observe that AM-KroSBL yields better NMSE and SRR performance with the noise variance estimation step. The lower estimation error can be caused because the optimal value of the noise variance, which corresponds to the best performance, does not necessarily have to be the true noise variance. A similar trend was as also observed in [76], demonstrating that the true noise variance can possibly lead to suboptimal recovery performance. Furthermore, different SBL algorithms can have different optimal noise

variance values, so changes for SVD-KroSBL and cSBL have different magnitudes [76].

We note that noise estimation slightly increases the runtime of SVD-KroSBL. This behavior is intuitive, as the noise variance estimation is computationally light. Moreover, from Figure 2.3(a), we observe that for the unknown noise variance case, the NMSE curve initially remains flat before gradually decreasing. This behavior is more pronounced for AM-KroSBL than for SVD-KroSBL, leading to a double computation time for AM-KroSBL when the noise variance is unprovided.

Table 2.1.: Computation time (in seconds) comparison with SNR = 25dB and measurement level $M = 8$ (underlined text for the best and boldface for **the second best**)

Sparsity level	$S=2$	$S=3$	$S=4$	$S=5$	$S=6$
AM-KroSBL	8.379	7.181	6.550	6.224	5.944
SVD-KroSBL	**2.164**	**1.759**	**1.605**	**1.557**	**1.691**
KSBL	11.318	10.313	9.669	9.041	8.659
cSBL	15.574	14.077	14.385	15.891	19.738
KOMP	*0.745*	*0.785*	*0.787*	*0.782*	*0.777*

2.5.5. Comparison of SVD-KroSBL and AM-KroSBL

We observe from Figure 2.2 that the SVD-KroSBL algorithm outperforms AM-KroSBL and cSBL with sufficient measurements regardless of its approximations. Here, we give the intuition behind the better performance of the SVD-KroSBL. In KroSBL, the M-step solves (2.15). SVD-KroSBL approximates (2.15) by first identifying (2.18) and then solving (2.19). Suppose $\boldsymbol{d}^{(r)} = \otimes_{i=1}^{I} \boldsymbol{d}_i^{(r)}$ for some nonnegative vectors $\boldsymbol{d}_i^{(r)} \in \mathbb{R}^{N\times 1}$. Then the optimal solution to (2.15) is attained at $\check{\boldsymbol{\gamma}}_i = \boldsymbol{d}_i^{(r)}/\|\boldsymbol{d}_i^{(r)}\|$ for $i = 1,2,\ldots,I-1$, and $\check{\boldsymbol{\gamma}}_I = \prod_{i=1}^{I-1} \|\boldsymbol{d}_i^{(r)}\| \boldsymbol{d}_I^{(r)}$. Since (2.18) attains the optimal value at $\check{\boldsymbol{\gamma}} = \boldsymbol{d}^{(r)}$ that satisfies the constraints of (2.15), $\check{\boldsymbol{\gamma}} = \boldsymbol{d}^{(r)}$ is also optimal for (2.15). Hence, the SVD-based is exact when $\boldsymbol{d}^{(r)}$ is Kronecker-structured. Further, we empirically verified the rank of rearranged/devectorized $\boldsymbol{d}^{(r)}$ when $I = 3$, where rank = 1 indicates $\boldsymbol{d}^{(r)} = \otimes_{i=1}^{3} \boldsymbol{d}_i^{(r)}$. Without noise, the SVD-based method drives $\boldsymbol{d}^{(r)}$ to this structure within a few EM iterations and retains this structure, as shown in Figure 2.5. Therefore, after a few EM iterations, the SVD-based method solves the M-step exactly and outperforms the iterative first-order optimization in the AM-KroSBL algorithm. Furthermore, Figure 2.5 also indicates that SVD-based converges to a Kronecker-structured $\boldsymbol{d}^{(r)}$ faster than AM-KroSBL and cSBL. Thus, SVD can be viewed as a stronger imposition of the Kronecker structure, achieving faster EM iterations than the AM-based algorithm.

2.6. Application and Extension

In this section, we discuss the application of our algorithm to channel estimation in an IRS-assisted MIMO system. In the first, we present the system model and the channel estimation problem. Then, we discuss how to apply KroSBL to solve the problem. After that, we discuss an extension of KroSBL to handle block sparsity. In the end, we motivate the block sparsity in the considered IRS-assisted MIMO channel model, and apply the block sparsity extension of KroSBL to the channel estimation problem.

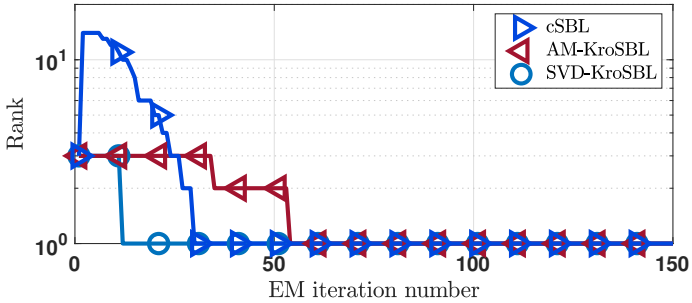


Figure 2.5.: The rank of reorganized $\boldsymbol{d}^{(r)}$ generated by different schemes in the noiseless setting with measurement level $M = 14$ and sparsity level $S = 4$.

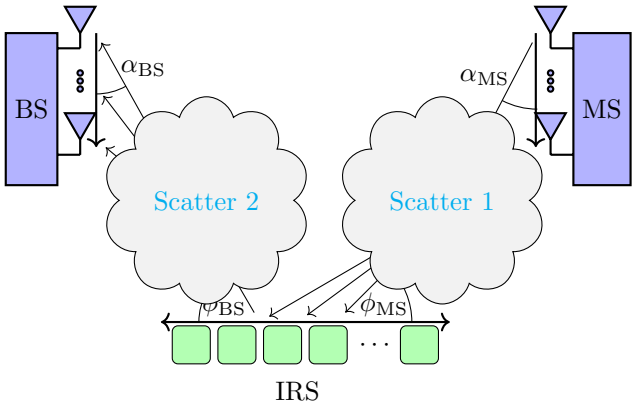


Figure 2.6.: An illustration of AoAs and AoDs in an IRS-aided channel.

2.6.1. Cascaded Channel Estimation for IRS-aided MIMO

Consider an uplink MIMO millimeter-wave/terahertz band system as shown in Figure 2.6, with a T-antenna transmitter MS, an R-antenna receiver BS, and an L-element uniform linear array IRS. Let $\boldsymbol{\Phi}_{\mathrm{MS}} \in \mathbb{C}^{L\times T}$ and $\boldsymbol{\Phi}_{\mathrm{BS}} \in \mathbb{C}^{R\times L}$ denote the MS-IRS and IRS-BS (narrowband) channels, respectively,

$$\boldsymbol{\Phi}_{\mathrm{MS}} = \left(\sum_{p=1}^{P_{\mathrm{MS}}} \sqrt{\frac{LT}{P_{\mathrm{MS}}}}\beta_{\mathrm{MS},p}\boldsymbol{a}_L(\phi_{\mathrm{MS},p})\right)\boldsymbol{a}_T(\alpha_{\mathrm{MS}})^{\mathsf{H}}, \tag{2.28}$$

$$\boldsymbol{\Phi}_{\mathrm{BS}} = \left(\sum_{p=1}^{P_{\mathrm{BS}}} \sqrt{\frac{RL}{P_{\mathrm{BS}}}}\beta_{\mathrm{BS},p}\boldsymbol{a}_R(\alpha_{\mathrm{BS},p})\right)\boldsymbol{a}_L(\phi_{\mathrm{BS}})^{\mathsf{H}}, \tag{2.29}$$

where P_{MS} and P_{BS} are the number of rays. Also, for any integer Q and angle ψ, steering vector $\boldsymbol{a}_Q(\psi) \in \mathbb{C}^{Q\times 1}$ with half-wavelength spacing is

$$\boldsymbol{a}_Q(\psi) = \frac{1}{\sqrt{Q}}\begin{bmatrix}1 & e^{j\pi\cos\psi} & \cdots & e^{j\pi(Q-1)\cos\psi}\end{bmatrix}^{\mathsf{T}}. \tag{2.30}$$

The angles $\phi_{\mathrm{MS},p}$, α_{MS}, $\alpha_{\mathrm{BS},p}$, and ϕ_{BS} denote the pth AoA of the IRS, and the pth AoD of the MS, the pth AoA of the BS, and the AoD of the IRS, respectively (see Figure 2.6). Then, the cascaded MS-IRS-BS channel is then given by $\boldsymbol{\Phi}_{\mathrm{BS}}\,\mathrm{diag}(\boldsymbol{\theta})\boldsymbol{\Phi}_{\mathrm{MS}}$ for a given IRS configuration $\boldsymbol{\theta} \in \mathbb{C}^{L\times 1}$. Here, the ith entry of $\boldsymbol{\theta}$ represents the gain and phase shift due to the ith IRS element. We aim to estimate the cascaded channel $\boldsymbol{\Phi}_{\mathrm{BS}}\,\mathrm{diag}(\boldsymbol{\theta})\boldsymbol{\Phi}_{\mathrm{MS}}$ for any $\boldsymbol{\theta}$.

2.6.2. Sparse Recovery Formulation

To estimate the channel, we send pilot symbols over K time slots over which $\boldsymbol{\Phi}_{\mathrm{MS}}$ and $\boldsymbol{\Phi}_{\mathrm{BS}}$ is assumed to be constant. We choose $K_{\mathrm{I}} < K$ IRS configurations, and for each configuration, we transmit pilot data $\boldsymbol{G} \in \mathbb{C}^{T\times K_{\mathrm{P}}}$ over K_{P} time slots such that $K = K_{\mathrm{I}}K_{\mathrm{P}}$. Hence, the received signal $\boldsymbol{Y}_k \in \mathbb{C}^{R\times K_{\mathrm{P}}}$ corresponding to the kth configuration $\boldsymbol{\theta}_k$ is [77, 78]

$$\boldsymbol{Y}_k = \boldsymbol{\Phi}_{\mathrm{BS}}\,\mathrm{diag}(\boldsymbol{\theta}_k)\boldsymbol{\Phi}_{\mathrm{MS}}\boldsymbol{G} + \boldsymbol{N}_k, \tag{2.31}$$

where $\boldsymbol{N}_k \in \mathbb{C}^{R\times K_{\mathrm{P}}}$ is the additive white Gaussian noise with zero mean and variance σ^2.

To formulate the sparse recovery problem, we disentangle $\boldsymbol{\theta}_k$ from the unknowns $\boldsymbol{\Phi}_{\mathrm{MS}}$ and $\boldsymbol{\Phi}_{\mathrm{BS}}$. For this, we vectorize both sides of (2.31) to obtain [77, 78]

$$\bar{\boldsymbol{y}}_k = \big((\boldsymbol{X}^{\mathsf{T}}\boldsymbol{\Phi}_{\mathrm{MS}}^{\mathsf{T}}) \odot \boldsymbol{\Phi}_{\mathrm{BS}}\big)\boldsymbol{\theta}_k + \bar{\boldsymbol{n}}_k \in \mathbb{C}^{RK_{\mathrm{P}}\times 1}, \tag{2.32}$$

So, the received data $\bar{\boldsymbol{Y}} \in \mathbb{C}^{RK_{\mathrm{P}}\times K_{\mathrm{I}}}$ for the IRS configurations $\boldsymbol{\Theta} = [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{K_{\mathrm{I}}}] \in \mathbb{C}^{L\times K_{\mathrm{I}}}$ is

$$\bar{\boldsymbol{Y}} = [\bar{\boldsymbol{y}}_1, \ldots, \bar{\boldsymbol{y}}_{K_{\mathrm{I}}}] = \big((\boldsymbol{X}^{\mathsf{T}}\boldsymbol{\Phi}_{\mathrm{MS}}^{\mathsf{T}}) \odot \boldsymbol{\Phi}_{\mathrm{BS}}\big)\boldsymbol{\Theta} + \bar{\boldsymbol{N}}, \tag{2.33}$$

where $\bar{\boldsymbol{N}} = [\bar{\boldsymbol{n}}_1, \ldots, \bar{\boldsymbol{n}}_{K_{\mathrm{I}}}] \in \mathbb{C}^{RK_{\mathrm{P}}\times K_{\mathrm{I}}}$. Next, we leverage angular sparsity in the channel matrices $\boldsymbol{\Phi}_{\mathrm{MS}}$ and $\boldsymbol{\Phi}_{\mathrm{BS}}$. For this, we apply the basis expansion model by discretizing the angular domain using a set of N grid angles $\{\psi_n\}_{n=1}^N$ such that $\cos(\psi_n) = 2n/N - 1$ [79]. Then, (2.28) and (2.29) reduce to

$$\boldsymbol{\Phi}_{\mathrm{BS}} = \boldsymbol{A}_R\boldsymbol{x}_{\mathrm{R}}\boldsymbol{x}_{\mathrm{L,d}}^{\mathsf{H}}\boldsymbol{A}_L^{\mathsf{H}} \quad \text{and} \quad \boldsymbol{\Phi}_{\mathrm{MS}} = \boldsymbol{A}_L\boldsymbol{x}_{\mathrm{L,a}}\boldsymbol{x}_{\mathrm{T}}^{\mathsf{H}}\boldsymbol{A}_T^{\mathsf{H}}, \tag{2.34}$$

where for any integer $Q > 0$, using (2.30), we define

$$\boldsymbol{A}_Q = \begin{bmatrix}\boldsymbol{a}_Q(\psi_1) & \boldsymbol{a}_Q(\psi_2) & \ldots & \boldsymbol{a}_Q(\psi_N)\end{bmatrix} \in \mathbb{C}^{Q\times N}.$$

Also, $\boldsymbol{x}_{\mathrm{R}}, \boldsymbol{x}_{\mathrm{L,d}}, \boldsymbol{x}_{\mathrm{L,a}}, \boldsymbol{x}_{\mathrm{T}} \in \mathbb{C}^{N\times 1}$ are the unknown sparse channel representations. The non-uniform grid points in the angular domain help to reduce the computational complexity of the estimation algorithm, which is discussed in (2.37). We also note that we use the same grid angle set $\{\psi_n\}_{n=1}^N$ for the AoAs and AoDs of the two channels for simplicity, but our framework and algorithm can also handle different grid sets.

Combining (2.34) and (2.33), and using the properties of the Khatri-Rao product [80,

Lemma A1], we disentangle the unknown sparse vectors from the known matrices as:

$$\bar{\boldsymbol{Y}} = \left[(\boldsymbol{X}^{\mathsf{T}}\boldsymbol{A}_T^*) \otimes \boldsymbol{A}_R\right]\left[(\boldsymbol{x}_{\mathrm{L,a}}\boldsymbol{x}_{\mathrm{T}}^{\mathsf{H}})^{\mathsf{T}} \otimes (\boldsymbol{x}_{\mathrm{R}}\boldsymbol{x}_{\mathrm{L,d}}^{\mathsf{H}})\right] \times (\boldsymbol{A}_L^{\mathsf{T}} \odot \boldsymbol{A}_L^{\mathsf{H}})\boldsymbol{\Theta} + \bar{\boldsymbol{W}}, \tag{2.35}$$

where $\otimes$ denotes the Kronecker product. Using Kronecker product's mixed-product property, we vectorize (2.35) to derive

$$\tilde{\boldsymbol{y}} = (\tilde{\boldsymbol{H}}_{\mathrm{L}} \otimes \boldsymbol{H}_{\mathrm{T}} \otimes \boldsymbol{H}_{\mathrm{R}})(\boldsymbol{x}_{\mathrm{L,a}} \otimes \boldsymbol{x}_{\mathrm{L,d}}^* \otimes \boldsymbol{x}_{\mathrm{T}}^* \otimes \boldsymbol{x}_{\mathrm{R}}) + \tilde{\boldsymbol{w}}, \tag{2.36}$$

where $\tilde{\boldsymbol{H}}_{\mathrm{L}} = \boldsymbol{\Theta}^{\mathsf{T}}(\boldsymbol{A}_L^{\mathsf{T}} \odot \boldsymbol{A}_L^{\mathsf{H}})^{\mathsf{T}}$, $\boldsymbol{H}_{\mathrm{T}} = \boldsymbol{X}^{\mathsf{T}}\boldsymbol{A}_T^*$, and $\boldsymbol{H}_{\mathrm{R}} = \boldsymbol{A}_R$. Further, we note that the only distinct columns of $\tilde{\boldsymbol{H}}_{\mathrm{L}}$ are its first N columns [81]. Hence, removing the redundant columns to reduce the dimension of the representation, we get

$$\tilde{\boldsymbol{y}} = (\boldsymbol{H}_{\mathrm{L}} \otimes \boldsymbol{H}_{\mathrm{T}} \otimes \boldsymbol{H}_{\mathrm{R}})\boldsymbol{x} + \tilde{\boldsymbol{n}} = \tilde{\boldsymbol{H}}\boldsymbol{x} + \tilde{\boldsymbol{n}} \in \mathbb{C}^{RK\times 1}, \tag{2.37}$$

where $\boldsymbol{H}_{\mathrm{L}} \in \mathbb{C}^{K_{\mathrm{I}}\times N}$ is the submatrix formed by the first N columns of $\tilde{\boldsymbol{H}}_{\mathrm{L}}$ and $\tilde{\boldsymbol{H}} = \boldsymbol{H}_{\mathrm{L}} \otimes \boldsymbol{H}_{\mathrm{T}} \otimes \boldsymbol{H}_{\mathrm{R}} \in \mathbb{C}^{RK\times N^3}$. Also, we define $\boldsymbol{x} = \boldsymbol{x}_{\mathrm{L}} \otimes \boldsymbol{x}_{\mathrm{T}}^* \otimes \boldsymbol{x}_{\mathrm{R}} \in \mathbb{C}^{N^3\times 1}$ with $\boldsymbol{x}_{\mathrm{L}} \in \mathbb{C}^{N\times 1}$ being the scaled version of the first N entries of $\boldsymbol{x}_{\mathrm{L,a}} \otimes \boldsymbol{x}_{\mathrm{L,d}}$. Hence, (2.37) translates the channel estimation problem into a sparse vector recovery problem with unknown $\boldsymbol{x}$. Using $\boldsymbol{g}$ and (2.34), we obtain the product term in the channel as

$$\mathrm{vec}(\boldsymbol{\Phi}_{\mathrm{MS}}^{\mathsf{T}} \odot \boldsymbol{\Phi}_{\mathrm{BS}}) = (\boldsymbol{H}_{\mathrm{A}} \otimes \boldsymbol{A}_M^* \otimes \boldsymbol{H}_{\mathrm{R}})\boldsymbol{x}, \tag{2.38}$$

where $\boldsymbol{H}_{\mathrm{A}}$ is the first N columns of $(\boldsymbol{A}_L^{\mathsf{T}} \odot \boldsymbol{A}_L^{\mathsf{H}})^{\mathsf{T}}$. Finally, the cascaded channel for a given IRS configuration $\boldsymbol{\theta}$ is computed as $(\boldsymbol{\Phi}_{\mathrm{MS}}^{\mathsf{T}} \odot \boldsymbol{\Phi}_{\mathrm{BS}})\boldsymbol{\theta}$. Thus, the rest of this section is devoted to derive an algorithm to estimate Kronecker-supported sparse $\boldsymbol{x}$ in (2.37).

Here, (2.37) is equivalent to (1.2) with $I = 3$, $\tilde{\boldsymbol{n}} = \boldsymbol{n}$, $M_1 = K_{\mathrm{I}}$, $M_2 = K_{\mathrm{P}}$, $M_3 = R$, and hence $\bar{M} = RK$. Also, we define $\boldsymbol{x} = \boldsymbol{x}_{\mathrm{L}} \otimes \boldsymbol{x}_{\mathrm{T}}^* \otimes \boldsymbol{x}_{\mathrm{R}} \in \mathbb{C}^{N^3\times 1}$ with $\boldsymbol{x}_{\mathrm{L}} \in \mathbb{C}^{N\times 1}$ being the scaled version of the first N entries of $\boldsymbol{x}_{\mathrm{L,a}} \otimes \boldsymbol{x}_{\mathrm{L,d}}^*$. Hence, (2.37) translates the channel estimation problem into the sparse recovery problem in (1.2) with unknown, Kronecker-supported vector $\boldsymbol{x}$. Now we can apply our algorithms (AM-KroSBL and SVD-KroSBL) with $I = 3$ to reconstruct the Kronecker-supported sparse channel vector $\boldsymbol{x}$.

Complexity Reduction

Although our AM-KroSBL and SVD-KroSBL are readily available, SBL-based algorithm is known to be computationally inefficient due to matrix inversion in (2.14). A technique to reduce the algorithm complexity was introduced in [31] using the Kronecker structure. We next present a novel technique to further improve the complexity, which can be applied to both AM-KroSBL and SVD-KroSBL. Specifically, invoking the matrix inversion lemma and the mixed product property of the Kronecker product and using the definition $\tilde{\boldsymbol{H}} = \boldsymbol{H}_{\mathrm{L}} \otimes \boldsymbol{H}_{\mathrm{T}} \otimes \boldsymbol{H}_{\mathrm{R}}$, we rewrite (2.14) as

$$\boldsymbol{\Sigma}_{\boldsymbol{x}} = \boldsymbol{\Gamma}^{(r-1)} - \boldsymbol{\Gamma}^{(r-1)}\tilde{\boldsymbol{H}}^{\mathsf{H}}(\sigma^2\boldsymbol{I} + (\boldsymbol{H}_{\mathrm{L}}\boldsymbol{\Gamma}_1^{(r-1)}\boldsymbol{H}_{\mathrm{L}}^{\mathsf{H}}) \otimes (\boldsymbol{H}_{\mathrm{T}}\boldsymbol{\Gamma}_2^{(r-1)}\boldsymbol{H}_{\mathrm{T}}^{\mathsf{H}}) \otimes (\boldsymbol{H}_{\mathrm{R}}\boldsymbol{\Gamma}_3^{(r-1)}\boldsymbol{H}_{\mathrm{R}}^{\mathsf{H}}))^{-1}\tilde{\boldsymbol{H}}\boldsymbol{\Gamma}^{(r-1)}, \tag{2.39}$$

where $\boldsymbol{\Gamma}_j^{(r-1)} = \text{diag}(\boldsymbol{\gamma}_j^{(r-1)})$, for $j = 1,2,3$. Let the eigenvalue decomposition of the three matrices in (2.39) be

$$\boldsymbol{H}_{\text{L}}\boldsymbol{\Gamma}_1^{(r-1)}\boldsymbol{H}_{\text{L}}^{\text{H}} = \boldsymbol{U}_1\boldsymbol{\Pi}_1\boldsymbol{U}_1^{\text{H}},$$
$$\boldsymbol{H}_{\text{T}}\boldsymbol{\Gamma}_2^{(r-1)}\boldsymbol{H}_{\text{T}}^{\text{H}} = \boldsymbol{U}_2\boldsymbol{\Pi}_2\boldsymbol{U}_2^{\text{H}},$$

and

$$\boldsymbol{H}_{\text{R}}\boldsymbol{\Gamma}_3^{(r-1)}\boldsymbol{H}_{\text{R}}^{\text{H}} = \boldsymbol{U}_3\boldsymbol{\Pi}_3\boldsymbol{U}_3^{\text{H}}.$$

Then, we derive

$$\boldsymbol{\Sigma}_{\boldsymbol{x}} = \boldsymbol{\Gamma}^{(r-1)}(\boldsymbol{I} - \tilde{\boldsymbol{H}}^{\text{H}}\boldsymbol{U}(\sigma^2\boldsymbol{I} + \boldsymbol{\Pi})^{-1}\boldsymbol{U}^{\text{H}}\tilde{\boldsymbol{H}}\boldsymbol{\Gamma}^{(r-1)}), \tag{2.40}$$

where $\boldsymbol{U} = \otimes_{j=1}^{3}\boldsymbol{U}_j$, and $\boldsymbol{\Pi} = \otimes_{j=1}^{3}\boldsymbol{\Pi}_j$. Combining (2.14) and (2.40), the posterior mean $\boldsymbol{\mu}_{\boldsymbol{x}}$ is

$$\boldsymbol{\mu}_{\boldsymbol{x}} = \sigma^{-2}(\boldsymbol{\Gamma}^{(r-1)}\tilde{\boldsymbol{H}}^{\text{H}}\tilde{\boldsymbol{y}} - \boldsymbol{\Gamma}^{(r-1)}\tilde{\boldsymbol{H}}^{\text{H}}\boldsymbol{U}(\sigma^2\boldsymbol{I} + \boldsymbol{\Pi})^{-1}\boldsymbol{U}^{\text{H}}\tilde{\boldsymbol{H}}\boldsymbol{\Gamma}^{(r-1)}\tilde{\boldsymbol{H}}^{\text{H}}\tilde{\boldsymbol{y}}). \tag{2.41}$$

The overall complexity of the different algorithms are summarized in Table 2.2. Here, R_{EM} is the number of EM iterations that varies across algorithms, and R_{AM} is the number of alternating iterations in (2.16). We see that the proposed schemes have lower complexity compared to the KroSBL. We also note that the complexity reduction method here can be seamlessly extended to higher orders with $I \geq 3$.

Table 2.2.: Time complexity of different versions of KroSBL

Method	Complexity
AM-KroSBL	$\mathcal{O}\big(R_{\text{EM}}(R_{\text{AM}}N^3 + N^3KB)\big)$
SVD-KroSBL	$\mathcal{O}\big(R_{\text{EM}}(N^4 + N^3KB)\big)$
KSBL in [31, 53]	$\mathcal{O}\big(R_{\text{EM}}N^6\big)$

Evaluation for Channel Estimation

Our simulation setting is as follows. We choose $R = 16$ BS antennas, $T = 6$ MS antennas, and $L = 256$ IRS elements. Each entry of the IRS configurations $\{\boldsymbol{\theta}_k\}_{k=1}^{K_{\text{I}}}$ is uniformly drawn from $\{-1/\sqrt{N}, 1/\sqrt{N}\}$ with $K_{\text{I}} \in \{4, 10\}$. For each IRS configuration, we send $K_{\text{P}} = 6$ pilot signals. The number of grid angles is $N = 18$ and all AoDs/AoAs are drawn uniformly from the grid angles. Further, the channel gains $\{\beta_{\text{BS},p}\}_{p=1}^{P_{\text{B}}}$ and $\{\beta_{\text{MS},p}\}_{p=1}^{P_{\text{M}}}$ in (2.28) and (2.29) are drawn from $\text{CN}(0,1)$ [82]. We use four performance metrics: NMSE, SRR, runtime, and symbol error rate (SER). Here, the channel estimation NMSE is given as

$$\frac{1}{K_{\text{I}}}\sum_{k=1}^{K_{\text{I}}}\frac{\|\boldsymbol{\Phi}_{\text{BS}}\,\text{diag}(\boldsymbol{\theta}_k)\boldsymbol{\Phi}_{\text{MS}} - \tilde{\boldsymbol{\Phi}}_{\text{BS}}\,\text{diag}(\boldsymbol{\theta}_k)\tilde{\boldsymbol{\Phi}}_{\text{MS}}\|_F^2}{\|\boldsymbol{\Phi}_{\text{BS}}\,\text{diag}(\boldsymbol{\theta}_k)\boldsymbol{\Phi}_{\text{MS}}\|_F^2}, \tag{2.42}$$

with $\tilde{\boldsymbol{\Phi}}_{\text{BS}}\,\text{diag}(\boldsymbol{\theta}_k)\tilde{\boldsymbol{\Phi}}_{\text{MS}}$ being the reconstructed channel, and SRR is the same as in (2.27). We use the classic SBL [62], OMP, and the KSBL in [31] as benchmarks.

From Figure 2.7, we observe that both SVD-KroSBL and AM-KroSBL outperform the other schemes in terms of NMSE, SRR and SER. Especially in the low SNR and low overhead (quantified by K_{I}) regimes, our algorithms have the best NMSE. From Figure 2.7

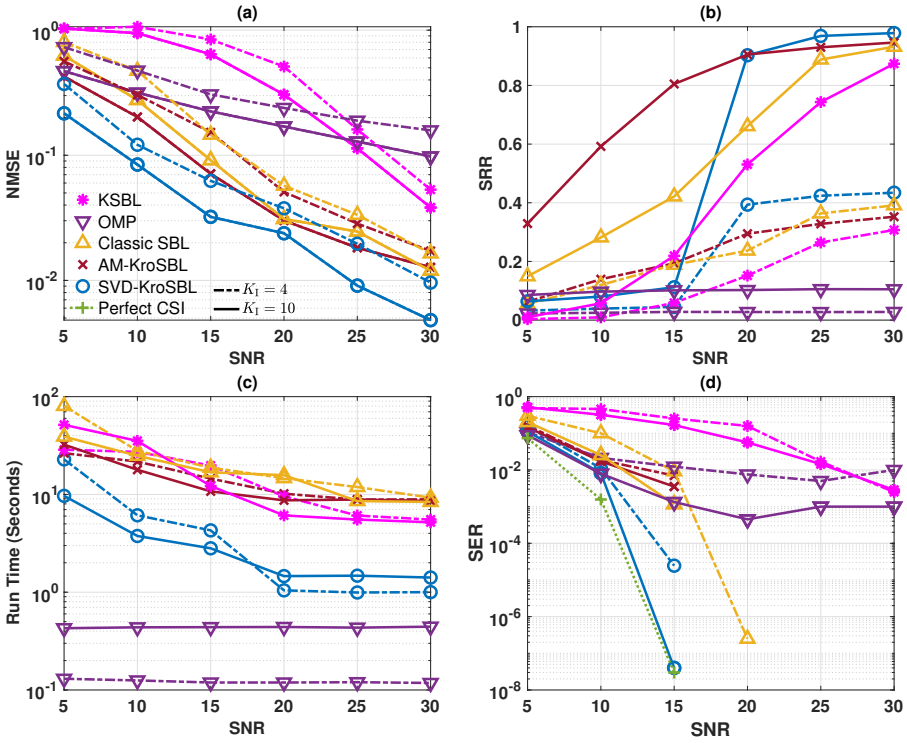


Figure 2.7.: Comparison of NMSE, SRR, runtime, and SER of AM-KroSBL and SVD-KroSBL with the competing schemes as a function SNR when $K_I = 4, 10$, $K_P = 6$ and $N = 18$.

(b), in the low SNR regime, the AM-KroSBL has the best SRR, while the SVD-KroSBL has the optimal performance in the high SNR case for both low and high overhead cases. We observe that in the low SNR regime, SVD-KroSBL outputs a sparse vector with many small terms, leading to a low SRR. But since the sparse vector is dominated by large values on the correct support, NMSE is still low. Further, Figure 2.7 (c) indicates that SVD-KroSBL, compared with other SBL-based methods, has one order less runtime. The high runtime of AM-KroSBL is expected due to the inner loop in the M-step, yet its runtime is comparable to the classic SBL and KSBL but with better NMSE. Finally, Figure 2.7 (d) shows that when $K_I = 4$, our schemes possess better SER than others. In contrast, when $K_I = 10$, only SVD-KroSBL has lower SER compared to the existing schemes and approaches the oracle scheme with perfect channel state information (CSI).

2.6.3. Extension of AM-KroSBL for Block Sparsity

In the IRS-aided channel model, the scatters lead to spreading AoAs (see Figure 2.6), causing clustered non-zero BEM coefficients. In our model, sparse vectors $\boldsymbol{x}_L$ and $\boldsymbol{x}_R$, containing the BEM coefficients of the AoAs of IRS and BS, possess block sparsity structures with unknown block boundaries. In this section, we extend KroSBL to handle block

Algorithm 4 PC-KroSBL

Input: Measurements $\boldsymbol{y}$, matrix $\boldsymbol{H}$, noise power σ^2, threshold ϵ, and ϵ_{AM}
Initialization: set $r=1$, $\{\boldsymbol{\gamma}_i\}^{(0)}=\mathbf{0}$, $\{\boldsymbol{\gamma}_i\}^{(1)}\in\mathscr{C}_+$
repeat
  Compute $\boldsymbol{d}^{(r)}$ using (2.13) and (2.14) with $\boldsymbol{\gamma}^{(r)}=\hat{\boldsymbol{\gamma}}^{(r)}$
  Set $t=1$, $\{\boldsymbol{\gamma}_i\}^{(r,1)}=\{\boldsymbol{\gamma}_i\}^{(r)}\in\mathscr{C}_+$, $\{\boldsymbol{\gamma}_i\}^{(r,0)}=\mathbf{0}$
  repeat
    Compute $\boldsymbol{\gamma}_1^{(r,t+1)}$ using (2.46)
    Compute $\tilde{\boldsymbol{\gamma}}_1=\hat{\boldsymbol{\gamma}}_1{}^{(r,t+1)}$ using (2.43)
    Compute $\{\boldsymbol{\gamma}_i\}^{(r,t+1)}$ using (2.16) and (2.17)
    Update AM iteration index $t\leftarrow t+1$
  until $\|\hat{\boldsymbol{\gamma}}^{(r,t)}-\hat{\boldsymbol{\gamma}}^{(r,t-1)}\|_2>\epsilon_{\mathsf{AM}}$
  $\{\boldsymbol{\gamma}_i\}^{(r+1)}=\{\boldsymbol{\gamma}_i\}^{(r,t)}$
  Update iteration index $r\leftarrow r+1$
until $\|\hat{\boldsymbol{\gamma}}^{(r)}-\hat{\boldsymbol{\gamma}}^{(r-1)}\|_2>\epsilon$
Output: $\boldsymbol{x}=\boldsymbol{\mu}_{\boldsymbol{x}}$ using (2.14)

sparsity. We term the new algorithm as pattern-coupled KroSBL (PC-KroSBL) and evaluate its performance for the channel estimation problem.

To tackle block sparsity, we draw inspiration from the PC-SBL algorithm [63] and impose a prior on each entry of the sparse vector, which not only depends on its hyperparameter but also the hyperparameters of its neighbors. This method connects the sparsity patterns of the adjacent entries, promoting block sparsity. We assume that $\boldsymbol{\gamma}_1$ exhibits block sparsity for ease of exposition. However, this idea can be readily extended when multiple hyperparameter vectors possess block sparsity. We adopt the prior on $\boldsymbol{x}$ with hyperparameters $\{\boldsymbol{\gamma}_i\}$ as $p(\boldsymbol{x}|\{\boldsymbol{\gamma}_i\})=\mathrm{CN}\big(0,\hat{\boldsymbol{\gamma}}\big)$, where $\hat{\boldsymbol{\gamma}}=\hat{\boldsymbol{\gamma}}_1\otimes(\otimes_{i=2}^{I}\boldsymbol{\gamma}_i)$ and

$$\hat{\boldsymbol{\gamma}}_1=\boldsymbol{C}_\beta\boldsymbol{\gamma}_1, \tag{2.43}$$

where $\boldsymbol{C}_\beta\in\mathbb{R}^{N\times N}$ is a tridiagonal Toeplitz matrix with ones along its diagonal and β along its first sub- and super-diagonals [83]. The parameter $\beta>0$ is the pattern-coupled coefficient. Using the new prior, the mean and variance of conditional distribution in the rth EM iteration are modified by replacing $\boldsymbol{\gamma}^{(r)}$ with $\hat{\boldsymbol{\gamma}}^{(r)}$ in (2.14). Thus, the optimization problem in the M-step is

$$\min_{\{\boldsymbol{\gamma}_i\}}\log|\operatorname{diag}(\hat{\boldsymbol{\gamma}})|+(\boldsymbol{d}^{(r)})^{\mathsf{T}}\hat{\boldsymbol{\gamma}}^{-1}\ \text{s.t.}\ \{\hat{\boldsymbol{\gamma}}_1,\{\boldsymbol{\gamma}_i\}_{i=2}^{I}\}\in\mathscr{C}_+. \tag{2.44}$$

We solve (2.44) using an iterative algorithm similar in spirit to AM-KroSBL by setting the gradient of the cost function with respect to all hyperparameter vectors to zero. Here, the update for $\{\boldsymbol{\gamma}_i\}_{i=2}^{I}$ is given by (2.16). However, due to the entanglement of hyperparameters in $\hat{\boldsymbol{\gamma}}_1$, the update for $\boldsymbol{\gamma}_1^{(r,t+1)}$ is

$$N^{I-1}\boldsymbol{C}_\beta(\hat{\boldsymbol{\gamma}}_1{}^{(r,t+1)})^{-1}=\boldsymbol{C}_\beta\operatorname{diag}\big((\hat{\boldsymbol{\gamma}}_1{}^{(r,t+1)})^{-2}\big)\times\Big(\boldsymbol{I}_N\otimes(\otimes_{i=2}^{I}(\boldsymbol{\gamma}_i^{(r,t)})^{-1})\Big)^{\mathsf{T}}\boldsymbol{d}^{(r)}. \tag{2.45}$$

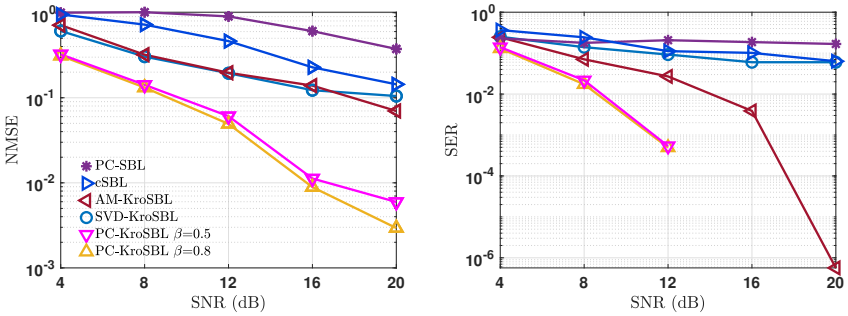


Figure 2.8.: NMSE of IRS-aided channel estimation and SER of different algorithms as functions of SNR

Solving (2.45) is not trivial due to the matrix-vector multiplication on both sides. However, if $\boldsymbol{C}_\beta$ is invertible, we can simplify (2.45) using (2.43) as

$$\boldsymbol{C}_\beta \boldsymbol{\gamma}_1^{(r,t+1)} = \hat{\boldsymbol{\gamma}}_1^{(r,t+1)} = N^{-I+1}(\boldsymbol{I}_N \otimes (\otimes_{i=2}^{I}(\boldsymbol{\gamma}_i^{(r,t)})^{-1}))^{\mathsf{T}} \boldsymbol{d}^{(r)}.$$

Also, we notice that the n-th eigenvalue of $\boldsymbol{C}_\beta$ is $\lambda_n = 1 + 2\beta\cos(\frac{n\pi}{N+1})$ for $n = 1, 2, \ldots, N$ [83]. Thus, we choose $\beta \neq -(2\cos(\frac{n\pi}{N+1}))^{-1}$ so that $\boldsymbol{C}_\beta$ is non-singular. With this choice, we update $\boldsymbol{\gamma}_1$ along with the constraint $\boldsymbol{\gamma}_1^{(r,t+1)} > 0$ with

$$\min_{\boldsymbol{\gamma}_1^{(r,t+1)} \geq 0} \|\boldsymbol{C}_\beta \boldsymbol{\gamma}_1^{(r,t+1)} - N^{-I+1}\left(\boldsymbol{I}_N \otimes (\otimes_{i=2}^{I}(\boldsymbol{\gamma}_i^{(r,t)})^{-1})\right)^{\mathsf{T}} \boldsymbol{d}^{(r)}\|_2^2. \tag{2.46}$$

The resulting algorithm, namely PC-KroSBL, is summarized in Algorithm 4.

Evaluation of PC-KroSBL for Channel Estimation

In this subsection, we present the observation from the results obtained when our PC-KroSBL is applied to the IRS-aided MIMO channel estimation, as discussed in Section 2.6. We choose the number of BS antennas $R = 16$, the number of MS antennas $T = 6$, and the number of IRS elements $L = 256$. Each entry of the IRS configurations $\{\boldsymbol{\theta}_k\}_{k=1}^{K_\mathrm{I}}$ is uniformly drawn from $\{-1/\sqrt{L}, 1/\sqrt{L}\}$ with $K_\mathrm{I} = 6$ and $K_\mathrm{P} = 3$. The number of grid angles is $N = 18$, and all AoDs/AoAs are drawn from the grids. We assume the angle spreads over three grid points, leading to three consecutive non-zeros in sparse vectors $\boldsymbol{x}_\mathrm{L}$ and $\boldsymbol{x}_\mathrm{R}$. For this, we choose one grid point uniformly at random and add the selected grid and its two neighboring grids to the support. The channel gains $\beta_{\mathrm{MS},p}$ and $\beta_{\mathrm{BS},p}$ defined in Section 2.6 are drawn from $\mathrm{CN}(0,1)$ [82]. The performance of our PC-KroSBL is compared with PC-SBL [63] and cSBL [62]. The pattern-coupled coefficient is 0.9 in the PC-SBL algorithm. We choose $\beta = 0.5$ and $\beta = 0.8$. The performance metrics are NMSE, SER, and runtime. Here, channel estimation NMSE is given as in (2.42). SER is computed over data transmission containing 10^5 uncoded QPSK symbols. The results are averaged over fifty independent channel realizations.

We focus on the low measurement regime with the under-sampling ratio $\bar{M}/\bar{N} = RK/N^3 \approx 5\%$. Figure 2.8 shows that PC-KroSBL outperforms AM- and SVD-KroSBL that

Table 2.3.: Computation time (in seconds) comparison for the IRS-aided MIMO channel estimation with $K_{\mathrm{I}} = 6$ and $K_{\mathrm{P}} = 3$ (underlined text for the best and boldface for **the second best**)

SNR	4dB	8dB	12dB	16dB	20dB
PC-KroSBL $\beta = 0.8$	*26.871*	14.392	9.821	7.979	7.380
PC-KroSBL $\beta = 0.5$	**27.904**	**13.675**	**8.771**	**6.670**	*5.441*
AM-KroSBL	44.480	19.672	11.973	12.533	11.915
SVD-KroSBL	30.194	*10.403*	*6.358*	*5.677*	**5.746**
PC-SBL	88.245	84.397	55.798	43.115	46.446
cSBL	62.844	22.455	14.478	11.258	9.008

do not account for block sparsity in terms of NMSE and SER. Comparing the performance of PC-KroSBL for different values of β, we infer that $\beta = 0.8$ achieves better accuracy and lower error than $\beta = 0.5$ case. Another interesting observation is that AM-KroSBL has higher NMSE but lower SER than SVD-KroSBL when SNR $= \{4, 8, 12, 16\}$. This is because SVD-KroSBL fails to identify correct AoAs and AoDs for some channel realizations, resulting in severe channel estimation errors. In such cases, SER is close to one, significantly affecting the overall SER performance of SVD-KroSBL. However, the AM-KroSBL does not completely fail even in the low measurement scenario and works better with increasing SNR. Moreover, PC-KroSBL can consistently and accurately estimate the channel and ensure low SER with limited measurements.

Further, PC-SBL fails in all SNR scenarios because it does not exploit the Kronecker structure, and hence, it needs more measurements than KroSBL algorithms for a graceful recovery. Also, while both sparse vectors $\boldsymbol{x}_{\mathrm{L}}$ and $\boldsymbol{x}_{\mathrm{R}}$ exhibit block sparsity, PC-SBL can only exploit the block sparsity of the $\boldsymbol{x}_{\mathrm{L}} \otimes \boldsymbol{x}_{\mathrm{R}}$. In contrast, PC-KroSBL can utilize the block sparsity of both $\boldsymbol{x}_{\mathrm{L}}$ and $\boldsymbol{x}_{\mathrm{R}}$, which is an added advantage of our algorithm. Finally, unlike PC-SBL, the cSBL algorithm involves no approximation, which is why it performs better than PC-SBL in the low measurement regime. However, our KroSBL outperforms cSBL, in which no prior knowledge is incorporated.

Table 2.3 includes the computation time of different schemes. PC-KroSBL has a comparable runtime to SVD-KroSBL and is faster than PC-SBL and cSBL despite its iterative M-step (inner loop). This is due to the complexity reduction technique [27] and the fast convergence of the inner loop.

2.7. Chapter Summary

In this chapter, we solved the Kronecker-structured linear inversion problem using the SBL framework with the EM procedure. To encourage the Kronecker-supported sparse vector, we designed a prior distribution parameterized by a Kronecker-structured hyperparameter vector and derived two Bayesian algorithms: an iterative AM-based and non-iterative SVD-based algorithm. The AM-based algorithm is theoretically guaranteed to converge to the stationary point of the SBL cost function, while the SVD-based algorithm is faster and has better reconstruction accuracy. We also analyzed the local minima of the KroSBL cost function. Numerical results on synthetic data demonstrated that our algorithms outperform the existing schemes in terms of accuracy and runtime. For applications, we applied our algorithm to the IRS-aided channel estimation for MIMO systems and extended the framework to the block sparsity case. After investigating the

recovery of the Kronecker-supported sparse vector, in the next chapter, we dive into the recovery of Kronecker-structured sparse vector $\boldsymbol{x}$, defined in (1.3). We will see that although KroSBL can be directly applied to this sparsity pattern to exploit the Kronecker-structured support of such an $\boldsymbol{x}$, it does not sufficiently enforce the additional Kronecker structure, i.e., structure on amplitude of $\boldsymbol{x}$. In the next chapter, to better exploit this property, we introduce a decomposition-based approach, resulting in not only better performance, but also better efficiency.

3

Efficient Off-Grid Bayesian Parameter Estimation for Kronecker-Structured Signals

In this chapter, we study the problem of estimating a Kronecker-structured sparse vector from an underdetermined linear inversion problem (1.2). Such a problem arises from the joint estimation of unknown parameters from Kronecker-structured multidimensional signals, motivated by applications like IRS-aided channel estimation. Exploiting the Kronecker structure, we decompose the estimation problem into smaller, independent subproblems across each dimension. Each subproblem is posed as a sparse recovery problem using basis expansion and solved using a novel off-grid SBL-based algorithm. Additionally, we derive probabilistic error bounds for the decomposition, quantify its denoising effect, and provide convergence analysis for off-grid SBL. Our simulations show that applying the algorithm to IRS-aided channel estimation improves accuracy and runtime compared to state-of-the-art methods through the low-complexity and denoising benefits of the decomposition step and the high-resolution estimation capabilities of off-grid SBL.

3.1. Introduction

This chapter addresses the problem of estimating a Kronecker-structured sparse vector $\boldsymbol{x}$ from the Kronecker product underdetermined linear system

$$\bar{\boldsymbol{y}} = \boldsymbol{H}\boldsymbol{x} + \bar{\boldsymbol{n}} = \left(\otimes_{i=1}^{I} \boldsymbol{H}_i\right)\boldsymbol{x} + \bar{\boldsymbol{n}},$$

as in (1.2). Here, $\bar{\boldsymbol{y}} \in \mathbb{R}^{\bar{M}}$ in this chapter denotes the measurement vector, $\boldsymbol{H}_i \in \mathbb{R}^{M_i \times N_i}$ is the ith factor matrix, $\boldsymbol{H} \in \mathbb{R}^{\bar{M} \times \bar{N}}$ is the known Kronecker product measurement matrix with $\prod_{i=1}^{I} M_i = \bar{M}$ and $\prod_{i=1}^{I} N_i = \bar{N}$, $\bar{\boldsymbol{n}}$ in this chapter denotes the measurement noise, and $\boldsymbol{x}$ is the Kronecker-structured sparse vector defined as

$$\boldsymbol{x} = \otimes_{i=1}^{I} \boldsymbol{x}_i,$$

as in (1.3). Next, we first introduce the application of estimating unknown parameters from Kronecker-structured multidimensional signals and explain its connection to the underdetermined linear inversion problem with Kronecker-structured sparse vector, and then present the related works and our contributions.

3.1.1. Kronecker-Structured Multidimensional Signal Parameter Estimation and Multidimensional BEM

As motivated in Section 1.1.2, the problem of estimating unknown parameters from Kronecker-structured multidimensional signals naturally leads to an underdetermined linear inversion problem, following the form of (1.2) with Kronecker-structured sparse vector as in (1.3). To be specific, consider the following fundamental model

$$\bar{\boldsymbol{y}} = \boldsymbol{y}_1 \otimes \boldsymbol{y}_2 \otimes \cdots \otimes \boldsymbol{y}_I + \bar{\boldsymbol{n}} = \otimes_{i=1}^{I} \boldsymbol{y}_i + \bar{\boldsymbol{n}}, \tag{3.1}$$

where $\otimes_{i=1}^{I} \boldsymbol{y}_i \in \mathbb{C}^{\bar{M}}$ is an I-dimensional signal, $\boldsymbol{y}_i \in \mathbb{C}^{M_i}$ represents the signal in each dimension, $\bar{\boldsymbol{n}}$ is the noise, $\bar{M} = \prod_{i=1}^{I} M_i$, and $\otimes$ is the Kronecker product. Each $\boldsymbol{y}_i$ encapsulates the signal in the corresponding dimension (e.g., AoA and AoD) and is expressed as

a weighted sum of nonlinear parametric functions,

$$\boldsymbol{y}_i = \sum_{s=1}^{S_i} \boldsymbol{h}_i(\bar{\psi}_{i,s})\bar{x}_{i,s}, \tag{3.2}$$

with parameters $\bar{\psi}_{i,s}$ and weights $\bar{x}_{i,s}$, for $s = 1, 2, \ldots, S_i$, where $\boldsymbol{h}_i(\cdot) \in \mathbb{C}^{M_i}$ is the nonlinear function. In the IRS-aided system example, the parameter $\bar{\psi}_{i,s}$ can be AoAs or AoDs, the nonlinear function $\boldsymbol{h}_i$ is related to the steering vector, and $\bar{x}_{i,s}$ represents the path gain corresponding to each AoA or AoD. Thus, the channel estimation problem reduces to estimating all $\bar{\psi}_{i,s}$'s and $\bar{x}_{i,s}$'s from the received signal $\bar{\boldsymbol{y}}$, where S_i's are also unknown. Hence, this paper focuses on the general problem of estimating the parameters and weights $\{\bar{\psi}_{i,s}, \bar{x}_{i,s}\}_{s=1}^{S_i}$ from measurements $\bar{\boldsymbol{y}}$ and the function $\boldsymbol{h}_i(\cdot)$ for all I dimensions.

Multidimensional BEM serves as a natural approach for this problem, as discussed in Section 1.1.1. It evaluates the nonlinear function over pre-sampled grids of unknown parameters in each dimension to express the multidimensional signal as the product of a known overcomplete Kronecker product dictionary of the basis functions and an unknown sparse coefficient vector as

$$\bar{\boldsymbol{y}} = \left(\otimes_{i=1}^{I} \boldsymbol{H}_i\right)\boldsymbol{x} + \bar{\boldsymbol{n}}. \tag{3.3}$$

Here, $\bar{\boldsymbol{y}} \in \mathbb{C}^{\bar{M}}$ is the measurement, $\boldsymbol{H}_i \in \mathbb{C}^{M_i \times N_i}$ is the overcomplete basis for $\boldsymbol{h}_i(\cdot)$, and $\otimes_{i=1}^{I} \boldsymbol{H}_i \in \mathbb{C}^{\bar{M} \times \bar{N}}$ is the overall dictionary with $\bar{M} < \prod_{i=1}^{I} N_i = \bar{N}$. Also, $\boldsymbol{x} \in \mathbb{C}^{\bar{N}}$ is the unknown Kronecker-structured sparse vector.

Apart from an effective way to exploit the Kronecker-structured prior knowledge, the mathematical model (3.3) also poses another issue, i.e., grid mismatch. This is because BEM adopts pre-sampled grids of unknown parameters to set up basis functions for the linear expression (3.3). However, the true parameters may fall off these grids, causing a *grid mismatch* issue [84, 85], which can degrade the estimation performance. Therefore, our approach should also handle the grid mismatch issue for improved performance. Next, we first present the prior effort to exploit Kronecker-structured prior knowledge and then demonstrate the progress made to mitigate grid mismatch.

3.1.2. Related Works

Solving (3.3) for a sparse vector $\boldsymbol{x}$ with Kronecker-structured support has been discussed in [18, 23, 31, 53]. A greedy method called KOMP generalizes the traditional OMP to multidimensional BEM [18]. It has low complexity but requires hand-tuning of a sensitive stopping threshold [23]. Another approach is called KroSBL, presented in Chapter 2. Although KroSBL can be readily applied here, it has two main drawbacks, as elaborated below.

First, KroSBL does not fully exploit prior knowledge (1.3). While KroSBL employs a Kronecker-structured covariance matrix, the variance only determines whether an entry is nonzero; in effect, KroSBL only exploits the Kronecker structure of the support vector. Consequently, KroSBL directly estimates the high-dimensional vector $\boldsymbol{x}$. Although the KroSBL algorithm employs some complexity reduction techniques [23], it still faces high overall complexity due to the high dimensionality of the Kronecker product. So, we seek

a method that can better exploit the Kronecker structure while significantly reducing complexity compared to KroSBL.

Second, KroSBL relies on multidimensional BEM, formulating the dictionary using predefined grids, causing grid mismatch. A popular way to address mismatch is to estimate the unknown parameters by optimizing the SBL cost function. However, this is challenging due to the *nonlinearity* of the SBL cost function. One approach is to use linearization methods that initialize the unknown parameters using pre-sampled values from the range of interest, locally approximate the nonlinear cost function using linearization techniques such as Taylor expansion [84, 86, 87], and solve for the first-order coefficients. However, such an approximation is only valid locally [88, 89]. Initialization with finer sampled values can only alleviate but not fully eliminate this issue [88–90], and it also induces higher-dimension dictionary with worse coherence condition. Alternatively, some works turn to marginal likelihood maximization by isolating the contribution of each variable to the likelihood, and then sequentially selecting the candidate that maximizes this isolated contribution [45, 88, 91, 92]. This greedy strategy starts with an empty dictionary and progressively adds the top-contributing candidates. So it is closely related to matching pursuit algorithms [45] and can suffer performance degradation as the number of unknowns increases [75]. The framework in [93] can also be leveraged to jointly construct Kronecker product dictionary and sparse vector. However, the variational Bayesian-based approach in [93] assumes unconstrained columns in dictionary, while the column of our dictionary is constrained by column function $\boldsymbol{h}_i(\cdot)$. In addition to the above off-grid methods, *gridless* methods, which operate directly in the continuous domain without discretization [85], are also studied in the literature. Example approaches include atomic norm-based methods [94–96] and variational Bayesian line spectral estimation techniques [97, 98]. However, atomic norm methods require a structured column function $\boldsymbol{h}_i(\cdot)$ to admit a Vandermonde decomposition for parameter recovery, while the distribution approximation used in variational Bayesian approaches is valid when $\boldsymbol{h}_i(\cdot)$ consists of complex sinusoidal components. Thus, they cannot be trivially generalized to different forms of $\boldsymbol{h}_i(\cdot)$. The drawbacks and limitations of existing multidimensional BEM, off-grid SBL algorithms, and gridless methods motivate novel approaches to solving our parameter estimation problem.

We aim to develop a method to solve (3.3) for estimating the parameters and weights $\{\bar{\psi}_{i,s}, \bar{x}_{i,s}\}_{s=1}^{S_i}$ for all I dimensions using $\bar{\boldsymbol{y}}$ in (3.1) and (3.2) with three key features: (i) utilizing the Kronecker structure in (3.1); (ii) overcoming the grid mismatch of linearization and marginal likelihood optimization; and (iii) achieving lower complexity compared to KroSBL.

3.1.3. Contributions

Our main contributions are as follows:

- *Decomposition-Based Algorithm*: We present two methods to decompose the measurement $\bar{\boldsymbol{y}}$ into multiple low-dimensional measurements, better utilizing the prior information of the Kronecker structure in Section 3.2.1. It transforms the joint multidimensional unknown parameter estimation into multiple separate subproblems in each dimension, leading to reduced complexity.

- *Off-grid Algorithm*: We use BEM for parameter estimation in each dimension and cast it into a sparse vector recovery problem solved using the EM-based SBL in Section 3.2.2. We further incorporate a grid optimization step in the EM iterations to address grid mismatch, implemented via alternating minimization. This approach fills gaps in prior work by enabling optimization without the linearization approximations used in [84, 86, 87] and jointly updating all variables, avoiding the greedy selection strategies of [45, 88, 91, 92].

- *Algorithm Analyses and Extensions*: We study the decomposition step and the iterative grid optimization in Section 3.3. We theoretically quantify the error bound of the decomposition step in the presence of noise and the denoising effect which we attribute to the better estimation performance. We discuss the convergence property of our algorithm and explore potential extensions and applicability of our decomposition strategy to alternative measurement structures that emerge in various practical scenarios in Section 3.3.4.

- *Application*: In Section 3.4, we analyze the signal model of a prototypical IRS-aided wireless communication system and explain the implementation of our algorithm for uplink cascaded IRS channel estimation.

- *Numerical Results*: We evaluate our schemes in three scenarios in Section 3.5. The first scenario highlights the computational efficiency and denoising benefits of the decomposition method. The second scenario demonstrates the high-resolution estimation capabilities of off-grid SBL. The third scenario focuses on IRS channel estimation, showcasing improved accuracy and reduced runtime, driven by the combined effects of decomposition and off-grid SBL.

In short, our algorithm recovers a Kronecker-structured sparse vector, which can be used to estimate parameters from Kronecker-structured multidimensional signals, tackling grid mismatch and high complexity through two key techniques: decomposition and off-grid SBL. These techniques are of independent interest and can be applied separately, depending on the specific signal model.

3.2. Off-Grid Sparse Recovery Algorithm for Kronecker-Structured Measurements

In this section, we study the parameter estimation problem with Kronecker-structured measurements. The signal model is

$$\bar{\boldsymbol{y}} = \otimes_{i=1}^{I} \bar{\boldsymbol{H}}_{i,\bar{\boldsymbol{\psi}}_i} \bar{\boldsymbol{x}}_i + \bar{\boldsymbol{n}} = \otimes_{i=1}^{I} \boldsymbol{y}_i + \bar{\boldsymbol{n}}, \tag{3.4}$$

where the noise term in $\bar{\boldsymbol{n}}$ need not be Kronecker-structured. For $i \in [I]$, the matrix $\bar{\boldsymbol{H}}_{i,\bar{\boldsymbol{\psi}}_i}$ is parameterized by $\bar{\boldsymbol{\psi}}_i \coloneqq \left[\bar{\psi}_{i,1}, \ldots, \bar{\psi}_{i,S_i}\right]^{\mathsf{T}} \in \mathbb{R}^{S_i}$ as follows,

$$\bar{\boldsymbol{H}}_{i,\bar{\boldsymbol{\psi}}_i} \coloneqq \begin{bmatrix} \boldsymbol{h}_i(\bar{\psi}_{i,1}) & \cdots & \boldsymbol{h}_i(\bar{\psi}_{i,S_i}) \end{bmatrix} \in \mathbb{C}^{M_i \times S_i},$$

where $\boldsymbol{h}_i \in \mathbb{C}^{M_i}$ is a *known* and *continuous* column function. The scalar S_i is the number of unknowns in $\bar{\boldsymbol{H}}_{i,\bar{\boldsymbol{\psi}}_i}$. We assume $\bar{\psi}_{i,s} \in [\psi_{i,\mathrm{l}}, \psi_{i,\mathrm{r}}]$, a *known* compact range of the unknown parameters, and the goal is to estimate $\bar{\boldsymbol{\psi}}_i$ and $\bar{\boldsymbol{x}}_i$ from (3.4).

To ensure identifiability of $\bar{\boldsymbol{\psi}}_i$, we assume $\boldsymbol{h}_i(\psi_p) \neq \boldsymbol{h}_i(\psi_q)$ for any $\psi_p \neq \psi_q$. Identifiability of $\bar{\boldsymbol{x}}_i$ is limited by the Kronecker structure, i.e., for scalars $\{\alpha_i\}_{i=1}^I$ with $\prod_{i=1}^I \alpha_i = 1$, the set of vectors $\{\bar{\boldsymbol{x}}_i\}_{i=1}^I$ and $\{\alpha_i \bar{\boldsymbol{x}}_i\}_{i=1}^I$ both result in $\bar{\boldsymbol{y}}$ when combined with a given noise vector $\bar{\boldsymbol{n}}$. However, in many applications (e.g., channel estimation [21, 27]), the goal is to recover the solution up to a scaling factor, as we later elaborate in Section 3.4. Therefore, we aim to jointly obtain $\bar{\boldsymbol{\psi}}_i$ and the coefficient $\bar{\boldsymbol{x}}_i$ up to scaling ambiguities, given i) measurement $\bar{\boldsymbol{y}}$, ii) vector function $\boldsymbol{h}_i$, and iii) range $[\psi_{i,\mathrm{l}}, \psi_{i,\mathrm{r}}]$ for $i \in [I]$.

We devise a two-step solution: the first step decomposes (3.4) into I subproblems, each estimating $\bar{\boldsymbol{\psi}}_i$ and $\bar{\boldsymbol{x}}_i$, and the second step solves these subproblems using an off-grid approach.

3.2.1. Step 1: Decomposition-Based Algorithm

To develop the decomposition algorithm, we use Lemma 2 for the noiseless set of linear equations, $\bar{\boldsymbol{y}} = \otimes_{i=1}^I \bar{\boldsymbol{H}}_{i,\bar{\boldsymbol{\psi}}_i} \bar{\boldsymbol{x}}_i$.

Lemma 2. *[23, Lemma 4] Consider linear equations* $\boldsymbol{y}_1 \otimes \boldsymbol{y}_2 = (\boldsymbol{H}_1 \otimes \boldsymbol{H}_2)(\boldsymbol{x}_1 \otimes \boldsymbol{x}_2) \neq \boldsymbol{0}$. *Solving for* $\boldsymbol{x}_1 \otimes \boldsymbol{x}_2$ *from the equations is equivalent to solving for* $\boldsymbol{x}_1$ *and* $\boldsymbol{x}_2$ *from* $\boldsymbol{H}_1(\alpha \boldsymbol{x}_1) = \boldsymbol{y}_1$ *and* $\boldsymbol{H}_2\left(\alpha^{-1}\boldsymbol{x}_2\right) = \boldsymbol{y}_2$, *for any scalar* $\alpha \neq 0$.

Lemma 2 indicates that we can estimate individual vectors $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, up to a scaling ambiguity α. Therefore, if $\bar{\boldsymbol{y}}$ is split into I low-dimensional vectors $\{\hat{\boldsymbol{y}}_i \in \mathbb{C}^{M_i}\}_{i=1}^I$, then (3.4) in the noiseless case ($\bar{\boldsymbol{n}} = \boldsymbol{0}$) can be decomposed into I subproblems, each with ambiguity $\{\alpha_i \neq 0\}_{i=1}^I$ with $\prod_{i=1}^I \alpha_i = 1$. This approach allows solving for $\bar{\boldsymbol{x}}_i$ individually, rather than jointly. We now discuss the decomposition of $\bar{\boldsymbol{y}}$ into low-dimensional vectors $\hat{\boldsymbol{y}}_i$'s, in both noiseless and noisy cases.

Noiseless Setting and Higher-Order SVD (HOSVD)

In the noiseless case, we aim to find $\{\hat{\boldsymbol{y}}_i\}_{i=1}^I$ so that $\bar{\boldsymbol{y}} = \boldsymbol{y} = \otimes_{i=1}^I \hat{\boldsymbol{y}}_i$. This can be achieved using HOSVD applied to the tensor representation of $\bar{\boldsymbol{y}}$. Using (3.1), $\bar{\boldsymbol{y}}$ can be represented as an Ith order tensor[1] $\boldsymbol{\mathcal{Y}} = \circ_{i=1}^I \boldsymbol{y}_i \in \mathbb{C}^{M_1 \times \cdots \times M_I}$, where $\circ$ is the tensor outer product. Its i^*th mode matricization for any $i^* \in [I-1]$ is

$$\boldsymbol{Y}_{(i^*)} = \boldsymbol{y}_{i^*} \left(\left(\otimes_{i=I}^{i^*+1} \boldsymbol{y}_i \right) \otimes \left(\otimes_{i=i^*-1}^{1} \boldsymbol{y}_i \right) \right)^{\mathsf{T}}, \tag{3.5}$$

where $\boldsymbol{y}_{i^*} \in \mathbb{C}^{M_{i^*}}$ is the i^*th component in the Kronecker product $\otimes_{i=1}^I \boldsymbol{y}_i$, and $(\cdot)^{\mathsf{T}}$ is the transpose operator. We use the index i^* in (3.5) and the subsequent discussion to avoid confusion with the iteration index i in the Kronecker product expression. Now, an estimate $\hat{\boldsymbol{y}}_{i^*}$ of $\boldsymbol{y}_{i^*}$ up to scaling ambiguities is the left leading singular vector $\boldsymbol{e}_{i^*}$ of the rank-one matrix $\boldsymbol{Y}_{(i^*)}$, i.e., $\hat{\boldsymbol{y}}_{i^*} = \boldsymbol{e}_{i^*}$, for $i^* \in [I-1]$. For $i^* = I$, the estimate $\hat{\boldsymbol{y}}_I$ is $\boldsymbol{e}_I$ multiplied by the

[1] From [99], $\mathrm{vec}(\boldsymbol{\mathcal{Y}}) = \mathrm{vec}(\circ_{i=1}^I \boldsymbol{y}_i) = \otimes_{i=I}^1 \boldsymbol{y}_i$ where the subscript is descending. For simplicity, we use ascending subscripts in the tensor outer product, resulting in $\boldsymbol{\mathcal{Y}}$ and $\bar{\boldsymbol{y}}$ containing identical entries, albeit reordered.

leading singular value of $\boldsymbol{Y}_{(I)}$, ensuring $\otimes_{i=1}^{I}\hat{\boldsymbol{y}}_i = \otimes_{i=1}^{I}\boldsymbol{y}_i$. The decomposition is called the HOSVD, assuming a multilinear rank of $(1,\cdots,1)$ due to the Kronecker structure [99–101].

Noisy Case and Truncated HOSVD

Extending to the noisy setting, the decomposition step becomes

$$\{\hat{\boldsymbol{y}}_i\}_{i=1}^{I} = \underset{\{\boldsymbol{z}_i\in\mathbb{C}^{M_i}\}_{i=1}^{I}}{\arg\min}\ \|\bar{\boldsymbol{y}} - \otimes_{i=1}^{I}\boldsymbol{z}_i\|_2, \tag{3.6}$$

where $\|\cdot\|_2$ is the vector ℓ_2 norm. We see that (3.6) is the same as seeking a tensor $\hat{\boldsymbol{\mathcal{Y}}} = \circ_{i=1}^{I}\boldsymbol{z}_i$ with multilinear rank $(1,\cdots,1)$ from measurement tensor $\bar{\mathscr{Y}}$ obtained from $\bar{\boldsymbol{y}}$ as

$$\min_{\hat{\boldsymbol{\mathcal{Y}}}} \|\bar{\boldsymbol{\mathcal{Y}}} - \hat{\boldsymbol{\mathcal{Y}}}\|_{\mathrm{F}} \ \text{ s.t. multilinear rank of } \hat{\boldsymbol{\mathcal{Y}}} \text{ is } (1,\cdots,1). \tag{3.7}$$

Unlike the noiseless case, here the ith mode matricization $\bar{\boldsymbol{Y}}_{(i)}$ of $\bar{\boldsymbol{\mathcal{Y}}}$ is not rank-one due to noise. We solve (3.7) through the truncated HOSVD, where only the left leading singular vector is selected. We obtain $\hat{\boldsymbol{\mathcal{Y}}} = \xi \times_1 \boldsymbol{e}_1 \cdots \times_I \boldsymbol{e}_I$ and $\xi = \bar{\boldsymbol{\mathcal{Y}}} \times_1 \boldsymbol{e}_1^{\mathsf{H}} \cdots \times_I \boldsymbol{e}_I^{\mathsf{H}}$, where $\boldsymbol{e}_i$ is the left leading singular vector of the ith mode matricization $\bar{\boldsymbol{Y}}_{(i)}$ of $\bar{\mathscr{Y}}$ for $i \in [I]$ [102]. Here, operator $\times_i$ is the ith tensor mode product and $(\cdot)^{\mathsf{H}}$ is the conjugate transpose. Then, a solution to (3.6) is $\hat{\boldsymbol{y}}_i = \boldsymbol{e}_i$ for $i \in [I-1]$ and $\hat{\boldsymbol{y}}_I = \xi \boldsymbol{e}_I$.

A Low-Complexity Approximation

When I and M_i are large, HOSVD can become computationally intensive due to SVD needed to obtain $\boldsymbol{e}_i$ for $i \in [I-1]$. Hence, we offer a low-complexity method using recursive SVD-based rank-one approximations,

$$(\hat{\boldsymbol{y}}_i, \bar{\boldsymbol{y}}_i) = \underset{(\boldsymbol{z}_i\in\mathbb{C}^{M_i}, \bar{\boldsymbol{z}}_i), \|\boldsymbol{z}_i\|_2=1}{\arg\min}\ \|\bar{\boldsymbol{y}}_{i-1} - \boldsymbol{z}_i \otimes \bar{\boldsymbol{z}}_i\|_2, \tag{3.8}$$

for $i \in [I-1]$ where $\bar{\boldsymbol{y}}_0 = \bar{\boldsymbol{y}}$ and $\bar{\boldsymbol{y}}_{I-1} = \hat{\boldsymbol{y}}_I$. For example, we consider the case when $i = 1$. We rearrange $\bar{\boldsymbol{y}}$ as $\bar{\boldsymbol{Y}} \in \mathbb{C}^{\bar{M}/M_1 \times M_1}$ where $\mathrm{vec}(\bar{\boldsymbol{Y}}) = \bar{\boldsymbol{y}}$. Since $\boldsymbol{z}_i \otimes \bar{\boldsymbol{z}}_i = \mathrm{vec}(\bar{\boldsymbol{z}}_i \boldsymbol{z}_i^{\mathsf{T}})$, (3.8) is equivalent to a rank-one approximation that minimizes $\|\bar{\boldsymbol{Y}} - \bar{\boldsymbol{z}}_i \boldsymbol{z}_i^{\mathsf{T}}\|_{\mathrm{F}}$, and $\hat{\boldsymbol{y}}_i$ is the leading singular vector of $\bar{\boldsymbol{Y}}$. Compared to HOSVD, here, the problem dimension decreases with i as $\bar{\boldsymbol{z}}_i \in \mathbb{C}^{\prod_{j>i} M_j}$, and the overall complexity is dominated by the first step, i.e., $i = 1$. Besides, in the noiseless case, (3.8) and HOSVD yield the same solution.

Combining the decomposition step for $\boldsymbol{y}$ obtaining $\{\hat{\boldsymbol{y}}_i\}_{i=1}^{I}$ with Lemma 2, we break down the original $\bar{M}$-dimensional problem into I subproblems of dimensions $\{M_i\}_{i=1}^{I}$,

$$\hat{\boldsymbol{y}}_i = \bar{\boldsymbol{H}}_{i,\bar{\boldsymbol{\psi}}_i} \bar{\boldsymbol{x}}_i + \bar{\boldsymbol{n}}_i, \quad i \in [I], \tag{3.9}$$

which can be solved in parallel. Here, we assume $\alpha_i = 1$ without loss of generality, as we seek solutions up to a scaling factor. The decomposition better exploits the Kronecker structure in the measurements, aiding denoising (see Section 3.3.1) and reducing the complexity (see Section 3.3.4). Before presenting these analyses, we first develop an algorithm to estimate $\bar{\boldsymbol{\psi}}_i$ and $\bar{\boldsymbol{x}}_i$ from (3.9) for a given i.

Algorithm 5 OffSBL

Input: Measurement $\hat{\boldsymbol{y}}$, the number of initial grids N, the range $[\psi_1, \psi_r]$, and thresholds $\epsilon_1, \epsilon_2 < 1$

Set $r = 0$, initialize $\boldsymbol{\gamma}^{(0)} = \mathbf{1}$, and initialize $\boldsymbol{\psi}^{(0)}$ with uniform samples $\{\theta_n\}_{n=1}^N$ from $[\psi_1, \psi_r]$.
repeat over r
  Compute $\boldsymbol{\mu}_x$ and $\boldsymbol{\Sigma}_x$ using (3.13)
  Update $\boldsymbol{\gamma}^{(r+1)}$ using (3.12)
  Set $t = 0$, initialize $\boldsymbol{\psi}^{(r,t)} = \boldsymbol{\psi}^{(r)}$
  repeat over t
    Obtain $\psi_{n^*}^{(r,t+1)}$ by solving (3.16), for $n^* \in [N]$
    Set $t = t + 1$
  until $\|\boldsymbol{\psi}^{(t+1)} - \boldsymbol{\psi}^{(t)}\| < \epsilon_1$
  Let $\boldsymbol{\psi}^{(r+1)} = \boldsymbol{\psi}^{(r,t)}$
  Obtain $(\sigma^2)^{(r+1)}$ using (3.14) and set $r = r + 1$
until $\|\boldsymbol{\gamma}^{(r+1)} - \boldsymbol{\gamma}^{(r)}\|_2 / \|\boldsymbol{\gamma}^{(r)}\|_2 < \epsilon_2$

Output: Estimated $\boldsymbol{x} = \boldsymbol{\mu}_x$.

Algorithm 6 Decomposition-Based SBL (dSBL)

Input: Measurement $\bar{\boldsymbol{y}}$, dictionaries $\boldsymbol{H}_i$ for $i \in [I]$

for $i \in [I]$ **do**
  Solve (3.7) or (3.8) to obtain $\hat{\boldsymbol{y}}_i$
  Solve (3.9) for $\boldsymbol{x}_i$ using Algorithm 5
end for

Output: For $i \in [I]$, $\bar{\boldsymbol{x}}_i$ corresponds to the nonzero entries of $\boldsymbol{x}_i$ and $\bar{\boldsymbol{\psi}}_i$ corresponds to the support of $\boldsymbol{x}_i$, and $\hat{\boldsymbol{x}}$ is obtained through $\otimes_{i=1}^I \boldsymbol{x}_i$.

3.2.2. Step 2: Off-Grid SBL-Based Estimation Algorithm

In each dimension, the subproblem takes the general form of $\hat{\boldsymbol{y}} = \bar{\boldsymbol{H}}_{\bar{\boldsymbol{\psi}}} \bar{\boldsymbol{x}} + \bar{\boldsymbol{n}}$ for a nonlinear function $\boldsymbol{h}$ when entries of $\bar{\boldsymbol{\psi}}$ belong to $[\psi_1, \psi_r]$, where we *drop the dimension index i* for simplicity. We adopt BEM by discretizing $[\psi_1, \psi_r]$ with a set of variables $\boldsymbol{\psi} \in \mathbb{R}^N$ with the nth variable ψ_n, forming the dictionary

$$\boldsymbol{H}(\boldsymbol{\psi}) := \begin{bmatrix} \boldsymbol{h}(\psi_1) & \cdots & \boldsymbol{h}(\psi_N) \end{bmatrix} \in \mathbb{R}^{M \times N}.$$

This leads to the BEM with coefficient vector $\boldsymbol{x}$ as

$$\hat{\boldsymbol{y}} = \boldsymbol{H}(\boldsymbol{\psi})\boldsymbol{x} + \boldsymbol{n}, \tag{3.10}$$

Only a few entries of $\boldsymbol{\psi}$ correspond to the true parameters $\bar{\boldsymbol{\psi}}$, making $\boldsymbol{x}$ sparse. However, solving (3.10) by fixing $\boldsymbol{\psi}$ at some predefined grid points, as in standard sparse recovery, leads to grid mismatch. Thus, we treat $\boldsymbol{\psi}$ as variable and jointly estimate $\boldsymbol{\psi}$ and sparse $\boldsymbol{x}$ using the SBL framework. Then, the nonzero entries of $\boldsymbol{x}$ and their corresponding ψ_n's are estimates of $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{\psi}}$, respectively.

SBL adopts a fictitious zero mean complex Gaussian distribution $\mathrm{CN}(\boldsymbol{x}|\boldsymbol{0},\boldsymbol{\Gamma})$ as prior on the sparse vector $\boldsymbol{x}$ with an unknown diagonal covariance matrix $\boldsymbol{\Gamma}$. Let $\boldsymbol{\Gamma}=\mathrm{diag}(\boldsymbol{\gamma})\in\mathbb{R}^{N\times N}$ with the diagonal entries $\boldsymbol{\gamma}\in\mathbb{R}^N$. We assume Gaussian noise $\boldsymbol{n}\sim\mathrm{CN}(0,\sigma^2\boldsymbol{I})$ with unknown variance σ^2. Using type II ML, we first estimate the hyperparameters $\boldsymbol{\gamma}$, $\boldsymbol{\psi}$, and σ^2, and then, the estimate of $\boldsymbol{x}$ is $\arg\max_{\boldsymbol{x}} p(\boldsymbol{x}|\hat{\boldsymbol{y}};\boldsymbol{\gamma},\boldsymbol{\psi},\sigma^2)$. The ML estimates of the hyperparameters are

$$\min_{\boldsymbol{\gamma}\geq\boldsymbol{0},\ \boldsymbol{\psi}\in[\psi_{\mathrm{l}},\psi_{\mathrm{r}}]^N,\ \sigma^2>0}\mathscr{L}\left(\boldsymbol{\gamma},\boldsymbol{\psi},\sigma^2\right), \tag{3.11}$$

where $\boldsymbol{\gamma}\geq\boldsymbol{0}$ indicates that the entries of $\boldsymbol{\gamma}$ are nonnegative. Using the SBL priors, negative log-likelihood function $\mathscr{L}$ is

$$\mathscr{L}\left(\boldsymbol{\gamma},\boldsymbol{\psi},\sigma^2\right)=\log p(\hat{\boldsymbol{y}};\boldsymbol{\gamma},\boldsymbol{\psi},\sigma^2)=\log|\boldsymbol{\Sigma}_{\mathrm{y}}|+\mathrm{tr}\left(\hat{\boldsymbol{y}}^{\mathsf{H}}\boldsymbol{\Sigma}_{\mathrm{y}}^{-1}\hat{\boldsymbol{y}}\right),$$

where $\boldsymbol{\Sigma}_{\mathrm{y}}=\sigma^2\boldsymbol{I}_M+\boldsymbol{H}(\boldsymbol{\psi})\boldsymbol{\Gamma}\boldsymbol{H}(\boldsymbol{\psi})^{\mathsf{H}}$. We resort to the EM algorithm to solve (3.11). Specifically, the rth iteration of EM is

$$\begin{aligned}
&\textbf{E-step: } Q(\boldsymbol{\gamma},\boldsymbol{\psi},\sigma^2):=\mathbb{E}_{\boldsymbol{x}|\hat{\boldsymbol{y}};\{\boldsymbol{\gamma},\boldsymbol{\psi},\sigma^2\}^{(r)}}\{\log p(\hat{\boldsymbol{y}},\boldsymbol{x};\boldsymbol{\gamma},\boldsymbol{\psi},\sigma^2)\},\\
&\textbf{M-step: }\{\boldsymbol{\gamma},\boldsymbol{\psi},\sigma^2\}^{(r+1)}=\underset{\boldsymbol{\gamma}>\boldsymbol{0},\boldsymbol{\psi}\in[\psi_{\mathrm{l}},\psi_{\mathrm{r}}]^N,\sigma^2>0}{\arg\max}\ Q(\boldsymbol{\gamma},\boldsymbol{\psi},\sigma^2).
\end{aligned}$$

Here, $\boldsymbol{\gamma}>\boldsymbol{0}$ means the entries of $\boldsymbol{\gamma}$ should be positive to avoid degenerate distributions. Further, we note that

$$Q(\boldsymbol{\gamma},\boldsymbol{\psi},\sigma^2)=\mathbb{E}_{\boldsymbol{x}|\hat{\boldsymbol{y}};\{\boldsymbol{\gamma},\boldsymbol{\psi},\sigma^2\}^{(r)}}\{\log p(\hat{\boldsymbol{y}}|\boldsymbol{x};\boldsymbol{\psi},\sigma^2)\}+\mathbb{E}_{\boldsymbol{x}|\hat{\boldsymbol{y}};\{\boldsymbol{\gamma},\boldsymbol{\psi},\sigma^2\}^{(r)}}\{\log p(\boldsymbol{x};\boldsymbol{\gamma})\},$$

and thus, the optimization problem in the M-step is separable in $\boldsymbol{\gamma}$ and $\{\boldsymbol{\psi},\sigma^2\}$. The optimization problem in $\boldsymbol{\gamma}$ is

$$\boldsymbol{\gamma}^{(r+1)}=\underset{\boldsymbol{\gamma}>\boldsymbol{0}}{\arg\min}\log|\mathrm{diag}(\boldsymbol{\gamma})|+(\boldsymbol{d}^{(r)})^{\mathsf{T}}\boldsymbol{\gamma}^{-1}=\boldsymbol{d}^{(r)}, \tag{3.12}$$

with $\boldsymbol{d}^{(r)}=\mathrm{diag}(\boldsymbol{\Sigma}_{\mathrm{x}}+\boldsymbol{\mu}_{\mathrm{x}}\boldsymbol{\mu}_{\mathrm{x}}^{\mathsf{H}})$ where $\mathrm{diag}(\cdot)$ returns the diagonal entries of the matrix input, and $\boldsymbol{\gamma}^{-1}$ representing element-wise inversion. Here, $\boldsymbol{\mu}_{\mathrm{x}}$ and $\boldsymbol{\Sigma}_{\mathrm{x}}$ are the mean and variance of conditional distribution $p(\boldsymbol{x}|\hat{\boldsymbol{y}};\{\boldsymbol{\gamma},\boldsymbol{\psi},\sigma^2\}^{(r)})$, respectively,

$$\begin{aligned}
\boldsymbol{\mu}_{\mathrm{x}}&=\left(\sigma^{-2}\right)^{(r)}\boldsymbol{\Sigma}_{\mathrm{x}}\boldsymbol{H}(\boldsymbol{\psi}^{(r)})^{\mathsf{H}}\hat{\boldsymbol{y}},\\
\boldsymbol{\Sigma}_{\mathrm{x}}&=\left[\left(\sigma^{-2}\right)^{(r)}\boldsymbol{H}(\boldsymbol{\psi}^{(r)})^{\mathsf{H}}\boldsymbol{H}(\boldsymbol{\psi}^{(r)})+\mathrm{diag}(\boldsymbol{\gamma}^{(r)})^{-1}\right]^{-1}.
\end{aligned} \tag{3.13}$$

Optimizing $\boldsymbol{\psi}$ and σ^2 in the M-step yields

$$\{\boldsymbol{\psi},\sigma^2\}^{(r+1)}=\underset{\boldsymbol{\psi}\in[\psi_{\mathrm{l}},\psi_{\mathrm{r}}]^N,\ \sigma^2>0}{\arg\min}\ N\log\sigma+\frac{1}{2\sigma^2}g(\boldsymbol{\psi}),$$

where $g(\boldsymbol{\psi}):=\|\hat{\boldsymbol{y}}-\boldsymbol{H}(\boldsymbol{\psi})\boldsymbol{\mu}_{\mathrm{x}}\|_2^2+\mathrm{tr}(\boldsymbol{\Sigma}_{\mathrm{x}}\boldsymbol{H}(\boldsymbol{\psi})^{\mathsf{H}}\boldsymbol{H}(\boldsymbol{\psi}))$, is independent of σ^2. Given $\boldsymbol{\psi}^{(r+1)}$,

we update σ^2 as

$$(\sigma^2)^{(r+1)} = g(\boldsymbol{\psi}^{(r+1)})/N. \tag{3.14}$$

Further, $\boldsymbol{\psi}^{(r+1)} = \arg\min_{\boldsymbol{\psi}} g(\boldsymbol{\psi})$ simplifies to

$$\boldsymbol{\psi}^{(r+1)} = \underset{\boldsymbol{\psi}\in[\psi_1,\psi_\Gamma]^N}{\arg\min}\ \mathrm{tr}\left(\boldsymbol{H}(\boldsymbol{\psi})\boldsymbol{\Sigma}\boldsymbol{H}(\boldsymbol{\psi})^{\mathsf{H}}\right) - 2\mathrm{Re}\left\{\mathrm{tr}\left(\boldsymbol{M}\boldsymbol{H}(\boldsymbol{\psi})\right)\right\}, \tag{3.15}$$

where $\boldsymbol{\Sigma} := \boldsymbol{\Sigma}_{\mathrm{x}} + \boldsymbol{\mu}_{\mathrm{x}}\boldsymbol{\mu}_{\mathrm{x}}^{\mathsf{H}}$ and $\boldsymbol{M} = \boldsymbol{\mu}_{\mathrm{x}}\hat{\boldsymbol{y}}^{\mathsf{H}}$. We use the alternating minimization method to solve (3.15), where we alternatively optimize one entry of $\boldsymbol{\psi}$ while keeping all others fixed. The tth iteration of the alternating minimization method updates the n^*th variable ψ_{n^*} by minimizing

$$f_{n^*}(\psi_{n^*}) = 2\mathrm{Re}\left\{\boldsymbol{v}_{n^*}^{\mathsf{H}}\boldsymbol{h}(\psi_{n^*})\right\} + \boldsymbol{\Sigma}_{n^*,n^*}\boldsymbol{h}(\psi_{n^*})^{\mathsf{H}}\boldsymbol{h}(\psi_{n^*}).$$

Here, $\boldsymbol{v}_{n^*} \in \mathbb{C}^M$ is defined as

$$\boldsymbol{v}_{n^*} = \sum_{n=1}^{n^*-1}\boldsymbol{\Sigma}_{n^*,n}\boldsymbol{h}(\psi_n^{(r,t+1)}) + \sum_{n=n^*+1}^{N}\boldsymbol{\Sigma}_{n^*,n}\boldsymbol{h}(\psi_n^{(r,t)}) - \boldsymbol{M}_{n^*,:}^{\mathsf{H}},$$

where $\boldsymbol{M}_{n^*,:}$ is the n^*th row of $\boldsymbol{M}$, and $\boldsymbol{\Sigma}_{n^*,n^*}$ is the (n^*,n^*)th entry of $\boldsymbol{\Sigma}$. We use index n^* to avoid confusion with iteration index n. Hence, the tth alternative minimization iterate $\psi_{n^*}^{(r,t+1)}$ in the rth EM iteration for index $n^* \in [N]$ is

$$\psi_{n^*}^{(r,t+1)} = \underset{\frac{\psi_{n^*}^{(r,t)}+\psi_{n^*-1}^{(r,t)}}{2}\le\psi\le\frac{\psi_{n^*}^{(r,t)}+\psi_{n^*+1}^{(r,t)}}{2}}{\arg\min}\ f_{n^*}(\psi). \tag{3.16}$$

The assumption $\boldsymbol{h}_i(\psi_p) \neq \boldsymbol{h}_i(\psi_q)$ for any $\psi_p \neq \psi_q$ ensures the solution identifiability of problem (3.16). Although $f_{n^*}(\psi)$ in (3.16) is a nonlinear function of ψ, we avoid local first-order approximations [90] by using the alternating minimization. It translates the problem (3.15) into simple *one dimension* subproblems (3.16). We solve (3.16) using a simple (one-dimensional) grid search, preserving accuracy and ensuring easy implementation. Our off-grid SBL (OffSBL) and the overall decomposition-based SBL (dSBL) are outlined in Algorithms 5 and 6, respectively.

So far, we discussed our dSBL framework. Next, we discuss how the recovered signals $\{\boldsymbol{x}_i\}_{i=1}^I$ are used to infer coefficients $\{\bar{\boldsymbol{x}}_i\}_{i=1}^I$, unknown parameters $\{\bar{\boldsymbol{\psi}}_i\}_{i=1}^I$, $\{S_i\}_{i=1}^I$, and the estimation of $\boldsymbol{x}$, denoted as $\hat{\boldsymbol{x}}$. Through automatic relevance determination mechanism [103], SBL promotes sparsity in $\boldsymbol{x}_i$ by learning a sparse hyperparameter $\boldsymbol{\gamma}_i$ for $i \in [I]$. We note that $\boldsymbol{\gamma}_i$ is the hyperparameter involved in (3.11) where the subscript i referring to the ith subproblem (3.9) is dropped. Then, the support of $\boldsymbol{x}_i$, indicated by the significant peaks in amplitude, reveals the estimated active components $\bar{\boldsymbol{\psi}}_i$, while its values are the estimates of $\bar{\boldsymbol{x}}_i$; the number of estimated active components is S_i. Due to computational limits, we cap the number of EM iterations, so the inactive components are not exactly zero but a very small value. We threshold $\boldsymbol{\gamma}_i$ (e.g., 10^{-4}) to prune the values below the threshold and retain the other values for estimating $\{\bar{\boldsymbol{x}}_i\}_{i=1}^I$ and $\{\bar{\boldsymbol{\psi}}_i\}_{i=1}^I$, as explained before. In the end, the estimation $\hat{\boldsymbol{x}}$ of $\boldsymbol{x}$ is obtained as $\hat{\boldsymbol{x}} = \otimes_{i=1}^I \boldsymbol{x}_i$.

3.3. Theoretical Analysis and Extensions

This section analyzes dSBL algorithm, covering the decomposition error bound and denoising effect of HOSVD and convergence results of OffSBL. We then present the complexity analysis of dSBL, showing its computational advantage compared to other methods. Finally, we discuss extensions of our algorithms to other similar signal models.

3.3.1. Analysis of Decomposition-Based Algorithm

We start with the decomposition error bound, where we quantify the error between the decomposed vectors $\{\hat{\boldsymbol{y}}_i\}_{i=1}^I$ and the true signal components $\{\boldsymbol{y}_i = \bar{\boldsymbol{H}}_{i,\bar{\boldsymbol{\psi}}_i}\bar{\boldsymbol{x}}_i\}_{i=1}^I$. We measure the error as the angle between $\boldsymbol{y}_i$ and $\hat{\boldsymbol{y}}_i$, accounting for scaling ambiguity in the decomposition step.

Theorem 4 (Decomposition Accuracy)**.** *Let $\bar{\boldsymbol{y}} = \otimes_{i=1}^I \boldsymbol{y}_i + \bar{\boldsymbol{n}} \in \mathbb{R}^{\bar{M}}$ be the noisy measurement as given in (3.4), where $\boldsymbol{y}_i \in \mathbb{R}^{M_i}$ and $\bar{\boldsymbol{n}}$ has independent zero-mean Gaussian entries with variance σ_{t}^2. Suppose the signal satisfies*

$$\lambda^2 = \| \otimes_{i=1}^I \boldsymbol{y}_i \|_2^2 / \sigma_{\mathrm{t}}^2 \geq C_{\mathrm{gap}} (\sqrt{\bar{M}} + \max_{1\leq i\leq I} M_i),$$

for a large constant $C_{\mathrm{gap}} > 0$. Then, there exist constants $c, C > 0$ such that with probability at least $1 - C\exp\{-cM_i\}$,

$$\sin(\vartheta_i) \leq C\left(\sqrt{M_i}\lambda^{-1} + \sqrt{\bar{M}}\lambda^{-2}\right), \quad i \in [I],$$

where $\vartheta_i := \arccos\left(|\boldsymbol{y}_i^{\mathsf{T}}\hat{\boldsymbol{y}}_i| / (\|\boldsymbol{y}_i\|_2 \|\hat{\boldsymbol{y}}_i\|_2)\right)$ is the angle between $\boldsymbol{y}_i$ and its estimate $\hat{\boldsymbol{y}}_i$ obtained by solving (3.6) and (3.7).

Proof. See Appendix B.1. □

We note that $\lambda/\bar{M}$ represents the SNR of our measurement model, and a higher SNR (i.e., a larger λ) improves estimation accuracy, as expected. As λ goes to ∞ (noiseless case), the error bounds approach zero. Conversely, when the signal strength is insufficient to meet the required condition, there is no consistent estimator for $\boldsymbol{y}_i$'s [101].

While the decomposition accuracy reflects how well $\hat{\boldsymbol{y}}_i$ aligns with the true signal $\boldsymbol{y}_i$, we can also access the noise level after decomposition. We next quantify the denoising effect of the decomposition step, which refers to noise reduction in the measurements, i.e., $\| \otimes_{i=1}^I \hat{\boldsymbol{y}}_i - \otimes_{i=1}^I \boldsymbol{y}_i \|_2^2$ is expected to be smaller than $\mathbb{E}\{\|\bar{\boldsymbol{n}}\|_2^2\} = \sigma_t^2 \bar{M}$, as summarized in the following result.

Theorem 5 (Denoising Effect)**.** *Let $\bar{\boldsymbol{y}} = \otimes_{i=1}^I \boldsymbol{y}_i + \bar{\boldsymbol{n}} \in \mathbb{R}^{\bar{M}}$ denote the noisy measurement as in (3.4), where $\boldsymbol{y}_i \in \mathbb{R}^{M_i}$ and $\bar{\boldsymbol{n}}$ has independent zero-mean Gaussian entries with variance σ_{t}^2. Let $\hat{\boldsymbol{y}}_i$ denote the estimate of $\boldsymbol{y}_i$ obtained by solving (3.6) and (3.7). Then, the estimates satisfy*

$$\| \otimes_{i=1}^I \hat{\boldsymbol{y}}_i - \otimes_{i=1}^I \boldsymbol{y}_i \|_2 \leq \sigma_{\mathrm{t}} \left(2 \sum_{i=1}^I \left[3\sqrt{M_i} + \sqrt{\bar{M}/M_i} \right] + 1 + 2\sqrt{\max_{1\leq j\leq I} M_j} \right), \tag{3.17}$$

with probability exceeding $1-3\sum_{i=1}^{I} e^{-M_i}$. *Moreover,*

$$\mathbb{E}\left\{\|\otimes_{i=1}^{I}\hat{\boldsymbol{y}}_i-\otimes_{i=1}^{I}\boldsymbol{y}_i\|_2^2\right\}\approx\sigma_t^2\left(\sum_{i=1}^{I}M_i+1-I\right). \tag{3.18}$$

Proof. See Appendix B.2. □

To gain insights from Theorem 5, suppose that $M_i=\mathcal{O}(M)$ for $i\in[I]$ for some value M. Then, (3.17) shows that the noise level $\|\otimes_{i=1}^{I}\hat{\boldsymbol{y}}_i-\otimes_{i=1}^{I}\boldsymbol{y}_i\|_2^2$ in the decomposed signal is dominated by the term $\mathcal{O}(\sigma_{\mathrm{t}}^2\bar{M}/M_i)=\mathcal{O}(M^{I-1}\sigma_t^2)$ for $I\geq 2$. Then, in the general situation with $I\geq 2$, compared to the noisy signal $\bar{\boldsymbol{y}}$, the noise level reduces from $\mathcal{O}(M^I\sigma_t^2)$ to $\mathcal{O}(M^{I-1}\sigma_t^2)$ after HOSVD. Besides, from (3.18), the average noise level can be approximated as $\mathcal{O}(MI\sigma_t^2)$ when $M>I$. Specifically, the noise level $\mathbb{E}\{\|\bar{\boldsymbol{n}}\|^2\}=\sigma_{\mathrm{t}}^2\bar{M}$ reduces approximately by

$$\frac{\mathbb{E}\left\{\|\otimes_{i=1}^{I}\hat{\boldsymbol{y}}_i-\otimes_{i=1}^{I}\boldsymbol{y}_i\|_2^2\right\}}{\mathbb{E}\{\|\bar{\boldsymbol{n}}\|_2^2\}}\approx\frac{\sum_{i=1}^{I}M_i+1-I}{\bar{M}}<1.$$

The probabilistic bound in Theorem 5 can also be interpreted as an error bound for HOSVD. Consider the simplest case $I=2$ and fix $\bar{M}=M_1M_2$ and σ_{t}. The upper bound in (3.17) (with respect to σ_{t}) can be bounded from below as

$$8(\sqrt{M_1}+\sqrt{M_2})+1+2\sqrt{\max\{M_1,M_2\}}$$
$$\geq 8\left[\min_{\substack{M_1,M_2\\ M_1M_2=\bar{M}}}\sqrt{M_1}+\sqrt{M_2}\right]+1+2\bar{M}^{1/4}=18\bar{M}^{1/4}+1,$$

where equality is achieved when $M_1=M_2$. Therefore, the bound in (3.17) is minimized when the vectors $\boldsymbol{y}_i$'s have the same size.

We present the next corollary on the low complexity approximation, obtained by setting $I=2$ and $M_2=\bar{M}/M_1$ in Theorem 5, noting that it has the same first step as HOSVD.

Corollary 1. *Under the assumptions of Theorem 5, if* $\bar{\boldsymbol{y}}_1$ *and* $\hat{\boldsymbol{y}}_1$ *are obtained from* $\bar{\boldsymbol{y}}$ *using the low complexity approximation (3.8), with probability at least* $1-3(e^{-M_1}+e^{-\bar{M}/M_1})$,

$$\|\hat{\boldsymbol{y}}_1\otimes\bar{\boldsymbol{y}}_1-\bar{\boldsymbol{y}}\|_2\leq\sigma_{\mathrm{t}}\left(1+18\sqrt{\max\{M_1,\bar{M}/M_1\}}\right),$$

and $\mathbb{E}\left\{\|\hat{\boldsymbol{y}}_1\otimes\bar{\boldsymbol{y}}_1-\bar{\boldsymbol{y}}\|_2^2\right\}\approx\sigma_{\mathrm{t}}^2(M_1+\bar{M}/M_1-1)$.

Corollary 1 shows that the first step of low complexity approximation also aids denoising. To intuitively see this, we reorganize $\otimes_{i=1}^{I}\boldsymbol{y}_i$ into a rank-one matrix $\boldsymbol{Y}:=(\otimes_{i=2}^{I}\boldsymbol{y}_i)\boldsymbol{y}_1^{\mathsf{T}}$

$$\bar{\boldsymbol{Y}}=\boldsymbol{Y}+\bar{\boldsymbol{N}}=\left(\otimes_{i=2}^{I}\boldsymbol{y}_i\right)\boldsymbol{y}_1^{\mathsf{T}}+\bar{\boldsymbol{N}},$$

with $\mathrm{vec}\,(\bar{\boldsymbol{Y}})=\bar{\boldsymbol{y}}$, $\mathrm{vec}(\boldsymbol{Y})=\otimes_{i=1}^{I}\boldsymbol{y}_i$, and $\mathrm{vec}\,(\bar{\boldsymbol{N}})=\bar{\boldsymbol{n}}$. Here, the noise term is unstructured $\bar{\boldsymbol{N}}$ and typically has full rank. The first step ($i=1$) of (3.8), yields $\hat{\boldsymbol{y}}_1$ and $\bar{\boldsymbol{y}}_1$, which estimate $\boldsymbol{Y}$

as $\bar{\boldsymbol{y}}_1\hat{\boldsymbol{y}}_1^{\mathsf{T}}$. Comparing this estimate with $\bar{\boldsymbol{Y}}$, we observe that $\bar{\boldsymbol{y}}_1\hat{\boldsymbol{y}}_1^{\mathsf{T}}$ preserves the rank-one structure of the signal. It discards the components that violate the rank-one constraint, which are often attributed to noise, effectively performing denoising. However, a drawback of the low-complexity approximation is that the error from one step can propagate to the subsequent steps, as later computations depend on estimates from the previous steps. In contrast, the error in HOSVD is independent in each subspace, leading to a better decomposition but with a higher computation cost.

3.3.2. Analysis of OffSBL Algorithm

This section discusses the convergence results for OffSBL in Algorithm 5. We note that OffSBL is a two-level iterative algorithm, where the outer EM iteration is given by (3.13) followed by (3.12) and (3.14), and the inner loop updates the grid points via (3.16). We first provide the guarantees for the inner loop for a given rth EM iteration.

Lemma 3 (Convergence of Update)**.** *Consider the alternating update (3.16) in the rth EM iteration. If $\sup_{\psi\in[\psi_1,\psi_{\mathrm{r}}]}\|\boldsymbol{h}(\psi)\|_2^2<\infty$, the sequence $\{g(\boldsymbol{\psi}^{(r,t)})\}_{t=0}^{\infty}$ is non-increasing and convergent. Its iterate $\{\boldsymbol{\psi}^{(r,t)}\}_{t=0}^{\infty}$ adopts at least one limit point.*

Proof. First, we note that there always exists an optimal solution to (3.16) due to the continuity of the function f_{n^*}. The extreme value theorem states that $f_{n^*}(\psi)$ must reach a minimum at least once within the closed and bounded constraint set. The solvability of (3.16) further indicates

$$g(\boldsymbol{\psi}^{(r,t+1)})\le g(\boldsymbol{\psi}^{(r,t)}). \tag{3.19}$$

Further, the cost function $g(\boldsymbol{\psi})$ is bounded from below as

$$\begin{aligned} g(\boldsymbol{\psi}) &= \|\hat{\boldsymbol{y}}-\boldsymbol{H}(\boldsymbol{\psi})\boldsymbol{\mu}_{\mathrm{x}}\|_2^2+\mathrm{tr}(\boldsymbol{\Sigma}_{\mathrm{x}}\boldsymbol{H}(\boldsymbol{\psi})^{\mathsf{H}}\boldsymbol{H}(\boldsymbol{\psi}))\\ &\ge -2\mathrm{Re}(\mathrm{tr}\left\{\boldsymbol{M}\boldsymbol{H}(\boldsymbol{\psi})\right\})\\ &\ge -\mathrm{tr}\left\{\boldsymbol{M}^{\mathsf{H}}\boldsymbol{M}\right\}-\mathrm{tr}\left\{\boldsymbol{H}^{\mathsf{H}}(\boldsymbol{\psi})\boldsymbol{H}(\boldsymbol{\psi})\right\}\\ &\ge -\mathrm{tr}\left\{\boldsymbol{M}^{\mathsf{H}}\boldsymbol{M}\right\}-N\sup_{\psi\in[\psi_1,\psi_{\mathrm{r}}]}\|\boldsymbol{h}(\psi)\|_2^2>-\infty. \end{aligned}$$

Thus, by the monotone convergence theorem, the sequence $\{g(\boldsymbol{\psi}^{(r,t)})\}_{t=0}^{\infty}$ converges. The monotonicity (3.19) also ensures that $\{\boldsymbol{\psi}^{(r,t)}\}_{t=0}^{\infty}$ belongs to the sublevel set of $\boldsymbol{\psi}^{(r,0)}$. Since $g(\boldsymbol{\psi})$ is continuous, its sublevel sets are compact, implying that the sequence $\{\boldsymbol{\psi}^{(r,t)}\}_{t=0}^{\infty}$ adopts at least one limit point. □

We next prove the convergence of our OffSBL algorithm. For the convergence result, we assume the zero-mean Gaussian noise $\boldsymbol{n}$ in (3.10) has a known variance $\sigma^2>0$ and present the convergence based on iterates $\{\boldsymbol{\gamma}^{(r)},\boldsymbol{\psi}^{(r)}\}_{r=0}^{\infty}$, i.e., we do not update σ^2 via (3.14), but use the true value of σ^2 in Algorithm 5. This assumption simplifies deriving a lower bound for the negative log-likelihood function (3.11), which is challenging when $\sigma^2=0$ or treated as a variable. Moreover, assuming $\sigma^2>0$ is standard in SBL analysis [72, 74] and the noiseless settings corresponds the limit where $\sigma^2\to 0$. Thus, the results below are asymptotically applicable to the noiseless case.

Theorem 6 (Convergence Property of OffSBL)**.** *Consider the problem (3.10) where* $\boldsymbol{n}$ *is zero-mean Gaussian noise with known variance* $\sigma^2 > 0$*, solved using OffSBL in Algorithm 5. If* $\sup_{\psi\in[\psi_\mathrm{l},\psi_\mathrm{r}]} \|\boldsymbol{h}(\psi)\|_2^2 < \infty$*, the cost function sequence* $\{\mathscr{L}(\boldsymbol{\gamma}^{(r)},\boldsymbol{\psi}^{(r)})\}_{r=0}^{\infty}$ *converges monotonically to some value* $\mathscr{L}^*$*, and the sequence* $\{\boldsymbol{\gamma}^{(r)},\boldsymbol{\psi}^{(r)}\}_{r=0}^{\infty}$ *converges to a set* $\mathscr{S}^*$ *with* $\mathscr{L}(\boldsymbol{\gamma},\boldsymbol{\psi}) = \mathscr{L}^*$ *for any* $\{\boldsymbol{\gamma},\boldsymbol{\psi}\} \in \mathscr{S}^*$*.*

Proof. We have $Q(\boldsymbol{\gamma}^{(r)},\boldsymbol{\psi}^{(r)}) \le Q(\boldsymbol{\gamma}^{(r+1)},\boldsymbol{\psi}^{(r+1)})$ in the rth EM iteration due to Lemma 3. Then, OffSBL is a generalized EM algorithm and the sequence $\{\mathscr{L}(\boldsymbol{\gamma}^{(r)},\boldsymbol{\psi}^{(r)})\}_{r=0}^{\infty}$ is nonincreasing [71]. Also, $\mathscr{L}(\boldsymbol{\gamma},\boldsymbol{\psi})$ is bounded from below as

$$\mathscr{L}\left(\boldsymbol{\gamma},\boldsymbol{\psi}\right) = \log|\boldsymbol{\Sigma}_\mathrm{y}| + \mathrm{tr}\left(\hat{\boldsymbol{y}}^\mathsf{H}\boldsymbol{\Sigma}_\mathrm{y}^{-1}\hat{\boldsymbol{y}}\right) \ge \log|\boldsymbol{\Sigma}_\mathrm{y}| \ge M\log\sigma^2,$$

where $\boldsymbol{\Sigma}_\mathrm{y}^{-1} = (\sigma^2\boldsymbol{I}_M + \boldsymbol{H}(\boldsymbol{\psi})\boldsymbol{\Gamma}\boldsymbol{H}(\boldsymbol{\psi})^\mathsf{H})^{-1}$ is positive definite for $\boldsymbol{\gamma} \ge \boldsymbol{0}$ and $\boldsymbol{\psi} \in [\psi_\mathrm{l},\psi_\mathrm{r}]^N$. The last inequality is because the eigenvalues of $\boldsymbol{\Sigma}_\mathrm{y}$ are lower bounded by σ^2 [23]. Thus, by the monotone convergence theorem, the sequence $\{\mathscr{L}(\boldsymbol{\gamma}^{(r)},\boldsymbol{\psi}^{(r)})\}_{r=0}^{\infty}$ converges to some value $\mathscr{L}^*$.

Further, the function $\mathscr{L}$ is a coercive function of $\boldsymbol{\gamma}$ and continuous in both $\boldsymbol{\gamma}$ and $\boldsymbol{\psi}$ [72, Lemma 3]. Thus, its sublevel sets are compact. The nonincreasing $\{\mathscr{L}(\boldsymbol{\gamma}^{(r)},\boldsymbol{\psi}^{(r)})\}_{r=0}^{\infty}$ indicates that $\{\boldsymbol{\gamma}^{(r)},\boldsymbol{\psi}^{(r)}\}_{r=0}^{\infty}$ adopts at least one limit point in the level set of $\mathscr{L}^*$, i.e., $\mathscr{S}^*$, which completes the proof. □

We conclude by noting that although the properties of limit points of the EM iterations are unknown, OffSBL offers a sequence over which the negative log-likelihood $\mathscr{L}$ is nonincreasing, aligning with the problem (3.11). We also empirically observe that OffSBL iterates also converge.

3.3.3. Algorithm Complexity

For simplicity and interpretability, we assume $M_i = \mathscr{O}(M)$, $N_i = \mathscr{O}(N)$, and $I < M < N$, where N_i is the number of grid points adopted in the BEM of (3.9). The time complexity of dSBL with HOSVD is $\mathscr{O}\left(R_\mathrm{EM}N^2MI + IM^{I+1}\right)$ while low-complexity approximation has $\mathscr{O}\left(R_\mathrm{EM}N^2MI + M^{I+1}\right)$. Here, R_EM denotes the number of EM iterations. The difference between HOSVD and low complexity approximation-based schemes is roughly of order I. Meanwhile, both have similar space complexity, i.e., $\mathscr{O}(M^I + MN + N^2)$. Also, all sparse recovery subproblems are independent of each other and can thus be solved in parallel. In that case, the time complexity changes to $\mathscr{O}(M^{I+1} + R_\mathrm{EM}N^2M)$ for low-complexity approximation based and $\mathscr{O}(IM^{I+1} + R_\mathrm{EM}N^2M)$ for HOSVD based, while the space complexity becomes $\mathscr{O}(M^I + IMN + IN^2)$. For comparison, the time complexity of AM- and SVD-KroSBL is $\mathscr{O}\big(R_\mathrm{EM}(R_\mathrm{AM}IN^I + (MN)^I)\big)$ and $\mathscr{O}\big(R_\mathrm{EM}(N^{I+1} + (MN)^I)\big)$, respectively, and the space complexity is $\mathscr{O}((MN)^I)$ for both. Here, R_AM denotes the number of AM iterations in AM-KroSBL. Therefore, both the time and space complexities of our algorithm are several orders less than the state-of-the-art KroSBL methods.

3.3.4. Extensions to Similar Structures

We reiterate that the decomposition and the OffSBL algorithms we presented are general algorithmic techniques and can also be applied independently. For instance, for estima-

tion tasks involving Kronecker-structured signals, if there is no grid mismatch and the parameters $\bar{\boldsymbol{\psi}}_i$ lie on a discrete set, one can combine the decomposition algorithm with any sparse recovery algorithm. Similarly, OffSBL is a stand-alone off-grid algorithm for conventional linear inversion problems (i.e., $I = 1$). Moreover, these techniques can be extended to other parameter estimation models, as discussed next.

Superposition of Kronecker-Structured Data

In several wideband orthogonal frequency division multiplexing (OFDM) MIMO systems [21, 22, 48], measurements $\bar{\boldsymbol{y}}$ takes the form

$$\bar{\boldsymbol{y}} = \sum_{u=1}^{U} \left(\otimes_{i=1}^{I} \bar{\boldsymbol{H}}_{u,i,\bar{\boldsymbol{\psi}}_{u,i}} \bar{\boldsymbol{x}}_{u,i} \right) + \bar{\boldsymbol{n}}. \tag{3.20}$$

where the special case of $U = 1$ reduces to (3.4). Here, we can rewrite (3.20) using the tensor form as

$$\boldsymbol{\mathcal{Y}} = \sum_{u=1}^{U} \circ_{i=1}^{I} \left(\bar{\boldsymbol{H}}_{u,i,\bar{\boldsymbol{\psi}}_{u,i}} \bar{\boldsymbol{x}}_{u,i} \right) + \bar{\mathcal{N}}.$$

Thus, each factor $\bar{\boldsymbol{H}}_{u,i,\bar{\boldsymbol{\psi}}_{u,i}} \bar{\boldsymbol{x}}_{u,i}$ for $u = [U]$ and $i \in [I]$ can be obtained by tensor canonical polyadic decomposition under mild conditions for uniqueness of the decomposition [99], followed by OffSBL for unknown parameter estimation.

Non-Kronecker-Structured Sparse Vector

Some applications in [17, 23, 31] lead to the measurement model

$$\bar{\boldsymbol{y}} = \left(\otimes_{i=1}^{I} \bar{\boldsymbol{H}}_{i,\bar{\boldsymbol{\psi}}_i} \right) \bar{\boldsymbol{x}} + \bar{\boldsymbol{n}},$$

where the coefficient vector $\bar{\boldsymbol{x}}$ is not Kronecker-structured. Here, direct decomposition of $\bar{\boldsymbol{y}}$ cannot be applied, but we can use the Kronecker product dictionary in multidimensional BEMs as $\boldsymbol{H} = \otimes_{i=1}^{I} \boldsymbol{H}_i \left(\boldsymbol{\psi}_{(i)} \right)$. Here, vector $\boldsymbol{\psi}_{(i)}$ collects all the variables in the ith BEM dictionary matrix $\boldsymbol{H}_i$. Then, we arrive at the sparse vector problem

$$\bar{\boldsymbol{y}} = \otimes_{i=1}^{I} \boldsymbol{H}_i \left(\boldsymbol{\psi}_{(i)} \right) \boldsymbol{x} + \bar{\boldsymbol{n}}, \tag{3.21}$$

which can be solved using the OffSBL algorithm. Specifically, we adopt a fictitious Gaussian prior distribution with covariance matrix $\boldsymbol{\Gamma} = \operatorname{diag}(\boldsymbol{\gamma}) \in \mathbb{R}^{\bar{N} \times \bar{N}}$ on $\boldsymbol{x}$. Then the estimates of $\boldsymbol{\gamma}$, $\{\boldsymbol{\psi}_{(i)}\}_{i=1}^{I}$ and σ^2 are determined by the type II ML with EM algorithm. The hyperparameter $\boldsymbol{\gamma}$ and the noise variance σ^2 can be similarly obtained using (3.12) and (3.14). We can exploit the Kronecker structure in (3.21) to simplify the alternating minimization of OffSBL for $\{\boldsymbol{\psi}_{(i)}\}_{i=1}^{I}$. The EM update step for $\{\boldsymbol{\psi}_{(i)}\}_{i=1}^{I}$ is equivalent to minimizing

$$g\left(\{\boldsymbol{\psi}_{(i)}\}_{i=1}^{I} \right) = \operatorname{tr}\left(\boldsymbol{H} \boldsymbol{\Sigma} \boldsymbol{H}^{\mathsf{H}} \right) - 2\operatorname{Re}\left\{ \operatorname{tr}\left(\boldsymbol{M} \boldsymbol{H} \right) \right\}.$$

Let ψ_{i^*,n^*} be the n^*th variable of the i^*th BEM dictionary matrix $\boldsymbol{H}_{i^*}$. Then, the alternating minimization step optimizes ψ_{i^*,n^*} with other $\psi_{i,n}$'s being fixed, as in OffSBL. To this

end, for any i^*, we can reorder the Kronecker product as

$$\boldsymbol{H} = \otimes_{i=1}^{I} \boldsymbol{H}_i = \boldsymbol{P}_{i^*}(\boldsymbol{H}_{i^*} \otimes \boldsymbol{S}_{i^*})\boldsymbol{Q}_{i^*},$$

where $\boldsymbol{S}_{i^*} = (\otimes_{i=i^*+1}^{I} \boldsymbol{H}_i) \otimes (\otimes_{i=1}^{i^*-1} \boldsymbol{H}_i)$ is independent of $\boldsymbol{H}_{i^*}$ and $\boldsymbol{P}_{i^*}$ and $\boldsymbol{Q}_{i^*}$ are the corresponding permutation matrices [104]. Thus, the update step for ψ_{i^*,n^*} minimizes

$$\begin{aligned} f_{i^*,n^*}(\psi_{i^*,n^*}) &= \operatorname{tr}\left(\left[\left(\boldsymbol{H}_{i^*}^{\mathsf{H}}\boldsymbol{H}_{i^*}\right) \otimes \left(\boldsymbol{S}_{i^*}^{\mathsf{H}}\boldsymbol{S}_{i^*}\right)\right]\boldsymbol{Q}_{i^*}\boldsymbol{\Sigma}\boldsymbol{Q}_{i^*}^{\mathsf{H}}\right) + \operatorname{tr}([\boldsymbol{H}_{i^*} \otimes \boldsymbol{S}_{i^*}]\boldsymbol{Q}_{i^*}\boldsymbol{M}\boldsymbol{P}_{i^*}) \\ &= 2\mathrm{Re}\{\boldsymbol{v}_{i^*,n^*}^{\mathsf{H}}\boldsymbol{h}_{i^*,n^*}\} + c_{i^*,n^*}\|\boldsymbol{h}_{i^*,n^*}\|_2^2 + \rho_{i_*,n^*}, \end{aligned}$$

where $\boldsymbol{v}_{i^*,n^*}$, c_{i^*,n^*} and ρ_{i_*,n^*} are independent of ψ_{i^*,n^*}. Thus, $f_{i^*,n^*}(\psi_{i^*,n^*})$ can be efficiently minimized with respect to ψ_{i^*,n^*} using a grid search. Also, alternating minimization sequentially updates ψ_{i^*,n^*} for different values of i^* and n^*, unlike dSBL where parallel updates are possible. Thus, OffSBL incurs a higher computational cost here.

3.4. Application: Channel Estimation for IRS-Aided MIMO System

In this section, we explore the application of our algorithm to cascaded channel estimation in an IRS-assisted MIMO system. We consider an uplink MIMO millimeter-wave system with a transmitter MS with T antennas, a receiver BS with R antennas, and a uniform linear array IRS with L elements. Let $\boldsymbol{\Phi}_{\mathrm{MS}} \in \mathbb{C}^{L\times T}$ and $\boldsymbol{\Phi}_{\mathrm{BS}} \in \mathbb{C}^{R\times L}$ denote the geometric narrowband MS-IRS and IRS-BS channel, respectively,

$$\boldsymbol{\Phi}_{\mathrm{MS}} = \sum_{p=1}^{P_{\mathrm{MS}}} \sqrt{\frac{LT}{P_{\mathrm{MS}}}}\beta_{\mathrm{MS},p}\boldsymbol{a}_L(\phi_{\mathrm{MS},p})\boldsymbol{a}_T(\alpha_{\mathrm{MS}})^{\mathsf{H}}, \tag{3.22}$$

$$\boldsymbol{\Phi}_{\mathrm{BS}} = \sum_{p=1}^{P_{\mathrm{BS}}} \sqrt{\frac{RL}{P_{\mathrm{BS}}}}\beta_{\mathrm{BS},p}\boldsymbol{a}_R(\alpha_{\mathrm{BS},p})\boldsymbol{a}_L(\phi_{\mathrm{BS}})^{\mathsf{H}}, \tag{3.23}$$

where we define the steering vector $\boldsymbol{a}_Q(\psi) \in \mathbb{C}^Q$ for an integer Q and angle ψ as

$$\boldsymbol{a}_Q(\psi) = 1/\sqrt{Q}[1, e^{j\frac{2\pi\Delta}{\eta}\cos\psi}, \cdots, e^{j\frac{2\pi\Delta}{\eta}(Q-1)\cos\psi}]^{\mathsf{T}}$$

Here, Δ is the distance between two adjacent elements, and η is the wavelength. We denote the number of rays in the scatter as P_{MS} and P_{BS}. The angles $\phi_{\mathrm{MS},p}$, α_{MS}, $\alpha_{\mathrm{BS},p}$, and ϕ_{BS} denote the pth AoA of the IRS, and the AoD of the MS, the pth AoA of the BS, and the AoD of the IRS, respectively (see [27, Figure 1]). Then, the cascaded MS-IRS-BS channel can be expressed as $\boldsymbol{\Phi}_{\mathrm{BS}}\operatorname{diag}(\boldsymbol{\omega})\boldsymbol{\Phi}_{\mathrm{MS}}$ for any IRS configuration $\boldsymbol{\omega} \in \mathbb{C}^L$. The lth entry of $\boldsymbol{\omega}$ represents the gain and phase shift due to the lth IRS element. Our goal is to estimate the cascaded channel $\boldsymbol{\Phi}_{\mathrm{BS}}\operatorname{diag}(\boldsymbol{\omega})\boldsymbol{\Phi}_{\mathrm{MS}}$, which is a function of angles $\phi_{\mathrm{MS},p}$, α_{MS}, $\alpha_{\mathrm{BS},p}$, and ϕ_{BS}, for a given $\boldsymbol{\omega}$.

We estimate the channel using pilot data transmitted over K time slots. We choose K_{I} IRS configurations, and for each configuration, transmit pilot data $\boldsymbol{G} \in \mathbb{C}^{T\times K_{\mathrm{P}}}$ over K_{P} time slots, where $K = K_{\mathrm{I}}K_{\mathrm{P}}$. For the kth configuration $\boldsymbol{\omega}_k$, the received signal $\boldsymbol{Y}_k =$

$\boldsymbol{\Phi}_{\mathrm{BS}}\,\mathrm{diag}(\boldsymbol{\omega}_k)\boldsymbol{\Phi}_{\mathrm{MS}}\boldsymbol{G}+\boldsymbol{N}_k \in \mathbb{C}^{R\times K_{\mathrm{P}}}$ where $\boldsymbol{N}_k \in \mathbb{C}^{R\times K_{\mathrm{P}}}$ is the noise. Using (3.22) and (3.23), we get

$$\boldsymbol{Y}_k = \zeta\sqrt{L}\boldsymbol{A}_{R,\mathrm{BS}}\boldsymbol{\beta}_{\mathrm{BS}}\boldsymbol{a}_L(\phi_{\mathrm{BS}})^{\mathsf{H}}\,\mathrm{diag}(\boldsymbol{\omega}_k)\times \boldsymbol{A}_{L,\mathrm{MS}}\boldsymbol{\beta}_{\mathrm{MS}}\boldsymbol{a}_T(\alpha_{\mathrm{MS}})^{\mathsf{H}}\boldsymbol{G}+\boldsymbol{N}_k,$$

where $\boldsymbol{A}_{R,\mathrm{BS}} \in \mathbb{C}^{R\times P_{\mathrm{BS}}}$ and $\boldsymbol{\beta}_{\mathrm{BS}} \in \mathbb{C}^{P_{\mathrm{BS}}}$ have $\boldsymbol{a}_R(\alpha_{\mathrm{BS},p})$ and $\beta_{\mathrm{BS},p}$ as their pth column and entry, respectively. Similarly, $\boldsymbol{A}_{L,\mathrm{MS}} \in \mathbb{C}^{R\times P_{\mathrm{MS}}}$ and $\boldsymbol{\beta}_{\mathrm{MS}} \in \mathbb{C}^{P_{\mathrm{MS}}}$ have $\boldsymbol{a}_L(\alpha_{\mathrm{MS},p})$ and $\beta_{\mathrm{MS},p}$ as their pth column and entry, respectively. Also, $\zeta \coloneqq \sqrt{\frac{LRT}{P_{\mathrm{MS}}P_{\mathrm{BS}}}}$. Vectorizing $\boldsymbol{Y}_k$ and using the properties of the Khatri-Rao product [80, Lemma A1] leads to (see [27] for detailed algebraic simplifications)

$$\begin{aligned}\boldsymbol{y}_k &= \zeta\sqrt{L}\left(\boldsymbol{A}_{L,\mathrm{MS}}\boldsymbol{\beta}_{\mathrm{MS}}\boldsymbol{a}_T(\alpha_{\mathrm{MS}})^{\mathsf{H}}\boldsymbol{G}\right)^{\mathsf{T}}\odot\left(\boldsymbol{A}_{R,\mathrm{BS}}\boldsymbol{\beta}_{\mathrm{BS}}\boldsymbol{a}_L(\phi_{\mathrm{BS}})^{\mathsf{H}}\right)\boldsymbol{\omega}_k+\boldsymbol{n}_k\\ &= \left[\boldsymbol{\omega}_k^{\mathsf{T}}\boldsymbol{A}_{L,\mathrm{I}}\boldsymbol{\beta}_{\mathrm{MS}}\right]\left[\zeta\boldsymbol{G}^{\mathsf{T}}\boldsymbol{a}_T^*(\alpha_{\mathrm{MS}})\right]\otimes\left[\boldsymbol{A}_{R,\mathrm{BS}}\boldsymbol{\beta}_{\mathrm{BS}}\right]+\boldsymbol{n}_k,\end{aligned}$$

where $\boldsymbol{A}_{L,\mathrm{I}} \in \mathbb{C}^{L\times P\mathrm{MS}}$ whose pth column is $\boldsymbol{a}_L(\phi_{\mathrm{MS},p}-\phi_{\mathrm{BS}})$, $\odot$ is the Khatri-Rao product, and $(\cdot)^*$ is conjugate. Here, the channel is given by

$$\mathrm{vec}(\boldsymbol{\Phi}_{\mathrm{BS}}\,\mathrm{diag}(\boldsymbol{\omega})\boldsymbol{\Phi}_{\mathrm{MS}}) = \left[\boldsymbol{\omega}^{\mathsf{T}}\boldsymbol{A}_{L,\mathrm{I}}\boldsymbol{\beta}_{\mathrm{MS}}\zeta\boldsymbol{a}_T^*(\alpha_{\mathrm{MS}})\right]\otimes\left[\boldsymbol{A}_{R,\mathrm{BS}}\boldsymbol{\beta}_{\mathrm{BS}}\right]. \tag{3.24}$$

Combining the received signals obtained for the K_{I} configurations, we obtain $\bar{\boldsymbol{y}} \in \mathbb{C}^{RK}$

$$\bar{\boldsymbol{y}} = \left[\boldsymbol{\Omega}^{\mathsf{T}}\boldsymbol{A}_{L,\mathrm{I}}\boldsymbol{\beta}_{\mathrm{MS}}\right]\otimes\left[\zeta\boldsymbol{G}^{\mathsf{T}}\boldsymbol{a}_T^*(\alpha_{\mathrm{MS}})\right]\otimes\left[\boldsymbol{A}_{R,\mathrm{BS}}\boldsymbol{\beta}_{\mathrm{BS}}\right]+\bar{\boldsymbol{n}}, \tag{3.25}$$

where $\boldsymbol{\Omega} \in \mathbb{C}^{L\times K_{\mathrm{I}}}$ with the kth column as $\boldsymbol{\omega}_k$. Therefore, from (3.24), the channel estimation problem reduces to recovering $\boldsymbol{A}_{L,\mathrm{I}}\boldsymbol{\beta}_{\mathrm{MS}}$, $\boldsymbol{a}_T(\alpha_{\mathrm{MS}})$, and $\boldsymbol{A}_{R,\mathrm{BS}}\boldsymbol{\beta}_{\mathrm{BS}}$ from $\bar{\boldsymbol{y}}$ up to a scaling factor, given that $\boldsymbol{G}$ and $\boldsymbol{\Omega}$ are known.

Comparing (3.25) with (3.4), we see the signal model here is the Kronecker product of three terms, i.e., $I=3$. The unknown parameters are AoA or AoD given by $\bar{\boldsymbol{\psi}}_1 \in \mathbb{R}^{P_{\mathrm{MS}}}$, $\bar{\boldsymbol{\psi}}_2 = \alpha_{\mathrm{MS}} \in \mathbb{R}$, and $\bar{\boldsymbol{\psi}}_3 \in \mathbb{R}^{P_{\mathrm{BS}}}$ where the pth entry of $\bar{\boldsymbol{\psi}}_1$ and $\bar{\boldsymbol{\psi}}_3$ are $\phi_{\mathrm{MS},p}-\phi_{\mathrm{BS}}$ and $\alpha_{\mathrm{BS},p}$, respectively. Also, $\psi_{i,\mathrm{l}}=0$ and $\psi_{i,\mathrm{r}}=\pi$ for all values of i. The basis functions are related to the steering vectors as

$$\boldsymbol{h}_1(\psi)=\boldsymbol{\Omega}^{\mathsf{T}}\boldsymbol{a}_L(\psi),\ \boldsymbol{h}_2(\psi)=\boldsymbol{G}^{\mathsf{T}}\boldsymbol{a}_T(\psi),\ \boldsymbol{h}_3(\psi)=\boldsymbol{a}_R(\psi).$$

Correspondingly, we have $\bar{\boldsymbol{x}}_1=\boldsymbol{\beta}_{\mathrm{MS}}$, $\bar{\boldsymbol{x}}_2=\zeta$, and $\bar{\boldsymbol{x}}_3=\boldsymbol{\beta}_{\mathrm{BS}}$. The channel estimation problem is now translated into estimating $\{\bar{\boldsymbol{\psi}}_i\}_{i=1}^3$ and $\{\bar{\boldsymbol{x}}_i\}_{i=1}^3$ from the noisy measurement $\bar{\boldsymbol{y}}$, where our dSBL (Algorithm 5) can be applied.

Using the decomposition step (Line 2) of Algorithm 5, we first decompose $\bar{\boldsymbol{y}}$ into three vectors, $\hat{\boldsymbol{y}}_1 \in \mathbb{C}^{K_{\mathrm{I}}}$, $\hat{\boldsymbol{y}}_2 \in \mathbb{C}^{K_{\mathrm{P}}}$, and $\hat{\boldsymbol{y}}_3 \in \mathbb{C}^{R}$, corresponding to the three terms in the Kronecker product in (3.25). We then use the dictionary for the steering vectors for a given integer Q as

$$\boldsymbol{A}_Q(\boldsymbol{\psi}) = \begin{bmatrix}\boldsymbol{a}_Q(\psi_1) & \boldsymbol{a}_Q(\psi_2) & \dots & \boldsymbol{a}_Q(\psi_N)\end{bmatrix} \in \mathbb{C}^{Q\times N}, \tag{3.26}$$

where $\boldsymbol{\psi}$ captures the unknown angles. This formulation leads to the following three

problems similar to (3.10),

$$\hat{\boldsymbol{y}}_i = \boldsymbol{H}_i(\boldsymbol{\psi}_{(i)})\boldsymbol{x}_i + \boldsymbol{n}_i,\ i = 1,2,3, \tag{3.27}$$

where $\boldsymbol{H}_1(\boldsymbol{\psi}_{(1)}) = \boldsymbol{\Omega}^{\mathsf{T}}\boldsymbol{A}_L(\boldsymbol{\psi}_{(1)})$, $\boldsymbol{H}_2(\boldsymbol{\psi}_{(2)}) = \boldsymbol{G}^{\mathsf{T}}\boldsymbol{A}_T(\boldsymbol{\psi}_{(2)})$, and $\boldsymbol{H}_3(\boldsymbol{\psi}_{(3)}) = \boldsymbol{A}_R(\boldsymbol{\psi}_{(3)})$. We solve them via OffSBL, initializing $\boldsymbol{\psi}_{(i)}$ by sampling the angular domain using N grid angles $\{\theta_n\}_{n=1}^{N}$ such that $\cos(\theta_n) = 2(n-1)/N - 1$. OffSBL provides estimates $(\hat{\boldsymbol{\psi}}_{(i)}, \hat{\boldsymbol{x}}_i)$ of $(\bar{\boldsymbol{\psi}}_i, \bar{\boldsymbol{x}}_i)$, for $i = 1,2,3$. Finally, using (3.24) we compute the channel estimate for a given configuration $\boldsymbol{\omega}$ as

$$\left[\boldsymbol{\omega}^{\mathsf{T}}\boldsymbol{A}_L(\hat{\boldsymbol{\psi}}_{(1)})\hat{\boldsymbol{x}}_1\boldsymbol{A}_T(\hat{\boldsymbol{\psi}}_{(2)})\hat{\boldsymbol{x}}_2\right] \otimes \left[\boldsymbol{A}_R(\hat{\boldsymbol{\psi}}_{(3)})\hat{\boldsymbol{x}}_3\right].$$

Here, the channel estimate is not affected by scaling ambiguity.

We reiterate that OffSBL is a standalone off-grid algorithm applicable to various linear inversion problems (i.e., $I = 1$). One notable example is direction-of-arrival estimation, where K far-field narrowband signals impinge on a uniform array with L elements ($K < L$), resulting in the received signal

$$\hat{\boldsymbol{y}} = \boldsymbol{A}_L(\boldsymbol{\psi})\boldsymbol{x} + \boldsymbol{n},$$

where the BEM dictionary $\boldsymbol{A}_L(\boldsymbol{\psi})$ defined in (3.26) has steering vectors as its columns, $\boldsymbol{\psi}$ captures the AoAs, and $\boldsymbol{x}$ is the sparse vector whose support corresponds to the true AoAs. This formulation is similar to $i = 3$ in (3.27) for the IRS-aided channel estimation problem. In several other applications, the BEM dictionary takes the form $\boldsymbol{B}\boldsymbol{A}_L(\boldsymbol{\psi})$. Specifically, for $i = 1$ and $i = 2$ in (3.27), $\boldsymbol{B}$ corresponds to $\boldsymbol{\Omega}^{\mathsf{T}}$ and $\boldsymbol{G}^{\mathsf{T}}$, respectively. In other cases, $\boldsymbol{B}$ can represent beamformers [21, 22, 105], combiners [21], IRS configurations [106], pilot data [22], or random matrices [107].

3.5. Performance Evaluation

We present numerical results to compare our algorithm with the state-of-the-art methods[2]. We present three sets of results. The first two demonstrate the decomposition step and OffSBL for parameter estimation. The third shows the combined results for IRS-aided channel estimation.

3.5.1. Decomposition-Based Sparse Vector Recovery

In this section, we highlight the advantages of the decomposition step in reducing computational complexity and enhancing the denoising effect. We focus on recovering the Kronecker-structured sparse vector (1.3) using a multidimensional BEM (4.1) in the on-grid setting, without requiring the OffSBL algorithm. By combining the decomposition step with SBL, we demonstrate the benefits of this approach. We compare our method's performance with other methods that do not use decomposition, such as classical SBL [62], classical OMP, AM-KroSBL, and SVD-KroSBL [27]. Specifically, AM- and

[2] Our code is found at https://github.com/YanbinHe/JournalDecomOffGrid. HOSVD is implemented using Tensorlab [108].

SVD-KroSBL only consider the Kronecker-structured support of the sparse vector and do not exploit the Kronecker structure in the nonzero entries as in (1.3).

We set $I = 3$ with $M_i = M$, and $N_i = 12$ for $i = 1,2,3$ in (3.3) and (1.3). So, we have $\boldsymbol{H} = \otimes_{i=1}^{3} \boldsymbol{H}_i$ and $\boldsymbol{x} = \otimes_{i=1}^{3} \boldsymbol{x}_i$ with $\boldsymbol{x}_i \in \mathbb{R}^{12}$. The columns of $\boldsymbol{H}_i \in \mathbb{C}^{M\times 12}$ for $i = 1,2,3$ are the steering vectors evaluated by the grids $\{\theta_n\}_{n=1}^{12}$ defined in Section 3.4. There are four nonzeros in each $\boldsymbol{x}_i$, whose positions are uniformly chosen from the grids and amplitudes are uniformly drawn from $[0.5, 1.5]$. Here, measurement level M is set to be $\{8,9,10\}$, labeled as `Low`, `Medium`, and `High` measurement case, controlling the number of measurements $\bar{M} = M^3$ and the undersampling ratio M^3/N^3. We adopt the additive white Gaussian noise with zero mean whose variance is determined by SNR (dB) $= 10\log_{10} \mathbb{E}\{\|\boldsymbol{H}\boldsymbol{x}\|_2^2/\|\boldsymbol{n}\|_2^2\}$ of $\{5,10,15,20,25,30\}$. Three metrics are considered for performance evaluation: NMSE, SRR, and runtime. Here, we similarly define NMSE and SRR as in (2.26) and (2.27), respectively. We limit the number of iterations for the SBL-based methods (dSBL, cSBL, AM-KroSBL, and SVD-KroSBL) to 200 and prune small entries in hyperparameters for faster convergence.

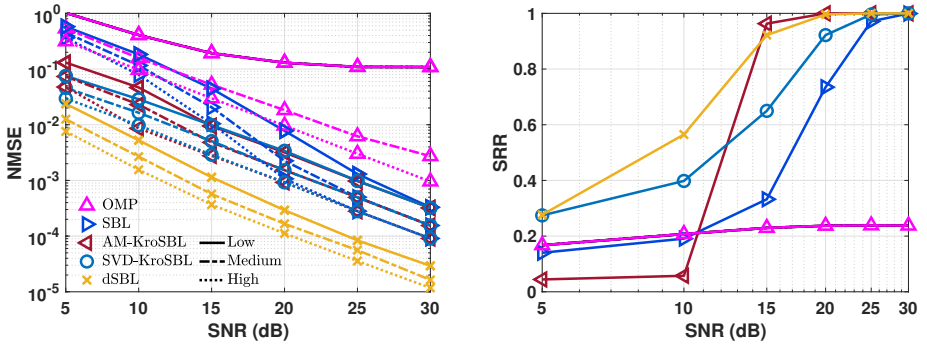


Figure 3.1.: NMSE and SRR of different algorithms as functions of SNR. For the SRR curves, we choose the low measurement case to avoid clutter.

The denoising effect of the decomposition step is shown in Table 3.1. Here, we compare the noise levels of *a*) the original noisy signal $\boldsymbol{y}$ in (3.3), *b*) the signal after decomposition $\hat{\boldsymbol{y}}_{\mathrm{h}} := \otimes_{i=1}^{3} \hat{\boldsymbol{y}}_i$ where $\hat{\boldsymbol{y}}_i$'s are obtained through (3.6) or (3.7), *c*) the signal after decomposition $\hat{\boldsymbol{y}}_{\mathrm{l}} := \otimes_{i=1}^{3} \hat{\boldsymbol{y}}_i$ where $\hat{\boldsymbol{y}}_i$'s are obtained through (3.8), and *d*) the result of (3.18). It can be seen that the noise level for both low complexity approximation and HOSVD is significantly reduced after decomposition. Low complexity approximation can reduce even more noise. It also closely matches the result in (3.18), validating our claim on the denoising effect discussed in Section 3.3.1.

Figure 3.1 shows that with higher SNR and more measurements, all algorithms yield better NMSE and SRR, as expected. Our dSBL algorithm outperforms other methods in NMSE and has the best SRR performance in most cases, demonstrating the efficacy of the decomposition idea. In contrast to the SVD-KroSBL algorithm that uses Kronecker-structured support, dSBL achieves superior NMSE by using the additional Kronecker structure in nonzero entries explicitly enforced via (1.3) through the decomposition step. The relatively lower performance of AM-KroSBL is attributed to its slow convergence,

Table 3.1.: Illustration of denoising with $M = 10$, using the noisy signal $\boldsymbol{y}$, reconstructed signal $\hat{\boldsymbol{y}}_{\mathrm{h}} = \otimes_{i=1}^{3} \hat{\boldsymbol{y}}_i$ using HOSVD, reconstructed signal $\hat{\boldsymbol{y}}_{\mathrm{l}} = \otimes_{i=1}^{3} \hat{\boldsymbol{y}}_i$ using low complexity approximation, and ground truth $\boldsymbol{y}_{\mathrm{o}}$.

Noise level \ SNR	5 dB	10 dB	15 dB	20 dB	25 dB	30 dB
$\|\boldsymbol{y} - \boldsymbol{y}_{\mathrm{o}}\|_2^2$	34.661	9.553	2.853	0.972	0.304	0.090
$\|\hat{\boldsymbol{y}}_{\mathrm{h}} - \boldsymbol{y}_{\mathrm{o}}\|_2^2$	0.991	0.274	0.080	0.027	0.009	0.002
$\|\hat{\boldsymbol{y}}_{\mathrm{l}} - \boldsymbol{y}_{\mathrm{o}}\|_2^2$	0.980	0.272	0.080	0.027	0.009	0.002
From (3.18)	0.929	0.258	0.077	0.026	0.008	0.002

Table 3.2.: Runtime of different schemes in seconds

SNR	5 dB	10 dB	15 dB	20 dB	25 dB	30 dB
OMP	0.599	0.602	0.605	0.603	0.604	0.603
cSBL	8.961	7.470	6.111	5.552	5.397	5.318
AM-KroSBL	8.528	8.516	7.249	5.424	4.520	4.093
SVD-KroSBL	4.534	3.360	2.840	2.668	2.627	2.608
dSBL	0.009	0.005	0.004	0.004	0.004	0.004

Table 3.3.: Runtime in seconds for unknown parameter estimation with $S_1 = 2$ and SNR= 30dB.

M	20	25	30	35	40	45	50
OffSBL	0.602	0.546	0.539	0.541	0.477	0.504	0.522
SBL	0.131	0.137	0.161	0.170	0.171	0.181	0.191
LWSSBL	0.032	0.033	0.035	0.037	0.038	0.041	0.043
OGSBI	0.250	0.269	0.274	0.286	0.290	0.305	0.315

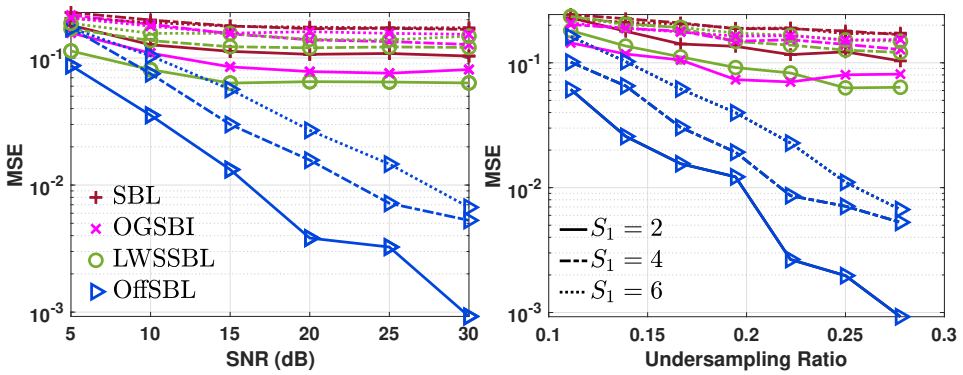
given that we fix the number of EM iterations, as pointed out in [23]. The lower SRR and NMSE observed in the low SNR regime are due to small nonzero values in the estimate at locations where the ground truth is zero. We only include the low measurement case in the SRR result in Figure 3.1, since all three measurement regimes exhibit similar trends; the others were omitted for clarity and better illustration.

Finally, Table 3.2 demonstrates that dSBL requires two-order less runtime than the other competing algorithms, corroborating the computational advantage of our decomposition.

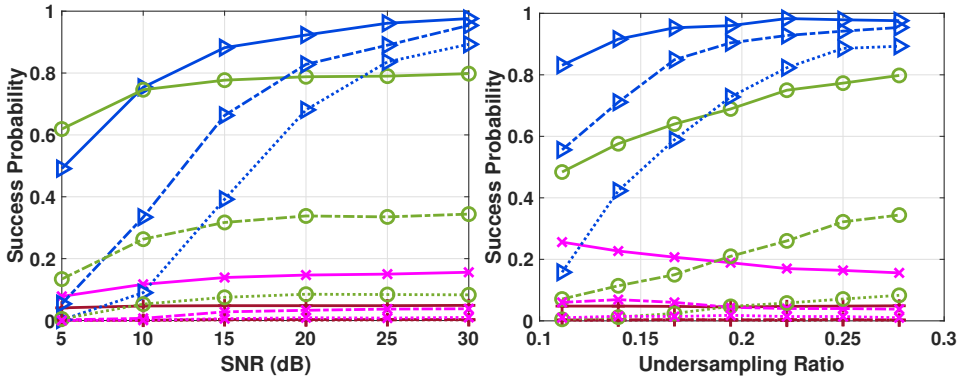
3.5.2. Off-Grid Parameter Estimation

In this section, we apply OffSBL to the unknown parameter estimation problem. The model we consider here is the case in (3.27) with $i = 1$, where the goal is to estimate angles $\boldsymbol{\psi}_{(i)}$ and coefficients $\boldsymbol{x}_i$. The column function is $\boldsymbol{h}_1(\psi) = \boldsymbol{\Omega}^{\mathsf{T}} \boldsymbol{a}_L(\psi)$ with $\boldsymbol{\Omega} \in \mathbb{C}^{L \times M}$, $L = 256$, and M being the number of measurements. Here, we opt for $M = \{20, 25, 30, 35, 40, 45, 50\}$ and controls the undersampling ratio defined as $\frac{M}{N}$ with $N = 180$. The matrix $\boldsymbol{\Omega}$ is randomly generated, whose entries take the form $e^{j\phi}$ where ϕ is drawn from a uniform distribution on $[0, \pi]$. We set the number of unknown parameters (angles) S_1 to be $\{2, 4, 6\}$, and the angles are drawn sequentially from a uniform distribution on $[-0.9, 0.9]$ ensuring a minimal separation of 0.1. The coefficients are drawn from $\mathrm{CN}(0, 2)$.

We use three benchmarks: i) classical (on-grid) SBL, ii) off-grid sparse Bayesian inference (OGSBI) using the first-order Taylor expansion [86], and iii) light-weight sequential SBL (LWSSBL), a state-of-the-art off-grid method using marginal likelihood optimization [92]. In our simulations, we do not provide the number of unknowns S_1 to all algorithms, but only an upper bound $\bar{S}$ of the number of unknowns. In practice, we only



(a) $M = 50$ (b) SNR = 30dB



(c) $M = 50$ (d) SNR = 30dB

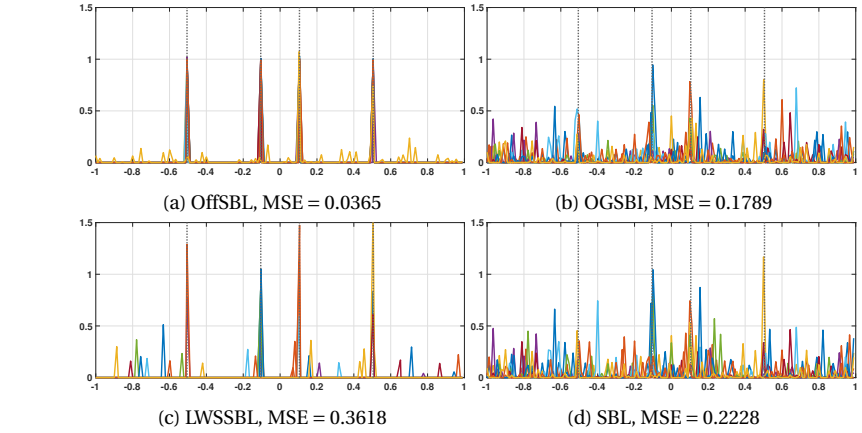
Figure 3.2.: MSE and success probability as a function of SNR and undersampling ratio $\frac{M}{N}$ for unknown parameter estimation with a varying number of unknown parameters S_1 and $N = 180$.

solve the problem (3.16) for the variables corresponding to $\bar{S}$ largest peaks of the hyperparameter $\boldsymbol{\gamma}$ instead of all variables. OffSBL estimates the noise variance using (3.14). SBL and OGSBI can also estimate the noise variance, while noise variance estimation for LWSSBL is not discussed [92]. So for LWSSBL, we set noise variance estimate as $0.1\|\bar{\boldsymbol{y}}\|_2^2/M$ as in [92]. We choose SNR $= 10\log_{10}\mathbb{E}\{\|\bar{\boldsymbol{H}}_{\bar{\boldsymbol{\psi}}}\bar{\boldsymbol{x}}\|_2^2/\|\bar{\boldsymbol{n}}\|_2^2\}$ as $\{5,10,15,20,25,30\}$ in dB. The performance metrics are mean squared error (MSE) and success probability, where

$$\text{MSE} = \mathbb{E}\left\{\frac{1}{S_1}\sum_{s=1}^{S_1}(\bar{\psi}_s - \hat{\psi}_s)^2\right\},$$

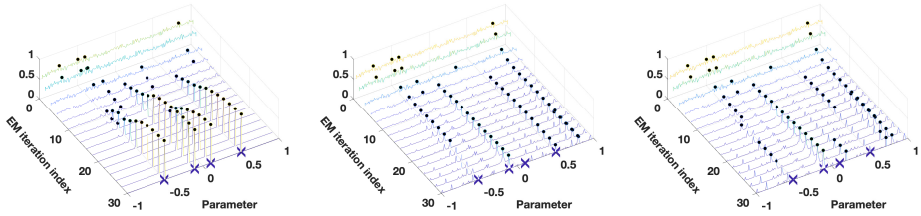
with expectation taken over 10^3 independent trials. Here, $\bar{\psi}_s$ and $\hat{\psi}_s$ denote the true value and the estimation, respectively. The success probability is defined as the fraction of trials with MSE smaller than 10^{-6}.

We compare MSE and recovery probability for different SNRs and undersampling ratios in Figure 3.2. We see that higher SNR and more measurements facilitate all algorithms, except OGSBI in Figs. 3.2(b) and 3.2(d). This is because OGSBI cannot effectively optimize



(a) OffSBL, MSE = 0.0365 (b) OGSBI, MSE = 0.1789

(c) LWSSBL, MSE = 0.3618 (d) SBL, MSE = 0.2228



(e) OffSBL (f) OGSBI (g) SBL

Figure 3.3.: Pseudospectrum $\boldsymbol{\gamma}$ of different algorithms in the worst-case scenario analysis. (a)-(d): pseudospectrum $\boldsymbol{\gamma}^{(1000)}$ of the 1000th EM iteration. Different colors mean different independent realizations of Gaussian noise. (e)-(g): dynamic evolution of pseudospectrum $\{\boldsymbol{\gamma}^{(r)}\}_{1\leq r\leq 30}$ for the EM iterations of EM-based SBL algorithms. Black dot: grid points chosen to be updated. Cross: true values. Although SBL operates as an on-grid method, we still include the grid points that correspond to the top four peaks, sorely for comparison purposes.

the grid points in this setting, as we show later in Figure 3.3. Among all candidates, our OffSBL has the best performance in both MSE and recovery probability in most cases. An exception is SNR = 5dB where LWSSBL has a higher success probability. However, LWSSBL often produces larger errors when it fails, making OffSBL superior in MSE.

In Figure 3.3, we present a worst-case scenario study. We set $M = 60$ measurements and SNR = 30dB. The unknowns are $[-0.5050, -0.1050, 0.1050, 0.5050]$, shown as vertical dashed lines in Figs. 3.3(a)-3.3(d). These values are intentionally selected to be midway between two grids to create a challenging case for grid optimization. All the coefficients are set to one. We provide the number of unknowns to all the algorithms but not the noise variance. We perform 10^3 EM iterations to facilitate the convergence of all algorithms. The input of all algorithms is the same noiseless signal but with ten independent Gaussian noise realizations. We plot the final pseudospectrum (hyperparameter $\boldsymbol{\gamma}^{(1000)}$) after 10^3 EM iterations for ten noise realizations in Figs. 3.3(a)-3.3(d) with different colors.

Comparing the different algorithms, we note that OffSBL consistently recovers all parameters, with minimal amplitude spikes appearing in the pseudospectrum corresponding to parameters other than the true values. LWSSBL also recovers the unknowns but with

a lower success probability. LWSSBL exhibits more peaks at parameters other than the true values, implying that it is more prone to being misled by incorrect columns in the dictionary due to its greedy nature. In contrast, while OffSBL takes longer to reach the final result (see Table 3.3), evaluating all columns rather than proceeding greedily reduces the risk of being misled by incorrect columns.

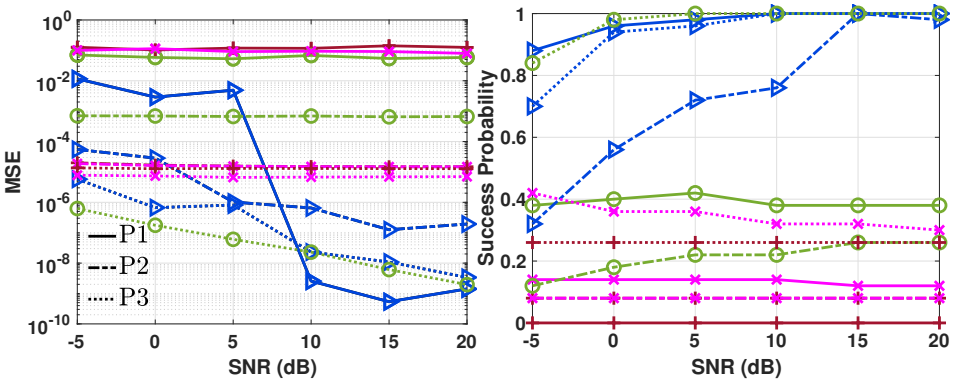
Further, there is little difference between OGSBI and the on-grid benchmark SBL, indicating that the first-order approximation is less effective in this case. However, OGSBI has some improvement over SBL as reflected by a lower MSE. These findings also highlight that algorithms relying on on-grid SBL for rough estimates and then refining peaks are likely to fail, as on-grid SBL often doesn't provide a reliable starting point, with peaks rarely matching the true parameters. This is likely due to the dictionary's structure, which takes the form $\boldsymbol{\Omega}^{\mathsf{T}}\boldsymbol{A}_L(\boldsymbol{\psi})$ for some integer L. When $\boldsymbol{\Omega}^{\mathsf{T}} \in \mathbb{C}^{M\times L}$ has fewer rows than columns ($M < L$), the compression effect from multiplication by $\boldsymbol{\Omega}^{\mathsf{T}}$ can lead to information loss, creating a challenging setting for off-grid sparse recovery [109]. However, in many applications, such as IRS channel estimation, where the value of M represents the number of time slots, M is typically limited. Thus, integrating grid updates into the EM iteration, as implemented in OffSBL, is essential.

Figs. 3.3(e)-3.3(g) present the pseudospectrum $\{\boldsymbol{\gamma}^{(r)}\}_{1\le r\le 30}$ for OffSBL, OGSBI, and SBL, along with the grid points that are updated dynamically throughout the EM iteration. Although SBL is an on-grid method, we pinpoint the grids of the top four peaks. All algorithms start from the same $\boldsymbol{\gamma}$. Our OffSBL demonstrates superior optimization of grid points, identifying the correct values and amplitudes, whereas SBL and OGSBI do not reveal the true parameters. The pseudospectrum highlights the effectiveness of our grid adjustment.

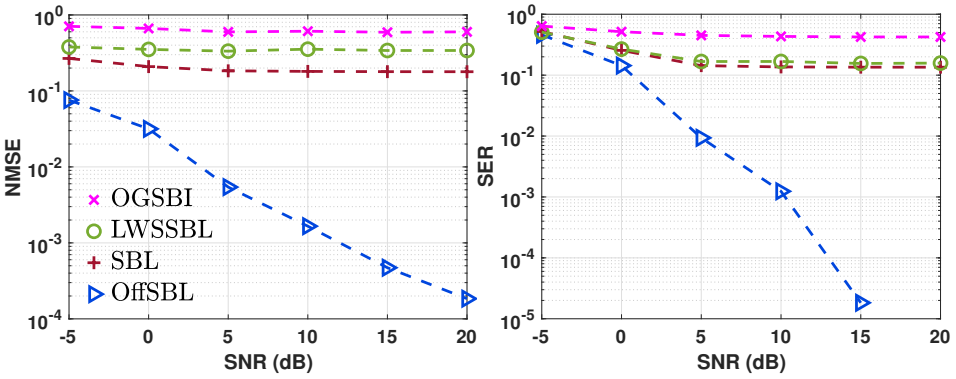
3.5.3. Application: IRS-Aided Wireless Channel Estimation

We focus on the IRS-aided channel estimation problem, as described in Section 3.4. Here, we first use the decomposition step and then turn to the BEM and apply OffSBL separately for $i = 1,2,3$ in (3.27). Thus, the channel estimation scheme can be viewed as a collective evaluation of the decomposition and OffSBL. For benchmarking, we apply the same decomposition step, and then solve (3.27) using the same algorithms as in Section 3.5.2. For simplicity, we denote the problem (3.27) with $i = 1,2,3$ as P1, P2, and P3, respectively.

For IRS-aided channel estimation, we use $R = 16$ BS antennas, $T = 6$ MS antennas, $L = 256$ IRS elements. We consider only one path between the BS and IRS [110–113], as the IRS is typically mounted in locations with fewer obstacles [8, 114], and the line-of-sight path is generally much stronger than the other paths. Therefore, we take $P_{\mathrm{BS}} = 1$ and $P_{\mathrm{MS}} = 3$. The IRS configuration entries $\{\boldsymbol{\omega}_k\}_{k=1}^{K_{\mathrm{I}}}$ are $1/\sqrt{L}e^{j\phi}$ where ϕ is drawn uniformly randomly from $[0,\pi]$ with $K_{\mathrm{I}} = 40$. We send $K_{\mathrm{P}} = 20$ pilot signals for each IRS configuration. Our OffSBL algorithm uses the dictionaries in P1, P2, and P3 with $N_1 = 180$, $N_2 = 50$, and $N_3 = 50$, respectively. The other algorithms set $N_1 = 180$, $N_2 = 150$, and $N_3 = 150$ grid points. The channel gains β_{BS} and $\{\beta_{\mathrm{MS},p}\}_{p=1}^{P_{\mathrm{MS}}}$ in (3.22) and (3.23) are drawn from standard complex Gaussian [82]. We randomly draw α_{MS}, $\{\phi_{\mathrm{MS},p}\}_{p=1}^{P_{\mathrm{MS}}}$, ϕ_{BS}, and α_{BS} from uniform distribution in $[0.3,0.5]$, $[-0.2,0.2]$, $[0.3,0.5]$, and $[0,0.5]$, respectively. We also assume that the angles are separated by at least 0.07. We opt for SNR $\{-5,0,5,10,15,20\}$ in dB.



(a) Angle estimation performance



(b) Channel estimation performance

Figure 3.4.: IRS-aided channel estimation. (a): MSE and recovery success probability for angle estimation in P1, P2, and P3. (b): NMSE of IRS-aided channel estimation and SER of different algorithms as functions of SNR.

Also, additional channel estimation metrics are NMSE and SER over 10^6 16-QAM symbols decoded using the estimated channel, where NMSE is defined in (2.42).

We first examine the angle estimation results in Figure 3.4. It can be seen that our OffSBL can achieve the best performance in solving P1 and P2 except for the low SNR case for P2. P3 reduces to the normal DoA estimation problem, where LWSSBL exhibits superior recovery ability. However, at higher SNRs, our algorithm is able to achieve comparable performance. As evident from the NMSE and SER plots, OffSBL consistently recovers the true angles and accurately retrieves the coefficients, leading to the best NMSE and SER. Although other algorithms can perform well in solving P2 and P3, the significant recovery errors in P1 affect the overall accuracy of channel retrieval.

3.6. Chapter Summary

In this chapter, we addressed the problem of estimating a Kronecker-structured sparse vector from a underdetermined linear inversion problem. Such a problem arose from

the joint estimation of unknown parameters and coefficients from Kronecker-structured measurements. Leveraging the Kronecker structure, we decomposed the problem into smaller independent subproblems. Each subproblem was solved with EM-based SBL integrated with a novel grid optimization method to reduce grid mismatch. We provided a theoretical analysis of the error bound for the decomposition step and established the algorithm's convergence. Our decomposition step also reduces the noise level in the measurements, which was also analyzed theoretically. Numerical results demonstrated that the decomposition step reduces complexity, while the grid optimization improves accuracy. After the investigation on Kronecker-structured sparsity, in the next chapter, we study a more general problem where the standard sparsity, hierarchical sparsity, and Kronecker-supported sparsity are considered.

4

Hierarchical Multi-Stage Recovery Framework for Kronecker Compressed Sensing

Having addressed the Kronecker compressed sensing problem with specific sparsity patterns, we now focus on a general framework for multiple sparsity patterns. In this chapter, we first introduce the *hierarchical view* of the Kronecker compressed sensing, showing that the Kronecker product measurement matrix probes the sparse vector from different levels, following a block-wise and hierarchical structure. Leveraging this insight, we develop a *versatile* multi-stage sparse recovery algorithmic framework and demonstrate its application to three different sparsity models: standard, hierarchical, and Kronecker-supported. We further analyze the RIP of Kronecker product matrices under different sparsity models, and provide theoretical recovery guarantees for our multi-stage algorithm. Simulations demonstrate that our method achieves comparable recovery performance to other state-of-the-art techniques while substantially reducing runtime owing to the hierarchical, multi-stage recovery process.

4.1. Introduction

In this section, we focus on the Kronecker compressed sensing problem

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{n} = (\boldsymbol{H}_I \otimes \boldsymbol{H}_{I-1} \otimes \cdots \otimes \boldsymbol{H}_1)\,\boldsymbol{x} + \boldsymbol{n} = \left(\otimes_{i=I}^{1} \boldsymbol{H}_i\right)\boldsymbol{x} + \boldsymbol{n}. \tag{4.1}$$

We note that (4.1) is mathematically equivalent to (1.2). The only difference is notational where we reverse the order of the Kronecker product to facilitate the tensor operation, as we will see later. Besides, instead of considering a structured $\boldsymbol{x}$ as in Chapter 2 and 3, we here assume no structure in the *unknown* sparse vector $\boldsymbol{x}$ yet, since we aim for a general framework for Kronecker compressed sensing with three prevalent sparsity models, i.e., the standard sparsity, the hierarchical sparsity, and the Kronecker-supported sparsity. In the next, we first discuss the prior efforts on recovering standard sparse, hierarchical sparse, and Kronecker-supported sparse vector from a Kronecker compressed sensing problem, and then present the contributions of this chapter.

4.1.1. Related Works

The Kronecker product measurement matrix is introduced for compressed imaging in [20]. Kronecker compressed sensing is formalized in [16] tailored to hyperspectral imaging, with an RIP analysis for Kronecker compressed sensing with standard sparsity [16, 55]. It bounds the RIC of the Kronecker product using the RIC of factor matrices $\boldsymbol{H}_i$. However, the recovery algorithm fails to leverage the Kronecker structure in $\boldsymbol{H}$. To leverage this structure, KroOMP [18] adopts tensor operations. Nonetheless, it still incurs a high complexity of $\mathcal{O}(N^I)$, and lacks theoretical analysis. Two algorithms are presented in [32, 56]: one uses tensor unfolding for sequential recovery in dimension, and the other uses approximate Tucker decomposition to recover along each dimension for compressible image and video representation and recovery. Still, both approaches are limited to standard sparsity. The unfolding-based approach is transformed into multiple *independent* subproblems for hyperspectral imaging in [26]. Yet, it fails to exploit joint sparsity patterns and is not immediately extendable to other sparsity patterns.

Kronecker compressed sensing with structured sparse recovery is also investigated in the literature. For *hierarchically sparse* vectors, HiHTP is discussed in [57], which adapts

classic HTP with a tailored RIP and coherence analysis for channel estimation for massive MIMO systems [115]. However, it fails to incorporate the Kronecker structure in $\boldsymbol{H}$, leading to higher computational costs. For *Kronecker-supported sparsity*, both greedy and Bayesian methods have been explored. An OMP-based algorithm offers reduced complexity [17, 18] but performs poorly in noisy settings [23]. Bayesian algorithms, designed for applications such as hyperspectral image processing [42] and wireless communication [23, 31, 53], use a structured prior distribution. They suffer from poor generalization and high complexity [59]. Besides, both OMP-based and Bayesian algorithms lack theoretical guarantees.

To summarize, existing approaches reveal several literature gaps. First, Kronecker compressed sensing methods mostly ignore the structure of $\boldsymbol{H}$, relying on generic solvers. Second, current methods are largely tailored to a single sparsity pattern and cannot be generalized. Third, many methods suffer from high computational complexity. To address these gaps, in this chapter, we specifically design a method to leverage the Kronecker structure through tensor operations to reduce the complexity, which also provides a unified framework for multiple patterns. Furthermore, no prior work offers a unified RIP analysis of Kronecker product matrices across various sparsity patterns, nor a recovery framework for different sparsity patterns with RIP-based guarantees, which are our central theoretical contributions.

4.1.2. Contributions

This chapter introduces a novel *hierarchical view* on Kronecker compressed sensing, showing how its dimension-wise measuring structure can be used to design and analyze efficient recovery methods to exploit structured sparsity efficiently. Our main contributions are as follows:

- *Hierarchical View*: We establish that when measuring via Kronecker product matrices, each factor matrix in the Kronecker product captures the vector at a distinct hierarchical level. It provides a unified perspective for handling different sparsity models within a single framework.

- *Unified Algorithm*: We design a multi-stage sparse recovery algorithm using the hierarchical view. By leveraging the Kronecker structure of $\boldsymbol{H}$ through tensor operation and investigating the underlying structure, our method achieves a significant complexity reduction, e.g., reducing from $\mathcal{O}((MN)^I)$ [23] to $\mathcal{O}(MN^I)$ regarding Kronecker-supported sparse vector recovery, and accommodates the mentioned sparsity patterns within a single, flexible framework.

- *Theoretical Guarantees*: We establish a unified RIP analysis for Kronecker compressed sensing covering the standard, hierarchical, and Kronecker-supported sparsity. It proves that sparsity at each hierarchical level, rather than total sparsity, drives the recovery. Our result improves the RIP-based bound for Kronecker compressed sensing with standard sparsity and provides a cohesive understanding of structured sparsity. We also provide a RIP-based recovery guarantee for our unified algorithm.

Next, we first focus on the case with $I = 2$ in Section 4.2. This is the most fundamental

scenario, providing a concrete basis for understanding the multidimensional analysis which is then demonstrated in Section 4.3, considering $I \geq 2$.

4.2. Special Case: Two Factor Matrices

Our hierarchical view relies on the structure of the measurement system in the Kronecker compressed sensing model (4.1), where the Kronecker product matrix $\boldsymbol{H}$ has two factor matrices, $\boldsymbol{H}_2$ and $\boldsymbol{H}_1$. These factor matrices operate at different levels: $\boldsymbol{H}_2$ captures block-level while $\boldsymbol{H}_1$ focuses on intra-block, following a hierarchical structure. To illustrate this view, we first partition $\boldsymbol{x}$ into N_1 blocks of size N_2. Let the ith block be denoted by $\boldsymbol{x}_i \in \mathbb{R}^{N_2}$. Then, we can rearrange $\boldsymbol{x}$ into $\boldsymbol{X} \in \mathbb{R}^{N_2 \times N_1}$ such that the ith column of $\boldsymbol{X}$ is $\boldsymbol{x}_i$. Similarly, we can rearrange $\boldsymbol{y}$ and $\boldsymbol{n}$ into matrices $\boldsymbol{Y} \in \mathbb{R}^{M_2 \times M_1}$ and $\boldsymbol{N} \in \mathbb{R}^{M_2 \times M_1}$, such that $\mathrm{vec}(\boldsymbol{Y}) = \boldsymbol{y}$ and $\mathrm{vec}(\boldsymbol{N}) = \boldsymbol{n}$, respectively. Here, $\mathrm{vec}(\cdot)$ denotes vectorization. Since $\mathrm{vec}(\boldsymbol{H}_1 \boldsymbol{X} \boldsymbol{H}_2^{\mathsf{T}}) = (\boldsymbol{H}_2 \otimes \boldsymbol{H}_1)\,\boldsymbol{x}$, we obtain

$$\boldsymbol{Y}^{\mathsf{T}} = \boldsymbol{H}_2\,(\boldsymbol{H}_1 \boldsymbol{X})^{\mathsf{T}} + \boldsymbol{N}^{\mathsf{T}}, \tag{4.2}$$

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

The above perspective can also be interpreted directly from (4.1). Recall that the Kronecker product matrix $\boldsymbol{H}$ possesses a column-block structure with a repetitive block pattern. Here, each block of columns is obtained by taking the Kronecker product of a column of $\boldsymbol{H}_2$ with $\boldsymbol{H}_1$. Also, the column-block structure of $\boldsymbol{H}$ matches with the blocks of $\boldsymbol{x}$. Hence, in this hierarchical framework, $\boldsymbol{H}_1$ first measures each block of $\boldsymbol{x}$. The resulting measurements of all blocks are then processed by $\boldsymbol{H}_2$, capturing information at a higher, global level. The relation (4.2) explicitly captures this measurement model, where intra-block measurement by $\boldsymbol{H}_2$ is followed by block-level measurement by $\boldsymbol{H}_1$.

4.2.1. Algorithm Development

The hierarchical view in the Kronecker compressed sensing problem (4.1) indicates that the sparse vector $\boldsymbol{x}$ 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 $(\boldsymbol{H}_1 \boldsymbol{X})^{\mathsf{T}}$ as unknown and solves (4.2). Also, $\boldsymbol{H}_1 \boldsymbol{X}$ exhibits a column-wise sparsity pattern, i.e., a nonzero column of $\boldsymbol{H}_1 \boldsymbol{X}$ corresponds to a nonzero block of $\boldsymbol{x}$ while a zero column corresponds to a zero block. Thus, $(\boldsymbol{H}_1 \boldsymbol{X})^{\mathsf{T}}$ is a row sparse matrix and recovering $(\boldsymbol{H}_1 \boldsymbol{X})^{\mathsf{T}}$ from (4.2) can be formulated as a multiple measurement vector (MMV) problem. It can be solved using any MMV variants of compressed sensing algorithms, such as OMP or SBL.

Let the estimate of $(\boldsymbol{H}_1 \boldsymbol{X})^{\mathsf{T}}$ after the first step be $\tilde{\boldsymbol{X}}$. In the next step, we treat $\tilde{\boldsymbol{X}}$ as measurements and recover $\boldsymbol{X}$ from

$$\tilde{\boldsymbol{X}}^{\mathsf{T}} = \boldsymbol{H}_1 \boldsymbol{X} + \tilde{\boldsymbol{N}}, \tag{4.3}$$

where $\tilde{\boldsymbol{N}}$ represents noise. For standard and hierarchical sparsity models, the support of the different blocks of $\boldsymbol{x}$ (or columns of $\boldsymbol{X}$) are different. So, we treat problem (4.3) 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 (4.3) is an MMV problem because the support is common across different blocks. The resulting algorithm, named Two-Stage Recovery (TSR), is summarized in Algorithm 7.

Algorithm 7 Two-stage sparse recovery

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

1: Re-order $\boldsymbol{y}$ to obtain $\boldsymbol{Y}$
2: Solve (4.2) to obtain $\tilde{\boldsymbol{X}}$ using any MMV algorithm
3: Solve (4.3) for $\boldsymbol{X}$ using any recovery algorithm (use MMV variant for Kronecker-supported sparsity model)

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

4.3. Generalization to Higher Order

Similarly to the $I = 2$ case, hierarchical view interprets the measurement matrix as probing the signal's sparsity across multiple block-wise and hierarchical levels. Specifically, factor matrices $\{\boldsymbol{H}_i\}_{i=I}^{1}$ operate at different levels: for any p, q with $p > q$, $\boldsymbol{H}_p$ captures higher block-level sparsity while $\boldsymbol{H}_q$ focuses on lower block-level sparsity, following a hierarchical structure. To illustrate this, we first introduce the hierarchical block partition of a sparse vector $\boldsymbol{x} \in \mathbb{R}^{\tilde{N}}$.

4.3.1. Hierarchical Partition

We first partition $\boldsymbol{x}$ in (4.1) into N_I equal-length blocks, denoting the Ith level blocks as $\{\boldsymbol{x}_{(n_I)}\}_{n_I=1}^{N_I} \in \mathbb{R}^{\prod_{i=1}^{I-1} N_i}$. Each $\boldsymbol{x}_{(n_I)}$ is further partitioned into N_{I-1} blocks, denoted as $(I-1)$th level blocks $\{\boldsymbol{x}_{(n_{I-1},n_I)}\}_{n_{I-1}=1}^{N_{I-1}} \in \mathbb{R}^{\prod_{i=1}^{I-2} N_i}$. We continue until we reach blocks of length N_1 at the second level. The first-level blocks are the individual entries of $\boldsymbol{x}$.

We use $\boldsymbol{x}_{\mathrm{n_j}}$ to denote a block in the jth level with length $\prod_{i=1}^{j-1} N_i$ and encapsulation $\mathrm{n_j} := (n_j, \cdots, n_{I-1}, n_I)$. An encapsulation $\mathrm{n_j} := (n_j, \cdots, n_{I-1}, n_I)$ can be viewed as a coordinate for blocks in this hierarchical block structure. Also, set $[\![\boldsymbol{x}_{\mathrm{n_j}}]\!]$ contains all N_j *child blocks* that share the same parent block at the level $j+1$ as that of $\boldsymbol{x}_{\mathrm{n_j}}$. We illustrate a hierarchical partition for $\boldsymbol{x} \in \mathbb{R}^{40}$ in Figure 4.1, where $[\![\boldsymbol{x}_{(1,3)}]\!] = [\![\boldsymbol{x}_{(2,3)}]\!] = \{\boldsymbol{x}_{(1,3)}, \boldsymbol{x}_{(2,3)}\}$ as they share the parent $\boldsymbol{x}_{(3)}$.

4.3.2. Hierarchical View

We first focus on the noiseless version of (4.1) reformulated using tensors,

$$\mathbf{T} := \mathbf{Y} = \mathbf{X} \times_1 \boldsymbol{H}_1 \cdots \times_I \boldsymbol{H}_I,$$

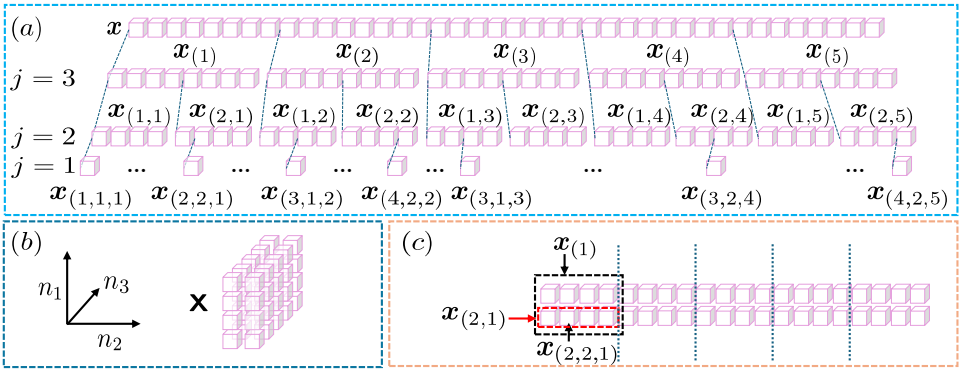


Figure 4.1.: (a) Hierarchical partition for $\boldsymbol{x} \in \mathbb{R}^{40}$ with $I = 3, N_1 = 5, N_2 = 2, N_1 = 4$, and $\tilde{N} = 40$. (b) Reordered tensor $\mathbf{X}$. (c) Mode unfolding $\boldsymbol{X}_{(2)}$ and the relation between the n_{I-1}th row within the n_Ith column block and the $(I-1)$th level child block $\boldsymbol{x}_{n_{\mathrm{I}-1}}$ with $n_{\mathrm{I}-1} = (2,1)$ and $I = 3$.

where the first mode unfolding satisfy $\mathrm{vec}(\boldsymbol{X}_{(1)}) = \boldsymbol{x}$ and $\mathrm{vec}(\boldsymbol{T}_{(1)}) = \mathrm{vec}(\boldsymbol{Y}_{(1)}) = \boldsymbol{y}$. Unfolding $\mathbf{T}$ on the Ith mode leads to

$$\boldsymbol{T}_{(I)} = \boldsymbol{H}_I \boldsymbol{X}_{(I)} \left(\otimes_{i=I-1}^{1} \boldsymbol{H}_i^{\mathsf{T}} \right) = \boldsymbol{H}_I \boldsymbol{U}_I \in \mathbb{R}^{M_I \times \prod_{i=I-1}^{I} M_i}.$$

Here, $\boldsymbol{U}_I = \boldsymbol{X}_{(I)} \left(\otimes_{i=I-1}^{1} \boldsymbol{H}_i^{\mathsf{T}} \right) \in \mathbb{R}^{N_I \times \prod_{i=I-1}^{I} M_i}$ and $\boldsymbol{X}_{(I)} \in \mathbb{R}^{N_I \times \prod_{i=I-1}^{I} N_i}$ whose n_Ith row is the Ith level block $\boldsymbol{x}_{\mathrm{n_I}}$ with $\mathrm{n_I} = (n_I)$. Therefore, matrix $\boldsymbol{H}_I$ acts on $\boldsymbol{U}_I$, and a zero row in $\boldsymbol{U}_I$ indicates that the corresponding Ith level block is entirely zero. Hence, matrix $\boldsymbol{H}_I$ captures the sparsity pattern of the Ith-level blocks.

For the $(I-1)$th level, we fold $\boldsymbol{U}_I$ into a new tensor $\mathbf{T}$, whose Ith mode unfolding $\boldsymbol{T}_{(I)} = \boldsymbol{U}_I$, as

$$\mathbf{T} = \mathbf{X} \times_1 \boldsymbol{H}_1 \cdots \times_{I-1} \boldsymbol{H}_{I-1} \times_I \boldsymbol{I}_{N_I}.$$

Unfolding $\mathbf{T}$ along its $(I-1)$th mode gives

$$\boldsymbol{T}_{(I-1)} = \boldsymbol{H}_{I-1} \boldsymbol{X}_{(I-1)} \left(\boldsymbol{I}_{N_I} \otimes \left(\otimes_{i=I-2}^{1} \boldsymbol{H}_i^{\mathsf{T}} \right) \right) = \boldsymbol{H}_{I-1} \boldsymbol{U}_{I-1} \in \mathbb{R}^{M_{I-1} \times N_I \prod_{i=I-2}^{I} M_i}.$$

Here, $\boldsymbol{X}_{(I-1)} \in \mathbb{R}^{N_{I-1} \times \tilde{N}/N_{I-1}}$ has N_I column blocks, with n_Ith block corresponding to $\boldsymbol{x}_{\mathrm{n_I}}$ with $\mathrm{n_I} = (n_I)$. Within the n_Ith column block, the n_{I-1}th row is the $(I-1)$th level child block $\boldsymbol{x}_{n_{\mathrm{I}-1}}$ with $n_{\mathrm{I}-1} = (n_{I-1}, n_I)$, as illustrated in Figure 4.1c. The Kronecker product $\boldsymbol{I}_{N_I} \otimes \left(\otimes_{i=I-2}^{1} \boldsymbol{H}_i^{\mathsf{T}} \right)$ is a block matrix and preserves the column block structure in $\boldsymbol{U}_{I-1}$. Column blocks of $\boldsymbol{U}_{I-1}$ are associated with I-level blocks, and the rows of a column block correspond to the $(I-1)$ level blocks. Hence, the zero rows in each column block of $\boldsymbol{U}_{I-1}$ indicate that the corresponding $(I-1)$th level blocks are entirely zero. Therefore, $\boldsymbol{H}_{I-1}$ captures the sparsity pattern at the $(I-1)$th-level blocks.

For a general jth level, we define $\mathbf{T} = \mathbf{X} \times_1 \boldsymbol{H}_1 \times_2 \boldsymbol{H}_2 \cdots \times_j \boldsymbol{H}_j \times_{j+1} \boldsymbol{I}_{N_{j+1}} \cdots \times_I \boldsymbol{I}_{N_I}$, and

$$\boldsymbol{T}_{(j)} = \boldsymbol{H}_j \boldsymbol{X}_{(j)} \left(\boldsymbol{I}_{\prod_{i=I}^{j+1} N_i} \otimes \left(\otimes_{i=j-1}^{1} \boldsymbol{H}_i \right) \right)^{\mathsf{T}} \in \mathbb{R}^{M_j \times \prod_{i=I}^{j+1} N_i \prod_{i=j-1}^{1} M_i},$$

is its jth unfolding. Similar to the column block structure at $(I-1)$th level, we have the following result.

Lemma 4. *Consider a sparse tensor* $\boldsymbol{\mathcal{X}}$ *reordered from a sparse vector* $\boldsymbol{x}$ *such that* $\mathrm{vec}(\boldsymbol{X}_{(1)}) = \boldsymbol{x}$. *For the* j*th mode unfolding of* $\boldsymbol{\mathcal{X}}$, *i.e.,* $\boldsymbol{X}_{(j)}$, *and with full row rank* $\boldsymbol{H}_i$*'s, the matrix*

$$\boldsymbol{U}_j := \boldsymbol{X}_{(j)} \left(\boldsymbol{I}_{\prod_{i=I}^{j+1} N_i} \otimes \left(\otimes_{i=j-1}^{1} \boldsymbol{H}_i \right) \right)^{\mathsf{T}} \in \mathbb{R}^{N_j \times \prod_{i=I}^{j+1} N_i \prod_{i=j-1}^{1} M_i},$$

can be divided into $\prod_{i=I}^{j+1} N_i$ *column blocks. Each block is indexed by an encapsulation* n_{j+1} *with* $\mathrm{n}_{j+1} = (n_{j+1}, \cdots, n_I)$ *for* $n_k \in [N_k]$ *for* $k = j+1, \ldots, I$. *The number of nonzero rows in a column block indexed by* n_{j+1} *equals the number of nonzero blocks in* $[\![\boldsymbol{x}_{\mathrm{n}_j}]\!]$ *with* $\mathrm{n}_j = (n_j, n_{j+1}, \cdots, n_I)$.

Lemma 4 implies that matrix $\boldsymbol{H}_j$ actually captures the sparsity at the jth level blocks, which we refer to as the *hierarchical view* of Kronecker compressed sensing. The above perspective can also be interpreted directly from (4.1). The Kronecker product matrix $\boldsymbol{H}$ has a recursive column-block structure: each block of columns is obtained by taking the Kronecker product of a column of $\boldsymbol{H}_I$ with $\otimes_{i=I-1}^{1} \boldsymbol{H}_i$, which itself has a column block structure. This recursive structure aligns with the hierarchical partition block of $\boldsymbol{x}$. Hence, in this hierarchical framework, factor matrices $\{\boldsymbol{H}_i\}_{i=I}^{1}$ operate at different levels: for any p, q with $p > q$, $\boldsymbol{H}_q$ first measures each qth level block of $\boldsymbol{x}$, the resulting measurements of all blocks are then processed by $\boldsymbol{H}_p$, which captures sparsity at a higher level.

4.4. Multi-Stage Sparse Recovery Algorithm

We aim to recover $\boldsymbol{x}$ in (4.1) from noisy measurement $\boldsymbol{y}$, given $\{\boldsymbol{H}_i\}_{i=1}^{I}$. Guided by the hierarchical view in Section 4.3, we next present a recovery framework that handles each $\boldsymbol{H}_i$ sequentially. Although different sparsity patterns have been introduced in Chapter 1, to facilitate the analysis, we rigorously define the standard, hierarchical, and Kronecker-supported sparsity patterns using the notations in this chapter as follows.

Sparsity 1 (Standard sparsity)**.** *A vector* $\boldsymbol{x} \in \mathbb{R}^{\tilde{N}}$ *is* s *sparse if* $\boldsymbol{x}$ *contains at most* s *nonzeros.*

Sparsity 2 (Hierarchical sparsity)**.** *A vector* $\boldsymbol{x} \in \mathbb{R}^{\tilde{N}}$ *is* $\boldsymbol{s}$ *hierarchically sparse with* $\boldsymbol{s} := (s_I, s_{I-1}, \cdots, s_1)$ *if it has a hierarchical partition defined by* $\{N_j\}_{j=1}^{I}$, *and at each level* $j \in [I]$, *every set* $[\![\boldsymbol{x}_{\mathrm{n}_j}]\!]$ *contains at most* s_i *nonzero blocks.*

Sparsity 3 (Kronecker-supported sparsity)**.** *A vector* $\boldsymbol{x} \in \mathbb{R}^{\tilde{N}}$ *is* $\boldsymbol{s}$ *Kronecker-supported sparse if its support is the Kronecker product of* s_j *sparse support vectors.*

We note that the Kronecker-supported sparsity is a special case of hierarchical sparsity, where at each level $j \in [I]$, the s_j nonzero blocks $\boldsymbol{x}_{\mathrm{n}_j}$ share the same support.

Our framework first solves for $\boldsymbol{U}_I = \boldsymbol{X}_{(I)} \left(\otimes_{i=I-1}^{1} \boldsymbol{H}_i \right)^{\mathsf{T}}$ from unfolding along Ith mode as

$$\boldsymbol{T}_{(I)} := \boldsymbol{Y}_{(I)} = \boldsymbol{H}_I \boldsymbol{U}_I + \boldsymbol{N}_{(I)}. \tag{4.4}$$

Here, $\boldsymbol{U}_I$ exhibits a row sparsity pattern where a zero row in $\boldsymbol{U}_I$ corresponds to an all-zero Ith level block $\boldsymbol{x}_{\mathrm{n_I}}$. Thus, recovering $\boldsymbol{U}_I$ from (4.4) is an MMV problem and solved using MMV algorithms such as simultaneous OMP (SOMP), simultaneous iterative hard thresholding (SIHT), simultaneous HTP (SHTP), or MMV-SBL.

Let the estimate of $\boldsymbol{U}_I$ be $\tilde{\boldsymbol{U}}_I$ with error $\boldsymbol{E}_I$ modeling the estimation error and residual noise, $\tilde{\boldsymbol{U}}_I = \boldsymbol{U}_I + \boldsymbol{E}_I$. In the second step, we treat $\tilde{\boldsymbol{U}}_I$ as the noisy measurement and $\boldsymbol{E}_I$ as noise, reorder them into tensor T and N such that $\mathsf{T}_{(I)} = \tilde{\boldsymbol{U}}_I$ and $\mathsf{N}_{(I)} = \boldsymbol{E}_I$, to obtain $\mathsf{T} = \mathsf{X} \times_1 \boldsymbol{H}_1 \cdots \times_{I-1} \boldsymbol{H}_{I-1} \times_I \boldsymbol{I}_{N_I} + \mathsf{N}$. Unfolding T along its $(I-1)$th mode as

$$\boldsymbol{T}_{(I-1)} = \boldsymbol{H}_{I-1}\boldsymbol{U}_{I-1} + \boldsymbol{N}_{(I-1)}. \tag{4.5}$$

For standard and hierarchical sparsity models, the supports of different $(I-1)$th level blocks of $\boldsymbol{x}$ are different. By Lemma 4, zero $(I-1)$th level blocks leads to the zero rows in each column block in $\boldsymbol{U}_{I-1}$, making it a concatenation of N_I row sparse matrices $[\boldsymbol{U}_{I-1}]_{\mathrm{n_I}} := [\boldsymbol{X}_{(I-1)}]_{\mathrm{n_I}} \left(\otimes_{i=I-2}^{1} \boldsymbol{H}_i\right)^{\mathsf{T}}$ for $\mathrm{n_I} = (n_I)$ and $n_I \in [N_I]$. We thus partition (4.5) into N_I independent MMV problems as

$$[\boldsymbol{T}_{(I-1)}]_{\mathrm{n_I}} = \boldsymbol{H}_{I-1}[\boldsymbol{U}_{I-1}]_{\mathrm{n_I}} + [\boldsymbol{N}_{(I-1)}]_{\mathrm{n_I}},$$

and solve them (sequentially or in parallel) using MMV solvers. Concatenating estimates $\tilde{\boldsymbol{U}}_{I-1} := [[\tilde{\boldsymbol{U}}_{I-1}]_1, [\tilde{\boldsymbol{U}}_{I-1}]_2, \cdots, [\tilde{\boldsymbol{U}}_{I-1}]_{N_I}]$ gives the final solution, where $[\tilde{\boldsymbol{U}}_{I-1}]_{\mathrm{n_I}} = [\boldsymbol{U}_{I-1}]_{\mathrm{n_I}} + [\boldsymbol{E}_{I-1}]_{\mathrm{n_I}}$. However, for the Kronecker-supported sparsity, (4.5) is a single MMV problem because the support is common across the $(I-1)$th level blocks.

For jth mode unfolding step, with measurement $\tilde{\boldsymbol{U}}_{j+1}$ from the previous step,

$$\tilde{\boldsymbol{U}}_{j+1} = \boldsymbol{U}_{j+1} + \boldsymbol{E}_{j+1} = \boldsymbol{X}_{(j+1)} \left(\boldsymbol{I}_{\prod_{i=I}^{j+2} N_i} \otimes \left(\otimes_{i=j}^{1} \boldsymbol{H}_i\right)\right)^{\mathsf{T}} + \boldsymbol{E}_{j+1}. \tag{4.6}$$

We unfold the measurement tensor formed from $\tilde{\boldsymbol{U}}_{j+1}$ along its jth mode as

$$\boldsymbol{T}_{(j)} = \boldsymbol{H}_j\boldsymbol{U}_j + \boldsymbol{N}_{(j)}. \tag{4.7}$$

Lemma 4 reduces (4.7) to $\prod_{i=I}^{j+1} N_i$ independent MMV problems for standard and hierarchical sparsity. Sparsity varies across MMVs for the standard model (defined via the total sparsity, not level-wise sparsity) but remains identical in the hierarchical model. For Kronecker-supported sparsity, (4.7) is a single MMV due to shared block support. While mixed models with single and multiple MMVs at different levels are possible, we focus on these three main cases for brevity, leading to the Multi-Stage Recovery (MSR) algorithm, summarized in Algorithm 8.

4.4.1. Complexity

We compare the complexity of MSR variants with existing methods for each sparsity model, assuming (4.7) is solved sequentially, and $M_i = \mathcal{O}(M)$, $N_i = \mathcal{O}(N)$ for $i \in [I]$ with $I < M < N$. For standard sparsity, MSR with OMP matches the time complexity of KroOMP [18], but reduces space complexity from $\mathcal{O}(N^I)$ to $\mathcal{O}(M^{I-1}N)$. For hierarchical sparsity,

Algorithm 8 Multi-Stage Recovery (MSR)

Input: Measurement $\boldsymbol{y}$, dictionaries $\{\boldsymbol{H}_i\}_{i=1}^{I} \in \mathbb{R}^{M_i \times N_i}$

1: Fold $\boldsymbol{y}$ to $\mathbf{Y}$ according to the dimensions of dictionaries $\{\boldsymbol{H}_i\}_{i=1}^{I}$, and initialize $\mathbf{T} = \mathbf{Y}$
2: **for** $j = I, I-1, \cdots, 1$ **do**
3: Obtain the jth mode unfolding of $\mathbf{T}$, i.e., $\boldsymbol{T}_{(j)}$
4: Solve (4.7) for $\boldsymbol{U}_j$ via a compressed sensing algorithm to get estimate $\tilde{\boldsymbol{U}}_j$
5: Fold $\tilde{\boldsymbol{U}}_j$ back to $\mathbf{T}$ such that the jth mode unfolding of $\mathbf{T}$, i.e., $\boldsymbol{T}_{(j)}$ is $\tilde{\boldsymbol{U}}_j$
6: **end for**

Output: Estimated sparse vector $\hat{\boldsymbol{x}} = \text{vec}(\tilde{\boldsymbol{U}}_1)$

our MSR with HTP has time complexity $\mathcal{O}(MN^I)$ and space complexity $\mathcal{O}(M^{I-1}N)$, improving over HiHTP [57] with time and space complexities of $\mathcal{O}(M^2N^2)$ for $I = 2$. For Kronecker-supported sparsity, MSR with SBL lowers time complexity to $\mathcal{O}(MN^I)$ and space complexity to $\mathcal{O}(N^I)$ compared to AM- and SVD-KroSBL [23] with both complexities $\mathcal{O}(M^I N^I)$. The improvements are due to i) the exploitation of the Kronecker structure through tensor operation, reducing the dimensionality; and ii) leveraging the MMV structure from Lemma 4. We refer to Table C.1 in Appendix C.6 for a comprehensive comparison.

4.5. Unified Analysis for Structured Sparsity Models

We establish a unified RIP analysis via a generalized notion of RIP called the $(\boldsymbol{s}, \mathrm{N})$-RIP condition with $\boldsymbol{s} := (s_I, s_{I-1}, \cdots, s_1)$ and $\mathrm{N} := (N_I, N_{I-1}, \cdots, N_1)$ defined by the dimension of factor matrices in Kronecker compressed sensing. To this end, we introduce the generalized $(\boldsymbol{s}, \mathrm{N})$ sparsity model, tailored to the Kronecker compressed sensing problem, which reflects a hierarchical view where sparsity at each level affects recovery.

Sparsity 4 (Generalized sparsity)**.** *Consider Kronecker compressed sensing with $\boldsymbol{H}_i \in \mathbb{R}^{M_i \times N_i}$. A vector $\boldsymbol{x} \in \mathbb{R}^{\tilde{N}}$ is $(\boldsymbol{s}, \mathrm{N})$ sparse if for tensor $\mathbf{X} \in \mathbb{R}^{N_1 \times \cdots \times N_I}$ reordered from $\boldsymbol{x}$ using $\mathrm{N} := (N_I, N_{I-1}, \cdots, N_1)$, the maximum number of nonzero rows of each of the column blocks of its jth mode unfolding $\boldsymbol{X}_{(j)}$ is s_j.*

Relation to other models: We relate the above model to the standard, hierarchical, Kronecker-supported, and block sparsity models. The standard sparsity model is not a special case of $(\boldsymbol{s}, \mathrm{N})$ sparsity, but the set of s sparse vectors is contained in a union of $(\boldsymbol{s}, \mathrm{N})$ sparse vectors.

Lemma 5. *Let set $\mathcal{S}$ contains all s standard sparse vectors in $\mathbb{R}^{\tilde{N}}$, and $\mathcal{S}_{\boldsymbol{s}}$ contains all $(\boldsymbol{s}, \mathrm{N})$ sparse vectors in $\mathbb{R}^{\tilde{N}}$ for a given $(\boldsymbol{s}, \mathrm{N})$. Then, $\mathcal{S} \subset \cup_{\boldsymbol{s} \in f_{\mathrm{N}}(s)} \mathcal{S}_{\boldsymbol{s}}$, where $f_{\mathrm{N}}(s) = \{\boldsymbol{s} : \sum_{i=1}^{I} s_i \leq s + (I-1),\ 1 \leq s_i \leq s\}$.*

Hierarchical sparsity is a special case of $(\boldsymbol{s}, \mathrm{N})$ sparsity when the hierarchical partition structure matches the dimensions of factor matrices in the Kronecker measurement matrix. If, additionally, all the column blocks of jth mode unfolding $\boldsymbol{X}_{(j)}$ share the same support regarding nonzero rows, then we arrive at the Kronecker-supported sparsity.

Block sparsity can also be viewed as $(\boldsymbol{s},\mathrm{N})$ sparsity with $I=2$ when the block boundary matches the hierarchical partition structure.

We next define the $(\boldsymbol{s},\mathrm{N})$-RIP condition for a Kronecker product matrix $\boldsymbol{H}$.

Definition 1 ($(\boldsymbol{s},\mathrm{N})$-RIP)**.** *A Kronecker product matrix $\boldsymbol{H}=\otimes_{i=I}^{1}\boldsymbol{H}_i$ with $\boldsymbol{H}_i\in\mathbb{R}^{M_i\times N_i}$ satisfies $(\boldsymbol{s},\mathrm{N})$-RIP if there exists $\delta\in(0,1)$ such that for all $(\boldsymbol{s},\mathrm{N})$ sparse $\boldsymbol{x}\in\mathbb{R}^{\bar{N}}$, it satisfies $(1-\delta)\|\boldsymbol{x}\|_2^2\le\|\boldsymbol{H}\boldsymbol{x}\|_2^2\le(1+\delta)\|\boldsymbol{x}\|_2^2$. The smallest feasible δ, denoted as $\delta_{(\boldsymbol{s},\mathrm{N})}(\boldsymbol{H})$, is the $(\boldsymbol{s},\mathrm{N})$-RIC of $\boldsymbol{H}$.*

Under our models, $(\boldsymbol{s},\mathrm{N})$-RIP is defined over the unions of subspaces, thus can be used to guarantee the success of recovery algorithms, such as IHT and HTP [116]. In general, such guarantees are established using the upper bound of the RICs. Therefore, we first derive the upper bound of $\delta_{(\boldsymbol{s},\mathrm{N})}(\boldsymbol{H})$, then discuss its implications for different sparsity models, and finally discuss the associated recovery algorithms and guarantees. Here, we denote the standard s-RIC of matrix $\boldsymbol{H}$ as $\delta_s(\boldsymbol{H})$.

Theorem 7. *The $(\boldsymbol{s},\mathrm{N})$-RIC of Kronecker product dictionary $\boldsymbol{H}=\otimes_{i=I}^{1}\boldsymbol{H}_i$, i.e., $\delta_{(\boldsymbol{s},\mathrm{N})}(\boldsymbol{H})$, satisfies $\delta_{(\boldsymbol{s},\mathrm{N})}(\boldsymbol{H})\le\prod_{i=I}^{1}(1+\delta_{s_i}(\boldsymbol{H}_i))-1$.*

The above result immediately applies to hierarchical and Kronecker-supported sparsity, as both are special cases of $(\boldsymbol{s},\mathrm{N})$ sparsity. For Kronecker-supported sparsity, a tighter bound could be expected due to its additional joint sparsity structure arising from the shared support across the nonzero block. However, improving the RIC bound by exploiting this additional joint sparsity is difficult. As noted in [117, 118], RIP analysis considers the worst-case performance and does not guarantee that MMV outperforms the SMV case. So, our bound shows no improvement, and deriving a stronger RIP-based condition for the MMV model is an open problem.

Theorem 7 can also be tailored to standard sparsity using Lemma 5.

Corollary 2. *Consider the Kronecker product $\boldsymbol{H}=\otimes_{i=I}^{1}\boldsymbol{H}_i$. For any s, the s-RIC of $\boldsymbol{H}$ satisfies $\delta_s(\boldsymbol{H})\le\max_{\boldsymbol{s}\in f_{\mathrm{N}}(s)}\delta_{(\boldsymbol{s},\mathrm{N})}(\boldsymbol{H})\le\max_{\boldsymbol{s}\in f_{\mathrm{N}}(s)}\prod_{i=1}^{I}(1+\delta_{s_i}(\boldsymbol{H}_i))-1$.*

The s-RIC bound corroborates that only the sparsity level at different level of blocks explicitly affects the s-RIC of the Kronecker product $\boldsymbol{H}$. Also, a known upper RIC bound is $\delta_s(\boldsymbol{H})\le\prod_{i=1}^{I}(1+\delta_s(\boldsymbol{H}_i))-1$ [16]. Our bound slightly improves this bound:

$$\max_{\boldsymbol{s}\in f_{\mathrm{N}}(s)}\prod_{i=1}^{I}(1+\delta_{s_i}(\boldsymbol{H}_i))-1\le\prod_{i=1}^{I}(1+\delta_s(\boldsymbol{H}_i))-1,$$

because δ_s is a non-decreasing function of s [119] and $s_i^*\le s$ for all $i\in[I]$ and the equality cannot be achieved simultaneously.

Maximum sparsity level: Corollary 2 indicates that recovering s standard sparse vectors via Kronecker compressed sensing with $M_i<N_i$ is only guaranteed when $s<\min_i N_i$, as it is a worst-case analysis. When $s=\min_i N_i$ with $j=\arg\min_i N_i$, a worst-case scenario is $s_j=s=N_j$ and $s_i=1$ for all $i\neq j$. To analyze this case, we recall an equivalent definition for RIC. The s_j-RIC $\delta_{s_j}(\boldsymbol{H}_j)$ is the smallest $\delta\ge0$ such that

$$\delta_{s_j}=\max_{\mathcal{T}_j\subset[N_j],\mathrm{card}(\mathcal{T}_j)\le s_j}\|[\boldsymbol{H}_j]_{\mathcal{T}_j}^{\mathsf{T}}[\boldsymbol{H}_j]_{\mathcal{T}_j}-\boldsymbol{I}_{s_j}\|_2,$$

where $\|\cdot\|_2$ is the matrix spectral norm. If $s_j = N_j$, then $\mathscr{T}_j = [N_j]$ and Then, $\delta_{s_j} = \|\boldsymbol{H}_j^\mathsf{T}\boldsymbol{H}_j - \boldsymbol{I}_{N_j}\|_2 \geq 1$. According to the Definition 1, linear transformation by $\boldsymbol{H}_j$ is a non-injective map and recovery impossible. This also indicates that it is only possible to recover block-sparse vectors with block length smaller than $\min_i N_i$. Intuitively, this is the most unbalanced case where all the nonzeros appear in one single jth level block, making it completely dense and hence cannot be recovered through an underdetermined system. However, it is not always impossible to recover *an s standard sparse vector* when $s \geq \min_i N_i$. since RIP analysis is a worst-case analysis. This represents the most unbalanced case, though recovery is still possible for $s \geq \min_i N_i$ in structured sparsity settings.

Measurement bounds for classical methods: We discuss the implications of Theorem 7 on measurement bounds for recovering $(\boldsymbol{s},\mathrm{N})$-sparse vectors using classical iterative algorithms, namely IHT and HTP. For both algorithms, at iteration k, the support is updated via thresholding operator $L_{\mathscr{S}}$ as $\mathscr{T}^{k+1} = L_{\mathscr{S}}\left(\boldsymbol{x}^k + \boldsymbol{H}^\mathsf{T}\left(\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}^k\right)\right)$. The thresholding operator depends on the sparsity model. For standard s sparse, $L_{\mathscr{S}}$ returns the support of the s largest entries of $\boldsymbol{x}$ in amplitude [119]. For $\boldsymbol{s}$ hierarchically sparse, it selects the top s_1 entries within each first-level block, then recursively picks top $s_2,\ldots,s_I$ blocks at higher levels based on the ℓ_2 norm, as in [57]. However, finding the thresholding operator $L_{\mathscr{S}}$ for $\boldsymbol{s}$ Kronecker-supported sparse vectors is NP-hard and not available in the literature. For example, when $I = 2$, it reduces to selecting rows and columns whose intersection maximizes the squared sum, equivalent to the NP-hard maximum weight biclique problem. A practical alternative is to first select the top s_I blocks at the Ith level by ℓ_2 norm, then recursively sum norms across matching indices at each lower level and select the top $s_{I-1},\ldots,s_1$ blocks; this is the approach we use in simulations for comparison. Then, IHT applies a simple projection while HTP solves a least-squares problem on the support,

$$\boldsymbol{x}^{k+1} = \left(\boldsymbol{x}^k + \boldsymbol{H}^\mathsf{T}\left(\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}^k\right)\right)_{\mathscr{T}^{k+1}}, \tag{IHT}$$

$$\boldsymbol{x}^{k+1} = \arg\min_{\boldsymbol{x}\in\mathbb{R}^{\bar{N}}} \|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}\|_2, \ \ \mathrm{supp}(\boldsymbol{x}) \in \mathscr{T}^{k+1}, \tag{HTP}$$

where operator $(\cdot)_{\mathscr{T}^{k+1}}$ only preserves the entries within the set $\mathscr{T}^{k+1}$ and sets the others to zero.

We next discuss the implications for measurement bounds. It is known that for IHT and HTP to recover a vector from a union of subspaces, tailoring the thresholding operator $L_{\mathscr{S}}$ to the union and having an RIC below $1/\sqrt{3}$ over that union is sufficient to guarantee convergence to the ground truth [57, 119]. So, our results shows that $\max_{\boldsymbol{s}\in f_{\mathrm{N}}(3s)} \delta_{(\boldsymbol{s},\mathrm{N})} < 1/\sqrt{3}$ (for s standard sparsity) and $\delta_{(3\boldsymbol{s},\mathrm{N})} < 1/\sqrt{3}$ (for $\boldsymbol{s}$ hierarchical sparsity) are sufficient for the success of IHT and HTP. However, it does not guarantee the recovery of the $\boldsymbol{s}$ Kronecker-supported sparse vectors as the thresholding operator is suboptimal.

To compare the measurement bound for Kronecker compressed sensing, we consider the simplest case with $I = 2$ and $s = \mathscr{O}(s_1 s_2)$ for $\boldsymbol{s} \in f_{\mathrm{N}}(s)$, and Gaussian factor matrices $\boldsymbol{H}_i$'s. For recovering s standard sparse vectors, our Corollary 2 implies that each $\boldsymbol{H}_i$ satisfies the s_i-RIP, requiring $M_i = \mathscr{O}(s_i \log N_i)$ [119]. So, the total measurement bound scales as $\bar{M} = \mathscr{O}(s_1 s_2 \log N_1 \log N_2)$ improving over the existing bound $\bar{M} = \mathscr{O}(s^2 \log N_1 \log N_2) = \mathscr{O}(s_1^2 s_2^2 \log N_1 \log N_2)$ [16]. In comparison, standard compressed sensing with *fully unstruc-*

tured Gaussian matrix requires only $\mathcal{O}(s_1 s_2 \log N_1 N_2)$ measurements, which is smaller due to greater flexibility and randomness in measurement. However, Kronecker compressed sensing exploits the multidimensional structure to reduce the computational complexity during recovery. For the recovery of $\boldsymbol{s}$ hierarchical sparse vectors, Corollary 2 suggests a measurement bound $\mathcal{O}(s_1 s_2 \log N_1 \log N_2)$, while a *fully unstructured Gaussian matrix* requires only $\mathcal{O}(s_1 s_2 \log N_1 + s_2 \log N_2)$ [57].

Measurement bounds for our MSR: We now establish recovery guarantees for MSR with IHT and HTP using the RICs of factor matrices.

Theorem 8. *Consider the sparse recovery problem, $\boldsymbol{y} = \left(\otimes_{i=I}^{1} \boldsymbol{H}_i\right)\boldsymbol{x} + \boldsymbol{n}$. Define tensors $\boldsymbol{\mathsf{X}}$ and $\boldsymbol{\mathsf{N}}$, which are reshaped from $\boldsymbol{x}$ and $\boldsymbol{n}$, respectively, using the dimensions of $\boldsymbol{H}_i$'s. If $\boldsymbol{x}$ is an s standard sparse vector and the factor matrices $\boldsymbol{H}_i$ for $i \in [I]$ satisfy $\delta_{3s_i}(\boldsymbol{H}_i) < 1/\sqrt{3}$ for $\forall\, \boldsymbol{s} \in f_{\mathrm{N}}(s)$, then the estimate $\hat{\boldsymbol{x}}$ of $\boldsymbol{x}$ using k-iteration IHT or HTP in Algorithm 8, satisfies*

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_2 \leq \max_{\boldsymbol{s} \in f_{\mathrm{N}}(s)} \sum_{n_2, \cdots, n_I} \left(\sum_{i=1}^{I} \prod_{j=1}^{i-1} \tau_j \alpha_i^k \left\| [\boldsymbol{U}_i]_{\mathrm{n}_{i+1}} \right\|_{\mathrm{F}} + \prod_{i=1}^{I} \tau_i \|\boldsymbol{\mathsf{N}}\|_{\mathrm{F}} \right),$$

where $[\boldsymbol{U}_i]_{\mathrm{n}_{i+1}} = [\boldsymbol{X}_{(i)}]_{\mathrm{n}_{i+1}} \left(\otimes_{l=i-1}^{1} \boldsymbol{H}_l\right)^{\mathsf{T}}$, and if $\boldsymbol{x}$ is an $\boldsymbol{s}$ hierarchically sparse vector, and the factor matrices $\boldsymbol{H}_i$ for $i \in [I]$ satisfy $\delta_{3s_i}(\boldsymbol{H}_i) < 1/\sqrt{3}$, then the estimate $\hat{\boldsymbol{x}}$ of $\boldsymbol{x}$ using k-iteration IHT or HTP in Algorithm 8, satisfies

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_2 \leq \sum_{n_2, \cdots, n_I} \left(\sum_{i=1}^{I} \prod_{j=1}^{i-1} \tau_j \alpha_i^k \left\| [\boldsymbol{U}_i]_{\mathrm{n}_{i+1}} \right\|_{\mathrm{F}} + \prod_{i=1}^{I} \tau_i \|\boldsymbol{\mathsf{N}}\|_{\mathrm{F}} \right),$$

and if $\boldsymbol{x}$ is an $\boldsymbol{s}$ Kronecker-supported sparse vector, there is

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_2 \leq \sum_{i=1}^{I} \prod_{j=1}^{i-1} \tau_j \alpha_i^k \|\boldsymbol{U}_i\|_{\mathrm{F}} + \prod_{i=1}^{I} \tau_i \|\boldsymbol{\mathsf{N}}\|_{\mathrm{F}},$$

where $\boldsymbol{U}_i = \boldsymbol{X}_{(i)} \left(\boldsymbol{I}_{\prod_{l=I}^{i+1} N_l} \otimes \left(\otimes_{l=i-1}^{1} \boldsymbol{H}_l\right) \right)^{\mathsf{T}}$, with $\alpha_i < 1$, and τ_i are

$$\textit{MSIHT: } \alpha_i = \sqrt{3}\delta_{3s_i}(\boldsymbol{H}_i);\ \tau_i = (1 - \alpha_i^k) \frac{\sqrt{3(1 + \delta_{2s_i}(\boldsymbol{H}_i))}}{1 - \alpha_i}, \textit{ and}$$

$$\textit{MSHTP: } \alpha_i = \sqrt{2\delta_{3s_i}^2(\boldsymbol{H}_i)/(1 - \delta_{2s_i}^2(\boldsymbol{H}_i))};\ \tau_i = (1 - \alpha_i^k) \frac{\sqrt{2(1 - \delta_{2s_i}(\boldsymbol{H}_i))} + \sqrt{1 + \delta_{s_i}(\boldsymbol{H}_i)}}{\left(1 - \delta_{2s_i}(\boldsymbol{H}_i)\right)(1 - \alpha_i)}.$$

As the number of iterations $k \to \infty$, the error bound reduces to $\tau_1 \prod_{i=2}^{I} \tau_i N_i \|\boldsymbol{\mathsf{N}}\|_{\mathrm{F}}$ for the standard and hierarchical sparsity, and $\prod_{i=1}^{I} \tau_i \|\boldsymbol{\mathsf{N}}\|_{\mathrm{F}}$ for Kronecker-supported sparsity. So, MSIHT and MSHTP approach the true value within a constant factor of measurement noise power. Although factors $\tau_1 \prod_{i=2}^{I} \tau_i N_i$ and $\prod_{i=1}^{I} \tau_i$ suggest error propagation as the algorithm proceeds from $j = I$ till $j = 1$ and scale with the problem dimension, this amplification is not observed in practice (see Figure 4.5). The bound for $\boldsymbol{s}$ Kronecker-supported sparsity is tighter than that for the other two models because it solves a single MMV problem, resulting a collective error bound, instead of a looser bounds due to the

sum of each individual MMV bound. While our MSR's measurement bound scales the same as classical methods due to a shared requirement on the s_i-RIP of $\boldsymbol{H}_i$'s, it can have a larger error from propagation, potentially requiring more iterations or $\boldsymbol{H}_i$'s with smaller s_i-RICs. However, a key advantage of MSR is that it provides recovery guarantees for the Kronecker-supported sparsity model, unlike classical IHT and HTP-based methods.

4.6. Numerical Evaluations

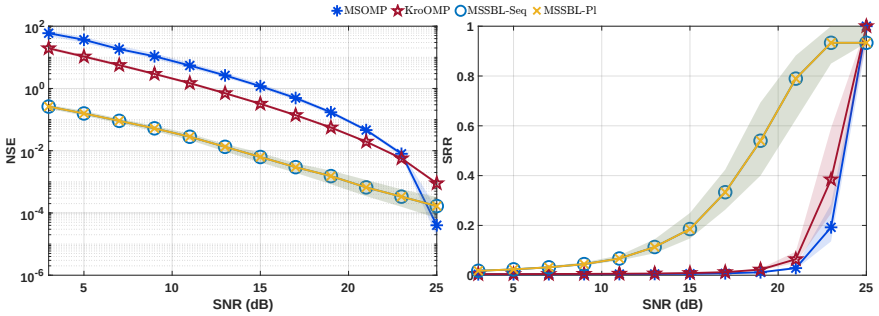
4.6.1. Evaluations Using Synthetic Signals

For numerical results, we combine MSR with MMV-SBL [120], SIHT [121], SHTP [121], and SOMP [122], and the resulting algorithms are referred to as MSSBL, MSIHT, MSHTP, and MSOMP, respectively. Our benchmark for the standard sparsity is KroOMP [18]. Here, we omit computationally intensive SBL and OMP whose results are identical to KroOMP. For hierarchical sparsity, our benchmark is the state-of-the-art HiHTP [57]. For Kronecker-structured support sparsity, we benchmark with the state-of-the-art AM- and SVD-KroSBL [23]. Unlike the OMP/SBL-based algorithms, the IHT/HTP-based algorithms need the true sparsity level $\boldsymbol{s}$ as input. We use Tensorlab [108] for tensor operation and `Seq` and `Pl` to represent the *sequential* and *parallel* (`parfor` in Matlab [123]) implementation of (4.7); they have the same recovery performance but different runtimes.

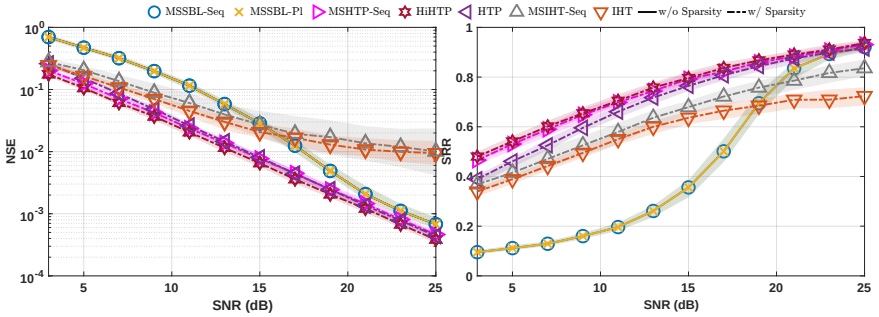
For fairness, we cap the number of EM iterations for SBL-based methods (MSSBL, AM-KroSBL, and SVD-KroSBL) to two hundred, for HTP based methods to one hundred, and for all IHT based methods to two hundred. For HTP-based algorithms, we stop the iterations if the detected support remains the same in two consecutive iterations [124], while IHT-based algorithms are terminated when the normalized difference between two consecutive estimations is smaller than 10^{-6}. For OMP-based algorithms, we stop when the norm of the residual is smaller than $\epsilon\|\boldsymbol{y}\|_2$. Here, the coefficient $\epsilon = 0.05$ is fixed for all OMP-based algorithms, which is empirically determined. We also prune small entries in hyperparameters for faster convergence for SBL-based algorithms.

For all three models, we set $M_i = M$, $N_i = N$, and $s_i = s$ for $i \in [I]$. For the s standard sparsity, we opt for $\boldsymbol{H} = \otimes_{i=I}^{1} \boldsymbol{H}_i$ with $I = 2$, $M = 64$, and $N = 80$. The entries of $\boldsymbol{H}_i$ and the nonzero entries of $\boldsymbol{x}$ are drawn independently from the standard normal distribution. We set $s = 15$, and the support is randomly drawn from a uniform distribution. For $\boldsymbol{s}$ hierarchically sparse vectors, we also opt for $I = 2$, $M = 64$, $N = 80$, and $s = 15$. Here, supports are generated by first selecting s blocks uniformly at random, then assigning support within each block uniformly. In the Kronecker-supported sparsity model, we opt for $I = 3$, $M = 15$, $N = 18$, and $s = 4$. The measurement noise is zero mean white Gaussian noise whose variance is determined by SNR (dB) $= 10\log_{10}\mathbb{E}\{\|\boldsymbol{H}\boldsymbol{x}\|_2^2/\|\boldsymbol{n}\|_2^2\}$ of $\{3,5,\cdots,23,25\}$.

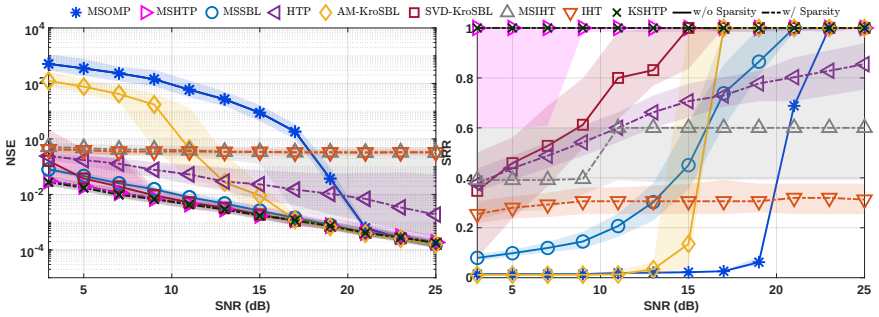
Our metrics are runtime, the normalized squared error NSE $= \|\boldsymbol{x}-\hat{\boldsymbol{x}}\|_2^2/\|\boldsymbol{x}\|_2^2$, where $\boldsymbol{x}$ is the ground truth and $\hat{\boldsymbol{x}}$ is the estimated vector, and SRR as in (2.27). We cap the number of EM iterations for SBL-based methods (MSSBL, AM-KroSBL, and SVD-KroSBL) to two hundred, for HTP based methods to one hundred, and for all IHT based methods to two hundred. We also prune small entries in hyperparameters for faster convergence for SBL-based algorithms. The results are shown in Figure 4.2 and Table 4.1, with the figure showing median and 25%/75% quartiles, and the table showing averages.



(a) s standard sparsity.



(b) $\boldsymbol{s}$ hierarchical sparsity.



(c) $\boldsymbol{s}$ Kronecker-supported sparsity.

Figure 4.2.: NSE and SRR as functions of SNR.

The NSE for recovering an s standard sparse vector is shown in Figure 4.2(a). Compared to KroOMP, MSOMP provides similar performance regarding NSE but needs one to three orders less runtime, as in Table 4.1. MSSBL outperforms KroOMP in all SNR cases with one or two orders less runtime. Regardless of different runtimes as in Table 4.1, sequential and parallel implementations provide identical NSE and SRR results. Regarding runtime, only in low SNR cases, MSSBL-`Pl` is faster than MSSBL-`Seq`. This is because in high SNR cases, the parallel overhead dominates, including data transfer and communication cost. As we see in Figure 4.5, when the computation cost dominates, there is a significant gain in computation time, as a trade-off for memory usage. The NSE for hierarchical

sparsity is shown in Figure 4.2(b). Our MSHTP/MSIHT offers similar performance as HTP and HiHTP, and IHT. However, MSHTP requires two orders less runtime than HTP and one order less runtime than HiHTP; and MSIHT requires two orders less runtime than IHT. MSSBL exhibits a worse performance in low SNR scenario because it does not require the true sparsity level $\boldsymbol{s}$ as an input, while for IHT/HTP-based algorithms, this prior knowledge is necessary. However, MSSBL is still able to offer a comparable performance in high SNR scenarios, making it a powerful candidate when the prior knowledge $\boldsymbol{s}$ is absent. The NSE for Kronecker-supported sparsity is shown in Figure 4.2(c). Our MSSBL consistently achieves a comparable NSE and is two or three orders faster than AM- and SVD-KroSBL. KSHTP is the algorithm we explained in HTP. Although the thresholding operator for Kronecker support is not optimal, KSHTP still offers the best SRR performance, followed by MSHTP. MSIHT has the least runtime, which is four orders less than its classic counterpart IHT. Overall, Figure 4.2 and Table 4.1 demonstrate that our MSR framework can offer comparable or better performance with significantly reduced runtime.

Table 4.1.: Averaged runtime. **Bold**: the best result.

SNR	3 dB	7 dB	11 dB	15 dB	19 dB	23 dB
Recovery of s sparse vectors						
MSOMP-Seq	**0.426**	**0.412**	0.383	0.333	0.220	**0.057**
KroOMP	130.541	108.053	76.694	39.984	11.577	0.753
MSSBL-Seq	1.819	1.102	0.576	**0.222**	**0.142**	0.114
MSSBL-P1	0.452	4.926	**0.266**	1.829	0.153	0.128
Recovery of $\boldsymbol{s}$ hierarchically sparse vectors						
MSSBL-Seq	2.493	2.013	1.250	0.610	0.197	0.111
MSSBL-P1	0.496	0.461	1.366	0.251	0.145	0.108
MSHTP-Seq	**0.038**	**0.031**	**0.030**	**0.025**	**0.019**	**0.017**
HiHTP	0.651	0.549	0.520	0.544	0.440	0.457
HTP	2.244	1.717	1.326	0.845	0.826	0.531
MSIHT-Seq	0.050	0.051	0.053	0.051	0.045	0.043
IHT	8.244	8.241	8.255	8.292	8.289	8.279
Recovery of $\boldsymbol{s}$ Kronecker-supported sparse vectors						
MSOMP	0.004	0.004	0.004	0.004	0.003	0.002
MSHTP	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**
MSSBL	0.073	0.059	0.045	0.028	0.012	0.005
SVD-KroSBL	37.123	26.982	14.241	8.604	5.407	4.068
AM-KroSBL	55.953	63.468	75.973	74.584	51.709	34.133
HTP	0.977	0.835	0.471	0.347	0.234	0.232
MSIHT	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**
IHT	6.077	6.076	6.076	6.069	6.068	6.054
KSHTP	0.102	0.073	0.067	0.081	0.087	0.088

We next evaluate the performance of different algorithms by fixing the SNR and varying the number of measurements. The setting is as follows. For the s standard sparsity, we opt for $\boldsymbol{H} = \otimes_{i=I}^{1} \boldsymbol{H}_i$ with $I = 2$, and set $M = \{48, 52, 56, \cdots, 72\}$ and $N = 80$. The entries of $\boldsymbol{H}_i$ and nonzero entries of $\boldsymbol{x}$ are drawn independently from the standard normal distribution. We set $s = 15$, and the support is randomly drawn from a uniform distribution. For $\boldsymbol{s}$ hierarchically sparse vectors, we also opt for $I = 2$, and set $M = \{48, 52, 56, \cdots, 72\}$, $N = 80$, and $s = 15$. In the Kronecker-supported sparsity model, we opt for $I = 3$, and set $M = \{12, 13, \cdots, 16\}$, $N = 18$, and $s = 4$. We adopt the additive white Gaussian noise with zero mean with SNR (dB) = 20. Ratio is defined as $\bar{M}/\bar{N} = \prod_{i=1}^{I} M_i/N_i = (M/N)^I$. We consider NSE, SRR, and runtime for performance evaluation. We follow the same way to

cap the number of iterations. Results in Figure 4.3 and Table 4.2 are obtained through two hundred independent trials. Overall, we observe similar trends as in Figure 4.2 and Table 4.1. Our MSR is able to provide comparable or better performance with reduced runtime, demonstrating the efficacy of exploiting the Kronecker product structure in the recovery process.

Table 4.2.: Averaged runtime in seconds. **Bold**: the best result.

Recovery of s sparse vectors							
M	48	52	56	60	64	68	72
MSOMP-Seq	**0.074**	**0.095**	**0.127**	0.157	0.194	0.231	0.292
KroOMP	0.714	1.194	2.114	3.795	7.409	12.544	22.415
MSSBL-Seq	0.173	0.174	0.144	**0.136**	**0.133**	**0.137**	**0.132**
MSSBL-P1	0.141	0.146	0.151	0.152	0.150	0.153	0.330

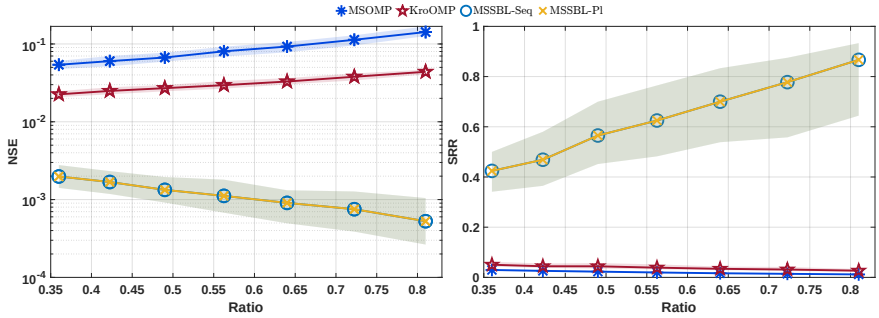
Recovery of $\boldsymbol{s}$ hierarchically sparse vectors							
M	48	52	56	60	64	68	72
MSSBL-Seq	0.298	0.259	0.203	0.169	0.155	0.141	0.127
MSSBL-P1	0.173	0.161	0.147	0.136	0.130	0.125	0.120
MSHTP-Seq	**0.020**	**0.020**	**0.019**	**0.019**	**0.018**	**0.017**	**0.017**
HiHTP	0.366	0.362	0.369	0.419	0.418	0.460	0.513
HTP	0.307	0.364	0.475	0.498	0.641	0.687	0.457
MSIHT-Seq	0.053	0.050	0.048	0.046	0.045	0.042	0.042
IHT	4.710	5.455	6.219	7.129	8.330	9.123	10.187

Recovery of $\boldsymbol{s}$ Kronecker-supported sparse vectors					
M	12	13	14	15	16
MSOMP	0.002	0.002	0.002	0.002	0.002
MSHTP	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**
MSSBL	0.019	0.014	0.012	0.009	0.008
SVD-KroSBL	5.313	4.893	4.725	5.078	5.309
AM-KroSBL	52.089	52.478	49.684	47.024	42.675
HTP	0.072	0.111	0.166	0.251	0.275
MSIHT	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**
IHT	3.458	4.050	4.949	6.019	7.286
KSHTP	0.118	0.091	0.105	0.089	0.083

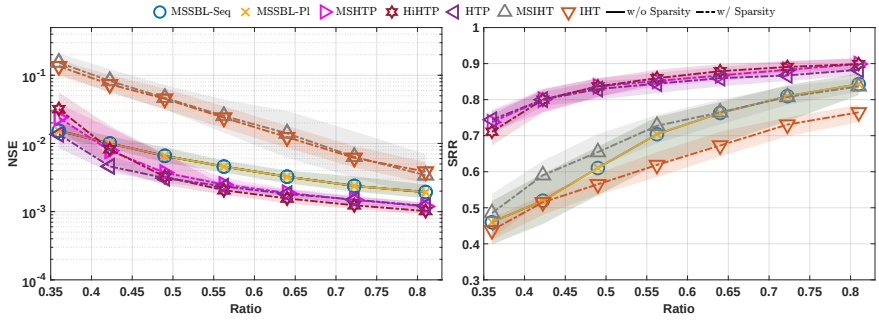
4.6.2. Comparisons With Traditional Compressed Sensing Approaches

In this section, we compare our MSR to traditional compressed sensing algorithms, including IHT, HTP, SBL, OMP, and the ℓ_1 norm-based basis pursuit denoising (BPDN) [119] (`basisPursuit` function in Matlab [123]).

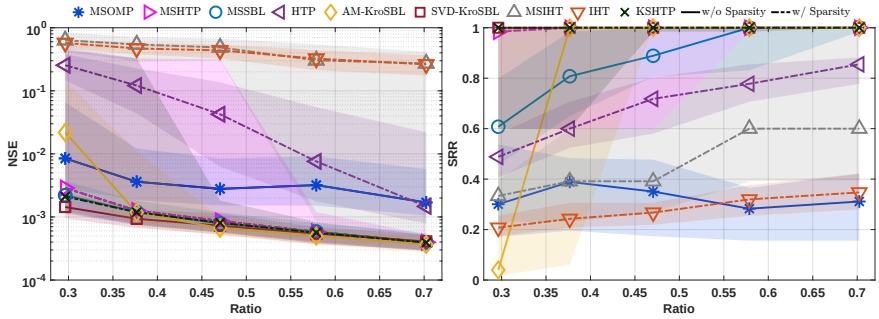
For all three models, we set $M_i = M$, $N_i = N$, and $s_i = s$ for $i \in [I]$. For the s standard sparsity, we opt for $\boldsymbol{H} = \otimes_{i=I}^{1} \boldsymbol{H}_i$ with $I = 3$, $M = 12$, and $N = 15$. The entries of $\boldsymbol{H}_i$ and the nonzero entries of $\boldsymbol{x}$ are drawn independently from the standard normal distribution. We set $s = 8$, and the support is randomly drawn from a uniform distribution. For $\boldsymbol{s}$ hierarchically sparse vectors, we opt for $I = 2$, $M = 35$, $N = 40$, and $s = 8$ per dimension. Here, supports are generated by first selecting s blocks uniformly at random, then assigning support within each block uniformly. In the $\boldsymbol{s}$ Kronecker-supported sparsity model, we opt for $I = 3$, $M = 12$, $N = 15$, and $s = 4$. In all models, the measurement noise is zero mean white Gaussian noise whose variance is determined by SNR (dB) $= 10\log_{10} \mathbb{E}\{\|\boldsymbol{Hx}\|_2^2 / \|\boldsymbol{n}\|_2^2\}$ of $\{0, 5, 10, 15, 20, 25\}$. We follow the same condition to cap the iterative algorithms, and for BPDN, we cap the number of iterations at fifty. Compared to Section 4.6.1, we downsize the measurement matrices mainly for computational feasibility. With the same condition



(a) s standard sparsity.



(b) $\boldsymbol{s}$ hierarchically sparsity.

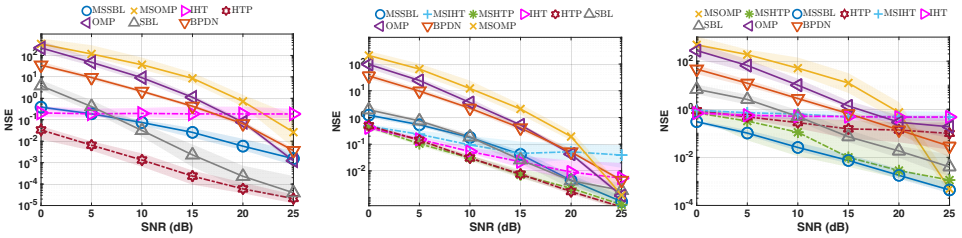


(c) $\boldsymbol{s}$ Kronecker-supported sparsity.

Figure 4.3.: NSE and SRR as functions of the number of measurements.

as in Section 4.6.1, it is hard to evaluate traditional algorithms such as BPDN and SBL. The NSE shown in Figure 4.4 are median and 25%/75% quartiles, while Table 4.3 shows the average runtime, both over fifty independent trials.

As shown in Figure 4.4(a), HTP achieves the lowest runtime and high accuracy but relies on prior knowledge of the true sparsity level s, which is generally unavailable in practice. While SBL yields higher reconstruction accuracy than MSSBL in the high SNR regime, it incurs a runtime two orders of magnitude higher, limiting its scalability. MSSBL emerges as the most robust solution: it outperforms traditional methods (IHT, OMP, BPDN) and has lower runtime than SBL, offering a balance between reconstruction accuracy and



(a) Standard sparsity. (b) Hierarchical sparsity. (c) Kronecker-supported sparsity.

Figure 4.4.: NSE as a function of SNR compared to traditional compressed sensing algorithms.

Table 4.3.: Average runtime for comparison with traditional compressed sensing algorithms. **Bold**: the best result.

SNR	0 dB	5 dB	10 dB	15 dB	20 dB	25 dB
Recovery of s sparse vectors						
MSOMP	0.063	0.061	0.062	0.058	0.039	**0.010**
MSSBL	0.455	0.410	0.336	0.193	0.097	0.058
HTP	**0.021**	**0.013**	**0.010**	**0.010**	**0.009**	0.011
IHT	2.258	2.263	2.263	2.258	2.247	2.243
OMP	22.438	20.169	15.747	8.983	2.341	0.036
SBL	37.921	21.528	13.248	10.986	10.075	9.492
BPDN	102.813	102.036	101.473	99.650	96.134	85.278
Recovery of $\boldsymbol{s}$ hierarchically sparse vectors						
MSOMP	0.025	0.025	0.023	0.023	0.017	**0.002**
MSHTP	**0.005**	**0.004**	**0.003**	**0.003**	**0.003**	0.003
MSIHT	0.007	0.007	0.007	0.008	0.009	0.009
MSSBL	0.469	0.400	0.253	0.154	0.042	0.017
HTP	0.197	0.123	0.078	0.062	0.036	0.027
IHT	0.482	0.483	0.484	0.481	0.481	0.482
OMP	6.524	5.793	4.475	2.624	0.786	0.037
SBL	18.312	16.188	8.372	3.235	2.172	1.860
BPDN	20.901	20.832	20.315	19.276	17.592	16.058
Recovery of $\boldsymbol{s}$ Kronecker-supported sparse vectors						
MSOMP	0.003	0.003	0.003	0.003	0.002	**0.001**
MSHTP	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**
MSIHT	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**
MSSBL	0.054	0.050	0.045	0.032	0.018	0.006
HTP	0.317	0.295	0.211	0.095	0.068	0.034
IHT	1.900	1.902	1.900	1.899	1.915	1.895
OMP	23.683	21.593	16.471	9.410	2.684	0.179
SBL	46.175	42.640	29.059	15.007	11.073	9.565
BPDN	108.037	108.129	108.027	106.957	102.110	91.115

computational efficiency, without requiring specific prior knowledge.

We illustrate the hierarchical sparsity recovery against traditional compressed sensing algorithms in Figure 4.4(b). Compared to HTP, MSHTP achieves almost the same reconstruction accuracy but with one to two orders of less runtime. This also happens to MSSBL compared to SBL. MSSBL, in this case, is also a balanced option when the true sparsity level is unknown, without sacrificing efficiency significantly. OMP and IHT are only slightly worse than their counterparts, i.e., MSOMP and MSIHT, but the gain in runtime is significant by two orders of magnitude. Finally, Figure 4.4(c) contains the results for Kronecker-supported sparsity recovery against traditional benchmarks. MSSBL constantly achieves the best performance, seconded by MSHTP and MSOMP (high SNR

case), with two to three orders less runtime than their counterparts.

4.6.3. Sparse Vector Recovery Performance With Varying N and I

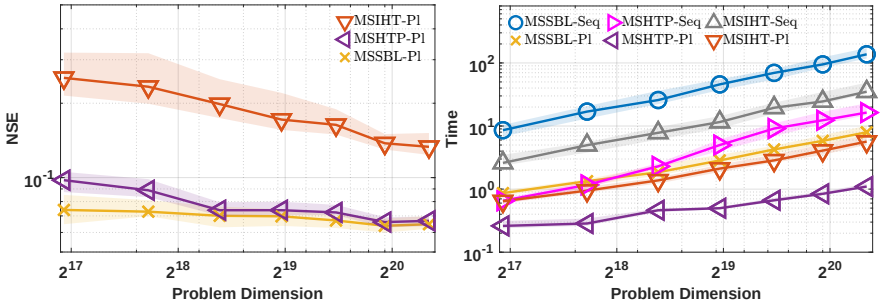


Figure 4.5.: NSE and runtime of MSR as functions of problem dimension $\bar{N}$.

Figure 4.5 shows how the NSE and runtime (median with 25%/75% quartiles) of MSSBL, MSHTP, and MSIHT scale with the problem dimension, focusing on hierarchical sparsity. We choose $I = 3$ and SNR as 20dB and vary $N = \{50, 60, \cdots, 110\}$, so that the problem dimension $\bar{N} = N^I = 125000, 216000, \cdots, 1331000$, with $M = \lceil (0.6\bar{N})^{1/I} \rceil$ and $s = \lceil (0.4\bar{N})^{1/I} \rceil$. As expected, parallel implementation is faster than sequential. MSSBL has the best NSE but is slower than MSIHT and MSHTP. The MSIHT is worse than MSHTP due to IHT's slow convergence [119]. Overall, our MSR efficiently handles large dimensional Kronecker compressed sensing problems.

Next we present results where we fix the size of each factor matrix $\boldsymbol{H}_i$ but vary I from 1 to 4, to demonstrate the ability of our MSR to handle arbitrary system order I. We consider $N_i = N$ and $M_i = M$ for $i \in [I]$, opt for $N = 50$, and determine M and s through $M = \lceil (0.6\bar{N})^{1/I} \rceil$ and $s = \lceil 0.4N \rceil$ as in Section 4.6. We fix SNR at 20dB and results in Figure 4.6 shows how the NSE and runtime (median with 25%/75% quartiles) of MSSBL, MSHTP, and MSIHT scale with different I for Kronecker-supported sparsity model.

In Figure 4.6, we observe similar trends as in Figure 4.5. Parallel and sequential implementations have identical recovery performance but different runtimes. Thus, for NSE, we only show the results for parallel implementation. The only exception in Figure 4.6 compared to Figure 4.5 is that for $I = 2$, parallel implementation for MSHTP and MSIHT requires more computation time than the sequential implementation. This is because when the problem is computationally light, parallel overhead dominates the time consumption, including data transfer and communication, rather than computation itself.

4.6.4. Application: Wideband Massive MIMO Channel Estimation

Massive MIMO has been a key enabler for the fifth generation communication. For data transmission, an important task is to estimate the channel by processing the received pilot signals sent from user. We focus on the OFDM-based wideband massive MIMO channel estimation, where we consider a base station with a half-wavelength spacing uniform

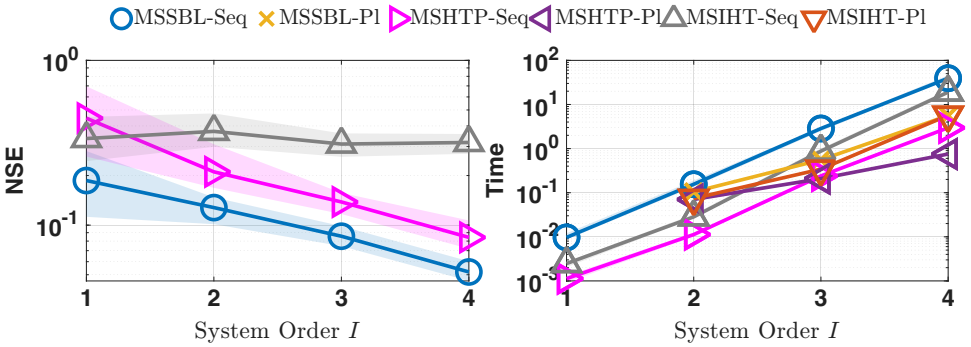


Figure 4.6.: NSE and runtime of MSR as functions of system order I.

linear array equipped with N_a elements serving one single antenna user. Due to the environment reflection, we consider L impinging angles, each containing up to K_L delays. The maximum delay is αT_s with $\alpha \leq 1$ where T_s is the OFDM symbol duration. The number of subcarriers of the OFDM symbol is N_s. The channel matrix $\boldsymbol{C}$ is the superposition of impinging waves characterized by delays and angles as $\boldsymbol{C} = \sum_{l=1}^{L}\sum_{k_l=1}^{K_L} \rho_{l,k_l}\boldsymbol{d}(\tau_{l,k_l})\boldsymbol{a}^{\mathsf{H}}(\theta_l)$ [125, 126], where $\rho_{l,k_l} \in \mathbb{C}$ is the complex gain of the path corresponding to the k_lth delay of the lth angle, $\boldsymbol{d}(\tau_{l,k_l}) := [1, e^{-j2\pi\tau_{l,k_l}/T_\mathrm{s}}, \cdots, e^{-j2\pi(N_\mathrm{s}-1)\tau_{l,k_l}/T_\mathrm{s}}]^{\mathsf{T}}$ is the delay manifold vector of the delay τ_{l,k_l}, while $\boldsymbol{a}(\theta_l) := [1, e^{-j2\pi\theta_l}, \cdots, e^{-j2\pi(N_\mathrm{a}-1)\theta_l}]^{\mathsf{T}}$ is the steering vector for $\theta_l \in [0,1]$ representing the equivalent lth impinging angle [115]. Due to the significant path loss, the received signal is transmitted through a limited number of paths, making the channel intrinsically sparse over two sparsifying bases. The first is obtained by sampling the delay range $[0, T_\mathrm{s}]$ with N_s samples as $\{nT_\mathrm{s}/N_\mathrm{s}\}_{n=0}^{N_\mathrm{s}-1}$, leading to a delay basis $\boldsymbol{H}_\mathrm{d} := [\boldsymbol{d}(0), \boldsymbol{d}(T_\mathrm{s}/N_\mathrm{s}), \cdots, \boldsymbol{d}((N_\mathrm{d}-1)T_\mathrm{s}/N_\mathrm{s})] \in \mathbb{C}^{N_\mathrm{s}\times N_\mathrm{d}}$, with $N_\mathrm{d} := \lfloor \alpha N_\mathrm{s} \rfloor$. The second is obtained by sampling the angular domain as $\{n/N_\mathrm{a}\}_{n=0}^{N_\mathrm{a}-1}$, yielding angle basis $\boldsymbol{H}_\mathrm{a} := [\boldsymbol{a}(0), \boldsymbol{a}(1/N_\mathrm{a}), \cdots, \boldsymbol{a}(1-1/N_\mathrm{a})] \in \mathbb{C}^{N_\mathrm{a}\times N_\mathrm{a}}$. Then the channel can be represented as $\boldsymbol{C} = \boldsymbol{H}_\mathrm{d}\boldsymbol{X}\boldsymbol{H}_\mathrm{a}^{\mathsf{H}}$, where $\boldsymbol{X} \in \mathbb{C}^{N_\mathrm{d}\times N_\mathrm{a}}$ is the sparse representation with up to LK_L nonzeros.

To reduce the overhead, one may use a subset of OFDM subcarriers and array elements for channel estimation. Denote the pilot as $\boldsymbol{p} \in \mathbb{C}^{M_\mathrm{d}}$, where $M_\mathrm{d} \leq N_\mathrm{s}$ is the number of subcarriers in a subset. Let $\boldsymbol{S}_\mathrm{d} \in \{0,1\}^{M_\mathrm{d}\times N_\mathrm{s}}$ be the sampling matrix for subcarriers and $\boldsymbol{S}_\mathrm{a} \in \{0,1\}^{M_\mathrm{a}\times N_\mathrm{a}}$ be the sampling matrix for array where only M_a out of N_a elements are chosen. We write the received signal as $\boldsymbol{Y} = \mathrm{diag}(\boldsymbol{p})\boldsymbol{S}_\mathrm{d}\boldsymbol{C}\boldsymbol{S}_\mathrm{a}^{\mathsf{T}} + \boldsymbol{N} \in \mathbb{C}^{M_\mathrm{d}\times M_\mathrm{a}}$ with $\boldsymbol{N}$ being noise [115]. Plugging in the sparse channel representation and vectorizing both sides of the equation, we have

$$\boldsymbol{y} = \left((\boldsymbol{S}_\mathrm{a}\boldsymbol{H}_\mathrm{a}^*) \otimes (\mathrm{diag}(\boldsymbol{p})\boldsymbol{S}_\mathrm{d}\boldsymbol{H}_\mathrm{d})\right)\boldsymbol{x} + \boldsymbol{n},$$

where $\boldsymbol{y} = \mathrm{vec}(\boldsymbol{Y})$, $\boldsymbol{x} = \mathrm{vec}(\boldsymbol{X})$, $\boldsymbol{n} = \mathrm{vec}(\boldsymbol{N})$, and $(\cdot)^*$ is the conjugate. Denoting $\boldsymbol{S}_\mathrm{a}\boldsymbol{H}_\mathrm{a}^* = \boldsymbol{H}_2 \in \mathbb{C}^{M_\mathrm{a}\times N_\mathrm{a}}$, $\mathrm{diag}(\boldsymbol{p})\boldsymbol{S}_\mathrm{d}\boldsymbol{H}_\mathrm{d} = \boldsymbol{H}_1 \in \mathbb{C}^{M_\mathrm{d}\times N_\mathrm{d}}$, and $I = 2$, the channel estimation problem is a Kronecker compressed sensing problem with $\boldsymbol{x}$ being $\boldsymbol{s} = (L, K_L)$ hierarchically sparse. For simulation, we consider $N_\mathrm{a} = 512$, $L = \{5, 10, 15, 20, 25, 35, 50, 75, 100\}$, $K_L = 3$, $N_\mathrm{s} = 1024$ OFDM subcarriers, and $\alpha = 0.5$ for the maximum delay. We fix $M_\mathrm{a} = \lceil 0.3N_\mathrm{a} \rceil$ and $M_\mathrm{d} = \lceil 0.1N_\mathrm{s} \rceil$, making $\boldsymbol{H}_2 \in \mathbb{C}^{154\times 512}$ and $\boldsymbol{H}_1 \in \mathbb{C}^{103\times 512}$. Both angles $\{\theta_l\}$ and delays $\{\tau_{l,k_l}\}$ are

generated independently and uniformly over the sampling grid, while path gains $\{\rho_{l,k_l}\}$ are drawn from a standard normal distribution [115]. The measurement noise is zero mean white Gaussian noise whose variance is determined by SNR (dB) $= 10\log_{10}\mathbb{E}\{\|((\boldsymbol{S}_{\mathrm{a}}\boldsymbol{H}_{\mathrm{a}}^{*})\otimes(\mathrm{diag}(\boldsymbol{p})\boldsymbol{S}_{\mathrm{d}}\boldsymbol{H}_{\mathrm{d}}))\boldsymbol{x}\|_2^2/\|\boldsymbol{n}\|_2^2\}$ of 20dB. We evaluate NSE $:= \|\boldsymbol{C}-\boldsymbol{H}_{\mathrm{d}}\hat{\boldsymbol{X}}\boldsymbol{H}_{\mathrm{a}}^{\mathsf{H}}\|_{\mathrm{F}}^2/\|\boldsymbol{C}\|_{\mathrm{F}}^2$ and runtime and compare MSHTP and MSOMP to HiIHT/HiHTP in [115]. Results are obtained by two hundred independent trials.

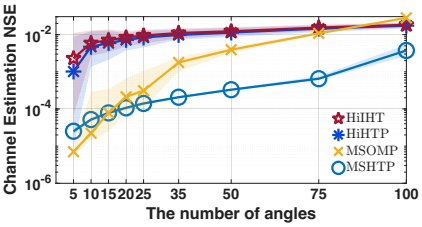


Figure 4.7.: NSE of different schemes.

#angles L	HiHTP	HiIHT	MSHTP	MSOMP
5	0.074	4.147	0.020	0.008
10	0.130	4.179	0.040	0.014
15	0.144	4.129	0.055	0.019
20	0.196	4.134	0.077	0.025
25	0.211	4.167	0.092	0.030
35	0.312	4.131	0.124	0.043
50	0.507	4.157	0.182	0.064
75	0.880	4.221	0.271	0.119
100	1.341	4.227	0.239	0.214

Table 4.4.: Average runtime in seconds.

We present NSE of channel estimation and average runtime in Figure 4.7 and Table 4.4, respectively. We observe that MSHTP and MSOMP provide better performance than HiHTP and HiIHT in most cases, with one or two orders less runtime. MSOMP's relatively higher NSE with large L is because it wrongly identifies many insignificant paths (smaller $|\rho_{l,k_l}|$), since it does not require the true sparsity level (L, K_L) as input. However, we still observe that the significant paths are estimated accurately and efficiently, making MSOMP a practical option for the channel estimation task.

4.6.5. Application: Channel Estimation for IRS-Aided Wireless System

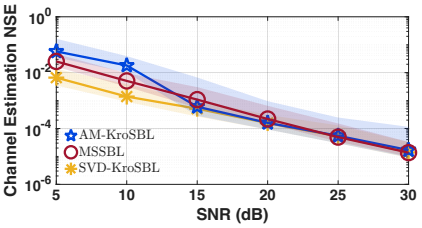


Figure 4.8.: NSE of different schemes.

SNR (dB)	AM-KroSBL	SVD-KroSBL	MSSBL
5	138.973	3.127	0.021
10	121.727	2.848	0.015
15	77.714	2.791	0.010
20	43.986	2.775	0.007
25	26.334	2.748	0.005
30	18.720	2.745	0.005

Table 4.5.: Average runtime in seconds.

An IRS is a reconfigurable meta-surface consisting of a large number of adjustable reflecting elements. By changing the reflection coefficients of elements, IRS can reflect the signal to a certain area to improve the coverage of the wireless communication systems operating at millimeter-wave/terahertz frequency bands. But properly configuring the IRS requires the channel state information. It is hence important to develop efficient and accurate channel estimation algorithms for the IRS-aided system. In this section, we consider an uplink narrowband IRS-aided MIMO system, consisting of a T half-wavelength spacing-antenna transmitter MS, an R half-wavelength spacing-antenna receiver BS, and an L half-wavelength spacing-element uniform linear array IRS. If the channel matrices of the MS-IRS, and the IRS-BS channel are denoted as $\boldsymbol{\Phi}_{\mathrm{MS}} \in \mathbb{C}^{L\times T}$ and $\boldsymbol{\Phi}_{\mathrm{BS}} \in \mathbb{C}^{R\times L}$, respectively, according to the geometric channel model [27, 81, 127, 128], $\boldsymbol{\Phi}_{\mathrm{MS}} \in \mathbb{C}^{L\times T}$ and

$\boldsymbol{\Phi}_{\mathrm{BS}} \in \mathbb{C}^{R\times L}$ can be formulated as

$$\boldsymbol{\Phi}_{\mathrm{MS}} = \sum_{p=1}^{P_{\mathrm{MS}}} \sqrt{\frac{LT}{P_{\mathrm{MS}}}} \beta_{\mathrm{MS},p} \boldsymbol{a}_L(\phi_{\mathrm{MS},p}) \boldsymbol{a}_T(\alpha_{\mathrm{MS}})^{\mathsf{H}}, \tag{4.8}$$

$$\boldsymbol{\Phi}_{\mathrm{BS}} = \sum_{p=1}^{P_{\mathrm{BS}}} \sqrt{\frac{RL}{P_{\mathrm{BS}}}} \beta_{\mathrm{BS},p} \boldsymbol{a}_R(\alpha_{\mathrm{BS},p}) \boldsymbol{a}_L(\phi_{\mathrm{BS}})^{\mathsf{H}}, \tag{4.9}$$

where P_{MS} and P_{BS} are the number of paths between MS and IRS, and IRS and BS, respectively. The angles $\phi_{\mathrm{MS},p}$, α_{MS}, $\alpha_{\mathrm{BS},p}$, and ϕ_{BS} represent the pth AoA of the IRS, and the pth AoD of the MS, the pth AoA of the BS, and the AoD of the IRS, respectively, while $\beta_{\mathrm{MS},p}$ and $\beta_{\mathrm{BS},p}$ are the complex path gains. Steering vector $\boldsymbol{a}_Q(\psi) \in \mathbb{C}^Q$ for any integer Q and angle ψ is defined as $\boldsymbol{a}_Q(\psi) = 1/\sqrt{Q}[1, e^{j\pi\cos\psi}, \cdots, e^{j\pi(Q-1)\cos\psi}]^{\mathsf{T}}$. Then, the cascaded MS-IRS-BS channel is given by $\boldsymbol{\Phi}_{\mathrm{BS}} \operatorname{diag}(\boldsymbol{\theta}) \boldsymbol{\Phi}_{\mathrm{MS}}$ for a given IRS configuration $\boldsymbol{\theta} \in \mathbb{C}^L$ whose ith entry of $\boldsymbol{\theta}$ models the reflection of the ith IRS element. Channel estimation problem targets to estimate the cascaded channel $\boldsymbol{\Phi}_{\mathrm{BS}} \operatorname{diag}(\boldsymbol{\theta}) \boldsymbol{\Phi}_{\mathrm{MS}}$ given any $\boldsymbol{\theta}$, which is sufficient for subsequent tasks such as beamforming [81].

Wireless channels operating on millimeter-wave/terahertz bands are intrinsically sparse due to severe path loss. To reveal this sparsity, we adopt three sparsifying bases $\boldsymbol{A}_R$, $\boldsymbol{A}_L$, and $\boldsymbol{A}_T$, corresponding to the angular domain of the array at BS, IRS, and MS, respectively. Such bases contain steering vectors evaluated over N grid angles $\{\psi_n\}_{n=1}^N$ such that $\cos(\psi_n) = 2n/N - 1$ [79], defined as $\boldsymbol{A}_Q = [\boldsymbol{a}_Q(\psi_1), \boldsymbol{a}_Q(\psi_2), \ldots, \boldsymbol{a}_Q(\psi_N)] \in \mathbb{C}^{Q\times N}$ for any integer $Q > 0$. Then, (4.8) and (4.9) reduce to

$$\boldsymbol{\Phi}_{\mathrm{BS}} = \boldsymbol{A}_R \boldsymbol{x}_{\mathrm{R}} \boldsymbol{x}_{\mathrm{L,d}}^{\mathsf{H}} \boldsymbol{A}_L^{\mathsf{H}} \quad \text{and} \quad \boldsymbol{\Phi}_{\mathrm{MS}} = \boldsymbol{A}_L \boldsymbol{x}_{\mathrm{L,a}} \boldsymbol{x}_{\mathrm{T}}^{\mathsf{H}} \boldsymbol{A}_T^{\mathsf{H}}, \tag{4.10}$$

where vectors $\boldsymbol{x}_{\mathrm{R}}, \boldsymbol{x}_{\mathrm{L,d}}, \boldsymbol{x}_{\mathrm{L,a}}, \boldsymbol{x}_{\mathrm{T}} \in \mathbb{C}^N$ are the unknown channel representations over the known sparsifying bases. They are sparse due to the intrinsic sparsity of the channel.

Channel estimation is performed by processing the received pilot signals, given the knowledge of the sent pilot signals and the training IRS configurations. Suppose we allocate K time slots for channel estimation, over which the channel is considered to be constant. We vary IRS configurations for K_{I} times, and for each different configuration, we transmit the same set of pilot signal $\boldsymbol{G} \in \mathbb{C}^{T\times K_{\mathrm{P}}}$ over K_{P} time slots such that $K = K_{\mathrm{I}} K_{\mathrm{P}}$. The received signal $\boldsymbol{Y}_k \in \mathbb{C}^{R\times K_{\mathrm{P}}}$ corresponding to the kth training configuration $\boldsymbol{\theta}_k$ is

$$\boldsymbol{Y}_k = \boldsymbol{\Phi}_{\mathrm{BS}} \operatorname{diag}(\boldsymbol{\theta}_k) \boldsymbol{\Phi}_{\mathrm{MS}} \boldsymbol{G} + \boldsymbol{N}_k, \tag{4.11}$$

where $\boldsymbol{N}_k \in \mathbb{C}^{R\times K_{\mathrm{P}}}$ is the noise. Substituting (4.10) into (4.11), and vectorizing the received signal $\{\boldsymbol{Y}_k\}_{k=1}^{K_I}$ followed by some algebraic operations [27], we have

$$\boldsymbol{y} = (\boldsymbol{H}_{\mathrm{L}} \otimes \boldsymbol{H}_{\mathrm{T}} \otimes \boldsymbol{H}_{\mathrm{R}})\, \boldsymbol{x} + \boldsymbol{n} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{n} \in \mathbb{C}^{RK}, \tag{4.12}$$

where $\boldsymbol{H}_{\mathrm{L}} \in \mathbb{C}^{K_{\mathrm{I}}\times N}$ is formed by the first N columns of $\boldsymbol{\Theta}^{\mathsf{T}} (\boldsymbol{A}_L^{\mathsf{T}} \odot \boldsymbol{A}_L^{\mathsf{H}})^{\mathsf{T}}$ whose N^2 columns are just N repetitions the columns of $\boldsymbol{H}_{\mathrm{L}}$, $\boldsymbol{H}_{\mathrm{T}} = \boldsymbol{X}^{\mathsf{T}} \boldsymbol{A}_T^*$, and $\boldsymbol{H}_{\mathrm{R}} = \boldsymbol{A}_R$, with $\odot$ being the Khatri-Rao product. We collect K_{I} IRS configurations $\{\boldsymbol{\theta}_k\}_{k=1}^{K_{\mathrm{I}}}$ in matrix $\boldsymbol{\Theta} \in \mathbb{C}^{L\times K_{\mathrm{I}}}$ and

define $\boldsymbol{x} = \boldsymbol{x}_{\mathrm{L}} \otimes \boldsymbol{x}_{\mathrm{T}}^{*} \otimes \boldsymbol{x}_{\mathrm{R}} \in \mathbb{C}^{N^3}$ with $\boldsymbol{x}_{\mathrm{L}} \in \mathbb{C}^{N}$ being the scaled version of the first N entries of $\boldsymbol{x}_{\mathrm{L,a}} \otimes \boldsymbol{x}_{\mathrm{L,d}}^{*}$, corresponding to the removal of redundant columns in $\boldsymbol{\Theta}^{\mathsf{T}}(\boldsymbol{A}_L^{\mathsf{T}} \odot \boldsymbol{A}_L^{\mathsf{H}})^{\mathsf{T}}$. Denoting $\boldsymbol{H}_{\mathrm{L}}$ as $\boldsymbol{H}_3$ with K_{I} as M_3, $\boldsymbol{H}_{\mathrm{T}}$ as $\boldsymbol{H}_2$ with K_{P} as M_2, $\boldsymbol{H}_{\mathrm{R}}$ as $\boldsymbol{H}_1$ with R as M_1, and $N = N_3 = N_2 = N_1$, the channel estimation problem is transformed into a Kronecker compressed sensing problem following the form of (4.1) with unknown $\boldsymbol{s} = (s_3, s_2, s_1) = (P_{\mathrm{MS}}, 1, P_{\mathrm{BS}})$ *Kronecker-supported sparse* vector $\boldsymbol{x}$. With the estimated $\hat{\boldsymbol{x}}$, the estimated channel for a given IRS configuration $\boldsymbol{\theta}$, i.e., $\hat{\boldsymbol{\Phi}}_{\mathrm{BS}}\,\mathrm{diag}(\boldsymbol{\theta})\hat{\boldsymbol{\Phi}}_{\mathrm{MS}}$, is obtained by reshaping $(\hat{\boldsymbol{\Phi}}_{\mathrm{MS}}^{\mathsf{T}} \odot \hat{\boldsymbol{\Phi}}_{\mathrm{BS}})\boldsymbol{\theta}$ with known size $R \times T$, where $\hat{\boldsymbol{\Phi}}_{\mathrm{MS}}^{\mathsf{T}} \odot \hat{\boldsymbol{\Phi}}_{\mathrm{BS}}$ with known size $RT \times L$ is reconstructed as $\mathrm{vec}(\hat{\boldsymbol{\Phi}}_{\mathrm{MS}}^{\mathsf{T}} \odot \hat{\boldsymbol{\Phi}}_{\mathrm{BS}}) = (\boldsymbol{\Phi}_{\mathrm{A}} \otimes \boldsymbol{A}_T^{*} \otimes \boldsymbol{A}_R)\hat{\boldsymbol{x}}$ with $\boldsymbol{\Phi}_{\mathrm{A}}$ being the first N columns of $(\boldsymbol{A}_L^{\mathsf{T}} \odot \boldsymbol{A}_L^{\mathsf{H}})^{\mathsf{T}}$.

We set $T = 6$, $R = 16$, and $L = 256$. To set up the sparsifying bases, we opt for $N = 18$. Pilot signals contained in $\boldsymbol{G}$ are randomly generated quadrature phase shift keying symbols using a uniform distribution while the IRS training configuration is randomly drawn from uniform distribution $\{\pm 1/\sqrt{L}\}$ [82]. Regarding the pilot signals and training IRS configurations, we consider $K_{\mathrm{I}} = K_{\mathrm{P}} = 10$, making $\boldsymbol{H}_3 \in \mathbb{C}^{10\times 18}$, $\boldsymbol{H}_2 \in \mathbb{C}^{10\times 18}$, and $\boldsymbol{H}_1 \in \mathbb{C}^{16\times 18}$. To model the scatters, we set $P_{\mathrm{MS}} = P_{\mathrm{BS}} = 3$ and all angles $\phi_{\mathrm{MS},p}$, α_{MS}, $\alpha_{\mathrm{BS},p}$, and ϕ_{BS} are drawn uniformly and independently from the grid points, while path gains $\beta_{\mathrm{MS},p}$ and $\beta_{\mathrm{BS},p}$ are drawn independently from complex standard normal distribution [27]. We compare MSSBL with SVD-/AM-KroSBL in [27] using the same iteration cap. Metrics include channel estimation NSE given by (2.42) and runtime.

Figure 4.8 shows 25%/50%/75% quartiles of NSE, while Table 4.5 shows the average runtime, both over fifty independent trials. We observe that all three algorithms provide comparable channel estimation performance, while MSSBL has two orders less runtime than SVD-KroSBL and four orders less runtime than AM-KroSBL, making it more efficient in this application scenario.

4.6.6. Application: Foreman Video Sequence Recovery

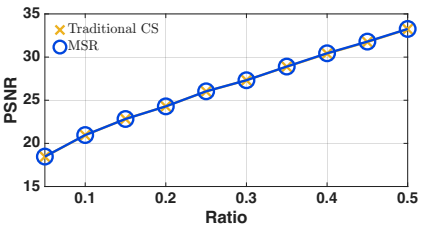


Figure 4.9.: PSNR of different schemes.

Ratio ζ	Traditional CS	MSR
0.05	77.529	10.559
0.10	91.276	9.001
0.15	70.156	9.754
0.20	75.829	9.920
0.25	67.998	11.689
0.30	99.262	11.138
0.35	79.100	11.111
0.40	109.850	11.165
0.45	92.898	7.771
0.50	66.310	6.772

Table 4.6.: Runtime in seconds.

This section presents the recovery results for *Foreman video sequence* to demonstrate MSR's ability to deal with a real-world dataset and its superiority over the traditional compressed sensing algorithm. In the experiment, we follow the settings in [16, 55] and to make this chapter self-contained, we provide a brief overview of these settings.

Tested frames are generated by cropping around the center to form a frame size of 128×128 pixels and there are in total eight frames used in the experiment. To spatially sparsify the image content within a single frame, we vectorize each frame and adopt a $2D$ inverse discrete wavelet transform basis $\boldsymbol{W}_1 \in \mathbb{R}^{16384\times 16384}$ applied to the sparse

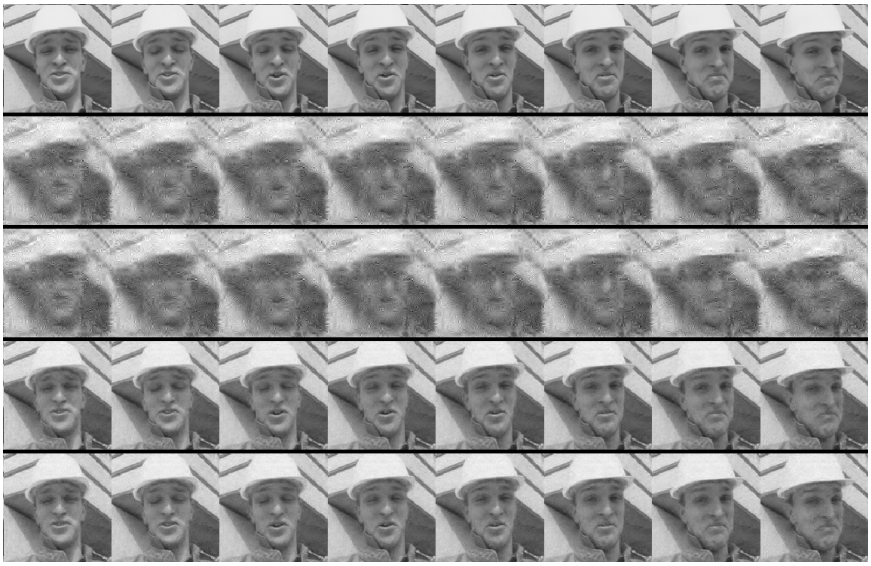


Figure 4.10.: Comparison of reconstructed Foreman video frames. Row 1: ground truth. Row 2: traditional CS with $\zeta = 0.1$. Row 3: MSR with $\zeta = 0.1$. Row 4: traditional CS with $\zeta = 0.5$. Row 5: MSR with $\zeta = 0.5$.

coefficients of each video frame. For the sparsity in the temporal dimension, we turn to a $1D$ inverse discrete wavelet transform basis $\boldsymbol{W}_2 \in \mathbb{R}^{8\times 8}$. It exploits the correlation between frames to sparsify the signal over time. Suppose the video sequence is denoted by $\boldsymbol{\theta} \in \mathbb{R}^{131072}$ where $131072 = 8$ (frames) $\times$ 128 (row pixels) $\times$ 128 (columns pixels), then its relation to sparsifying bases and the sparse coefficient vector $\boldsymbol{x} \in \mathbb{R}^{131072}$ is $\boldsymbol{\theta} = (\boldsymbol{W}_2 \otimes \boldsymbol{W}_1)\boldsymbol{x}$. To compress the video sequence, we use a measurement matrix as $\boldsymbol{I} \otimes \boldsymbol{S}$, where $\boldsymbol{I}$ means that there is no temporal compression while $\boldsymbol{S} \in \mathbb{R}^{M\times 16384}$ from a subsampled permuted Hadamard transform denotes the spatial compression. Here, M is the number of measurements taken in one frame. This leads to the following measurement/sparsifying model

$$\boldsymbol{y} = (\boldsymbol{W}_2 \otimes \boldsymbol{S}\boldsymbol{W}_1)\boldsymbol{x}. \tag{4.13}$$

The goal is to obtain sparse coefficient $\boldsymbol{x}$ using the compressed measurement $\boldsymbol{y}$, and finally we reconstruct the video sequence as $\boldsymbol{\theta} = (\boldsymbol{W}_2 \otimes \boldsymbol{W}_1)\boldsymbol{x}$.

Denoting $\boldsymbol{W}_2$ as $\boldsymbol{H}_2$, $\boldsymbol{S}\boldsymbol{W}_1$ as $\boldsymbol{H}_1$, and $I = 2$, we note that (4.13) is mathematically equivalent to (4.1), hence can be solved using our MSR. The benchmark in this experiment is ℓ_1-based basis pursuit [16], where (4.13) is treated as a traditional compressed sensing problem and the Kronecker structure of $\boldsymbol{W}_2 \otimes \boldsymbol{S}\boldsymbol{W}_1$ is ignored. For a fair comparison, we adopt the same ℓ_1 solver [129, 130] in MSR as in [16] with the same stopping criterion. We determine the number of measurements M as $M = \lfloor 16384\zeta \rceil$ with measurement ratio $\zeta \in \{0.05, 0.10, \cdots, 0.45, 0.50\}$. We use the peak signal-to-noise ratio (PSNR) and runtime as evaluation metrics. Results are shown in Figure 4.9 and Table 4.6, where traditional CS refers to traditional compressed sensing algorithm. We observe that MSR achieves the same PSNR as that of traditional CS with roughly one order of magnitude less runtime, ef-

fectively demonstrating the efficacy of our MSR on a real-world dataset. We also compare all used video frames in Figure 4.10.

4.7. Chapter Summary

We investigated the Kronecker compressed sensing problem for signals with multiple sparsity structures. We presented a novel hierarchical view, comprehending that each factor matrix in the Kronecker product dictionary senses the sparse signal at a different level, obeying a hierarchical structure. This insight led to a computationally efficient, multi-stage recovery framework that achieved performance comparable to state-of-the-art methods with one order or less runtime. On the theoretical front, we unified the RIP analysis for Kronecker product matrices across various structured sparsity models, and also established the recovery guarantee for our multi-stage recovery algorithm. This hierarchical framework opens promising avenues for designing new algorithms to accommodate more structured patterns and provide efficient solutions to many applications.

5

Conclusion

In this concluding chapter, we summarize our contributions to Kronecker compressed sensing and outline avenues for future research.

5.1. Summary of the Contributions

In this dissertation, our focus has been on the sparse recovery with Kronecker product measurement matrix, aiming for efficient, theoretically guaranteed sparse recovery algorithms for different sparsity patterns stemmed from various applications, such as massive machine-type communication and channel estimation problem for IRS-aided wireless communication system.

In Chapter 2, we considered Kronecker compressed sensing for Kronecker-supported sparsity pattern, presenting the theoretically guaranteed recovery algorithms. Adhering to the Bayesian philosophy, our solution was developed within the SBL framework. To enforce the Kronecker-supported sparsity, we adopted a prior distribution on the sparse vector parameterized by a Kronecker-structured hyperparameter. The Kronecker-structured hyperparameter resulted in a non-convex underlying optimization problem, which was then solved using two approaches, i.e., AM-based, and SVD-based. The former approach enjoyed the convergence guarantee, while the latter one converged faster, thus was more efficient and practical. Relying on the Kronecker structure of the measurement matrix, we designed a novel technique to further reduce the complexity of the algorithm. Such a technique replaced the inversion of a large dimensional matrix with the eigenvalue decompositions of multiple low dimensional matrices and multiple diagonal matrices inversions. Beyond such algorithmic contributions and theoretical guarantees, we also studied the property of the SBL cost function. One notable result was that by incorporating the Kronecker-structure support as prior knowledge, the number of local minima of the cost function was significantly reduced, compared to the cost function without the prior knowledge. This result aided to understand why exploiting prior knowledge could lead to a better performance.

In Chapter 3, we considered Kronecker compressed sensing for Kronecker-structured sparsity pattern, presenting the theoretically guaranteed recovery algorithms. To better exploit both the Kronecker structure of $\boldsymbol{H}$ and the Kronecker-structured sparsity in $\boldsymbol{x}$, through the Kronecker product mixture property, we turned to a decomposition-based solution. Here, we decomposed a large dimensional problem into multiple small dimensional subproblems through the decomposition, significantly reducing the computational complexity. Each subproblem was a traditional compressed sensing problem which could be solved using any compressed sensing algorithm. In the end, the final solution was the Kronecker product of the solutions to the subproblems, effectively aligning with the Kronecker structure prior knowledge. Furthermore, we also identified the grid mismatch issue when the decomposition-based approach was applied to parameter estimation problem together with BEM. Hence, we presented a novel off-grid SBL algorithm to address the grid mismatch, of which the performance was then validated through numerical evaluations. To theoretically guarantee the success of the decomposition step, we presented a probabilistic error bound. We also rigorously demonstrated the convergence property of the off-grid SBL. Besides, to understand why the decomposition step offered a better performance, we investigated the its denoising effect, showing that the noise level of

the signal was reduced after the decomposition step. This could contribute to the better performance in terms of sparse vector recovery.

After investigating the solutions to specific sparsity patterns in Chapter 2 and Chapter 3, we moved to the unified model for Kronecker compressed sensing. In Chapter 4, we first introduced the hierarchical view of the Kronecker compressed sensing, showing that the Kronecker product measurement matrix probed the sparse vector from different levels, following a block-wise and hierarchical structure. Leveraging this insight, we developed a multi-stage sparse recovery algorithmic framework, relying on the tensor operations and being readily adaptable to different sparsity models: standard, hierarchical, and Kronecker-supported. Further, through the hierarchical view, we analyzed the RIP of Kronecker product matrices under different sparsity models, compared the measurement bounds, and provided theoretical recovery guarantees for our multi-stage algorithm for all three sparsity patterns.

Beyond the above contributions of this dissertation, i.e., the algorithmic designs and theoretical guarantees, we gained the following insight that emerged during this research journey.

- *While the concept of incorporating prior knowledge to enhance algorithm performance is straightforward, its correct practical execution is challenging, and validating its theoretical benefits remains difficult. And it may also compromise the solution's flexibility.*

This insight is drawn from the common thread throughout this dissertation, i.e., exploiting the structure in both the measurement matrix and the sparsity pattern. By incorporating such prior knowledge, we can design more efficient algorithms with better performance. However, approaching it methodically is challenging, as we need to carefully design the algorithm to effectively leverage the prior knowledge. Moreover, it takes more effort to understand why it works than just showing it works empirically. Theorem 3 in Chapter 2 demonstrated that by incorporating the prior knowledge, the number of local minima of the cost function was significantly reduced compared to the cost function without the prior knowledge. This result is more exciting than just an algorithm with better performance, since it provided a deeper understanding on why such an approach worked. From the theoretical perspective, providing theoretical guarantees becomes more complicated when prior knowledge is involved, as we need to account for the additional structure in our analysis. On the other hand, incorporating prior knowledge may reduce the flexibility of the solution, as the algorithm may be tailored to a specific structure, making it less adaptable, even impossible for other scenarios. Therefore, there is a trade-off between leveraging prior knowledge for better performance and maintaining flexibility in the algorithm design.

5.2. Open Avenues for Future Research

The results in this dissertation have also triggered many open avenues for future research and we outline a few potential directions which we consider significant for further exploration.

5.2.1. Exploring Sparsity Patterns in Kronecker Compressed Sensing

The main theme of this dissertation is to exploit the structure in both the measurement matrix and the sparsity pattern in Kronecker compressed sensing. Four different structures were considered in this dissertation, i.e., standard, hierarchical, Kronecker-supported, and Kronecker-structured sparsity. A natural extension is to consider more sparsity patterns.

There are two aspects regarding the structured sparsity. First, in Chapter 4, we have shown that the multi-stage algorithmic framework can be readily adapted to different sparsity patterns. However, this adaptability was based on the assumption that the hierarchical level of the sparse vector aligned with that of the Kronecker product measurement matrix. An interesting direction is when such alignment does not hold. For example, in [115], the authors considered the recovery of a hierarchically sparse vector with three hierarchical levels while the measurement matrix was only a Kronecker product of two factor matrices. How to effectively handle such misalignment is an open question.

Second, apart from the sparsity patterns considered in this dissertation, there are many other general sparsity patterns that have been observed in various applications. For example, due to the physical scattering structure of massive MIMO channel, the sparse coefficient of the channel in the angular domain has burst sparsity, meaning that the nonzero entries appear in the burst and usually form connected blocks [131, 132]. Also the hidden Markov tree structure is observed in the wavelet coefficients of discrete wavelet transformation of image signals [133, 134]. Therefore, developing algorithms that can effectively leverage these general structures would be a valuable direction for future research.

More importantly, as the RIP analysis is strongly tied to considered sparsity patterns, how to adapt the newly considered structures for the theoretical analysis is also an open question. Investigating this theoretical aspect can lead to new insights into the properties of Kronecker compressed sensing for such newly considered patterns, e.g., recovery conditions as in Section 4.5.

5.2.2. Concentration Inequality for Kronecker Product Matrices

We have presented the RIP analysis for Kronecker product measurement matrices in Chapter 4, where our analyses provided the establishment of the RIP regarding those of the factor matrices. However, we did not see the difference in the measurement bounds between hierarchical sparsity and Kronecker-supported sparsity. As in [119, 135], concentration inequality is a powerful tool to establish the RIP of random measurement matrices. Hence, investigating the concentration inequality of Kronecker product matrices would be an interesting direction for future research. Such results can directly lead to the measurement bounds for the RIP of Kronecker product matrices to hold, and reveal the difference in the measurement bounds between different sparsity patterns due to the count of subspaces.

There have been several works studying the concentration inequality of certain classes of structured random matrices that arise in certain applications [136]. Some examples are subsampled bounded orthonormal systems [119], partial random circulant and Toeplitz matrices [137], Kronecker product of an identity matrix and a sub-Gaussian random

matrix [117], and the column-wise Khatri-Rao product of two matrices [138]. The structured measurement matrix can better reflect the system's physical property [117, 136, 139] or stems from the mathematical model of such problems [138]. In terms of Kronecker-wise structure, the Kronecker product of an identity matrix and a sub-Gaussian random matrix arises in distributed compressed sensing where the block diagonal structure of the measurement matrix aligns with the block structure of the sparse vector and measurements [117, 136, 139]. Another example is the Khatri-Rao product used in co-LASSO for wireless channel estimation [140] and tensor decomposition in direction-of-arrival estimation [141]. In particular, one can start with the Kronecker product of one deterministic matrix and one random matrix with independent and identically distributed sub-Gaussian entries. The deterministic-random Kronecker product pertains to many practical applications such as channel estimation in wireless communications, where the deterministic matrix can be the array response matrix while the random matrix can be the pilot signal matrix. For more general case, one can consider the Kronecker product of two or more random matrices with independent and identically distributed sub-Gaussian entries. However, since the entries of the output matrix are no longer independent, and the resulting distribution is no longer sub-Gaussian but more heavy-tailed (e.g., sub-Gamma distribution) [142, 143], new techniques need to be developed to tackle these challenges.

A

Appendix of Chapter 2

A.1. Proof of Proposition 1

The proof uses Zangwill's Convergence Theorem A [73] and the following proposition to arrive at the desired results:

Proposition 2. *Let $f(\boldsymbol{x}) : \mathscr{X} \to \mathbb{R}$ be a continuous function. If*

$$f(\boldsymbol{x}) \to +\infty \text{ as } \mathscr{X} \ni \boldsymbol{x} \to \mathrm{bd}(\mathscr{X}) \text{ or } \|\boldsymbol{x}\| \to +\infty, \tag{A.1}$$

where $\mathrm{bd}(\mathscr{X})$ *denotes the boundary of* $\mathscr{X}$*, then any sublevel set* $\mathscr{S} : \{\boldsymbol{x} | f(\boldsymbol{x}) \leq c\}$ *is compact.*

Proof. This proof is adopted from [144]. A set is compact if it is closed and bounded [144]. We start with the proof of boundedness. Suppose $\mathscr{S}$ is unbounded, then there must exist a sequence $\{\boldsymbol{x}_n\} \subset \mathscr{S}$ with $\|\boldsymbol{x}_n\| \to +\infty$. Then, (A.1) indicates $f(\boldsymbol{x}_n) \to +\infty$. Therefore, there exists an integer $n_0 > 0$ such that $f(\boldsymbol{x}_n) > c$, for $n > n_0$, which is a contradiction to the assumption $\{\boldsymbol{x}_n\} \subset \mathscr{S}$. So $\mathscr{S}$ is bounded.

We next complete the proof by establishing the closedness of $\mathscr{S}$. Suppose $\mathscr{S}$ is not a closed set. Then, there exists at least one sequence $\{\boldsymbol{x}_n \in \mathscr{S}\}$ converging to $\bar{\boldsymbol{x}} \notin \mathscr{S}$, which implies either $\bar{\boldsymbol{x}} \in \mathscr{X}$ with $f(\bar{\boldsymbol{x}}) > c$, or $\bar{\boldsymbol{x}} \in \mathrm{bd}(\mathscr{X})$. However, $\bar{\boldsymbol{x}} \in \mathrm{bd}(\mathscr{X})$ further implies that $f(\boldsymbol{x}_n) \to +\infty$ from (A.1), which is a contradiction. If $\bar{\boldsymbol{x}} \in \mathscr{X}$ with $f(\bar{\boldsymbol{x}}) > c$, then there exists a neighbourhood $\mathscr{V}_0$ of $\bar{\boldsymbol{x}}$ such that $f(\boldsymbol{x}) > c$ for any $\boldsymbol{x} \in \mathscr{V}_0$. The existence of $\mathscr{V}_0$ is guaranteed because $f(\boldsymbol{x})$ is continuous at $\bar{\boldsymbol{x}}$. However, because $\{\boldsymbol{x}_n \in \mathscr{S}\}$ converges to $\bar{\boldsymbol{x}} \notin \mathscr{S}$, there exists an integer n_0 such that $\boldsymbol{x}_n \in \mathscr{V}_0$, for $n > n_0$. Hence, $f(\boldsymbol{x}_n) > c$, leading to a contradiction to the assumption that $\{\boldsymbol{x}_n\} \subset \mathscr{S}$. Thus, $\mathscr{S}$ contains all its limit points, proving that $\mathscr{S}$ is closed and the proof is complete.

□

Next, we prove the proposition using the above results. We let the sequence $\{\boldsymbol{\gamma}_i\}^{(t)}|_{t=1}^{\infty}$ be generated by the AM in Algorithm 2 with the starting point $\{\boldsymbol{\gamma}_i\}^{(1)} \in \mathscr{C}_+$, where we omit the EM index r for brevity. We also define the mapping from $\{\boldsymbol{\gamma}_i\}^{(t)}$ to $\{\boldsymbol{\gamma}_i\}^{(t+1)}$ in Algorithm 2 as a function $M(\cdot)$, i.e., $M(\{\boldsymbol{\gamma}_i\}^{(t)}) = \{\boldsymbol{\gamma}_i\}^{(t+1)}$. We prove the result using the Zangwill's convergence theorem [73]. Suppose the following conditions from the Zangwill's convergence theorem hold,

1. If the sequence $\{\boldsymbol{\gamma}_i\}^{(t)}|_{t=1}^{\infty}$ is in a compact set $\mathscr{S} \subset \mathscr{C}_+$.
2. If $\{\boldsymbol{\gamma}_i\}^{(t)}$ is not a stationary point of $Q(\{\boldsymbol{\gamma}_i\})$,
$$Q(\{\boldsymbol{\gamma}_i\}^{(t)}) > Q(\{\boldsymbol{\gamma}_i\}^{(t+1)}). \tag{A.2}$$
3. If $\{\boldsymbol{\gamma}_i\}^{(t)}$ is a stationary point of $Q(\{\boldsymbol{\gamma}_i\})$, the AM step terminates or
$$Q(\{\boldsymbol{\gamma}_i\}^{(t)}) \geq Q(\{\boldsymbol{\gamma}_i\}^{(t+1)}). \tag{A.3}$$
4. Function $Q(\{\boldsymbol{\gamma}_i\})$ is continuous in $\{\boldsymbol{\gamma}_i\}$ and $M(\cdot)$ is continuous at $\{\boldsymbol{\gamma}_i\}^{(t)}$ if $\{\boldsymbol{\gamma}_i\}^{(t)}$ is not a stationary point.

Then, Zangwill's theorem [73] guarantees that the AM algorithm stops at a stationary point of $Q(\{\boldsymbol{\gamma}_i\})$. Consequently, in the remainder of the proof, we verify Conditions 1-4.

We begin with Condition 1. From Lemma 1, we deduce that $\{\boldsymbol{\gamma}_i\}^{(t)}|_{t=1}^{\infty} \subset \mathscr{S}$ where $\mathscr{S}$ is the sublevel set of $Q(\{\boldsymbol{\gamma}_i\}^{(0)})$. So to check Condition 1, we prove that $\mathscr{S}$ is compact. To this end, we establish that $\mathscr{S}$ is compact using the following result. Invoking Proposition 2, it is enough to verify that $Q(\{\boldsymbol{\gamma}_i\})$ satisfies (A.1) on its domain $\mathscr{C}_+$ to ensure compactness of $\mathscr{S}$. For this, we notice that $Q(\{\boldsymbol{\gamma}_i\})$ in (3.12) can be rewritten as

$$Q(\{\boldsymbol{\gamma}_i\}) = \sum_{j=1}^{N^I} \log[\boldsymbol{\gamma}]_j + [\boldsymbol{d}^{(r)}]_j [\boldsymbol{\gamma}]_j^{-1} \tag{A.4}$$

$$= \log[\boldsymbol{\gamma}]_{j_*} + [\boldsymbol{d}^{(r)}]_{j_*} [\boldsymbol{\gamma}]_{j_*}^{-1} + \sum_{j \neq j_*} \log[\boldsymbol{\gamma}]_j + [\boldsymbol{d}^{(r)}]_j [\boldsymbol{\gamma}]_j^{-1}, \tag{A.5}$$

for any $j_* = 1, 2, \ldots, N^I$. Since $\boldsymbol{d}^{(r)} > 0$ and the minimum value of function $f(x) = \log x + a/x$ is attained at $x = a$ and $f(x) \geq f(a) = \log a + 1$, for any $a > 0$, we get

$$Q(\{\boldsymbol{\gamma}_i\}) \geq \log[\boldsymbol{\gamma}]_{j_*} + [\boldsymbol{d}^{(r)}]_{j_*} [\boldsymbol{\gamma}]_{j_*}^{-1} + \sum_{j \neq j_*} \left(\log[\boldsymbol{d}^{(r)}]_j + 1\right). \tag{A.6}$$

The assumption $\boldsymbol{d}^{(r)} > 0$ also implies that the second term, i.e., $\sum_{j \neq j_*} \left(\log[\boldsymbol{d}]_j + 1\right)$, in the lower bound is finite. Therefore, if $[\boldsymbol{\gamma}]_{j_*} \to 0$ or $[\boldsymbol{\gamma}]_{j_*} \to +\infty$, the lower bound on Q goes to $+\infty$, and so, $Q \to +\infty$.

Clearly, $Q \to +\infty$ when $\|\{\boldsymbol{\gamma}_i\}\| = \left(\sum_{i=1}^{I} \|\boldsymbol{\gamma}_i\|_2^2\right)^{1/2} \to +\infty$ because at least one entry $\boldsymbol{\gamma}_{j_*}$ of $\boldsymbol{\gamma}$ satisfies $[\boldsymbol{\gamma}]_{j_*} \to +\infty$. Similarly, when $\mathscr{C}_+ \ni \{\boldsymbol{\gamma}_i\} \to \mathrm{bd}(\mathscr{C}_+)$, there exists an index i_* such that at least one entry of $\boldsymbol{\gamma}_{i_*}$ goes to zero. In that case, we can choose j_* such that

$$[\boldsymbol{\gamma}]_{j_*} = \min_l [\boldsymbol{\gamma}_{i_*}]_l \prod_{i \neq i_*} \|\boldsymbol{\gamma}_i\|_\infty. \tag{A.7}$$

Then, $[\boldsymbol{\gamma}]_{j_*} \to 0$ if $\prod_{i \neq i_*} \|\boldsymbol{\gamma}_i\|_\infty$ is finite; otherwise $[\boldsymbol{\gamma}]_{j_*} \to \infty$. In both cases, $Q \to +\infty$. Hence, (A.1) holds for $Q(\{\boldsymbol{\gamma}_i\})$, and due to Proposition 2, $\mathscr{S}$ is compact. Thus, Condition 1 is established.

We next examine Condition 2. A stationary point $\{\boldsymbol{\gamma}_i^{\mathsf{s}}\} \in \mathscr{C}_+$ of Q satisfies

$$\frac{\partial Q(\{\boldsymbol{\gamma}_i\}|\boldsymbol{\gamma}^{(r)})}{\partial \{\boldsymbol{\gamma}_i\}}\Big|_{\{\boldsymbol{\gamma}_i^{\mathsf{s}}\}} = \mathbf{0}. \tag{A.8}$$

If $\{\boldsymbol{\gamma}_i\}^{(t)} \in \mathscr{C}_+$ is not a stationary point of $Q(\{\boldsymbol{\gamma}_i\})$, there exists at least one index $i_* \in \{1,2,\ldots,I\}$ such that

$$\boldsymbol{\gamma}_{i_*}^{(t)} \neq N^{-I+1}\big[(\otimes_{j=1}^{i_*-1}(\boldsymbol{\gamma}_j^{(t)})^{-1}) \otimes \boldsymbol{I}_N \otimes (\otimes_{j=i_*+1}^{I}(\boldsymbol{\gamma}_j^{(t)})^{-1})\big]^{\mathsf{T}}\boldsymbol{d}^{(r)}, \tag{A.9}$$

$$\boldsymbol{\gamma}_{i}^{(t)} = N^{-I+1}\big[(\otimes_{j=1}^{i-1}(\boldsymbol{\gamma}_j^{(t)})^{-1}) \otimes \boldsymbol{I}_N \otimes (\otimes_{j=n+1}^{I}(\boldsymbol{\gamma}_j^{(t)})^{-1})\big]^{\mathsf{T}}\boldsymbol{d}^{(r)}, \tag{A.10}$$

for $i = 1,2,\ldots,i_*-1$. Therefore, from the AM update (2.16),

$$\tilde{\boldsymbol{\gamma}}_i = \boldsymbol{\gamma}_i^{(t)}, \text{ for } i = 1,2,\ldots,i^*-1. \tag{A.11}$$

Substituting this relation in (2.16) with $i = i_*$, we obtain

$$\tilde{\boldsymbol{\gamma}}_{i_*} = N^{-I+1}\big[(\otimes_{i=1}^{i_*-1}(\boldsymbol{\gamma}_i^{(t)})^{-1}) \otimes \boldsymbol{I}_N \otimes (\otimes_{i=i_*+1}^{I}(\boldsymbol{\gamma}_i^{(t)})^{-1})\big]^{\mathsf{T}}\boldsymbol{d}^{(r)}. \tag{A.12}$$

The AM update in (2.16) also guarantees

$$\tilde{\boldsymbol{\gamma}}_{i_*} = \arg\min_{\boldsymbol{\gamma}_{i_*}} Q\left(\{\tilde{\boldsymbol{\gamma}}_i\}_{i=1}^{i_*-1}, \boldsymbol{\gamma}_{i_*}, \{\boldsymbol{\gamma}_i^{(t)}\}_{i=i_*+1}^{I}\right) \neq \boldsymbol{\gamma}_{i_*}^{(t)}. \tag{A.13}$$

So using (A.11) and the above relation, we conclude

$$Q\left(\{\boldsymbol{\gamma}_i\}^{(t)}\right) = Q\left(\{\tilde{\boldsymbol{\gamma}}_i\}_{i=1}^{i_*-1}, \boldsymbol{\gamma}_{i_*}^{(t)}, \{\boldsymbol{\gamma}_i^{(t)}\}_{i=i_*+1}^{I}\right) \tag{A.14}$$

$$> Q\left(\{\tilde{\boldsymbol{\gamma}}_i\}_{i=1}^{i_*-1}, \tilde{\boldsymbol{\gamma}}_{i_*}, \{\boldsymbol{\gamma}_i^{(t)}\}_{i=i_*+1}^{I}\right) \geq Q(\{\boldsymbol{\gamma}_i\}^{(t+1)}), \tag{A.15}$$

using arguments similar to (2.22). Thus, Condition 2 is verified.

Further, to check Condition 3, we note that if $\{\boldsymbol{\gamma}_i\}^{(t)} \in \mathscr{C}_+$ is a stationary point, then it satisfies (A.10) for $i = 1,2,\ldots,I$. Then, $M\left(\{\boldsymbol{\gamma}_i\}^{(t)}\right) = \{\boldsymbol{\gamma}_i\}^{(t)}$ and $Q\left(\{\boldsymbol{\gamma}_i\}^{(t)}\right) = Q\left(\{\boldsymbol{\gamma}_i\}^{(t+1)}\right)$. So, Condition 3 holds.

Finally, we note that $M(\cdot)$ and Q involve the operations $\det(\cdot)$, $\log(\cdot)$, and $(\cdot)^{-1}$ that are continuous on $\mathscr{C}_+$. Hence, $M(\cdot)$ and $Q(\{\boldsymbol{\gamma}_i\})$ are also continuous, satisfying Condition 4. Thus, the proof is complete.

A.2. Proof of Theorem 1

Theorem 1 is proven using the convergence guarantees for the generalized EM (GEM) [71, Theorem 6]. GEM is an iterative algorithm similar to EM where the numerically infeasible M-step is replaced with a point-to-set map such that for every iteration r,

$$Q(\{\boldsymbol{\gamma}_i\}^{(r)}|\{\boldsymbol{\gamma}_i\}^{(r)}) \geq Q(\{\boldsymbol{\gamma}_i\}^{(r+1)}|\{\boldsymbol{\gamma}_i\}^{(r)}), \tag{A.16}$$

always holds for Q defined in (3.12). Unlike EM, the GEM algorithm does not require achieving the global minimum for the M-step optimization problem. We recall from Lemma 1 AM-KroSBL is a GEM algorithm. Now, suppose the following conditions of [71, Theorem 6] holds,

1. the domain of $\mathscr{L}$ is a subset in $\bar{N}$-th dimensional Euclidean space $\mathbb{R}^{\bar{N}}$,
2. KroSBL cost function $\mathscr{L}$ is continuous in its domain and differentiable in the interior of the domain,
3. the sublevel set of $\mathscr{L}(\{\boldsymbol{\gamma}_i\}^{(1)})$ is compact for any initialization $\{\boldsymbol{\gamma}_i\}^{(1)}$ with $\mathscr{L}(\{\boldsymbol{\gamma}_i\}^{(1)}) < +\infty$,
4. the sequence $\{\boldsymbol{\gamma}_i\}^{(r)}|_{r=1}^{\infty}$ generated by AM-KroSBL has the additional property $\nabla Q = \frac{\partial}{\partial\{\boldsymbol{\gamma}_i\}^{(r+1)}} Q(\{\boldsymbol{\gamma}_i\}^{(r+1)}|\boldsymbol{\gamma}^{(r)}) = 0$,
5. ∇Q is continuous in both $\{\boldsymbol{\gamma}_i\}^{(r+1)}$ and $\{\boldsymbol{\gamma}_i\}^{(r)}$.

Then, Theorem 6 in [71] ensures that the sequence $\{\boldsymbol{\gamma}_i\}^{(r)}|_{r=1}^{\infty}$ generated by AM-KroSBL converges to the stationary points of $\mathscr{L}$. Here, Condition 1 is trivially satisfied because the domain of $\mathscr{L}$, i.e., $\mathscr{C}$, is a subset of the $\bar{N}$-dimensional Euclidean space. Similarly, $\mathscr{L} = \log|\boldsymbol{\Sigma}_{\boldsymbol{y}}| + \boldsymbol{y}^{\mathsf{H}}\boldsymbol{\Sigma}_{\boldsymbol{y}}^{-1}\boldsymbol{y}$ is a continuous function of $\{\boldsymbol{\gamma}_i\}$ because matrix inversion and determinant are continuous in its entries [145, Theorems 5.19 and 5.20]. Further, the derivative of $\mathscr{L}$ exists everywhere in $\mathscr{C}$, and thus, Condition 2 is also satisfied. So, in the remainder of the proof, we verify if Conditions 3-5 to establish the convergence guarantee of AM-KroSBL.

To verify Condition Conditions 3, we first show the compactness of the sublevel set of $\mathscr{L}$. Compactness is established via the coercivity of the KroSBL cost function $\mathscr{L}$ since the sublevel sets of coercive functions are compact [144]. By definition [146], function $\mathscr{L}$ is coercive if $\lim_{\|\{\boldsymbol{\gamma}_i\}\|\to+\infty}\mathscr{L} = +\infty$, where $\|\{\boldsymbol{\gamma}_i\}\| = \left(\sum_{i=1}^{I}\|\boldsymbol{\gamma}_i\|_2^2\right)^{1/2}$. Further, since $\boldsymbol{\Sigma}_{\boldsymbol{y}} = \sigma^2\boldsymbol{I}_{\bar{M}} + \boldsymbol{H}\boldsymbol{\Gamma}\boldsymbol{H}^{\mathsf{H}}$ is positive-definite (PD) when $\sigma^2 > 0$, we have

$$\mathscr{L} = \log|\boldsymbol{\Sigma}_{\boldsymbol{y}}| + \boldsymbol{y}^{\mathsf{H}}\boldsymbol{\Sigma}_{\boldsymbol{y}}^{-1}\boldsymbol{y} > \log|\boldsymbol{\Sigma}_{\boldsymbol{y}}| = \sum_{j=1}^{\bar{M}}\log(\sigma^2+\lambda_j), \tag{A.17}$$

where $\lambda_j \geq 0$ is the jth eigenvalue of $\boldsymbol{H}\boldsymbol{\Gamma}\boldsymbol{H}^{\mathsf{H}}$. So $\mathscr{L}$ is coercive if

$$\lim_{\|\{\boldsymbol{\gamma}_i\}\|\to+\infty}\sum_{j=1}^{\bar{M}}\log(\sigma^2+\lambda_j) = +\infty, \tag{A.18}$$

which is true if at least one of the λ_j's goes to infinity as $\|\{\boldsymbol{\gamma}_i\}\|$ goes to $+\infty$. Moreover, from the boundedness assumption on the norm of the dictionary columns, we derive

$$\sum_{j=1}^{\bar{M}}\lambda_j = \mathrm{tr}(\boldsymbol{H}\boldsymbol{\Gamma}\boldsymbol{H}^{\mathsf{H}}) = \sum_{i=1}^{N^I}[\boldsymbol{\gamma}]_i\|[\boldsymbol{H}]_i\|_2^2 \geq \epsilon^2\sum_{i=1}^{N^I}[\boldsymbol{\gamma}]_i \geq \epsilon^2\|\boldsymbol{\gamma}\|_{\infty}. \tag{A.19}$$

So $\lim_{\|\{\boldsymbol{\gamma}_i\}\|\to+\infty}\sum_{j=1}^{\tilde{M}}\lambda_j = +\infty$, meaning at least one eigenvalue goes to $+\infty$, proving (A.18). Thus, KroSBL cost function $\mathscr{L}$ is a coercive function on $\mathscr{C}$, and its sublevel set of the starting point $\{\boldsymbol{\gamma}_i\}^{(1)}$ is compact. So, Condition 3 is true.

We next verify Condition 4, which requires the point $\{\boldsymbol{\gamma}_i\}^{(r+1)}$ to be a stationary point of $Q(\{\boldsymbol{\gamma}_i\}^{(r+1)}|\boldsymbol{\gamma}^{(r)})$. According to Proposition 1, it holds if $\boldsymbol{d}^{(r)} > 0$. So, we next show that $\boldsymbol{d}^{(r)} = \text{diag}(\boldsymbol{\Sigma}_{\boldsymbol{x}} + \boldsymbol{\mu}_{\boldsymbol{x}}\boldsymbol{\mu}_{\boldsymbol{x}}^{\mathsf{H}}) > 0$. Since $\text{diag}(\boldsymbol{\mu}_{\boldsymbol{x}}\boldsymbol{\mu}_{\boldsymbol{x}}^{\mathsf{H}}) \geq 0$, it suffices to show that $\text{diag}(\boldsymbol{\Sigma}_{\boldsymbol{x}}) > 0$. Also, since PD matrices have positive diagonal entries, from (3.13), it is enough to verify that $\boldsymbol{\Sigma}_{\boldsymbol{x}}^{-1} = \sigma^{-2}\boldsymbol{H}^{\mathsf{H}}\boldsymbol{H} + \text{diag}(\boldsymbol{\gamma}^{(r)})^{-1}$ is PD. So, $\boldsymbol{d}^{(r)} > 0$ if $\{\boldsymbol{\gamma}_i\}^{(r)} > 0$ because $\sigma^{-2}\boldsymbol{H}^{\mathsf{H}}\boldsymbol{H}$ is PSD [147]. From this observation, we prove the condition $\boldsymbol{d}^{(r)} > 0$ using induction. Since $\{\boldsymbol{\gamma}_i\}^{(1)} \in \mathscr{C}_+$, we have $\boldsymbol{d}^{(1)} > 0$. Next, we assume that $\boldsymbol{d}^{(r)} > 0$, for some $r > 0$. From Proposition 1, $\{\boldsymbol{\gamma}_i\}^{(r+1)}$ generated by the AM algorithm is bounded away from $\text{bd}(\mathscr{C}_+)$ when $\boldsymbol{d}^{(r)} > 0$. Thus, we conclude that $\{\boldsymbol{\gamma}_i\}^{(r+1)} > 0$, which in turn implies $\boldsymbol{d}^{(r+1)} > 0$. Condition 4 holds in our case.

Finally, we show Condition 5 by first computing the gradient of Q with respect to $\boldsymbol{\gamma}_i$ as

$$\frac{\partial}{\partial\boldsymbol{\gamma}_i}Q(\{\boldsymbol{\gamma}_i\}|\boldsymbol{\gamma}^{(r)}) = -N^{I-1}\boldsymbol{\gamma}_i^{-1} + \text{diag}(\boldsymbol{\gamma}_i)^{-2} \times\left[\left(\otimes_{l=1}^{i-1}(\boldsymbol{\gamma}_l)^{-1}\right)\otimes\boldsymbol{I}_N\otimes\left(\otimes_{l=i+1}^{I}(\boldsymbol{\gamma}_l^{(r,t)})^{-1}\right)\right]^{\mathsf{T}}\boldsymbol{d}^{(r)}. \quad \text{(A.20)}$$

Here, operators $(\cdot)^{-1}$ and $(\cdot)^{-2}$ are continuous in $(0,+\infty)$. Thus, the gradient is continuous in $\{\boldsymbol{\gamma}_i\}$. Finally, in ∇Q, only $\boldsymbol{d}^{(r)}$ depends on $\{\boldsymbol{\gamma}_i\}^{(r)}$ as in (3.13). Since the matrix inversion is continuous in its entries [145, Theorem 5.20], $\boldsymbol{\Sigma}_{\boldsymbol{x}}$ and $\boldsymbol{\mu}_{\boldsymbol{x}}$ are continuous in $\{\boldsymbol{\gamma}_i\}^{(r)}$. Therefore, ∇Q is continuous in $\{\boldsymbol{\gamma}_i\}^{(r)}$. Thus, Condition 5 is established, and the proof is complete.

A.3. Proof of Theorem 2

To prove the sparsity of local minima, we start with a few supporting lemmas.

Lemma 6. $\log|\boldsymbol{\Sigma}_{\boldsymbol{y}}|$ *is concave with respect to* $\{\boldsymbol{\gamma}_i\}$ *in the noiseless case, i.e.,* $\sigma^2 = 0$.

Proof. When $\sigma^2 = 0$, $\boldsymbol{\Sigma}_{\boldsymbol{y}} = \boldsymbol{H}\boldsymbol{\Gamma}\boldsymbol{H}^{\mathsf{H}} = \otimes_{i=1}^{I}\boldsymbol{H}_i\boldsymbol{\Gamma}_i\boldsymbol{H}_i^{\mathsf{H}}$. We have

$$\log|\boldsymbol{\Sigma}_{\boldsymbol{y}}| = \log|\otimes_{i=1}^{I}\boldsymbol{H}_i\boldsymbol{\Gamma}_i\boldsymbol{H}_i^{\mathsf{H}}| = \sum_{i=1}^{I}\left(\prod_{j\neq i}M_j\right)\log|\boldsymbol{H}_i\boldsymbol{\Gamma}_i\boldsymbol{H}_i^{\mathsf{H}}|. \quad \text{(A.21)}$$

Since $\boldsymbol{H}_i\boldsymbol{\Gamma}_i\boldsymbol{H}_i^{\mathsf{H}}$ is a PSD matrix and affine in $\boldsymbol{\gamma}_i$, and function $\log|\cdot|$ is a concave function in the space of PSD matrices, $\log|\boldsymbol{H}_i\boldsymbol{\Gamma}_i\boldsymbol{H}_i^{\mathsf{H}}|$ is concave in $\boldsymbol{\gamma}_i$. Thus, $\log|\boldsymbol{\Sigma}_{\boldsymbol{y}}|$ is concave because the sum of concave functions is also concave. □

Lemma 7. *If* $\{\boldsymbol{\gamma}_i\}$ *satisfies* $\boldsymbol{b} = \boldsymbol{A}(\otimes_{i=1}^{I}\boldsymbol{\gamma}_i)$, *where*

$$\boldsymbol{b} = \boldsymbol{y} - \sigma^2\boldsymbol{u}; \quad \boldsymbol{A} = \boldsymbol{H}\,\text{diag}(\boldsymbol{H}^{\mathsf{H}}\boldsymbol{u}), \quad \text{(A.22)}$$

and $\boldsymbol{u}$ *is any fixed vector such that* $\boldsymbol{y}^{\mathsf{H}}\boldsymbol{u} = C$, *then* $\boldsymbol{y}^{\mathsf{H}}\boldsymbol{\Sigma}_{\boldsymbol{y}}^{-1}\boldsymbol{y}$ *is a constant* C *for any value of* $\sigma^2 \geq 0$.

Proof. We combine $\boldsymbol{b} = \boldsymbol{A}(\otimes_{i=1}^{I} \boldsymbol{\gamma}_i)$ and (A.22) to obtain

$$\boldsymbol{y} = \boldsymbol{A}(\otimes_{i=1}^{I} \boldsymbol{\gamma}_i) + \sigma^2 \boldsymbol{u} = \boldsymbol{H}\,\mathrm{diag}(\boldsymbol{H}^{\mathsf{H}}\boldsymbol{u})(\otimes_{i=1}^{I} \boldsymbol{\gamma}_i) + \sigma^2 \boldsymbol{u} \tag{A.23}$$

$$= \boldsymbol{H}\boldsymbol{\Gamma}\boldsymbol{H}^{\mathsf{H}}\boldsymbol{u} + \sigma^2 \boldsymbol{u} = \boldsymbol{\Sigma}_{\boldsymbol{y}}\boldsymbol{u}, \tag{A.24}$$

where $\boldsymbol{\Gamma} = \mathrm{diag}(\otimes_{i=1}^{I} \boldsymbol{\gamma}_i)$. Then, $\boldsymbol{y}^{\mathsf{H}}\boldsymbol{\Sigma}_{\boldsymbol{y}}^{-1}\boldsymbol{y} = \boldsymbol{y}^{\mathsf{H}}\boldsymbol{u} = C$, for any value $\sigma^2 \geq 0$.

□

Lemma 8. *Consider the set of linear equations with* $\boldsymbol{t}_1, \boldsymbol{t}_2 \neq \boldsymbol{0}$,

$$(\boldsymbol{\Phi}_1 \otimes \boldsymbol{\Phi}_2)\,(\boldsymbol{w}_1 \otimes \boldsymbol{w}_2) = \boldsymbol{t}_1 \otimes \boldsymbol{t}_2. \tag{A.25}$$

Seeking $\boldsymbol{w}_1$ *and* $\boldsymbol{w}_2$ *that satisfy (A.25) is equivalent to solving*

$$\boldsymbol{\Phi}_1 \boldsymbol{w}_1 = \alpha \boldsymbol{t}_1;\ \boldsymbol{\Phi}_2 \boldsymbol{w}_2 = \alpha^{-1} \boldsymbol{t}_2, \tag{A.26}$$

where α *is any non-zero scalar.*

Proof. We rewrite (A.25) as $(\boldsymbol{\Phi}_1 \boldsymbol{w}_1) \otimes (\boldsymbol{\Phi}_2 \boldsymbol{w}_2) = \boldsymbol{t}_1 \otimes \boldsymbol{t}_2$. Now, we can arrange this equation as $(\boldsymbol{\Phi}_2 \boldsymbol{w}_2)\,(\boldsymbol{\Phi}_1 \boldsymbol{w}_1)^{\mathsf{T}} = \boldsymbol{t}_2 \boldsymbol{t}_1^{\mathsf{T}}$, which is a rank-one matrix. Here, $\boldsymbol{t}_2 \boldsymbol{t}_1^{\mathsf{T}}$ has at least one nonzero column since $\boldsymbol{t}_1, \boldsymbol{t}_2 \neq \boldsymbol{0}$, and every column of $\boldsymbol{t}_2 \boldsymbol{t}_1^{\mathsf{T}}$ is a scaled version $\boldsymbol{t}_2$. Therefore, the solution to the system of equations is given by (A.26), leading to the desired conclusion.

□

With all the mentioned lemmas, we present the proof of Theorem 2. First, we pose another optimization problem:

$$\min_{\{\boldsymbol{\gamma}_i\} \geq 0} \log|\boldsymbol{\Sigma}_{\boldsymbol{y}}| \text{ s.t. } \boldsymbol{A}\left(\otimes_{i=1}^{I} \boldsymbol{\gamma}_i\right) = \boldsymbol{b}, \tag{A.27}$$

where $\boldsymbol{A} = \boldsymbol{H}\,\mathrm{diag}(\boldsymbol{H}^{\mathsf{H}}\boldsymbol{u})$ and $\boldsymbol{b} = \boldsymbol{y} - \sigma^2 \boldsymbol{u}$. As Lemma 7, the constraint $\boldsymbol{A}(\otimes_{i=1}^{I} \boldsymbol{\gamma}_i) = \boldsymbol{b}$ holds the second term of $\mathscr{L}$ constant, and we minimize the first term of $\mathscr{L}$, which is concave, over a bounded convex polytope. Then, any local minimum of (2.9) denoted by $\{\boldsymbol{\gamma}_i^*\}$, must also be a local minimum of (A.27) with

$$\boldsymbol{y}^{\mathsf{H}}\left(\sigma^2 \boldsymbol{I}_{\bar{M}} + \boldsymbol{H}\,\mathrm{diag}(\otimes_{i=1}^{I} \boldsymbol{\gamma}_i^*)\boldsymbol{H}^{\mathsf{H}}\right)^{-1} \boldsymbol{y} = C^* = \boldsymbol{y}^{\mathsf{H}}\boldsymbol{u}^*, \tag{A.28}$$

as long as there exists a vector $\boldsymbol{u}^*$ satisfying $C^* = \boldsymbol{y}^{\mathsf{H}}\boldsymbol{u}^*$ to construct $\boldsymbol{A}$ and $\boldsymbol{b}$. A candidate for $\boldsymbol{u}^*$ in the noiseless setting is

$$\boldsymbol{u}^* = C^* \frac{\boldsymbol{y}}{\|\boldsymbol{y}\|_2^2} = C^* \frac{\otimes_{i=1}^{I} \boldsymbol{H}_i \boldsymbol{x}_i}{\|\boldsymbol{y}\|_2^2}, \tag{A.29}$$

where we also use (4.1), (1.2) and the assumption that $\boldsymbol{x} = \otimes_i^I \boldsymbol{x}_i$. Thus, we obtain

$$\boldsymbol{A} = \frac{C^*}{\|\boldsymbol{y}\|_2^2} \boldsymbol{H}\,\mathrm{diag}(\boldsymbol{H}^{\mathsf{H}} \otimes_{i=1}^{I} \boldsymbol{H}_i \boldsymbol{x}_i) = \frac{C^*}{\|\boldsymbol{y}\|_2^2} \otimes_{i=1}^{I} \boldsymbol{H}_i\,\mathrm{diag}(\boldsymbol{H}_i^{\mathsf{H}} \boldsymbol{H}_i \boldsymbol{x}_i).$$

Similarly, we have

$$\boldsymbol{b} = \boldsymbol{y} = \otimes_{i=1}^{I} \boldsymbol{H}_i \boldsymbol{x}_i. \tag{A.30}$$

Thus, using Lemma 8 with the scaling factor $\alpha = 1$, the constraint $\boldsymbol{A}(\otimes_{i=1}^{I} \boldsymbol{\gamma}_i) = \boldsymbol{b}$ can be written as

$$\boldsymbol{A}_i \boldsymbol{\gamma}_i = \boldsymbol{b}_i, \forall i = 1, 2, \ldots, I, \tag{A.31}$$

where $\boldsymbol{A}_i = \left(C^* / \|\boldsymbol{y}\|_2^2\right)^{1/I} \boldsymbol{H}_i \operatorname{diag}(\boldsymbol{H}_i^{\mathsf{H}} \boldsymbol{H}_i \boldsymbol{x}_i)$, and $\boldsymbol{b}_i = \boldsymbol{H}_i \boldsymbol{x}_i$. Problem (A.27) can be transformed into I separated problems as

$$\min_{\boldsymbol{\gamma}_i} \log|\boldsymbol{H}_i \boldsymbol{\Gamma}_i \boldsymbol{H}_i^{\mathsf{H}}| \quad \text{s.t. } \boldsymbol{A}_i \boldsymbol{\gamma}_i = \boldsymbol{y}_i, \boldsymbol{\gamma}_i \geq 0, \tag{A.32}$$

for $i = 1, 2, \ldots, I$. Any local minimum of $\mathscr{L}$, e.g., $\{\boldsymbol{\gamma}_i\}^*$ is also a local minimum of (A.32). Furthermore, all local minima of (A.32) are achieved at extreme points, which are also basic feasible solutions with at most M_i non-zeros for each $\boldsymbol{\gamma}_i$ [148]. Thus, $\{\boldsymbol{\gamma}_i^*\}$ is sparse when noise is absent.

A.4. Proof of Theorem 3

The proof is based on the following lemma:

Lemma 9. *If $\boldsymbol{H} = \otimes_{i=1}^{I} \boldsymbol{H}_i$ satisfies URP, then $\boldsymbol{H}_i$ for all i also satisfy URP.*

Proof. Suppose we choose any M_i columns of $\boldsymbol{H}_i$ indexed by $\mathscr{M}_i$, for $i = 1, 2, \ldots, I$. Then, the Kronecker product of the corresponding submatrices, given by $\otimes_{i=1}^{I} [\boldsymbol{H}_i]_{\mathscr{M}_i} \in \mathbb{C}^{\bar{M} \times \bar{M}}$, is a submatrix of $\boldsymbol{H} = \otimes_{i=1}^{I} \boldsymbol{H}_i$. Since $\boldsymbol{H}$ satisfies URP, any subset of $\bar{M}$ columns of $\boldsymbol{H}$ are linearly independent, we get

$$\bar{M} = \operatorname{rank}\left([\boldsymbol{H}]_{\otimes_{i=1}^{I} \mathscr{M}_i}\right) = \prod_{i=1}^{I} \operatorname{rank}\left([\boldsymbol{H}_i]_{\mathscr{M}_i}\right) \leq \prod_{i=1}^{I} M_i = \bar{M}. \tag{A.33}$$

The last step follows as $\operatorname{rank}\left([\boldsymbol{H}_i]_{\mathscr{M}_i}\right) \leq M_i$. So, (A.33) holds if and only if $\operatorname{rank}\left([\boldsymbol{H}_i]_{\mathscr{M}_i}\right) = M_i$ for $i = 1, 2, \ldots, I$. Hence, any subset of M_i columns of $\boldsymbol{H}_i$ is linearly independent, and $\boldsymbol{H}_i$ satisfies URP, for $i = 1, 2, \ldots, I$. □

Lemma 9 ensures that $\boldsymbol{H}_i$ satisfies the URP for all i. Then, we next show that for any index set $\mathscr{M}_i$ such that $|\mathscr{M}_i| = M_i$, if we restrict the nonzero values of $\boldsymbol{\gamma}_i$ to the set $\mathscr{M}_i$, there can only be one minimum for $\mathscr{L}(\{[\boldsymbol{\gamma}_i]_{\mathscr{M}_i}\})$. For this, we note that

$$\mathscr{L}(\{[\boldsymbol{\gamma}_i]_{\mathscr{M}_i}\}) = \log\left|\otimes_{i=1}^{I} [\boldsymbol{H}_i]_{\mathscr{M}_i} [\boldsymbol{\Gamma}_i]_{\mathscr{M}_i} [\boldsymbol{H}_i]_{\mathscr{M}_i}^{\mathsf{H}}\right| + \boldsymbol{y}^{\mathsf{H}} \left(\otimes_{i=1}^{I} [\boldsymbol{H}_i]_{\mathscr{M}_i} [\boldsymbol{\Gamma}_i]_{\mathscr{M}_i} [\boldsymbol{H}_i]_{\mathscr{M}_i}^{\mathsf{H}}\right)^{-1} \boldsymbol{y} \tag{A.34}$$

$$= \sum_{i=1}^{I} \left(\prod_{j \neq i} M_j\right) \log\left|[\boldsymbol{H}_i]_{\mathscr{M}_i} [\boldsymbol{\Gamma}_i]_{\mathscr{M}_i} [\boldsymbol{H}_i]_{\mathscr{M}_i}^{\mathsf{H}}\right| \tag{A.35}$$

$$+ \boldsymbol{y}^{\mathsf{H}} \otimes_{i=1}^{I} \left(\left([\boldsymbol{H}_i]_{\mathscr{M}_i}^{\mathsf{H}}\right)^{-1} [\boldsymbol{\Gamma}_i]_{\mathscr{M}_i}^{-1} [\boldsymbol{H}_i]_{\mathscr{M}_i}^{-1}\right) \boldsymbol{y}. \tag{A.36}$$

Here, the second term can be simplified using the assumption

$$\boldsymbol{y} = \left(\otimes_{i=1}^{I} [\boldsymbol{H}_i]_{\mathcal{M}_i}\right)\left(\otimes_{i=1}^{I} [\boldsymbol{x}_i]_{\mathcal{M}_i}\right) = \otimes_{i=1}^{I} ([\boldsymbol{H}_i]_{\mathcal{M}_i} [\boldsymbol{x}_i]_{\mathcal{M}_i}),$$

as follows:

$$\boldsymbol{y}^{\mathsf{H}} \otimes_{i=1}^{I} \left([\boldsymbol{H}_i]_{\mathcal{M}_i} [\boldsymbol{\Gamma}_i]_{\mathcal{M}_i} [\boldsymbol{H}_i]_{\mathcal{M}_i}^{\mathsf{H}}\right)^{-1} \boldsymbol{y}$$
$$= \otimes_{i=1}^{I} [\boldsymbol{x}_i]_{\mathcal{M}_i}^{\mathsf{H}} [\boldsymbol{\Gamma}_i]_{\mathcal{M}_i}^{-1} [\boldsymbol{x}_i]_{\mathcal{M}_i} = \prod_{i=1}^{I} [\boldsymbol{x}_i]_{\mathcal{M}_i}^{\mathsf{H}} [\boldsymbol{\Gamma}_i]_{\mathcal{M}_i}^{-1} [\boldsymbol{x}_i]_{\mathcal{M}_i}. \quad \text{(A.37)}$$

Therefore, from (A.36), we arrive at

$$\mathscr{L}(\{[\boldsymbol{\gamma}_i]_{\mathcal{M}_i}\}) = \sum_{i=1}^{I} \prod_{j \neq i} M_j \log \left| [\boldsymbol{H}_i]_{\mathcal{M}_i} [\boldsymbol{H}_i]_{\mathcal{M}_i}^{\mathsf{H}} \right|$$
$$+ \sum_{i=1}^{I} \prod_{j \neq i} M_j \log \left| [\boldsymbol{\Gamma}_i]_{\mathcal{M}_i} \right| + \prod_{i=1}^{I} [\boldsymbol{x}_i]_{\mathcal{M}_i}^{\mathsf{H}} [\boldsymbol{\Gamma}_i]_{\mathcal{M}_i}^{-1} [\boldsymbol{x}_i]_{\mathcal{M}_i}. \quad \text{(A.38)}$$

Setting the derivative of the above function with respect to $\boldsymbol{\gamma}_i$ to zero gives

$$\prod_{j \neq i} M_j [\boldsymbol{\gamma}_i]_{\mathcal{M}_i} = [\boldsymbol{x}_i]_{\mathcal{M}_i}^2 \prod_{j \neq i} [\boldsymbol{x}_j]_{\mathcal{M}_j}^{\mathsf{H}} [\boldsymbol{\Gamma}_j]_{\mathcal{M}_j}^{-1} [\boldsymbol{x}_j]_{\mathcal{M}_j}, \quad \text{(A.39)}$$

for $i = 1, 2, \ldots, I$. However, since $\{\boldsymbol{\gamma}_i\} \in \mathscr{C}$, we obtain the unique minimum of $\mathscr{L}(\{[\boldsymbol{\gamma}_i]_{\mathcal{M}_i}\})$ at

$$[\boldsymbol{\gamma}_i]_{\mathcal{M}_i} = \begin{cases} \frac{[\boldsymbol{x}_i]_{\mathcal{M}_i}^2}{\|[\boldsymbol{x}_i]_{\mathcal{M}_i}^2\|}, & \text{if } i < I, \\ [\boldsymbol{x}_I]_{\mathcal{M}_I}^2 \prod_{j=1}^{I-1} \|[\boldsymbol{x}_j]_{\mathcal{M}_j}^2\|, & \text{if } i = I. \end{cases} \quad \text{(A.40)}$$

Therefore, every set of $\{\mathcal{M}_I\}_{i=1}^{I}$ corresponds to one unique local minimum, and counting for all possible index set combinations, we get $\mathscr{N} \leq \prod_{i=1}^{I} \binom{N}{M_i}$. Further, all index set combinations containing a degenerate solution share the same minimum [62]. Accounting for such repetitions, we derive (2.25). Thus, the proof is complete.

B

Appendix of Chapter 3

B.1. Proof of Theorem 4

We define $\boldsymbol{P}_{\boldsymbol{U}}$ as the projection matrix onto the column space of a given matrix $\boldsymbol{U}$ and $\boldsymbol{U}_{\perp}$ as the projection onto its orthogonal subspace. Also, $\|\cdot\|$ is the matrix spectral norm. We need the below lemma for the proof.

Lemma 10. *[149, Supplement Sec. 1.2] Suppose $\boldsymbol{X} \in \mathbb{R}^{p_1 \times p_2}$ is a rank-r matrix and $\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{Z}$ where the entries of $\boldsymbol{Z} \in \mathbb{R}^{p_1 \times p_2}$ follow a zero mean Gaussian distribution with unit variance. We denote $\boldsymbol{V}, \hat{\boldsymbol{V}} \in \mathbb{R}^{p_2 \times r}$ as the matrix of the right singular vectors of $\boldsymbol{X}$ and the matrix of the top r right singular vectors of $\boldsymbol{Y}$, respectively. Suppose the r th right singular value σ_r^2 of $\boldsymbol{X}$ satisfies $\sigma_r^2 \geq C_{\text{gap}}\sigma^2(\sqrt{p_1 p_2} + p_2)$ for some large constant $C_{\text{gap}} > 0$. Then, for all $x \geq 0$, there exist constants $C, c > 0$ such that*

$$\mathbb{P}\left\{\|\boldsymbol{P}_{\boldsymbol{Y}\boldsymbol{V}}\boldsymbol{Y}\boldsymbol{V}_{\perp}\| \leq x\right\} \geq 1 - C\exp\left\{-c(\sigma_r^2 + p_1)\right\} - C\exp\left\{Cp_2 - c\min\left(x^2, x\sqrt{\sigma_r^2 + p_1}\right)\right\},$$

and with probability exceeding $1 - C\exp\left\{-c\sigma_r^4/(\sigma_r^2 + p_1)\right\}$,

$$\|\sin\Theta(\hat{\boldsymbol{V}}, \boldsymbol{V})\|^2 \leq C(\sigma_r^2 + p_1)\sigma_r^{-4}\|\boldsymbol{P}_{\boldsymbol{Y}\boldsymbol{V}}\boldsymbol{Y}\boldsymbol{V}_{\perp}\|^2,$$

Here, $\Theta(\hat{\boldsymbol{V}}, \boldsymbol{V}) = \text{diag}(\arccos(\sigma_1), \cdots, \arccos(\sigma_r))$ where $\sigma_1 \geq \cdots \geq \sigma_r \geq 0$ are the singular values of $\boldsymbol{V}^{\top}\hat{\boldsymbol{V}}$.

We prove Theorem 4 for $i = 1$, and $i = 2, \cdots, I$ follow similarly. Also, we consider the decomposition of $\bar{\boldsymbol{y}}/\sigma_{\text{t}}$ instead of $\bar{\boldsymbol{y}}$. This scaling does not alter the subspaces obtained after decomposition but ensures that the noise entries follow a zero-mean, unit-variance Gaussian distribution, as in Lemma 10.

For $i = 1$, the true and estimated subspaces are spanned by $\boldsymbol{y}_1/\|\boldsymbol{y}_1\|$ and $\hat{\boldsymbol{y}}_1/\|\hat{\boldsymbol{y}}_1\|$, respectively. The first mode matricization of the tensor $\boldsymbol{\mathcal{Y}}$, as defined in (3.5), is $\boldsymbol{Y}_{(1)}$. Setting $r = 1$, $p_1 = \bar{M}/M_1$ and $p_2 = M_1$, and consequently, $\sigma_r = \lambda$ in Lemma 10, we derive

$$\sin^2\vartheta_1 = \|\sin\Theta(\hat{\boldsymbol{y}}_1/\|\hat{\boldsymbol{y}}_1\|, \boldsymbol{y}_1/\|\boldsymbol{y}_1\|)\| \leq C(\lambda^2 + \bar{M}/M_1)\lambda^{-4}\|\boldsymbol{P}_{\boldsymbol{Y}_{(1)}^{\top}\boldsymbol{y}_1}\boldsymbol{Y}_{(1)}^{\top}\boldsymbol{y}_{1\perp}\|^2, \tag{B.1}$$

with probability at least $1 - Ce^{-c\frac{\lambda^4}{\lambda^2 + p_1}}$.

Further, we bound $\|\boldsymbol{P}_{\boldsymbol{Y}_{(1)}^\top \boldsymbol{y}_1} \boldsymbol{Y}_{(1)}^\top \boldsymbol{y}_{1_\perp}\|^2$ using Lemma 10 by setting $x = \sqrt{\tilde{C} M_1}$ where $\tilde{C} < C/c < C_{\text{gap}}$,

$$\mathbb{P}\left\{\|\boldsymbol{P}_{\boldsymbol{Y}\boldsymbol{V}} \boldsymbol{Y} \boldsymbol{V}_\perp\| \le \tilde{C}\sqrt{M_1}\right\} \ge 1 - C\exp\left\{-c(\sigma_r^2 + p_1)\right\} - C\exp\left\{CM_1 - c\min\left(\tilde{C}M_1, \sqrt{\tilde{C}M_1(\sigma_r^2 + p_1)}\right)\right\}. \quad \text{(B.2)}$$

Then, we simplify the right-hand side of (B.2) using

$$\sigma_r^2 + p_1 \ge \sigma_r^2 = \lambda^2 \ge C_{\text{gap}}(\sqrt{p_1 M_1} + M_1) \ge C_{\text{gap}} M_1. \quad \text{(B.3)}$$

Then, (B.2) is simplified as

$$\mathbb{P}\left\{\|\boldsymbol{P}_{\boldsymbol{Y}\boldsymbol{V}} \boldsymbol{Y} \boldsymbol{V}_\perp\| \le \tilde{C}\sqrt{M_1}\right\} \ge 1 - C\exp\left\{-cC_{\text{gap}} M_1\right\} - C\exp\left\{\left[C - c\min(\tilde{C}, \sqrt{\tilde{C}C_{\text{gap}}})\right] M_1\right\} \ge 1 - Ce^{-\tilde{c}M_1},$$

for some constant $\tilde{c}$. So, (B.1) and the union bound implies

$$\sin^2\vartheta_1 \le \frac{C(\lambda^2 + \bar{M}/M_1)}{\lambda^4}\tilde{C}M_1 = \frac{C\tilde{C}M_1}{\lambda^2} + \frac{C\tilde{C}\bar{M}}{\lambda^4}, \quad \text{(B.4)}$$

with probability exceeding $1 - Ce^{-c\frac{\lambda^4}{\lambda^2 + p_1}} - Ce^{-\tilde{c}M_1}$.

Furthermore, since $\lambda^2 \ge C_{\text{gap}}(\sqrt{p_1 M_1} + M_1)$, with $\bar{C} = \min(C_{\text{gap}}, C_{\text{gap}}^2)$, we derive

$$\lambda^2 \ge \sqrt{\bar{C}p_1 M_1} + \bar{C}M_1 \ge \frac{\bar{C}M_1 + \sqrt{\bar{C}^2 M_1^2 + 4\bar{C}p_1 M_1}}{2}.$$

Then, $\lambda^4 - \bar{C}M_1\lambda^2 - \bar{C}p_1 M_1 \ge 0$ because λ^2 is greater than both roots of the quadratic function in λ^2. Thus, we deduce $\frac{\lambda^4}{\lambda^2 + p_1} \ge \bar{C}M_1$. So, from (B.4), we arrive at the desired result,

$$\mathbb{P}\left\{\sin\vartheta_1 \le \frac{\sqrt{C\tilde{C}M_1}}{\lambda} + \frac{\sqrt{C\tilde{C}\bar{M}}}{\lambda^2}\right\} \ge 1 - 2Ce^{-\max(c\bar{C}, \tilde{c})M_1}.$$

B.2. Proof of Theorem 5

Our proof uses the following lemmas.

Lemma 11. *[101, Lemma 6]: Suppose $\boldsymbol{X}, \boldsymbol{Z}$ are two matrices, and the projection matrix orthogonal to the subspace spanned by the leading r left singular vectors of $\boldsymbol{X} + \boldsymbol{Z}$ is $\boldsymbol{U}$. Then, $\|\boldsymbol{U}\boldsymbol{X}\|_{\text{F}} \le 2\sqrt{r}\|\boldsymbol{Z}\|$, where r is the rank of $\boldsymbol{X}$.*

Lemma 12. *[150, Corollary 5.35] Let $\boldsymbol{Z} \in \mathbb{R}^{p_1 \times p_2}$ whose entries are independent Gaussian random variables with zero mean unit variance. Then, for any $x \ge 0$, the matrix satisfies $\|\boldsymbol{Z}\| \ge \sqrt{p_1} + \sqrt{p_2} + x$, with probability less than $2e^{-x^2/2}$.*

Lemma 13. *[151, Lemma 8.1] Suppose X satisfies the non-central $\chi^2_d(\nu)$ distribution with d degrees of freedom and non-centrality parameter ν. Then, for all $x>0$, it satisfies $X \geq (d+\nu)+2\sqrt{(d+2\nu)x}+2x$ with probability less than e^{-x}.*

Lemma 14. *[100] Consider tensors $\mathbf{Y},\mathbf{X}$, and $\mathbf{Z}$, such that $\mathbf{Y}=\mathbf{X}\times_1 \boldsymbol{A}_1\cdots\times_I \boldsymbol{A}_I$, where $\mathbf{X}=\mathbf{Z}\times_1 \boldsymbol{B}_1\cdots\times_I \boldsymbol{B}_I$, for any compatible matrices $\{\boldsymbol{A}_i,\boldsymbol{B}_i\}_{i=1}^I$. Then, $\mathbf{Y}=\mathbf{Z}\times_1(\boldsymbol{A}_1\boldsymbol{B}_1)\cdots\times_I(\boldsymbol{A}_I\boldsymbol{B}_I)$.*

Setting the tensor order $R=I$ and r-ranks as $p_i=1$ for $i\in[I]$ in [102, Eq. (19)] leads to (3.18). To prove the probabilistic bound, we note that using tensor notation, $\|\otimes_{i=1}^I \hat{\boldsymbol{y}}_i - \otimes_{i=1}^I \boldsymbol{y}_i\|_2 = \|\hat{\mathbf{Y}}-\mathbf{Y}\|_{\mathrm{F}}$, where $\hat{\mathbf{Y}}$ is HOSVD output and $\mathbf{Y}=\mathbf{Y}+\bar{\mathbf{N}}$, with $\mathbf{Y},\hat{\mathbf{Y}}$ and $\bar{\mathbf{N}}$ being the measurement, the noiseless signal, and noise tensors, respectively, from (3.4).

HOSVD reconstructs $\hat{\mathbf{Y}}=\xi\times_1 \boldsymbol{e}_1\cdots\times_I \boldsymbol{e}_I$. Here, $\boldsymbol{e}_i$, the leading left singular vectors of the ith mode matricization of $\bar{\mathbf{Y}}$ given by $\boldsymbol{Y}_{(i)}+\boldsymbol{N}_{(i)}$, with $\boldsymbol{Y}_{(i)}$ and $\boldsymbol{N}_{(i)}$ are the ith mode matricization of $\mathbf{Y}$ and $\bar{\mathbf{N}}$, respectively. Also, $\xi=\bar{\mathbf{Y}}\times_1 \boldsymbol{e}_1^{\mathsf{T}}\cdots\times_I \boldsymbol{e}_I^{\mathsf{T}}$, as the signal is real. Lemma 14 implies

$$\hat{\mathbf{Y}}=\left(\mathbf{Y}+\bar{\mathbf{N}}\right)\times_1(\boldsymbol{e}_1\boldsymbol{e}_1^{\mathsf{T}})\times_2\cdots\times_I(\boldsymbol{e}_I\boldsymbol{e}_I^{\mathsf{T}})=\bar{\mathbf{N}}_{(I)}+\mathbf{Y}_{(I)},$$

where we define $\bar{\mathbf{N}}_{(I)}=\bar{\mathbf{N}}\times_1(\boldsymbol{e}_1\boldsymbol{e}_1^{\mathsf{T}})\cdots\times_I(\boldsymbol{e}_I\boldsymbol{e}_I^{\mathsf{T}})$ and $\mathbf{Y}_{(i)}=\mathbf{Y}\times_1(\boldsymbol{e}_1\boldsymbol{e}_1^{\mathsf{T}})\cdots\times_i(\boldsymbol{e}_i\boldsymbol{e}_i^{\mathsf{T}})$ with $\mathbf{Y}_{(0)}=\mathbf{Y}$. Therefore,

$$\|\otimes_{i=1}^I \hat{\boldsymbol{y}}_i-\otimes_{i=1}^I \boldsymbol{y}_i\|_2=\left\|\hat{\mathbf{Y}}-\mathbf{Y}\right\|_{\mathrm{F}}\leq\left\|\mathbf{Y}_{(I)}-\mathbf{Y}\right\|_{\mathrm{F}}+\left\|\bar{\mathbf{N}}_{(I)}\right\|_{\mathrm{F}}. \tag{B.5}$$

To bound the first term in (B.5), let $\boldsymbol{P}_{i_\perp}$ be the projection matrix orthogonal to $\boldsymbol{e}_i$ so that $\mathbf{Y}=\mathbf{Y}\times_1(\boldsymbol{P}_{1_\perp}+(\boldsymbol{e}_1\boldsymbol{e}_1^{\mathsf{T}}))$, leading to

$$\begin{aligned}\mathbf{Y}&=\mathbf{Y}_{(0)}\times_1\boldsymbol{P}_{1_\perp}+\mathbf{Y}_{(1)}=\mathbf{Y}_{(0)}\times_1\boldsymbol{P}_{1_\perp}+\mathbf{Y}_{(1)}\times_2\boldsymbol{P}_{2_\perp}+\mathbf{Y}_{(2)}\\&=\sum_{i=1}^I\mathbf{Y}_{(i-1)}\times_i\boldsymbol{P}_{i_\perp}+\mathbf{Y}_{(I)}.\end{aligned}$$

Therefore, using triangle inequality, we obtain

$$\begin{aligned}\left\|\mathbf{Y}_{(I)}-\mathbf{Y}\right\|_{\mathrm{F}}&\leq\sum_{i=1}^I\left\|\mathbf{Y}_{(i-1)}\times_i\boldsymbol{P}_{i_\perp}\right\|_{\mathrm{F}}\leq\sum_{i=1}^I\left\|\mathbf{Y}\times_i\boldsymbol{P}_{i_\perp}\right\|_{\mathrm{F}}\\&=\sum_{i=1}^I\left\|\boldsymbol{P}_{i_\perp}\boldsymbol{Y}_{(i)}\right\|_{\mathrm{F}}\leq\sum_{i=1}^I 2\|\boldsymbol{N}_{(i)}\|.\end{aligned} \tag{B.6}$$

The last step follows from Lemma 11, as $\boldsymbol{P}_{i_\perp}$ is the projection matrix orthogonal to $\boldsymbol{e}_i$ and the rank of $\boldsymbol{Y}_{(i)}$ is 1 from (3.5). Also, Lemma 12 with $x=\sqrt{2M_i}$ and $\boldsymbol{Z}=\sigma_{\mathsf{t}}^{-1}\boldsymbol{N}_{(i)}$ implies that with probability at least $1-2e^{-M_i}$

$$\|\sigma_{\mathsf{t}}^{-1}\boldsymbol{N}_{(i)}\|\leq\sqrt{M_i}+\sqrt{\bar{M}/M_i}+\sqrt{2M_i}\leq 3\sqrt{M_i}+\sqrt{\bar{M}/M_i}.$$

From (B.6), with probability exceeding $1-2\sum_{i=1}^{I} e^{-M_i}$,

$$\left\|\mathbf{Y}_{(I)}-\mathbf{Y}\right\|_{\mathrm{F}} \leq 2\sigma_{\mathrm{t}} \sum_{i=1}^{I}\left[3\sqrt{M_i}+\sqrt{\bar{M}/M_i}\right]. \tag{B.7}$$

Next, we bound the second term in (B.5). We note that $\|\sigma_{\mathrm{t}}^{-1}\bar{\mathbf{N}}_{(I)}\|_{\mathrm{F}}^2$ is an 1-dimensional projection of a zero mean unit variance Gaussian tensor and follows $\chi_1^2(0)$ [152, Supplement Sec. C.4]. Lemma 13 states that for any $x>0$, $\sigma_{\mathrm{t}}^{-1}\left\|\bar{\mathbf{N}}_{(I)}\right\|_{\mathrm{F}} \leq \sqrt{1+2\sqrt{x}+2x} \leq 1+2\sqrt{x}$ with probability exceeding $1-e^{-x}$. Setting $x=\max_{1\leq i\leq I} M_i$ and combining with (B.5) and (B.7) using the union bound yields (3.17) with probability exceeding $1-2\sum_{i=1}^{I} e^{-M_i}-e^{-\max_{1\leq i\leq I} M_i} \geq 1-3\sum_{i=1}^{I} e^{-M_i}$. Hence, the proof is complete.

C

Appendix of Chapter 4

C.1. Proof of Lemma 4

The proof proceeds in two parts: first, establishing the column block structure, and second, analyzing the sparsity of each block.

First, we observe that matrix $\left(\boldsymbol{I}_{\prod_{i=I}^{j+1} N_i} \otimes \left(\otimes_{i=j-1}^{1} \boldsymbol{H}_i\right)\right)^\top$ is a *block-diagonal matrix*. It has $\prod_{i=I}^{j+1} N_i$ identical diagonal blocks, each equal to $\left(\otimes_{i=j-1}^{1} \boldsymbol{H}_i\right)^\top$. To match this structure, we partition the columns of the unfolded matrix $\mathbf{X}_{(j)}$ into $\prod_{i=I}^{j+1} N_i$ column blocks. The standard column ordering in tensor unfolding places elements with higher-level indices $(n_{j+1}, \ldots, n_I)$ further apart. Thus, we can partition $\mathbf{X}_{(j)}$ into $\prod_{i=j+1}^{I} N_i$ column blocks, where each block corresponds to a unique *encapsulation* $\mathrm{n_{j+1}} = (n_{j+1}, \ldots, n_I)$ as

$$\boldsymbol{X}_{(j)} = \left[\boldsymbol{X}_{(j),(1,\ldots,1)} \quad \ldots \quad \boldsymbol{X}_{(j),(N_{j+1},\ldots,N_I)}\right].$$

Since $\left(\boldsymbol{I}_{\prod_{i=I}^{j+1} N_i} \otimes \left(\otimes_{i=j-1}^{1} \boldsymbol{H}_i\right)\right)^\top$ is block-diagonal, the multiplication with $\boldsymbol{X}_{(j)}$ decouples and operates on each of these blocks independently,

$$\begin{aligned}\boldsymbol{X}_{(j)} &\left(\boldsymbol{I}_{\prod_{i=I}^{j+1} N_i} \otimes \left(\otimes_{i=j-1}^{1} \boldsymbol{H}_i\right)\right)^\top \\ &= \left[\boldsymbol{X}_{(j),(1,\ldots,1)} \left(\otimes_{i=j-1}^{1} \boldsymbol{H}_i\right)^\top \quad \ldots \quad \boldsymbol{X}_{(j),(N_{j+1},\ldots,N_I)} \left(\otimes_{i=j-1}^{1} \boldsymbol{H}_i\right)^\top\right].\end{aligned}$$

This confirms that the resulting matrix is also composed of $\prod_{i=j+1}^{I} N_i$ column blocks, each indexed by $\mathrm{n_{j+1}}$ and given by $\boldsymbol{X}_{(j),\mathrm{n_{j+1}}} \left(\otimes_{i=j-1}^{1} \boldsymbol{H}_i\right)^\top$.

Second, we consider a column block indexed by a fixed $\mathrm{n_{j+1}}$, i.e., $\boldsymbol{X}_{(j),\mathrm{n_{j+1}}} \left(\otimes_{i=j-1}^{1} \boldsymbol{H}_i\right)^\top$. The rows of this block are indexed by $n_j \in [N_j]$. The kth row of $\boldsymbol{X}_{(j),\mathrm{n_{j+1}}} \left(\otimes_{i=j-1}^{1} \boldsymbol{H}_i\right)^\top$ will be nonzero *if and only if* the kth row of $\boldsymbol{X}_{(j),\mathrm{n_{j+1}}}$ contains nonzeros due to the full row rankness. Moreover, the kth row of $\boldsymbol{X}_{(j),\mathrm{n_{j+1}}}$ is the hierarchical block $\boldsymbol{x}_{\mathrm{n_j}}$ where the

encapsulation is $\mathrm{n_j} = (k, n_{j+1}, \ldots, n_I)$. This equivalence follows because the indices of the entries in the kth row of $\boldsymbol{X}_{(j),\mathrm{n_{j+1}}}$ align exactly with those of the hierarchical block $\boldsymbol{x}_{\mathrm{n_j}}$ with $\mathrm{n_j} = (k, n_{j+1}, \ldots, n_I)$. Hence, the kth row of $\boldsymbol{X}_{(j),\mathrm{n_{j+1}}}$ and the hierarchical block $\boldsymbol{x}_{\mathrm{n_j}}$ with $\mathrm{n_j} = (k, n_{j+1}, \ldots, n_I)$ contain identical entries with identical order. Thus, the number of non-zero rows in $\boldsymbol{X}_{(j),\mathrm{n_{j+1}}} \left(\otimes_{i=j-1}^{1} \boldsymbol{H}_i\right)^\top$ is the number of hierarchical blocks $\{\boldsymbol{x}_{\mathrm{n_j}}\}$ (within the parent block defined by $\mathrm{n_{j+1}}$) that contain at least one non-zero element, which concludes the proof.

Illustrative Example: We provide an example in Figure C.1 for the proof of Lemma 4. In Figure C.1(a), we consider the same vector $\boldsymbol{x} \in \mathbb{R}^{40}$ with $I = 3, N_3 = 5, N_2 = 2, N_1 = 4, \bar{N} = 40$ as in Figure 4.1, and mark $s = 3$ nonzero entries using colored cubes. Figure C.1(b), (c), and (d) illustrate the reordered tensor $\boldsymbol{\mathcal{X}}$, its mode unfolding $\boldsymbol{X}_{(j)}$ with $j = 2$, and how $\boldsymbol{U}_2$ is computed, respectively.

The first step of the proof corresponds to Figure C.1(c). To see why there is a block column structure, we first investigate how the unfolding matrix $\boldsymbol{X}_{(2)}$ is obtained. Since the unfolding tensor mode is $j = 2$, the row of the unfolding matrix $\boldsymbol{X}_{(2)}$ is indexed by $n_2 = 1, 2$. The column index k is determined by n_3 and n_1 jointly as $k = 1 + (n_1 - 1) + N_1(n_3 - 1)$, according to the definition of tensor unfolding [153]. When n_1 increments by one, k increases by 1; when n_3 increments by one, k increases by N_1.

To arrange the columns of $\boldsymbol{X}_{(2)}$, we fix n_3 and let n_1 runs through $1, 2, 3, 4$, and then increase n_3 by one and let n_1 runs through $1, 2, 3, 4$ again, as shown in Figure C.1(c). This indicates that n_3 indexes $\prod_{i=I}^{j+1} N_i = N_3 = 5$ column blocks, each containing $\prod_{i=1}^{j-1} N_i = N_1 = 4$ columns. Besides, the matrix $\left(\boldsymbol{I}_{\prod_{i=I}^{j+1} N_i} \otimes \left(\otimes_{i=j-1}^{1} \boldsymbol{H}_i\right)\right)^\top$ in this case reduces to the block diagonal matrix $(\boldsymbol{I}_5 \otimes \boldsymbol{H}_1)^\top$ in Figure C.1(d), matching the column block structure of $\boldsymbol{X}_{(2)}$. Therefore, $\boldsymbol{U}_2$ can be divided into $N_3 = 5$ column blocks given by the product of the column blocks of $\boldsymbol{X}_{(2)}$ and $\boldsymbol{H}_1$, where each block of $\boldsymbol{U}_2$ is also indexed by an encapsulation $\mathrm{n_3} = (n_3)$ for $n_3 \in [N_3]$.

For the second step, to understand why the number of nonzero rows in a column block indexed by $\mathrm{n_{j+1}}$ in $\boldsymbol{U}_j$ equals the number of nonzero blocks in $[\![\boldsymbol{x}_{\mathrm{n_j}}]\!]$ with $\mathrm{n_j} = (n_j, n_{j+1}, \cdots, n_I)$, we examine Figure C.1(d). Consider $\mathrm{n_3} = 4$. In $\boldsymbol{U}_2$, it corresponds to the fourth column block given by $\boldsymbol{X}_{(2),(4)} \boldsymbol{H}_1^\top$, where $\boldsymbol{X}_{(2),(4)}$ is the fourth column block of $\boldsymbol{X}_{(2)}$. Also, we have

$$[\![\boldsymbol{x}_{\mathrm{n_2}}]\!] = [\![\boldsymbol{x}_{(1,4)}]\!] = \{\boldsymbol{x}_{(1,4)}, \boldsymbol{x}_{(2,4)}\}.$$

Each element of $[\![\boldsymbol{x}_{(1,4)}]\!]$ corresponds to one row of the column block $\boldsymbol{X}_{(2),(4)}$. Only $\boldsymbol{x}_{(2,4)}$ is nonzero leading to a nonzero row in $\boldsymbol{X}_{(2),(4)} \boldsymbol{H}_1^\top$. This demonstrates that the number of nonzero rows in a column block indexed by $\mathrm{n_{j+1}}$ in $\boldsymbol{U}_j$ equals the number of nonzero blocks in $[\![\boldsymbol{x}_{\mathrm{n_j}}]\!]$ with $\mathrm{n_j} = (n_j, n_{j+1}, \cdots, n_I)$.

C.2. Illustrations of Different Sparsity Patterns

In this section, we provide examples of sparsity patterns considered in this paper in Figure C.2, using the same vector shown in Figure 4.1.

In Figure C.2(a), we present the standard sparsity with $s = 3$. Three nonzero entries $\boldsymbol{x}_{(3,1,1)}$, $\boldsymbol{x}_{(4,1,3)}$, and $\boldsymbol{x}_{(3,2,4)}$ are arbitrarily positioned. Take $\boldsymbol{x}_{(3,2,4)}$ as an example. Its encap-

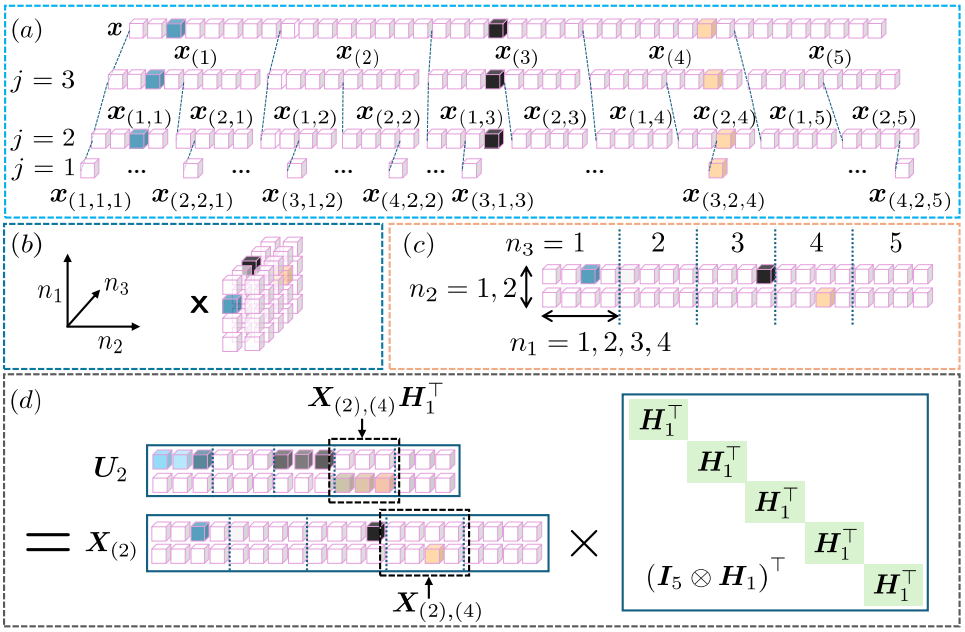


Figure C.1.: An illustrative example of the proof of Lemma 4 with $\boldsymbol{H}_1 \in \mathbb{R}^{3\times 4}$.

sulation $(3,2,4)$ means $\boldsymbol{x}_{(3,2,4)}$ is the third entry of the block indexed by encapsulation $(2,4)$, i.e., $\boldsymbol{x}_{(2,4)}$, while $\boldsymbol{x}_{(2,4)}$ means it is the second block of the block indexed by encapsulation $\boldsymbol{x}_{(4)}$. Then $\boldsymbol{x}_{(4)}$ is the fourth block of vector $\boldsymbol{x}$.

In Figure C.2(b), we show an example of $\boldsymbol{s} = (s_3, s_2, s_1) = (2,1,2)$ hierarchical sparsity. For the third level blocks, the set $[\![\boldsymbol{x}_{(2)}]\!]$ contains all blocks that share the same parent block as $\boldsymbol{x}_{(2)}$, meaning $[\![\boldsymbol{x}_{(2)}]\!] = \{\boldsymbol{x}_{(1)}, \boldsymbol{x}_{(2)}, \boldsymbol{x}_{(3)}, \boldsymbol{x}_{(4)}, \boldsymbol{x}_{(5)}\} = [\![\boldsymbol{x}_{(1)}]\!] = [\![\boldsymbol{x}_{(3)}]\!] = [\![\boldsymbol{x}_{(4)}]\!] = [\![\boldsymbol{x}_{(5)}]\!]$. Since $s_3 = 2$, according to the definition of Sparsity 2, $[\![\boldsymbol{x}_{(2)}]\!]$ contains at most $s_3 = 2$ nonzero blocks, which are $\boldsymbol{x}_{(2)}$ and $\boldsymbol{x}_{(4)}$. For the second level sparsity s_2, we take $\boldsymbol{x}_{(2)}$ and its child blocks $\boldsymbol{x}_{(1,2)}$ and $\boldsymbol{x}_{(2,2)}$ as an example. Since $s_2 = 1$, it means that $[\![\boldsymbol{x}_{(1,2)}]\!] = \{\boldsymbol{x}_{(1,2)}, \boldsymbol{x}_{(2,2)}\} = [\![\boldsymbol{x}_{(2,2)}]\!]$ contains at most s_2 nonzero block, which is $\boldsymbol{x}_{(1,2)}$. Similarly, $[\![\boldsymbol{x}_{(1,4)}]\!]$ contains at most s_2 nonzero block. For the first level sparsity $s_1 = 2$, we take $\boldsymbol{x}_{(1,2)}$ and its child blocks as an example. There should be at most $s_1 = 2$ nonzero blocks in the set of children of $\boldsymbol{x}_{(1,2)}$, which are $\boldsymbol{x}_{(2,1,2)}$ and $\boldsymbol{x}_{(3,1,2)}$ in $[\![\boldsymbol{x}_{(3,1,2)}]\!]$. Since this is the first level, a block corresponds to an individual element of $\boldsymbol{x}$.

Figure C.2(c) illustrates the $\boldsymbol{s} = (s_3, s_2, s_1) = (2,1,2)$ Kronecker-supported sparsity with $\boldsymbol{b}_3 = [0,1,0,1,0]$, $\boldsymbol{b}_2 = [1,0]$, and $\boldsymbol{b}_1 = [0,1,1,0]$. Its support is then $\boldsymbol{b}_3 \otimes \boldsymbol{b}_2 \otimes \boldsymbol{b}_1$.

C.3. Proof of Lemma 5

Let $\boldsymbol{x}$ be an s sparse vector. We denote k_j as the total number of nonzero blocks within all jth-level blocks of $\boldsymbol{x}$. Clearly, $k_1 = s$ and $k_{J+1} = 1$. Then, each nonzero block in the $j+1$th level can have at most $k_j - (k_{j+1} - 1)$ number of nonzero jth level blocks. This occurs in the most unbalanced case, where $k_{j+1} - 1$ blocks have only one nonzero jth

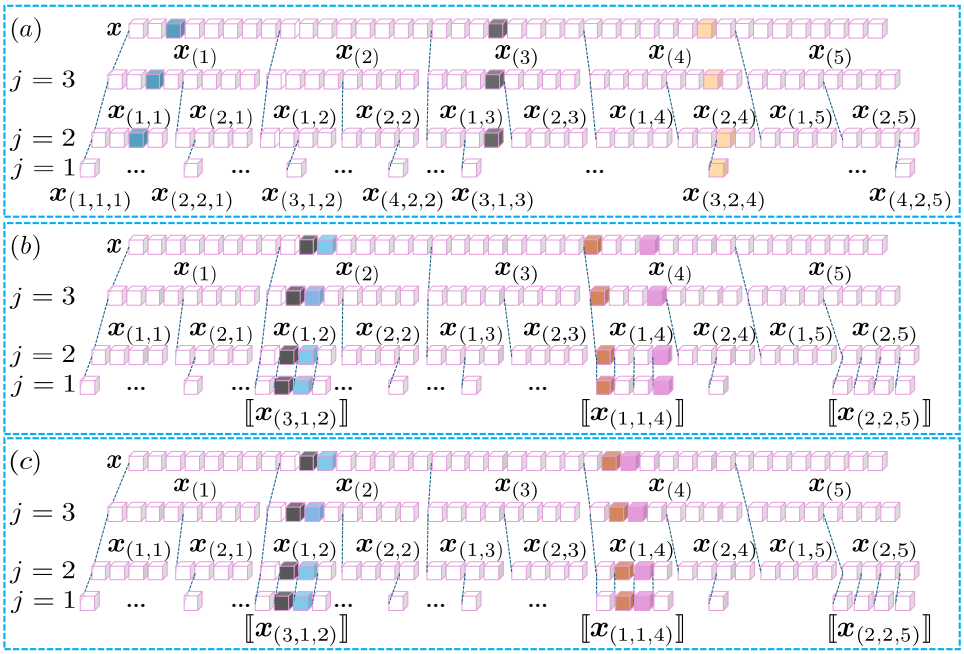


Figure C.2.: Examples of different sparsity patterns. (a) Standard sparsity. (b) $(2,1,2)$ hierarchical sparsity. (c) $(2,1,2)$ Kronecker-supported sparsity.

level block while the remaining block has $k_j - (k_{j+1} - 1)$ nonzero jth level blocks. This observation leads to the upper bound for the sparsity level, $s_j \leq k_j - (k_{j+1} - 1)$, which yields $\sum_{i=1}^{I} s_i \leq s + (I-1)$. So, any s sparse vector $\boldsymbol{x}$ belongs to $\cup_{\boldsymbol{s} \in f_{\mathrm{N}}(s)} \mathscr{S}_{\boldsymbol{s}}$.

C.4. Proof of Theorem 7

For any $\boldsymbol{x}$, we note that Definition 1 bounds $\|\boldsymbol{H}\boldsymbol{x}\|_2^2$. Following the hierarchical view, we note

$$\|\boldsymbol{H}\boldsymbol{x}\|_2^2 = \|\boldsymbol{\mathsf{X}} \times_1 \boldsymbol{H}_1 \cdots \times_I \boldsymbol{H}_I\|_{\mathrm{F}}^2 = \|\boldsymbol{H}_I \boldsymbol{X}_{(I)} \left(\otimes_{i=I-1}^{1} \boldsymbol{H}_i\right)^\top\|_{\mathrm{F}}^2.$$

Using the RIC of $\boldsymbol{H}_I$, we have

$$(1-\delta_{s_I})\|\boldsymbol{X}_{(I)} \left(\otimes_{i=I-1}^{1} \boldsymbol{H}_i\right)^\top\|_{\mathrm{F}}^2 \leq \|\boldsymbol{H}\boldsymbol{x}\|_2^2 \leq (1+\delta_{s_I})\|\boldsymbol{X}_{(I)} \left(\otimes_{i=I-1}^{1} \boldsymbol{H}_i\right)^\top\|_{\mathrm{F}}^2.$$

We also note that $\|\boldsymbol{X}_{(I)} \left(\otimes_{i=I-1}^{1} \boldsymbol{H}_i\right)^\top\|_{\mathrm{F}}^2 = \|\boldsymbol{H}_{I-1}\boldsymbol{X}_{(I-1)} \left(\boldsymbol{I}_{N_I} \otimes \left(\otimes_{i=I-2}^{1} \boldsymbol{H}_i\right)\right)^\top\|_{\mathrm{F}}^2$ due to the tensor folding and unfolding. Therefore, using RIC of $\boldsymbol{H}_{I-1}$, we arrive at

$$\begin{aligned}(1-\delta_{s_I})(1-\delta_{s_{I-1}}))\|\boldsymbol{X}_{(I-1)} \left(\boldsymbol{I}_{N_I} \otimes \left(\otimes_{i=I-2}^{1} \boldsymbol{H}_i\right)\right)^\top\|_{\mathrm{F}}^2 \\ \leq \|\boldsymbol{H}\boldsymbol{x}\|_2^2 \leq (1+\delta_{s_I})(1+\delta_{s_{I-1}})\|\boldsymbol{X}_{(I)} \left(\otimes_{i=I-1}^{1} \boldsymbol{H}_i\right)^\top\|_{\mathrm{F}}^2\end{aligned}$$

Repeating these steps recursively following the analysis in hierarchical view, we obtain

$$\prod_{i=1}^{I}(1-\delta_{s_i})\|\boldsymbol{X}_{(1)}\left(\otimes_{i=I-1}^{1}\boldsymbol{I}_{N_i}\right)^{\top}\|_{\mathrm{F}}^2 \leq \|\boldsymbol{H}\boldsymbol{x}\|_2^2 \leq \prod_{i=1}^{I}(1+\delta_{s_i})\|\boldsymbol{X}_{(1)}\left(\otimes_{i=I-1}^{1}\boldsymbol{I}_{N_i}\right)^{\top}\|_{\mathrm{F}}^2.$$

Since $\boldsymbol{X}_{(1)}\left(\otimes_{i=I-1}^{1}\boldsymbol{I}_{N_i}\right)^{\top} = \boldsymbol{X}_{(1)}$ and $\|\boldsymbol{X}_{(1)}\|_{\mathrm{F}}^2 = \|\boldsymbol{x}\|_2^2$,

$$\prod_{i=1}^{I}(1-\delta_{s_i})\|\boldsymbol{x}\|_2^2 \leq \|\boldsymbol{H}\boldsymbol{x}\|_2^2 \leq \prod_{i=1}^{I}(1+\delta_{s_i})\|\boldsymbol{x}\|_2^2.$$

Hence, we derive

$$\delta_{(\boldsymbol{s},\mathrm{N})}(\boldsymbol{H}) \leq \max\{1-\prod_{i=1}^{I}(1-\delta_{s_i})), \prod_{i=1}^{I}(1+\delta_{s_i})-1\} = \prod_{i=1}^{I}(1+\delta_{s_i})-1,$$

which completes the proof.

Remark: We note that the high-level proof strategy of our Theorem 7 and [57, Theorem 4] is similar in that both aim to sequentially unwrap the effect of the Kronecker product. The key difference is that we employ tensor representations and operations such as tensor unfolding, enabling a straightforward, flip-operator-free proof. This formulation clearly demonstrates how the sparse signal $\boldsymbol{x}$ (or its tensor form $\boldsymbol{\mathcal{X}}$ and its unfolding $\boldsymbol{X}_{(j)}$) is measured by factor matrix $\boldsymbol{H}_j$ through a linear transformation $\boldsymbol{X}_{(j)}\left(\boldsymbol{I}_{\prod_{i=I}^{j+1}N_i}\otimes\left(\otimes_{i=j-1}^{1}\boldsymbol{H}_i\right)\right)^{\top}$. The row sparsity of $\boldsymbol{X}_{(j)}\left(\boldsymbol{I}_{\prod_{i=I}^{j+1}N_i}\otimes\left(\otimes_{i=j-1}^{1}\boldsymbol{H}_i\right)\right)^{\top}$ is dictated by the sparsity of our hierarchical block partition as in Section 4.4. The aspect of this multi-stage measurement framework is missing in [57]. Thus, the result in [57, Theorem 4] focuses solely on hierarchical sparsity while our multi-stage framework provides a general perspective that defines generalized sparsity, where standard, hierarchical, and Kronecker-supported sparsity are special cases for analysis and recovery. This proof also explains why standard RIP cannot be improved beyond hierarchical sparsity, clarifies the maximum achievable sparsity level, and shows why the corresponding bounds are fundamentally tight. It further provides insight into why proofs for Kronecker-supported sparsity can be strengthened, drawing analogies to standard RIP and MMV analyses.

C.5. The Number of Measurements

In this section, we present the measurement bounds for unstructured $\boldsymbol{H}$ with different sparsity patterns. Let $\boldsymbol{H} \in \mathbb{R}^{\bar{M}\times\bar{N}}$ has independent and identically distributed standard Gaussian. For

$$\bar{M} = \mathcal{O}\left(s\ln(\frac{e\bar{N}}{s})\right)$$

where c is a positive constant, s sparse vectors can be recovered from the measurement of $\boldsymbol{H}$ with high probability [119]. Also, if

$$\bar{M} = \mathcal{O}\left(\sum_{i=1}^{I}\prod_{j=i}^{I} s_j \ln(\frac{eN_i}{s_i}) + \prod_{i=1}^{I} s_i\right),$$

$\boldsymbol{s}$ hierarchical sparse vectors can be recovered from the measurement of $\boldsymbol{H}$ with high probability [57]. These two results lead to the measurement bounds in Section 4.5.

C.6. Complexity Comparison

We comprehensively analyze the complexity of our MSR algorithm to demonstrate the benefit of exploiting the Kronecker structure of $\boldsymbol{H}$ via the hierarchical view. We consider MSR combined with MMV-SBL [120], SIHT [121], SHTP [121], and SOMP [122] as sparse recovery algorithms, referred to as MSSBL, MSIHT, MSHTP, and MSOMP, respectively. We also use `Seq` and `Pl` to represent the *sequential* and *parallel* implementation of (4.7). Assume M_i's are $\mathcal{O}(M)$, N_i's are $\mathcal{O}(N)$ for $i \in [I]$, and $I < M < N$. We compare the time and space complexity of our algorithms with other state-of-the-art algorithms. For the recovery of s sparse vectors, we include SBL [62], OMP, and KroOMP [18] as benchmarks. For the recovery of $\boldsymbol{s}$ hierarchically sparse vectors, HiHTP [57], IHT, and HTP are used as benchmarks. We note that only the exact implementation of HiHTP for $I = 2$ is given in [57]. Regarding recovering $\boldsymbol{s}$ Kronecker-supported sparse vectors, we consider AM- and SVD-KroSBL [23] for benchmarking.

For the recovery of s standard sparse vectors, our MSSBL and MSOMP substantially reduce both the time and space complexity compared to their traditional counterparts. In terms of time complexity, our MSSBL ($\mathcal{O}(M^2N^I)$ for `Seq` and $\mathcal{O}(M^IN)$ for `Pl`) is superior than SBL ($\mathcal{O}(M^{2I}N^I)$), while the time complexity of MSOMP ($\mathcal{O}(MN^I)$ for `Seq` and $\mathcal{O}(M^IN)$ for `Pl`) is also lower than OMP with $\mathcal{O}(M^IN^I)$. Moreover, both MSSBL and MSOMP avoid $\mathcal{O}(MN)^I$ in space complexity and have $\mathcal{O}(M^{I-1}N)$ for `Seq` and $\mathcal{O}(MN^I)$ for `Pl`. Compared to KroOMP with time complexity $\mathcal{O}(MN^I)$ and space complexity $\mathcal{O}(N^I)$, MSOMP-`Seq` achieves the same time complexity but with a much lower space complexity $\mathcal{O}(M^{I-1}N)$. Alternatively, we can achieve a much lower time complexity $\mathcal{O}(M^IN)$ by parallel implementation, at the cost of a slightly higher space complexity of $\mathcal{O}(MN^I)$.

The computational gains are particularly significant regarding structured sparsity. For both hierarchically sparse and Kronecker-supported sparse vectors, classical methods like IHT and HTP exhibit a time and space complexity of $\mathcal{O}(M^IN^I)$. Our MSIHT-`Seq`, MSHTP-`Seq`, and MSSBL-`Seq` have time complexity $\mathcal{O}(MN^I)$, $\mathcal{O}(MN^I)$, and $\mathcal{O}(M^2N^I)$, respectively, and $\mathcal{O}(M^{I-1}N)$ for space complexity. Compared to HiHTP, our MSSBL-`Seq` has the same time complexity $\mathcal{O}(M^2N^2)$ while MSSBL-`Pl` has a lower space complexity ($\mathcal{O}(MN^2)$ compared to $\mathcal{O}(M^2N^2)$ of HiHTP.

Similarly, for Kronecker-supported sparse recovery, when compared to AM-KroSBL and SVD-KroSBL, MSSBL algorithm demonstrates lower time complexity from $\mathcal{O}(M^IN^I)$ to $\mathcal{O}(MN^I)$ and space complexity from $\mathcal{O}(M^IN^I)$ to $\mathcal{O}(N^I)$. MSIHT and MSHTP exhibit the same or even lower time and space complexity than MSSBL, hence lower than AM-KroSBL and SVD-KroSBL, demonstrating the superiority of our multi-stage framework. We list all

the time and space complexity of the algorithms in Table C.1. We use R_{EM}, R_{OMP}, R_{HTP}, R_{IHT}, and R_{AM} to denote the number of EM, OMP, HTP, IHT, and AM iterations. All these values can vary for different algorithms and experiment settings.

Table C.1.: Complexity of different algorithms in different sparse recovery problems.

Method	Time Complexity	Space Complexity
Recovery of s sparse vectors		
MSSBL-Seq	$\mathcal{O}\left(R_{\text{EM}}(M^2N^I + MN^I)\right)$	$\mathcal{O}(M^{I-1}N)$
MSSBL-Pl	$\mathcal{O}\left(R_{\text{EM}}(IM^2N + M^IN)\right)$	$\mathcal{O}(MN^I)$
MSOMP-Seq	$\mathcal{O}\left(R_{\text{OMP}}MN^I + R_{\text{OMP}}^3N^{I-1} + R_{\text{OMP}}^2MN^{I-1}\right)$	$\mathcal{O}(M^{I-1}N)$
MSOMP-Pl	$\mathcal{O}\left(R_{\text{OMP}}M^IN + R_{\text{OMP}}^2M^I + R_{\text{OMP}}^3M^{I-1}\right)$	$\mathcal{O}(MN^I)$
KroOMP	$\mathcal{O}\left(R_{\text{OMP}}MN^I + R_{\text{OMP}}^2M^I + R_{\text{OMP}}^2MN + R_{\text{OMP}}^3\right)$	$\mathcal{O}(N^I)$
SBL	$\mathcal{O}\left(R_{\text{EM}}M^{2I}N^I\right)$	$\mathcal{O}((MN)^I)$
OMP	$\mathcal{O}\left(R_{\text{OMP}}(MN)^I + R_{\text{OMP}}^3 + R_{\text{OMP}}^2M^I\right)$	$\mathcal{O}((MN)^I)$
Recovery of $\boldsymbol{s}$ hierarchically sparse vectors		
MSSBL-Seq	$\mathcal{O}\left(R_{\text{EM}}(M^2N^I + MN^I)\right)$	$\mathcal{O}(M^{I-1}N)$
MSSBL-Pl	$\mathcal{O}\left(R_{\text{EM}}(IM^2N + M^IN)\right)$	$\mathcal{O}(MN^I)$
MSHTP-Seq	$\mathcal{O}\left(R_{\text{HTP}}(MN^I + \max_i s_i^2MN^{I-1})\right)$	$\mathcal{O}(M^{I-1}N)$
MSIHT-Seq	$\mathcal{O}\left(R_{\text{IHT}}MN^I\right)$	$\mathcal{O}(M^{I-1}N)$
HiHTP [57] ($I=2$)	$\mathcal{O}\left(R_{\text{HTP}}((s_1s_2)^2M^2 + (MN)^2)\right)$	$\mathcal{O}((MN)^2)$
IHT	$\mathcal{O}\left(R_{\text{IHT}}(MN)^I\right)$	$\mathcal{O}((MN)^I)$
HTP	$\mathcal{O}\left(R_{\text{HTP}}((MN)^I + (\prod_{i=1}^I s_i)^2M^I)\right)$	$\mathcal{O}((MN)^I)$
Recovery of $\boldsymbol{s}$ Kronecker-supported sparse vectors		
MSSBL	$\mathcal{O}\left(R_{\text{EM}}(IM^2N + MN^I)\right)$	$\mathcal{O}(N^I)$
MSIHT	$\mathcal{O}\left(R_{\text{IHT}}MN^I\right)$	$\mathcal{O}(N^I)$
MSHTP	$\mathcal{O}\left(R_{\text{HTP}}MN^I + R_{\text{HTP}}M\sum_{i=1}^I s_i^2)\right)$	$\mathcal{O}(N^I)$
MSOMP	$\mathcal{O}\left(R_{\text{OMP}}^3N^{I-1} + R_{\text{OMP}}^2MN^{I-1} + R_{\text{OMP}}MN^I\right)$	$\mathcal{O}(N^I)$
AM-KroSBL [23]	$\mathcal{O}\left(R_{\text{EM}}(R_{\text{AM}}IN^I + (MN)^I)\right)$	$\mathcal{O}((MN)^I)$
SVD-KroSBL [23]	$\mathcal{O}\left(R_{\text{EM}}(N^{I+1} + (MN)^I)\right)$	$\mathcal{O}((MN)^I)$
IHT	$\mathcal{O}\left(R_{\text{IHT}}(MN)^I\right)$	$\mathcal{O}((MN)^I)$
HTP	$\mathcal{O}\left(R_{\text{HTP}}((MN)^I + (\prod_{i=1}^I s_i)^2M^I)\right)$	$\mathcal{O}((MN)^I)$

Table C.1 compares the complexity of different versions of MSR to different traditional compressed sensing algorithms. The conclusion can be extended to a more general case. Consider $N > M > 1$ and $I > 1$ with $M, N, I \in \mathbb{Z}$. In a prototypical compressed sensing problem $\boldsymbol{y}_{\text{p}} = \boldsymbol{H}_{\text{p}}\boldsymbol{x}_{\text{p}}$ with $\boldsymbol{H}_{\text{p}} \in \mathbb{R}^{M_{\text{p}} \times N_{\text{p}}}$ being a dense, unstructured measurement matrix, consider a general compressed sensing algorithm that has complexity $\mathcal{O}(M_{\text{p}}^a N_{\text{p}}^b)$ to recover $\boldsymbol{x}_{\text{p}}$ with $a, b \geq 1$. We note that $a, b \geq 1$ is a fair consideration since computing $\boldsymbol{H}_{\text{p}}\boldsymbol{x}_{\text{p}}$ already requires $\mathcal{O}(M_{\text{p}}N_{\text{p}})$. Then applying this algorithm to (4.1) induces a time complexity of $\mathcal{O}(M^{aI}N^{bI})$. If we combine the same compressed sensing algorithm with our MSR, *regardless of the special structure or the MMV property of* $\boldsymbol{U}_j$, step 4 in Algorithm 8 is simply solving $N^{I-j}M^{j-1}$ compressed sensing subproblems where each has M measurements and N unknowns, inducing a per level complexity of $\mathcal{O}(M^aN^bN^{I-j}M^{j-1})$. Considering all steps from $j = I$ to 1, the total complexity is given by

$$\sum_{j=1}^{I} M^aN^bN^{I-j}M^{j-1} = M^{a-1}N^{b+I}\frac{M(N^I - M^I)}{N^I(N-M)} = M^aN^b\frac{N^I - M^I}{N-M}.$$

To compare to $M^{aI}N^{bI}$, we consider the ratio

$$\frac{M^a N^b (N^I - M^I)}{M^{aI} N^{bI} (N-M)} = \frac{N^I - M^I}{M^{a(I-1)} N^{b(I-1)} (N-M)} \overset{(i)}{\leq} \frac{N^I - M^I}{M^{I-1} N^{I-1} (N-M)} \overset{(ii)}{<} \frac{I}{M^{I-1}} \overset{(iii)}{\leq} 1,$$

where (i) holds since the ratio is a decreasing function of a and b, (ii) is due to

$$N^I - M^I = (N-M) \sum_{i=1}^{I} N^{I-i} M^{i-1} < I N^{I-1} (N-M),$$

and (iii) holds since $M^{I-1} \geq 2^{I-1} \geq I$ for $\forall I, M > 1, I, M \in \mathbb{Z}$, we conclude that $\frac{I}{M^{I-1}} \leq 1$, and thus $M^a N^b \frac{N^I - M^I}{N-M} < M^{aI} N^{bI}$ for any $N > M > 1$, $I > 1$, and $a, b \geq 1$ with $M, N, I \in \mathbb{Z}$. Hence, in general, our MSR has lower computational complexity than traditional compressed sensing algorithms when both are applied to (4.1). We note that the argument also holds for compressed sensing algorithms with complexity $\mathcal{O}(N^b)$ with $b \geq 2$ or with $\mathcal{O}(N \log N)$. When $I = 1$, (4.1) reduces to the traditional compressed sensing problem and MSR has identical complexity to that of a traditional compressed sensing algorithm.

C.7. Proof of Theorem 8

Before the proof of Theorem 8, we introduce four aiding lemmas.

Lemma 15. *[119, Lemma 6.16] Given a vector $\boldsymbol{v} \in \mathbb{R}^N$ and an index set $\mathcal{T} \subset [N]$, there is*

$$\|((\boldsymbol{I}_N - \boldsymbol{H}^\top \boldsymbol{H})\boldsymbol{v})_{\mathcal{T}}\|_2 \leq \delta_t \|\boldsymbol{v}\|_2,$$

if the cardinality of the union of $\mathcal{T}$ and the support set of $\boldsymbol{v}$ is not exceeding t.

Lemma 16. *[119, Lemma 6.20] Given vector $\boldsymbol{n} \in \mathbb{R}^N$ and set $\mathcal{T} \subset [N]$ with cardinality not exceeding s, then*

$$\|(\boldsymbol{H}^\top \boldsymbol{n})_{\mathcal{T}}\|_2 \leq \sqrt{1+\delta_s} \|\boldsymbol{n}\|_2.$$

Lemma 17. *For sparse matrix $\boldsymbol{X}$ with row support $\mathcal{T}$ with* $\mathrm{card}(\mathcal{T}) \leq s$, *and $\boldsymbol{N} \in \mathbb{R}^{M \times N}$, the sequence $\{\boldsymbol{X}^k\}$ defined by SIHT or SHTP for solving an MMV problem $\boldsymbol{Y} = \boldsymbol{H}\boldsymbol{X} + \boldsymbol{N}$ with $\boldsymbol{X}^0 = \boldsymbol{0}$, satisfies for any $k \geq 0$,*

$$\|\boldsymbol{X}^k - \boldsymbol{X}\|_{\mathrm{F}} \leq \alpha^k \|\boldsymbol{X}\|_{\mathrm{F}} + \tau \|\boldsymbol{N}\|_{\mathrm{F}},$$

where

$$\text{for SIHT: } \alpha = \sqrt{3}\delta_{3s}, \tau = \sqrt{3(1+\delta_{2s})} \frac{1-\alpha^k}{1-\alpha}, \text{ and}$$

$$\text{for SHTP: } \alpha = \sqrt{\frac{2\delta_{3s}^2}{1-\delta_{2s}^2}}, \tau = \frac{(\sqrt{2(1-\delta_{2s})} + \sqrt{1+\delta_s})(1-\alpha^k)}{(1-\delta_{2s})(1-\alpha)}.$$

Proof. The proof closely follows the technique in [119, Theorem 6.18] and [121]. Here, we extend the SMV case in [119, Theorem 6.18] to the MMV case. In the MMV case, the

thresholding operator retains the rows of $\boldsymbol{X}^k + \boldsymbol{H}^\top\left(\boldsymbol{Y} - \boldsymbol{H}\boldsymbol{X}^k\right)$ with the s largest row ℓ_2 norms, and then we have

$$\|\left(\boldsymbol{X}^k + \boldsymbol{H}^\top\left(\boldsymbol{Y} - \boldsymbol{H}\boldsymbol{X}^k\right)\right)_{\mathcal{T}}\|_{\mathrm{F}}^2 \leq \|\left(\boldsymbol{X}^k + \boldsymbol{H}^\top\left(\boldsymbol{Y} - \boldsymbol{H}\boldsymbol{X}^k\right)\right)_{\mathcal{T}^{k+1}}\|_{\mathrm{F}}^2.$$

Removing the common rows from both sides, we arrive at

$$\|\left(\boldsymbol{X}^k + \boldsymbol{H}^\top\left(\boldsymbol{Y} - \boldsymbol{H}\boldsymbol{X}^k\right)\right)_{\mathcal{T}\setminus\mathcal{T}^{k+1}}\|_{\mathrm{F}}^2 \leq \|\left(\boldsymbol{X}^k + \boldsymbol{H}^\top\left(\boldsymbol{Y} - \boldsymbol{H}\boldsymbol{X}^k\right)\right)_{\mathcal{T}^{k+1}\setminus\mathcal{T}}\|_{\mathrm{F}}^2.$$

IHT proceeds with $\boldsymbol{X}^{k+1} = \left(\boldsymbol{X}^k + \boldsymbol{H}^\top\left(\boldsymbol{Y} - \boldsymbol{H}\boldsymbol{X}^k\right)\right)_{\mathcal{T}^{k+1}}$. Since we have $(\boldsymbol{X}^{k+1})_{\mathcal{T}\setminus\mathcal{T}^{k+1}} = \boldsymbol{0}$ and $(\boldsymbol{X})_{\mathcal{T}^{k+1}\setminus\mathcal{T}} = \boldsymbol{0}$, we get

$$\begin{aligned}&\|\left(\boldsymbol{X} - \boldsymbol{X}^{k+1} + \boldsymbol{X}^k - \boldsymbol{X} + \boldsymbol{H}^\top\left(\boldsymbol{Y} - \boldsymbol{H}\boldsymbol{X}^k\right)\right)_{\mathcal{T}\setminus\mathcal{T}^{k+1}}\|_{\mathrm{F}}\\ \leq&\|\left(\boldsymbol{X}^k - \boldsymbol{X} + \boldsymbol{H}^\top\left(\boldsymbol{Y} - \boldsymbol{H}\boldsymbol{X}^k\right)\right)_{\mathcal{T}^{k+1}\setminus\mathcal{T}}\|_{\mathrm{F}}.\end{aligned}$$

Applying reverse triangle inequality to the left-hand side and rearranging, we arrive at

$$\begin{aligned}\|\left(\boldsymbol{X} - \boldsymbol{X}^{k+1}\right)_{\mathcal{T}\setminus\mathcal{T}^{k+1}}\|_{\mathrm{F}} &\leq \|\left(\boldsymbol{X}^k - \boldsymbol{X} + \boldsymbol{H}^\top\left(\boldsymbol{Y} - \boldsymbol{H}\boldsymbol{X}^k\right)\right)_{\mathcal{T}^{k+1}\setminus\mathcal{T}}\|_{\mathrm{F}}\\ &\quad + \|\left(\boldsymbol{X}^k - \boldsymbol{X} + \boldsymbol{H}^\top\left(\boldsymbol{Y} - \boldsymbol{H}\boldsymbol{X}^k\right)\right)_{\mathcal{T}\setminus\mathcal{T}^{k+1}}\|_{\mathrm{F}}\\ &\leq \sqrt{2}\|\left(\boldsymbol{X}^k - \boldsymbol{X} + \boldsymbol{H}^\top\left(\boldsymbol{Y} - \boldsymbol{H}\boldsymbol{X}^k\right)\right)_{\mathcal{T}\Delta\mathcal{T}^{k+1}}\|_{\mathrm{F}},\end{aligned}$$

where $\mathcal{T}\Delta\mathcal{T}^{k+1} = (\mathcal{T}\setminus\mathcal{T}^{k+1})\cup(\mathcal{T}^{k+1}\setminus\mathcal{T})$ denoting the symmetric difference of the sets $\mathcal{T}$ and $\mathcal{T}^{k+1}$. Therefore, we obtain the error in the kth iteration as

$$\begin{aligned}\|\boldsymbol{X}^{k+1} - \boldsymbol{X}\|_{\mathrm{F}}^2 &= \|\left(\boldsymbol{X}^{k+1} - \boldsymbol{X}\right)_{\mathcal{T}^{k+1}}\|_{\mathrm{F}}^2 + \|\left(\boldsymbol{X}^{k+1} - \boldsymbol{X}\right)_{\mathcal{T}\setminus\mathcal{T}^{k+1}}\|_{\mathrm{F}}^2\\ &= \|\left(\boldsymbol{X}^k + \boldsymbol{H}^\top\left(\boldsymbol{Y} - \boldsymbol{H}\boldsymbol{X}^k\right) - \boldsymbol{X}\right)_{\mathcal{T}^{k+1}}\|_{\mathrm{F}}^2 + \|\left(\boldsymbol{X}^{k+1} - \boldsymbol{X}\right)_{\mathcal{T}\setminus\mathcal{T}^{k+1}}\|_{\mathrm{F}}^2\\ &\leq \|\left(\boldsymbol{X}^k + \boldsymbol{H}^\top\left(\boldsymbol{Y} - \boldsymbol{H}\boldsymbol{X}^k\right) - \boldsymbol{X}\right)_{\mathcal{T}^{k+1}}\|_{\mathrm{F}}^2\\ &\quad + 2\|\left(\boldsymbol{X}^k - \boldsymbol{X} + \boldsymbol{H}^\top\left(\boldsymbol{Y} - \boldsymbol{H}\boldsymbol{X}^k\right)\right)_{\mathcal{T}\Delta\mathcal{T}^{k+1}}\|_{\mathrm{F}}^2\\ &\leq 3\|\left(\boldsymbol{X}^k - \boldsymbol{X} + \boldsymbol{H}^\top\left(\boldsymbol{Y} - \boldsymbol{H}\boldsymbol{X}^k\right)\right)_{\mathcal{T}\cup\mathcal{T}^{k+1}}\|_{\mathrm{F}}^2.\end{aligned}$$

Considering $\boldsymbol{Y} = \boldsymbol{H}\boldsymbol{X} + \boldsymbol{N}$, we then have

$$\begin{aligned}\|\boldsymbol{X}^{k+1} - \boldsymbol{X}\|_{\mathrm{F}} &\leq \sqrt{3}\|\left(\boldsymbol{X}^k - \boldsymbol{X} + \boldsymbol{H}^\top\left(\boldsymbol{Y} - \boldsymbol{H}\boldsymbol{X}^k\right)\right)_{\mathcal{T}\cup\mathcal{T}^{k+1}}\|_{\mathrm{F}}\\ &= \sqrt{3}\|\left(\left(\boldsymbol{I} - \boldsymbol{H}^\top\boldsymbol{H}\right)\left(\boldsymbol{X}^k - \boldsymbol{X}\right) + \boldsymbol{H}^\top\boldsymbol{N}\right)_{\mathcal{T}\cup\mathcal{T}^{k+1}}\|_{\mathrm{F}}\\ &\leq \sqrt{3}\|\left(\left(\boldsymbol{I} - \boldsymbol{H}^\top\boldsymbol{H}\right)\left(\boldsymbol{X}^k - \boldsymbol{X}\right)\right)_{\mathcal{T}\cup\mathcal{T}^{k+1}}\|_{\mathrm{F}} + \sqrt{3}\|\left(\boldsymbol{H}^\top\boldsymbol{N}\right)_{\mathcal{T}\cup\mathcal{T}^{k+1}}\|_{\mathrm{F}}\\ &\leq \sqrt{3}\delta_{3s}\|\boldsymbol{X}^k - \boldsymbol{X}\|_{\mathrm{F}} + \sqrt{3(1+\delta_{2s})}\|\boldsymbol{N}\|_{\mathrm{F}},\end{aligned}$$

where the last step is the direct consequence of Lemma 15 and Lemma 16. To see this, we

note

$$\begin{aligned}\|\left(\left(\boldsymbol{I}-\boldsymbol{H}^{\top}\boldsymbol{H}\right)\left(\boldsymbol{X}^{k}-\boldsymbol{X}\right)\right)_{\mathcal{T}\cup\mathcal{T}^{k+1}}\|_{\mathrm{F}}^{2}&=\sum_{n}\|\left(\left(\boldsymbol{I}-\boldsymbol{H}^{\top}\boldsymbol{H}\right)[\boldsymbol{X}^{k}-\boldsymbol{X}]_{n}\right)_{\mathcal{T}\cup\mathcal{T}^{k+1}}\|_{2}^{2}\\&\leq\delta_{3s}^{2}\sum_{n}\|[\boldsymbol{X}^{k}-\boldsymbol{X}]_{n}\|_{2}^{2}=\delta_{3s}^{2}\|\boldsymbol{X}^{k}-\boldsymbol{X}\|_{\mathrm{F}}^{2},\end{aligned}$$

where $[\boldsymbol{X}^{k}-\boldsymbol{X}]_{n}$ is the nth column of matrix $\boldsymbol{X}^{k}-\boldsymbol{X}$. We can derive similar argument for $\sqrt{1+\delta_{2s}}\|\boldsymbol{N}\|_{\mathrm{F}}$. Conclusion for HTP has been given in [121, Theorem 3]. This concludes the proof.

□

Lemma 18. *For the sparse recovery problem in the stage of unfolding jth ($j\leq I-1$) mode of tensor*

$$\mathsf{T}=\mathsf{X}\times_{1}\boldsymbol{H}_{1}\times_{2}\boldsymbol{H}_{2}\cdots\times_{j}\boldsymbol{H}_{j}\times_{j+1}\boldsymbol{I}_{N_{j+1}}\cdots\times_{I}\boldsymbol{I}_{N_{I}}+\mathsf{N},$$

where the sparse tensor X corresponds to s standard sparse $\boldsymbol{x}$ or $\boldsymbol{s}$ hierarchically sparse $\boldsymbol{x}$. Its jth mode unfolding leads to

$$\boldsymbol{T}_{(j)}=\boldsymbol{H}_{j}\boldsymbol{U}_{j}+\boldsymbol{N}_{(j)}=\boldsymbol{H}_{j}\boldsymbol{X}_{(j)}\left(\boldsymbol{I}_{\prod_{i=I}^{j+1}N_{i}}\otimes\left(\otimes_{i=j-1}^{1}\boldsymbol{H}_{i}\right)\right)^{\top}+\boldsymbol{N}_{(j)}.\tag{C.1}$$

Then the estimate of $\boldsymbol{U}_{j}$, denoted as $\tilde{\boldsymbol{U}}_{j}$ and obtained through IHT or HTP, satisfies

$$\|[\tilde{\boldsymbol{U}}_{j}]_{\mathrm{n}_{\mathrm{j+1}}}-[\boldsymbol{U}_{j}]_{\mathrm{n}_{\mathrm{j+1}}}\|_{\mathrm{F}}\leq\alpha_{j}^{k}\|[\boldsymbol{U}_{j}]_{\mathrm{n}_{\mathrm{j+1}}}\|_{\mathrm{F}}+\tau_{j}\|[\tilde{\boldsymbol{U}}_{j+1}-\boldsymbol{U}_{j+1}]_{\mathrm{n}_{\mathrm{j+2}}}\|_{\mathrm{F}},\tag{C.2}$$

where $[\tilde{\boldsymbol{U}}_{j}]_{\mathrm{n}_{\mathrm{j+1}}}$ and $[\boldsymbol{U}_{j}]_{\mathrm{n}_{\mathrm{j+1}}}:=[\boldsymbol{X}_{(j)}]_{\mathrm{n}_{\mathrm{j+1}}}\left(\otimes_{i=j-1}^{1}\boldsymbol{H}_{i}\right)^{\top}$ denote the $\mathrm{n}_{\mathrm{j+1}}$th column block of $\tilde{\boldsymbol{U}}_{j}$ and $\boldsymbol{U}_{j}$, respectively. Here, encapsulation $\mathrm{n}_{\mathrm{j+1}}:=(n_{j+1},\cdots,n_{I-1},n_{I})$ is the index for the column block. The indices $\{n_{i}\}_{i=j+2}^{I}$ in encapsulation $\mathrm{n}_{\mathrm{j+2}}:=(n_{j+2},\cdots,n_{I-1},n_{I})$ have the same value as the indices $\{n_{i}\}_{i=j+2}^{I}$ in encapsulation $\mathrm{n}_{\mathrm{j+1}}$, i.e., the block indexed by $\mathrm{n}_{\mathrm{j+2}}$ should be a parent block of the block indexed by $\mathrm{n}_{\mathrm{j+1}}$. Constants μ_{j} and τ_{j} depend on the iteration number k and matrix $\boldsymbol{H}_{j}$.

Proof. According to Lemma 4, we solve (C.1) by separating it into $\prod_{i=I}^{j+1}N_{i}$ MMV problem, where each MMV problem is indexed by an encapsulation $\mathrm{n}_{\mathrm{j+1}}$. Suppose we consider a fixed encapsulation $\mathrm{n}_{\mathrm{j+1}}^{*}=(n_{j+1}^{*},\cdots,n_{I}^{*})$, and consider the MMV problem indexed by $\mathrm{n}_{\mathrm{j+1}}^{*}$ as

$$[\boldsymbol{T}_{(j)}]_{\mathrm{n}_{\mathrm{j+1}}^{*}}=\boldsymbol{H}_{j}[\boldsymbol{U}_{j}]_{\mathrm{n}_{\mathrm{j+1}}^{*}}+[\boldsymbol{N}_{(j)}]_{\mathrm{n}_{\mathrm{j+1}}^{*}}=\boldsymbol{H}_{j}[\boldsymbol{X}_{(j)}]_{\mathrm{n}_{\mathrm{j+1}}^{*}}\left(\otimes_{i=j-1}^{1}\boldsymbol{H}_{i}\right)^{\top}+[\boldsymbol{N}_{(j)}]_{\mathrm{n}_{\mathrm{j+1}}^{*}}.$$

According to Lemma 17 and denoting the solution as $[\tilde{\boldsymbol{U}}_{j}]_{\mathrm{n}_{\mathrm{j+1}}^{*}}$ with k IHT or HTP iterations, we have

$$\|[\tilde{\boldsymbol{U}}_{j}]_{\mathrm{n}_{\mathrm{j+1}}^{*}}-[\boldsymbol{U}_{j}]_{\mathrm{n}_{\mathrm{j+1}}^{*}}\|_{\mathrm{F}}\leq\alpha_{j}^{k}\|[\boldsymbol{U}_{j}]_{\mathrm{n}_{\mathrm{j+1}}^{*}}\|_{\mathrm{F}}+\tau_{j}\|[\boldsymbol{N}_{(j)}]_{\mathrm{n}_{\mathrm{j+1}}^{*}}\|_{\mathrm{F}},$$

where μ_{j} and τ_{j} relate to the RICs of matrix $\boldsymbol{H}_{j}$. The only step left is to bound $\|[\boldsymbol{N}_{(j)}]_{\mathrm{n}_{\mathrm{j+1}}^{*}}\|_{\mathrm{F}}$ using $[\boldsymbol{E}_{j+1}]_{\mathrm{n}_{\mathrm{j+2}}^{*}}$, where $\boldsymbol{E}_{j+1}=\tilde{\boldsymbol{U}}_{j+1}-\boldsymbol{U}_{j+1}$.

We recall that (C.1) is obtained by unfolding the measurement tensor formed from the matrix $\tilde{\boldsymbol{U}}_{j+1} = \boldsymbol{U}_{j+1} + \boldsymbol{E}_{j+1}$ along its jth mode. Hence, $\boldsymbol{N}_{(j)}$ is simply reordered version of $\boldsymbol{E}_{j+1}$. Consequently, the entries of the matrix $[\boldsymbol{N}_{(j)}]_{\mathrm{n}^*_{j+1}}$ are essentially entries of the n^*_{j+1}th row of $[\boldsymbol{E}_{j+1}]_{\mathrm{n}^*_{j+2}}$, leading to

$$\|[\boldsymbol{N}_{(j)}]_{\mathrm{n}^*_{j+1}}\|_{\mathrm{F}} \leq \|[\boldsymbol{E}_{j+1}]_{\mathrm{n}^*_{j+2}}\|_{\mathrm{F}},$$

and we arrive at the desired result.

To elaborate, we first investigate the indices of the entries of $[\boldsymbol{N}_{(j)}]_{\mathrm{n}^*_{j+1}}$. The entries of the n^*_jth row of matrix $[\boldsymbol{N}_{(j)}]_{\mathrm{n}^*_{j+1}}$ are obtained by i) fixing $n_j = n^*_j$ (row index) and $n_{j+1} = n^*_{j+1}, \cdots, n_I = n^*_I$ (encapsulation), and ii) running $n_1, \cdots, n_{j-1}$ from one till $N_1, \cdots, N_{j-1}$, respectively. Thus, the entries of matrix $[\boldsymbol{N}_{(j)}]_{\mathrm{n}^*_{j+1}}$ can be obtained by i) fixing $n_{j+1} = n^*_{j+1}, \cdots, n_I = n^*_I$, ii) running $n_1, \cdots, n_{j-1}$ from one till $N_1, \cdots, N_{j-1}$, respectively, and iii) running $n_j = 1, \cdots, N_j$ (going over all rows). Given such knowledge, we start investigating the n^*_{j+1}th row of matrix $[\boldsymbol{E}_{j+1}]_{\mathrm{n}^*_{j+2}}$. The entries of this row are obtained by i) fixing $n_{j+1} = n^*_{j+1}$ (row index), ii) fixing $n_{j+2} = n^*_{j+2}, \cdots, n_I = n^*_I$ (fixed encapsulation), and iii) running $n_1, \cdots, n_j$ from one till $N_1, \cdots, N_j$, respectively. By comparing how indices are arranged, we can see that the entries of the matrix $[\boldsymbol{N}_{(j)}]_{\mathrm{n}^*_{j+1}}$ are essentially entries of the n^*_{j+1}th row of $[\boldsymbol{E}_{j+1}]_{\mathrm{n}^*_{j+2}}$, inferring $\|[\boldsymbol{N}_{(j)}]_{\mathrm{n}^*_{j+1}}\|_{\mathrm{F}} \leq \|[\boldsymbol{N}_{j+1}]_{\mathrm{n}^*_{j+2}}\|_{\mathrm{F}}$. □

As we have described before, (C.1) is solved through multiple independent MMV problems. Thus, the error bound is also given regarding each individual MMV problem. Further, not only the jth step, but also the $j+1$th step is solved through multiple independent MMV problems. Thus, we do not have the upper bound for $\boldsymbol{E}_{j+1}$ in (4.6) as a whole but only the upper bound for each column block $[\boldsymbol{E}_{j+1}]_{\mathrm{n}_{j+2}}$. Fortunately, since all the noise entries in the jth step are contained as one single row of the noise block in the previous step, having the upper bound for each column block $[\boldsymbol{E}_{j+1}]_{\mathrm{n}_{j+2}}$ is sufficient to derive the noise bound for the jth step, which is shown in Lemma 18 and (C.2).

Now, we proceed to the proof of Theorem 8. Generally speaking, Theorem 8 is obtained by recursively applying Lemma 18. Particularly, focusing on the $\boldsymbol{s}$ hierarchical sparse vectors, for the last step, i.e., *the first mode unfolding*, we solve

$$\boldsymbol{T}_{(1)} = \boldsymbol{H}_1 \boldsymbol{X}_{(1)} + \boldsymbol{N}_{(1)},$$

leading to $\prod_{j=2}^{I} N_j$ SMV problems. They are SMV because there is only one column in each column block, and hence the MMV problem reduces to the SMV problem. Lemma 18 indicates that

$$\begin{aligned}
\|\tilde{\boldsymbol{U}}_1 - \boldsymbol{U}_1\|_{\mathrm{F}} &\leq \sum_{n_2,\cdots,n_I} \|[\tilde{\boldsymbol{U}}_1]_{\mathrm{n}_2} - [\boldsymbol{U}_1]_{\mathrm{n}_2}\|_{\mathrm{F}} \leq \sum_{n_2,\cdots,n_I} \alpha_1^k \|[\boldsymbol{U}_1]_{\mathrm{n}_2}\|_{\mathrm{F}} + \tau_1 \|[\boldsymbol{E}_2]_{\mathrm{n}_3}\|_{\mathrm{F}} \\
&\leq \sum_{n_2,\cdots,n_I} \alpha_1^k \|[\boldsymbol{U}_1]_{\mathrm{n}_2}\|_{\mathrm{F}} + \tau_1 \left(\alpha_2^k \|[\boldsymbol{U}_2]_{\mathrm{n}_3}\|_{\mathrm{F}} + \tau_2 \|[\boldsymbol{E}_3]_{\mathrm{n}_4}\|_{\mathrm{F}} \right) \\
&\leq \sum_{n_2,\cdots,n_I} \left(\sum_{i=1}^{I} \prod_{j=1}^{i-1} \tau_j \alpha_i^k \left\| [\boldsymbol{U}_i]_{\mathrm{n}_{i+1}} \right\|_{\mathrm{F}} + \prod_{i=1}^{I} \tau_i \|[\boldsymbol{E}_I]_{\mathrm{n}_{I+1}}\|_{\mathrm{F}} \right).
\end{aligned}$$

We note that $I+1$th level contains only one block, leading to $[\boldsymbol{E}_I]_{\mathrm{n}_{I+1}} = \boldsymbol{E}_I$. Using the relation $\tilde{\boldsymbol{U}}_I = \boldsymbol{U}_I + \boldsymbol{E}_I$ and Lemma 17 leads to $\|\boldsymbol{E}_I\|_{\mathrm{F}} \leq \alpha_I^k \|\boldsymbol{U}_I\|_{\mathrm{F}} + \tau_I \|\boldsymbol{N}_{(1)}\|_{\mathrm{F}}$. This concludes the proof for $\boldsymbol{s}$ hierarchical sparsity. For s standard sparsity, the upper bound for all s standard sparse vectors is the worst upper bound among all possible $\boldsymbol{s}$ corresponding to the sparsity level s. Therefore, taking the maximum over $\forall \boldsymbol{s} \in f_{\mathrm{N}}(s)$ concludes the proof.

For $\boldsymbol{s}$ Kronecker-supported sparsity, since the support is shared among different blocks in the same level, it is unnecessary to introduce multiple MMV problems, but to solve only one MMV problem. Thus, recursively applying Lemma 17 leads to the final result. For the last step, we solve

$$\boldsymbol{T}_{(1)} = \boldsymbol{H}_1 \boldsymbol{X}_{(1)} + \boldsymbol{N}_{(1)},$$

which leads to the following relations,

$$\begin{aligned}
\|\tilde{\boldsymbol{U}}_1 - \boldsymbol{U}_1\|_{\mathrm{F}} &\leq \alpha_1^k \|\boldsymbol{U}_1\|_{\mathrm{F}} + \tau_1 \|\boldsymbol{N}_{(1)}\|_{\mathrm{F}} \\
&\leq \alpha_1^k \|\boldsymbol{X}_{(1)}\|_{\mathrm{F}} + \tau_1 \left(\alpha_2^k \|\boldsymbol{U}_2\|_{\mathrm{F}} + \tau_2 \|\boldsymbol{N}_{(2)}\|_{\mathrm{F}} \right) \\
&\leq \sum_{i=1}^{I} \prod_{j=1}^{i-1} \alpha_i^k \tau_j \|\boldsymbol{U}_i\|_{\mathrm{F}} + \prod_{i=1}^{I} \tau_i \|\mathbf{N}\|_{\mathrm{F}}.
\end{aligned}$$

Thus, the proof is complete.

Bibliography

[1] E. J. Candès, J. Romberg and T. Tao. 'Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information'. In: *IEEE Transactions on Information Theory* 52.2 (2006), pp. 489–509.

[2] D. L. Donoho. 'Compressed sensing'. In: *IEEE Transactions on Information Theory* 52.4 (2006), pp. 1289–1306.

[3] F. Marvasti, A. Amini, F. Haddadi, M. Soltanolkotabi, B. H. Khalaj, A. Aldroubi, S. Sanei and J. Chambers. 'A unified approach to sparse signal processing'. In: *EURASIP Journal on Advances in Signal Processing* 2012.1 (2012), p. 44.

[4] Y. Niu, Y. Li, D. Jin, L. Su and A. V. Vasilakos. 'A survey of millimeter wave communications (mmWave) for 5G: Opportunities and challenges'. In: *Wireless Networks* 21.8 (2015), pp. 2657–2676.

[5] X. Wang, L. Kong, F. Kong, F. Qiu, M. Xia, S. Arnon and G. Chen. 'Millimeter wave communication: A comprehensive survey'. In: *IEEE Communications Surveys & Tutorials* 20.3 (2018), pp. 1616–1653.

[6] M. Xiao, S. Mumtaz, Y. Huang, L. Dai, Y. Li, M. Matthaiou, G. K. Karagiannidis, E. Björnson, K. Yang, A. Ghosh *et al.* 'Millimeter wave communications for future mobile networks'. In: *IEEE Journal on Selected Areas in Communications* 35.9 (2017), pp. 1909–1935.

[7] C. Pan, G. Zhou, K. Zhi, S. Hong, T. Wu, Y. Pan, H. Ren, M. Di Renzo, A. L. Swindlehurst, R. Zhang *et al.* 'An overview of signal processing techniques for RIS/IRS-aided wireless systems'. In: *IEEE Journal of Selected Topics in Signal Processing* 16.5 (2022), pp. 883–917.

[8] B. Zheng, C. You, W. Mei and R. Zhang. 'A survey on channel estimation and practical passive beamforming design for intelligent reflecting surface aided wireless communications'. In: *IEEE Communications Surveys and Tutorials* 24.2 (2022), pp. 1035–1071.

[9] K. Hammernik, T. Klatzer, E. Kobler, M. P. Recht, D. K. Sodickson, T. Pock and F. Knoll. 'Learning a variational network for reconstruction of accelerated MRI data'. In: *Magnetic Resonance in Medicine* 79.6 (2018), pp. 3055–3071.

[10] M. Lustig, D. Donoho and J. M. Pauly. 'Sparse MRI: The application of compressed sensing for rapid MR imaging'. In: *Magnetic Resonance in Medicine* 58.6 (2007), pp. 1182–1195.

[11] S. Geethanath, R. Reddy, A. S. Konar, S. Imam, R. Sundaresan, R. B. DR and R. Venkatesan. 'Compressed sensing MRI: A review'. In: *Critical Reviews™ in Biomedical Engineering* 41.3 (2013).

[12] K. G. Hollingsworth. 'Reducing acquisition time in clinical MRI by data undersampling and compressed sensing reconstruction'. In: *Physics in Medicine & Biology* 60.21 (2015), R297.

[13] C. D. Austin, E. Ertin and R. L. Moses. 'Sparse signal methods for 3-D radar imaging'. In: *IEEE Journal of Selected Topics in Signal Processing* 5.3 (2010), pp. 408–423.

[14] Ç. Önen, A. Pandharipande, G. Joseph and N. J. Myers. 'Occupancy grid mapping for automotive driving exploiting clustered sparsity'. In: *IEEE Sensors Journal* 24.7 (2024), pp. 9240–9250.

[15] P. Zhai, G. Joseph, N. J. Myers, Ç. Önen and A. Pandharipande. 'Spatial sparsity-aware radar-liDAR fusion for occupancy grid mapping in automotive driving'. In: *IEEE Sensors Journal* (2025).

[16] M. F. Duarte and R. G. Baraniuk. 'Kronecker compressive sensing'. In: *IEEE Transactions on Image Processing* 21.2 (2011), pp. 494–504.

[17] C. F. Caiafa and A. Cichocki. 'Block sparse representations of tensors using Kronecker bases'. In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*. 2012, pp. 2709–2712.

[18] C. F. Caiafa and A. Cichocki. 'Computing sparse representations of multidimensional signals using Kronecker bases'. In: *Neural Computation* 25.1 (2013), pp. 186–220.

[19] C. F. Caiafa and A. Cichocki. 'Multidimensional compressed sensing and their applications'. In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 3.6 (2013), pp. 355–380.

[20] Y. Rivenson and A. Stern. 'Compressed imaging with a separable sensing operator'. In: *IEEE Signal Processing Letters* 16.6 (2009), pp. 449–452.

[21] 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'. In: *IEEE Journal on Selected Areas in Communications* 35.7 (2017), pp. 1524–1538.

[22] R. Wang, H. Ren, C. Pan, G. Zhou and J. Wang. 'Tensor decomposition-based time varying channel estimation for mmWave MIMO-OFDM systems'. In: *arXiv preprint arXiv:2403.02942* (2024).

[23] Y. He and G. Joseph. 'Bayesian algorithms for Kronecker-structured sparse vector recovery with application to IRS-MIMO channel estimation'. In: *IEEE Transactions on Signal Processing* 73 (2025), pp. 142–157.

[24] X. Liu, W. Wang, X. Gong, X. Fu, X. Gao and X.-G. Xia. 'Structured hybrid message passing based channel estimation for massive MIMO-OFDM systems'. In: *IEEE Transactions on Vehicular Technology* 72.6 (2023), pp. 7491–7507.

[25] A. Alkhateeb, G. Leus and R. W. Heath. 'Compressed sensing based multi-user millimeter wave systems: How many measurements are needed?' In: *Proceedings of the IEEE international conference on acoustics, speech and signal processing*. IEEE. 2015, pp. 2909–2913.

[26] Q. Li and E. A. Bernal. 'An algorithm for parallel reconstruction of jointly sparse tensors with applications to hyperspectral imaging'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2017, pp. 24–31.

[27] Y. He and G. Joseph. 'Structure-aware sparse Bayesian learning-based channel estimation for intelligent reflecting surface-aided MIMO'. In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*. 2023, pp. 1–5.

[28] M. F. Duarte and R. G. Baraniuk. 'Kronecker product matrices for compressive sensing'. In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*. 2010, pp. 3650–3653.

[29] N. D. Sidiropoulos and A. Kyrillidis. 'Multi-way compressed sensing for sparse low-rank tensors'. In: *IEEE Signal Processing Letters* 19.11 (2012), pp. 757–760.

[30] G. Ortiz-Jiménez, M. Coutino, S. P. Chepuri and G. Leus. 'Sparse sampling for inverse problems with tensors'. In: *IEEE Transactions on Signal Processing* 67.12 (2019), pp. 3272–3286.

[31] W.-C. Chang and Y. T. Su. 'Sparse Bayesian learning based tensor dictionary learning and signal recovery with application to MIMO channel estimation'. In: *IEEE Journal of Selected Topics in Signal Processing* 15.3 (2021), pp. 847–859.

[32] S. Friedland, Q. Li and D. Schonfeld. 'Compressive sensing of sparse tensors'. In: *IEEE Transactions on Image Processing* 23.10 (2014), pp. 4438–4447.

[33] T. Hastie, R. Tibshirani and J. Friedman. *The elements of statistical learning: Data mining, inference, and prediction*. Vol. 2. Springer, 2009.

[34] L. He and L. Carin. 'Exploiting structure in wavelet-based Bayesian compressive sensing'. In: *IEEE Transactions on Signal Processing* 57.9 (2009), pp. 3488–3497.

[35] L. He, H. Chen and L. Carin. 'Tree-structured compressive sensing with variational Bayesian analysis'. In: *IEEE Signal Processing Letters* 17.3 (2009), pp. 233–236.

[36] X. Ma and G. B. Giannakis. 'Maximum-diversity transmissions over doubly selective wireless channels'. In: *IEEE Transactions on Information Theory* 49.7 (2003), pp. 1832–1840.

[37] I. Barhumi, G. Leus and M. Moonen. 'MMSE estimation of basis expansion models for rapidly time-varying channels'. In: *Proceedings of the European Signal Processing Conference*. 2005, pp. 1–5.

[38] G. B. Giannakis and C. Tepedelenlioglu. 'Basis expansion models and diversity techniques for blind identification and equalization of time-varying channels'. In: *Proceedings of the IEEE* 86.10 (1998), pp. 1969–1986.

[39] J. Han, L. Zhang, Q. Zhang and G. Leus. 'Low-complexity equalization of orthogonal signal-division multiplexing in doubly-selective channels'. In: *IEEE Transactions on Signal Processing* 67.4 (2018), pp. 915–929.

[40] C. Li and B. Adcock. 'Compressed sensing with local structure: Uniform recovery guarantees for the sparsity in levels class'. In: *Applied and Computational Harmonic Analysis* 46.3 (2019), pp. 453–477.

[41] B. Adcock, S. Brugiapaglia and M. King-Roskamp. 'The benefits of acting locally: Reconstruction algorithms for sparse in levels signals with stable and robust recovery guarantees'. In: *IEEE Transactions on Signal Processing* 69 (2021), pp. 3160–3175.

[42] R. Zhao, Q. Wang, J. Fu and L. Ren. 'Exploiting block-sparsity for hyperspectral Kronecker compressive sensing: A tensor-based Bayesian method'. In: *IEEE Transactions on Image Processing* 29 (2019), pp. 1654–1668.

[43] L. Sun, D. Huang, H. Sun and J.-X. Wang. 'Bayesian spline learning for equation discovery of nonlinear dynamics with quantified uncertainty'. In: *Proceedings of Advances in Neural Information Processing Systems* 35 (2022), pp. 6927–6940.

[44] Y. Yuan, X. Tang, W. Zhou, W. Pan, X. Li, H.-T. Zhang, H. Ding and J. Goncalves. 'Data driven discovery of cyber physical systems'. In: *Nature Communications* 10.1 (2019), p. 4894.

[45] S. E. Ament and C. P. Gomes. 'Sparse Bayesian learning via stepwise regression'. In: *Proceedings of the International Conference on Machine Learning*. 2021, pp. 264–274.

[46] C. R. Berger, Z. Wang, J. Huang and S. Zhou. 'Application of compressive sensing to sparse channel estimation'. In: *IEEE Communications Magazine* 48.11 (2010), pp. 164–174.

[47] Z. Xiao, S. Cao, L. Zhu, Y. Liu, B. Ning, X.-G. Xia and R. Zhang. 'Channel estimation for movable antenna communication systems: A framework based on compressed sensing'. In: *IEEE Transactions on Wireless Communications* 23.9 (2024), pp. 11814–11830.

[48] D. C. Araújo, A. L. De Almeida, J. P. Da Costa and R. T. de Sousa. 'Tensor-based channel estimation for massive MIMO-OFDM systems'. In: *IEEE Access* 7 (2019), pp. 42133–42147.

[49] G. Wunder, I. Roth, R. Fritschek and J. Eisert. 'HiHTP: A custom-tailored hierarchical sparse detector for massive MTC'. In: *Proceedings of the Asilomar Conference on Signals, Systems, and Computers*. IEEE. 2017, pp. 1929–1934.

[50] I. Roth, A. Flinth, R. Kueng, J. Eisert and G. Wunder. 'Hierarchical restricted isometry property for Kronecker product measurements'. In: *Proceedings of the Annual Allerton Conference on Communication, Control, and Computing*. IEEE. 2018, pp. 632–638.

[51] B. Groß, A. Flinth, I. Roth, J. Eisert and G. Wunder. 'Hierarchical sparse recovery from hierarchically structured measurements with application to massive random access'. In: *Proceedings of the IEEE Statistical Signal Processing Workshop*. IEEE. 2021, pp. 531–535.

[52] G. Wunder, Č. Stefanović, P. Popovski and L. Thiele. 'Compressive coded random access for massive MTC traffic in 5G systems'. In: *Proceedings of the Asilomar Conference on Signals, Systems and Computers*. IEEE. 2015, pp. 13–17.

[53] X. Xu, S. Zhang, F. Gao and J. Wang. 'Sparse Bayesian learning based channel extrapolation for RIS assisted MIMO-OFDM'. In: *IEEE Transactions on Communications* 70.8 (2022), pp. 5498–5513.

[54] S. Yang, M. Wang, P. Li, L. Jin, B. Wu and L. Jiao. 'Compressive hyperspectral imaging via sparse tensor and nonlinear compressed sensing'. In: *IEEE Transactions on Geoscience and Remote Sensing* 53.11 (2015), pp. 5943–5957.

[55] M. F. Duarte and R. G. Baraniuk. 'Kronecker product matrices for compressive sensing'. In: *Rice University, Houston, Technical Report* (2011). URL: http://www.ecs.umass.edu/mduarte/images/KroneckerCS-TREE1105.pdf.

[56] S. Friedland, Q. Li, D. Schonfeld and E. A. Bernal. 'Two algorithms for compressed sensing of sparse tensors'. In: *Compressed Sensing and its Applications: MATHEON Workshop 2013*. Springer. 2015, pp. 259–281.

[57] I. Roth, M. Kliesch, A. Flinth, G. Wunder and J. Eisert. 'Reliable recovery of hierarchically sparse signals for Gaussian and Kronecker product measurements'. In: *IEEE Transactions on Signal Processing* 68 (2020), pp. 4002–4016.

[58] Y. He and G. Joseph. 'Kronecker-structured sparse vector recovery with application to IRS-MIMO channel estimation'. In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*. 2025, pp. 1–5.

[59] Y. He and G. Joseph. 'Efficient off-grid Bayesian parameter estimation for Kronecker structured signals'. In: *IEEE Transactions on Signal Processing* 73 (2025), pp. 2616–2630.

[60] Y. He and G. Joseph. 'A hierarchical view of structured sparsity in Kronecker compressive sensing'. In: *Proceedings of the European Signal Processing Conference*. 2025, pp. 1–5.

[61] Y. He and G. Joseph. 'Hierarchical multi-stage recovery framework for Kronecker compressed sensing'. In: *Proceedings of the International Conference on Learning Representations*. 2026.

[62] D. P. Wipf and B. D. Rao. 'Sparse Bayesian learning for basis selection'. In: *IEEE Transactions on Signal Processing* 52.8 (2004), pp. 2153–2164.

[63] J. Fang, Y. Shen, H. Li and P. Wang. 'Pattern-coupled sparse Bayesian learning for recovery of block-sparse signals'. In: *IEEE Transactions on Signal Processing* 63.2 (2014), pp. 360–372.

[64] 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'. In: *IEEE Transactions on Signal Processing* 66.10 (2018), pp. 2759–2771.

[65] D. Prasanna and C. R. Murthy. 'mmWave channel estimation via compressive covariance estimation: Role of sparsity and intra-vector correlation'. In: *IEEE Transactions on Signal Processing* 69 (2021), pp. 2356–2370.

[66] X. Wu, S. Ma, X. Yang and G. Yang. 'Clustered sparse Bayesian learning based channel estimation for millimeter-wave massive MIMO systems'. In: *IEEE Transactions on Vehicular Technology* 71.12 (2022), pp. 12749–12764.

[67] A. Băltoiu and B. Dumitrescu. 'Sparse Bayesian learning algorithm for separable dictionaries'. In: *Digital Signal Processing* 111 (2021), p. 102990.

[68] K. Kreutz-Delgado, B. Rao, I. Fedorov and S. Das. 'Dictionaries in machine learning'. In: *Signal Processing and Machine Learning Theory*. Elsevier, 2024, pp. 1073–1159.

[69] Z. Zhang and B. D. Rao. 'Extension of SBL algorithms for the recovery of block sparse signals with intra-block correlation'. In: *IEEE Transactions on Signal Processing* 61.8 (2013), pp. 2009–2015.

[70] C. M. Bishop and N. M. Nasrabadi. *Pattern recognition and machine learning*. Vol. 4. 4. Springer, 2006.

[71] C. J. Wu. 'On the convergence properties of the EM algorithm'. In: *Annals of Statistics* (1983), pp. 95–103.

[72] G. Joseph and C. R. Murthy. 'On the convergence of a Bayesian algorithm for joint dictionary learning and sparse recovery'. In: *IEEE Transactions on Signal Processing* 68 (2019), pp. 343–358.

[73] W. I. Zangwill. *Nonlinear programming: A unified approach*. Vol. 52. Prentice-Hall Englewood Cliffs, NJ, 1969.

[74] S. Khanna and C. R. Murthy. 'On the support recovery of jointly sparse Gaussian sources via sparse Bayesian learning'. In: *IEEE Transactions on Information Theory* 68.11 (2022), pp. 7361–7378.

[75] A. Lin, A. H. Song, B. Bilgic and D. Ba. 'Covariance-free sparse Bayesian learning'. In: *IEEE Transactions on Signal Processing* 70 (2022), pp. 3818–3831.

[76] Z. Zhang and B. D. Rao. 'Clarify some issues on the sparse Bayesian learning for sparse signal recovery'. In: *University of California, San Diego, Technical Report* (2011).

[77] H. Guo and V. K. Lau. 'Uplink cascaded channel estimation for intelligent reflecting surface assisted multiuser MISO systems'. In: *IEEE Transactions on Signal Processing* 70 (2022), pp. 3964–3977.

[78] Y. Zhu, H. Guo and V. K. Lau. 'Bayesian channel estimation in multi-user massive MIMO with extremely large antenna array'. In: *IEEE Transactions on Signal Processing* 69 (2021), pp. 5463–5478.

[79] Z. Mao, X. Liu and M. Peng. 'Channel estimation for intelligent reflecting surface assisted massive MIMO systems—A deep learning approach'. In: *IEEE Communications Letters* 26.4 (2022), pp. 798–802.

[80] C. R. Rao. 'Estimation of heteroscedastic variances in linear models'. In: *Journal of the American Statistical Association* 65.329 (1970), pp. 161–172.

[81] P. Wang, J. Fang, H. Duan and H. Li. 'Compressed channel estimation for intelligent reflecting surface-assisted millimeter wave systems'. In: *IEEE Signal Processing Letters* 27 (2020), pp. 905–909.

[82] Y. Lin, S. Jin, M. Matthaiou and X. You. 'Channel estimation and user localization for IRS-assisted MIMO-OFDM systems'. In: *IEEE Transactions on Wireless Communications* 21.4 (2021), pp. 2320–2335.

[83] S. Noschese, L. Pasquini and L. Reichel. 'Tridiagonal Toeplitz matrices: Properties and novel applications'. In: *Numerical Linear Algebra with Applications* 20.2 (2013), pp. 302–326.

[84] H. Zhu, G. Leus and G. B. Giannakis. 'Sparsity-cognizant total least-squares for perturbed compressive sampling'. In: *IEEE Transactions on Signal Processing* 59.5 (2011), pp. 2002–2016.

[85] G. Tang, B. N. Bhaskar, P. Shah and B. Recht. 'Compressed sensing off the grid'. In: *IEEE Transactions on Information Theory* 59.11 (2013), pp. 7465–7490.

[86] Z. Yang, L. Xie and C. Zhang. 'Off-grid direction of arrival estimation using sparse Bayesian inference'. In: *IEEE Transactions on Signal Processing* 61.1 (2012), pp. 38–43.

[87] K. You, W. Guo, T. Peng, Y. Liu, P. Zuo and W. Wang. 'Parametric sparse Bayesian dictionary learning for multiple sources localization with propagation parameters uncertainty'. In: *IEEE Transactions on Signal Processing* 68 (2020), pp. 4194–4209.

[88] Y. Mao, Q. Guo, J. Ding, F. Liu and Y. Yu. 'Marginal likelihood maximization based fast array manifold matrix learning for direction of arrival estimation'. In: *IEEE Transactions on Signal Processing* 69 (2021), pp. 5512–5522.

[89] J. Dai, A. Liu and V. K. Lau. 'FDD massive MIMO channel estimation with arbitrary 2D-array geometry'. In: *IEEE Transactions on Signal Processing* 66.10 (2018), pp. 2584–2599.

[90] M. Ibrahim, F. Römer, R. Alieiev, G. Del Galdo and R. S. Thomä. 'On the estimation of grid offsets in CS-based direction-of-arrival estimation'. In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*. 2014, pp. 6776–6780.

[91] A. Faul and M. Tipping. 'Analysis of sparse Bayesian learning'. In: *Advances in Neural Information Processing Systems* 14 (2001).

[92] R. R. Pote and B. D. Rao. 'Light-weight sequential SBL algorithm: An alternative to OMP'. In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*. 2023, pp. 1–5.

[93] C. K. Thomas and D. Slock. 'Space alternating variational estimation and Kronecker structured dictionary learning'. In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE. 2019, pp. 5556–5560.

[94] T. Jia, H. Liu, C. Gao and J. Yan. 'Bayesian direction of arrival estimation using atomic norm minimization with prior knowledge'. In: *IEEE Transactions on Aerospace and Electronic Systems* (2024).

[95] Z. Yang and L. Xie. 'Frequency-selective Vandermonde decomposition of Toeplitz matrices with applications'. In: *Signal Processing* 142 (2018), pp. 157–167.

[96] Z. Yang and L. Xie. 'On gridless sparse methods for line spectral estimation from complete and incomplete data'. In: *IEEE Transactions on Signal Processing* 63.12 (2015), pp. 3139–3153.

[97] M.-A. Badiu, T. L. Hansen and B. H. Fleury. 'Variational Bayesian inference of line spectra'. In: *IEEE Transactions on Signal Processing* 65.9 (2017), pp. 2247–2261.

[98] J. Zhu, Q. Zhang, P. Gerstoft, M.-A. Badiu and Z. Xu. 'Grid-less variational Bayesian line spectral estimation with multiple measurement vectors'. In: *Signal Processing* 161 (2019), pp. 155–164.

[99] A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa and H. A. Phan. 'Tensor decompositions for signal processing applications: From two-way to multiway component analysis'. In: *IEEE Signal Processing Magazine* 32.2 (2015), pp. 145–163.

[100] L. De Lathauwer, B. De Moor and J. Vandewalle. 'A multilinear singular value decomposition'. In: *SIAM Journal on Matrix Analysis and Applications* 21.4 (2000), pp. 1253–1278.

[101] A. Zhang and D. Xia. 'Tensor SVD: Statistical and computational limits'. In: *IEEE Transactions on Information Theory* 64.11 (2018), pp. 7311–7338.

[102] E. R. Balda, S. A. Cheema, J. Steinwandt, M. Haardt, A. Weiss and A. Yeredor. 'First-order perturbation analysis of low-rank tensor approximations based on the truncated HOSVD'. In: *Proceedings of the Asilomar Conference on Signals, Systems, and Computers*. 2016, pp. 1723–1727.

[103] M. E. Tipping. 'Sparse Bayesian learning and the relevance vector machine'. In: *Journal of Machine Learning Research* 1.Jun (2001), pp. 211–244.

[104] C. F. Van Loan. 'The ubiquitous Kronecker product'. In: *Journal of Computational and Applied Mathematics* 123.1-2 (2000), pp. 85–100.

[105] R. Schroeder, J. He, H. Djelouat and M. Juntti. 'Low-complexity near-field channel estimation for hybrid RIS assisted systems'. In: *arXiv preprint arXiv:2404.17411* (2024).

[106] J. Wang, J. Fang and H. Li. 'Intelligent reflecting surface-assisted NLOS sensing via tensor decomposition'. In: *Proceedings of the European Signal Processing Conference*. 2024, pp. 1–5.

[107] Y. Wang, G. Leus and A. Pandharipande. 'Direction estimation using compressive sampling array processing'. In: *IEEE Workshop on Statistical Signal Processing*. 2009, pp. 626–629.

[108] N. Vervliet, O. Debals, L. Sorber, M. Van Barel and L. De Lathauwer. *Tensorlab 3.0*. Available online. 2016. URL: https://www.tensorlab.net.

[109] M. Guo, Y. D. Zhang and T. Chen. 'DOA estimation using compressed sparse array'. In: *IEEE Transactions on Signal Processing* 66.15 (2018), pp. 4133–4146.

[110] D. Dampahalage, K. S. Manosha, N. Rajatheva and M. Latva-Aho. 'Supervised learning based sparse channel estimation for RIS aided communications'. In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*. 2022, pp. 8827–8831.

[111] J. He, M. Leinonen, H. Wymeersch and M. Juntti. 'Channel estimation for RIS-aided mmWave MIMO systems'. In: *Proceedings of the IEEE Global Communications Conference*. 2020, pp. 1–6.

[112] Z. Wan, Z. Gao, F. Gao, M. Di Renzo and M.-S. Alouini. 'Terahertz massive MIMO with holographic reconfigurable intelligent surfaces'. In: *IEEE Transactions on Communications* 69.7 (2021), pp. 4732–4750.

[113] L. Yashvanth and C. R. Murthy. 'Cascaded channel estimation for distributed IRS Aided mmWave massive MIMO systems'. In: *Proceedings of the IEEE Global Communications Conference*. 2022, pp. 717–723.

[114] Y. Liu, X. Liu, X. Mu, T. Hou, J. Xu, M. Di Renzo and N. Al-Dhahir. 'Reconfigurable intelligent surfaces: Principles and opportunities'. In: *IEEE Communications Surveys and Tutorials* 23.3 (2021), pp. 1546–1577.

[115] G. Wunder, S. Stefanatos, A. Flinth, I. Roth and G. Caire. 'Low-overhead hierarchically sparse channel estimation for multiuser wideband massive MIMO'. In: *IEEE Transactions on Wireless Communications* 18.4 (2019), pp. 2186–2199.

[116] T. Blumensath. 'Sampling and reconstructing signals from a union of linear subspaces'. In: *IEEE Transactions on Information Theory* 57.7 (2011), pp. 4660–4671.

[117] B. Li and A. P. Petropulu. 'Structured sampling of structured signals'. In: *Proceedings of the IEEE Global Conference on Signal and Information Processing*. IEEE. 2013, pp. 1009–1012.

[118] Y. C. Eldar and M. Mishali. 'Robust recovery of signals from a structured union of subspaces'. In: *IEEE Transactions on Information Theory* 55.11 (2009), pp. 5302–5316.

[119] S. Foucart and H. Rauhut. *A mathematical introduction to compressive sensing*. Birkhäuser Basel, 2013. ISBN: 0-8176-4947-6.

[120] D. P. Wipf and B. D. Rao. 'An empirical Bayesian strategy for solving the simultaneous sparse approximation problem'. In: *IEEE Transactions on Signal Processing* 55.7 (2007), pp. 3704–3716.

[121] J. D. Blanchard, M. Cermak, D. Hanle and Y. Jing. 'Greedy algorithms for joint sparse recovery'. In: *IEEE Transactions on Signal Processing* 62.7 (2014), pp. 1694–1704.

[122] J. A. Tropp, A. C. Gilbert and M. J. Strauss. 'Algorithms for simultaneous sparse approximation. Part I: Greedy pursuit'. In: *Signal Processing* 86.3 (2006), pp. 572–588.

[123] T. M. Inc. *MATLAB version: 24.2.0.2923080 (R2024b)*. Natick, Massachusetts, United States, 2024.

[124] S. Foucart. 'Hard thresholding pursuit: An algorithm for compressive sensing'. In: *SIAM Journal on Numerical Analysis* 49.6 (2011), pp. 2543–2563.

[125] S. Haghighatshoar and G. Caire. 'Massive MIMO pilot decontamination and channel interpolation via wideband sparse channel estimation'. In: *IEEE Transactions on Wireless Communications* 16.12 (2017), pp. 8316–8332.

[126] Z. Chen and C. Yang. 'Pilot decontamination in wideband massive MIMO systems by exploiting channel sparsity'. In: *IEEE Transactions on Wireless Communications* 15.7 (2016), pp. 5087–5100.

[127] Y. You, L. Zhang, M. Yang, Y. Huang, X. You and C. Zhang. 'Structured OMP for IRS-Assisted mmWave channel estimation by exploiting angular spread'. In: *IEEE Transactions on Vehicular Technology* 71.4 (2022), pp. 4444–4448.

[128] A. Alkhateeb, O. El Ayach, G. Leus and R. W. Heath. 'Channel estimation and hybrid precoding for millimeter wave cellular systems'. In: *IEEE Journal of Selected Topics in Signal Processing* 8.5 (2014), pp. 831–846.

[129] E. van den Berg and M. P. Friedlander. 'Probing the Pareto frontier for basis pursuit solutions'. In: *SIAM Journal on Scientific Computing* 31.2 (2008), pp. 890–912.

[130] E. van den Berg and M. P. Friedlander. *SPGL1: A solver for large-scale sparse reconstruction*. 2019.

[131] A. Liu, V. K. Lau and W. Dai. 'Exploiting burst-sparsity in massive MIMO with partial channel support information'. In: *IEEE Transactions on Wireless Communications* 15.11 (2016), pp. 7820–7830.

[132] L. Chen, A. Liu and X. Yuan. 'Structured turbo compressed sensing for massive MIMO channel estimation using a Markov prior'. In: *IEEE Transactions on Vehicular Technology* 67.5 (2017), pp. 4635–4639.

[133] L. Qin, Z. Wang, Q. Guan, X. Guo and G. Wang. 'Variational Bayesian image restoration with multi-structured model of wavelet transform coefficients'. In: *Signal Processing: Image Communication* 72 (2019), pp. 1–8.

[134] S. Som and P. Schniter. 'Compressive imaging using approximate message passing and a Markov-tree prior'. In: *IEEE Transactions on Signal Processing* 60.7 (2012), pp. 3439–3448.

[135] R. Baraniuk, M. Davenport, R. DeVore and M. Wakin. 'A simple proof of the restricted isometry property for random matrices'. In: *Constructive Approximation* 28 (2008), pp. 253–263.

[136] J. Y. Park, H. L. Yap, C. J. Rozell and M. B. Wakin. 'Concentration of measure for block diagonal matrices with applications to compressive signal processing'. In: *IEEE Transactions on Signal Processing* 59.12 (2011), pp. 5859–5875.

[137] H. Rauhut. 'Circulant and Toeplitz matrices in compressed sensing'. In: *arXiv preprint arXiv:0902.4394* (2009).

[138] S. Khanna and C. R. Murthy. 'On the restricted isometry of the columnwise Khatri–Rao product'. In: *IEEE Transactions on Signal Processing* 66.5 (2017), pp. 1170–1183.

[139] N. Koep, A. Behboodi and R. Mathar. 'The restricted isometry property of block diagonal matrices for group-sparse signal recovery'. In: *Applied and Computational Harmonic Analysis* 60 (2022), pp. 333–367.

[140] P. Pal and P. Vaidyanathan. 'Pushing the limits of sparse support recovery using correlation information'. In: *IEEE Transactions on Signal Processing* 63.3 (2014), pp. 711–726.

[141] D. Nion and N. D. Sidiropoulos. 'A PARAFAC-based technique for detection and localization of multiple targets in a MIMO radar system'. In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE. 2009, pp. 2077–2080.

[142] M. Rudelson and R. Vershynin. 'Hanson-Wright inequality and sub-Gaussian concentration'. In: *Electronic Communications in Probability* 18.none (2013), pp. 1–9.

[143] R. Vershynin. *High-dimensional probability: An introduction with applications in data science*. Vol. 47. Cambridge university press, 2018.

[144] G. C. Calafiore and L. El Ghaoui. *Optimization models*. Cambridge university press, 2014.

[145] J. R. Schott. *Matrix analysis for statistics*. John Wiley & Sons, 2016.

[146] I. Fedorov. *Structured learning with scale mixture priors*. Ph.D. Dissertation. 2018.

[147] E. Million. 'The hadamard product'. In: *Course Notes* 3.6 (2007).

[148] D. G. Luenberger, Y. Ye *et al. Linear and nonlinear programming*. Vol. 2. Springer, 1984.

[149] T. T. Cai and A. Zhang. 'Rate-optimal perturbation bounds for singular subspaces with applications to high-dimensional statistics'. In: *Annals of Statistics* 46.1 (2018), pp. 60–89.

[150] R. Vershynin. 'Introduction to the non-asymptotic analysis of random matrices'. In: *arXiv preprint arXiv:1011.3027* (2010).

[151] L. Birgé. 'An alternative point of view on Lepski's method'. In: *Lecture Notes-Monograph Series* (2001), pp. 113–133.

[152] A. Zhang and R. Han. 'Optimal sparse singular value decomposition for high-dimensional high-order data'. In: *Journal of the American Statistical Association* (2019).

[153] T. G. Kolda and B. W. Bader. 'Tensor decompositions and applications'. In: *SIAM Review* 51.3 (2009), pp. 455–500.

[154] Y. He and G. Joseph. 'Total variation sparse Bayesian learning for block sparsity via majorization-minimization'. In: *arXiv preprint arXiv:2602.04623* (2026).

Acknowledgements

The Acknowledgements section is the most important part of a dissertation.

– Yanbin He

Many years later, as I finalize this dissertation, I am compelled to remember that distant moment when I first decided to pursue a Ph.D. Pursuing a Ph.D. degree was perceived then by myself as a painful procedure by default. While I would not deny the inevitable stress, failure, and pain, this journey has nonetheless turned out to be one of much joy and happiness. And this is all due to the lovely people I met.

First of all, I want to express my deep gratitude to my supervisor, Geethu Joseph. Geethu, thank you for giving me this opportunity to do scientific research in SPS from the beginning. After all these years, I cannot even begin to express how much I have benefited from your words. You not only taught me how to conduct research and to effectively communicate my output both orally and in writing, but also made me feel secure whenever things went wrong and helped me to stay positive in the face of failure. Your feedback and comments were always *incredibly* swift, penetrating, and helpful, and it was the interaction with you that made my Ph.D. such a joyful journey. I also liked and enjoyed each and every one of our discussions and conversations, not only the fruitful ones on research, but also the ones about your life and mine. I can feel your passion in both scientific exploration and your life, especially with your family and your adorable kids, which ignited my own passion for life. It was such a great pleasure and privilege to know and work with you as a student. And for the readers who arrive at this place, do not be jealous. I am just lucky.

Second, I also want to thank my promotor, Alle-Jan van der Veen. Alle-Jan, when I met you six years ago in the statistical and signal processing course, I could not imagine that I would work with you one day. You were always there when I needed help, serving as my promotor, annually evaluating my progress, and providing prompt feedback on my dissertation. I also like your stories of the history of our group and what happens in the department and faculty. You should really think about writing a booklet or a memoir based on these stories! They would be valuable, just like your booklet for the course on array processing.

Geert, your great questions and the discussions we shared over these years were always inspiring. I really appreciate you being my committee member, your help during my Master's study, especially recommending me for the cum laude, and also the help when I was looking for a Ph.D. position. Mario, you set a great example for me of how to approach the Ph.D. journey. I apologize for never making good on our promised hotpot dinner. But

the good news is, I know many excellent places now, and I will definitely invite you for a long-overdue hotpot.

I would like to express my sincere gratitude to the members of my doctoral committee: Prof. Emil Björnson, Prof. Richard Heusdens, Prof. Geert Leus, Prof. Chandra Murthy, Prof. Esa Ollila, and Prof. Alexander Yarovoy. I am truly grateful for their time, their thorough review of this manuscript, and their valuable insights during the defense.

I also want to give credit to my friends, who have been a significant and inseparable part of my life. Sofia, I will never forget my very first trip in Louvain-la-Neuve, and thank you for inviting me to join the following journey in Brussels, which initiated such a long-lasting friendship between us. Anurodh, you are the cherry on top (second only to Sofia :)). Aybüke, thank you for making every coffee break so cozy, and for your insightful suggestions on job hunting. Sofia, Anurodh, and Aybüke, thank you for being good friends, for listening to my rambling nonsense, for making me feel at home, and for saving my social life from sinking into a deep void of dullness. Zhonggang and Yingtao, you are for sure the people that I shared most of my spare time with (except for Zhaoyi). It is my privilege to know you and become friends with you. I will never forget our wonderful experience in Luxembourg, Germany, and Texel. I hope that we will have a chance to visit more places, and I can get my license and car soon so you do not have to drive all the time. Peiyuan and Sinian, thank you for inviting me for dinner and board games, for the fantastic cakes and desserts, and for all the cherishable moments together. Metin, your philosophical attitude towards life has always been attractive to me, and thank you for sharing your secret handshake with me. Ellen, thank you for all the gossips and random chats between us, which truly helped me relax my brain from equations and codes. Ids, you are my office mate from the beginning, thank you for making this office a nice place. Peiyuan, Ellen, and Ids, we should definitely continue our office meeting once in a while, even after we all leave. Maosheng, my longest friend in The Netherlands, you are a true gem. Changheng, it was a pleasure to discuss papers, equations, and math problems with you, and we indeed make a good team in PUBG. Thank you Yujie and Liming, for our great journey in Norway. And it is a great pleasure to collaborate with you, Yujie. Thank you, Cristian, for all the nice drinks and Italian food. I would like to thank my colleagues Costas, Seline, Alberto, Yanbo, Ruben, Didem, Shuoyan, Giovanni, Jordi, Hanie, Mostafa, Bishwadeep, Hamed, Avijit, Chen, Ali, Pascal, Rupam, TingYing, Gabriel, Yongsheng, Shao-Hsuan, Admitos, Ankush, Carlo, Varun, Çansu, Edoardo, Nan, Sen, Kaiwen ... for making this place so *gezellig*. And I also want to thank senior members of SPS, Richard, Gerard, Raj, Justin, Bori, and Jac for the company at lunch. For the friends outside Delft, I would like to mention Jiyue, Chi, Xiaoyao, Kunlei, Jiayi, Weiwen, and Bokai, for bringing me cherished memories in Europe. I would like to mention the members of my football team Real Boys: ZeNanGei, NiuGe, YueYue, Jiayu, Leizi, ZhouGei, and ZheRui. I would like to mention my friends from the university, Qianyi, Peihang, Shikao, and Zhiyu. I thank Laura for organizing many memorable events, and Antoon for setting up our computation server. The preparation of the computational framework, tables, and figures throughout this dissertation should be largely attributed to your efforts. I thank the Department of Microelectronics of Delft University of Technology for the generous scholarship for my Master's study that fulfilled my dream of studying abroad.

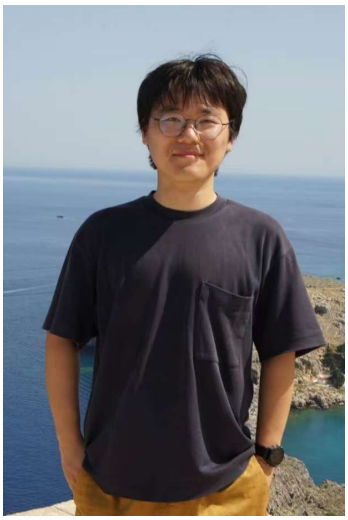
I want to thank SCBOY, who provided me with endless, hilarious moments of joy. I also

want to thank Lionel Messi. I have been a fan of yours for seventeen years, and you have been supporting me emotionally for that entire time. People talk about your talent, but I know that it is the diligence and determination that make your career brilliant. Being a good and honest person should always be prioritized before any other aspect of life.

Last but not least, I want to thank my parents for their endless support and love, and for making a true sanctuary for me. During my Ph.D. journey, I was fortunate to meet Zhaoyi. Zhaoyi, thank you for appearing in my life, for being my best friend, and for all your patience and love. You complete me. Life is not easy, but with you, I have the courage and confidence to carry on fearlessly.

Curriculum Vitæ

Yanbin He (贺彦斌) was born in Xianyang (咸阳), Shaanxi (陕西), China (中国). He received his Bachelor of Engineering degree in Electronic and Information Engineering from Beihang University (北京航空航天大学) in 2019. Following his undergraduate studies, he moved to The Netherlands (*Nederland*) to pursue a master's degree at Delft University of Technology (*Technische Universiteit Delft*). In 2021, he received his M.Sc. degree *cum laude* in Electrical Engineering from the Circuits and Systems (now known as Signal Processing Systems (SPS)) section. Since SPS is such a nice place, he continued his academic journey as a Ph.D. candidate in the same section, under the supervision of dr. Geethu Joseph and prof. dr. ir. Alle-Jan van der Veen, where he works on signal processing and structured compressed sensing.



List of Publications

Journal papers:

1. Y. He and G. Joseph. 'Bayesian algorithms for Kronecker-structured sparse vector recovery with application to IRS-MIMO channel estimation'. In: *IEEE Transactions on Signal Processing* 73 (2025), pp. 142–157

2. Y. He and G. Joseph. 'Efficient off-grid Bayesian parameter estimation for Kronecker structured signals'. In: *IEEE Transactions on Signal Processing* 73 (2025), pp. 2616–2630

Conference papers:

1. Y. He and G. Joseph. 'Structure-aware sparse Bayesian learning-based channel estimation for intelligent reflecting surface-aided MIMO'. In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*. 2023, pp. 1–5

2. Y. He and G. Joseph. 'Kronecker-structured sparse vector recovery with application to IRS-MIMO channel estimation'. In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*. 2025, pp. 1–5

3. Y. He and G. Joseph. 'A hierarchical view of structured sparsity in Kronecker compressive sensing'. In: *Proceedings of the European Signal Processing Conference*. 2025, pp. 1–5

4. Y. He and G. Joseph. 'Hierarchical multi-stage recovery framework for Kronecker compressed sensing'. In: *Proceedings of the International Conference on Learning Representations*. 2026

Preprint:

1. Y. He and G. Joseph. 'Total variation sparse Bayesian learning for block sparsity via majorization-minimization'. In: *arXiv preprint arXiv:2602.04623* (2026)