

ISLET: Fast and Optimal Low-Rank Tensor Regression via Importance Sketching*

Anru R. Zhang[†], Yuetian Luo[†], Garvesh Raskutti[†], and Ming Yuan[‡]

Abstract. In this paper, we develop a novel procedure for low-rank tensor regression, namely *Importance Sketching Low-rank Estimation for Tensors* (ISLET). The central idea behind ISLET is *importance sketching*, i.e., carefully designed sketches based on both the responses and low-dimensional structure of the parameter of interest. We show that the proposed method is sharply minimax optimal in terms of the mean-squared error under low-rank Tucker assumptions and under the randomized Gaussian ensemble design. In addition, if a tensor is low-rank with group sparsity, our procedure also achieves minimax optimality. Further, we show through numerical study that ISLET achieves comparable or better mean-squared error performance to existing state-of-the-art methods while having substantial storage and run-time advantages including capabilities for parallel and distributed computing. In particular, our procedure performs reliable estimation with tensors of dimension $p = O(10^8)$ and is 1 or 2 orders of magnitude faster than baseline methods.

Key words. dimension reduction, high-order orthogonal iteration, minimax optimality, sketching, tensor regression

AMS subject classifications. 15A69, 62H12, 65A99

DOI. 10.1137/19M126476X

1. Introduction. The past decades have seen a large body of work on tensors or multiway arrays [63, 100, 30, 67]. Tensors arise in numerous applications involving multiway data (e.g., brain imaging [132], hyperspectral imaging [70], or recommender system design [11]). In addition, tensor methods have been applied to many problems in statistics and machine learning where the observations are not necessarily tensors, such as topic and latent variable models [2], additive index models [5], and high-order interaction pursuit [53], among others. In many of these settings, the tensor of interest is *high-dimensional* in that the ambient dimension, i.e., the dimension of the target parameter, is substantially larger than the sample size. However, in practice, the tensor parameter often has intrinsic dimension-reduced structure, such as low-rankness and sparsity [63, 105, 114], which makes inference possible. How to exploit such structure for tensors poses new *statistical* and *computational challenges* [96].

From a statistical perspective, a key question is how many samples are required to learn the suitable dimension-reduced structure and what the optimal mean-squared error rates are.

*Received by the editors May 29, 2019; accepted for publication (in revised form) February 25, 2020; published electronically June 3, 2020.

<https://doi.org/10.1137/19M126476X>

Funding: The work of the first and second authors was partially by NSF grants DMS-1811868 and CAREER-1944904 and NIH grant R01 GM131399. The work of the third author was partially supported by NSF grant DMS-1811767, ARO grant W911NF-17-1-0357, and NGA grant HM0476-17-1-2003. The work of the fourth author was partially supported by NSF grants DMS-1265202 and DMS-1721584.

[†]Department of Statistics, University of Wisconsin-Madison, Madison, WI 53706 (anruzhang@stat.wisc.edu, yluo86@wisc.edu, raskutti@stat.wisc.edu).

[‡]Department of Statistics, Columbia University, New York, NY 10027 (my2550@columbia.edu).

Prior work has developed various tensor-based methods with theoretical guarantees based on regularization approaches [68, 84, 96, 110], the spectral method and projected gradient descent [27], alternating gradient descent [69, 106, 132], stochastic gradient descent [45], and power iteration methods [2]. However, a number of these methods are not statistically optimal. Furthermore, some of these methods rely on evaluation of a full gradient, which is typically costly in the high-dimensional setting. This leads to computational challenges including both the *storage* of tensors and *run time* of the algorithm.

From a computational perspective, one approach to addressing both the storage and run-time challenge is *randomized sketching*. Sketching methods have been widely studied (see, e.g., [3, 4, 8, 14, 31, 32, 33, 35, 36, 54, 76, 85, 90, 92, 93, 95, 103, 104, 107, 111, 117, 118]). Many of these prior works on matrix or tensor sketching mainly focused on relative approximation error [14, 32, 85, 95] after randomized sketching which either may not yield optimal mean-squared error rates under statistical settings [95] or requires multiple sketching iterations [93, 94].

In this article, we address both computational and statistical challenges by developing a novel sketching-based estimating procedure for tensor regression. The proposed procedure is provably fast and sharply minimax optimal in terms of mean-squared error under randomized Gaussian design. The central idea lies in constructing specifically designed structural sketches, namely *importance sketching*. In contrast with randomized sketching methods, importance sketching utilizes both the response and structure of the target tensor parameter and reduces the dimension of parameters (i.e., the number of columns) instead of samples (i.e., the number of rows), which leads to statistical optimality while maintaining the computational advantages of many randomized sketching methods. See more comparison between importance sketching in this work and sketching in prior literature in section 1.3.

1.1. Problem statement. Specifically, we focus on the following low-rank tensor regression model:

$$(1.1) \quad y_j = \langle \mathcal{X}_j, \mathcal{A} \rangle + \varepsilon_j, \quad j = 1, \dots, n,$$

where y_j and ε_j are responses and observation noise, respectively; $\{\mathcal{X}_j\}_{j=1}^n$ are tensor covariates with randomized design; and $\mathcal{A} \in \mathbb{R}^{p_1 \times \dots \times p_d}$ is the order- d tensor with parameters aligned in d ways. Here $\langle \cdot, \cdot \rangle$ stands for the usual vectorized inner product. The goal is to recover \mathcal{A} based on observations $\{y_j, \mathcal{X}_j\}_{j=1}^n$. In particular, when $d = 2$, this becomes a low-rank matrix regression problem, which has been widely studied in recent years [23, 64, 97]. The main focus of this paper is solving the underdetermined equation system, where the sample size n is much smaller than the number of coefficients $\prod_{i=1}^d p_i$. This is because many applications belong to this regime. In particular, in the real data example to be discussed later, one MRI image is 121-by-145-by-121, which includes 2,122,945 parameters. Typically we can collect far fewer MRI images in practice.

The general regression model (1.1) includes specific problem instances with different choices of design \mathcal{X} . Examples include matrix/tensor regression with general random or deterministic design [27, 71, 96, 132], matrix trace regression [6, 23, 41, 43, 64, 97], and matrix sparse recovery [123]. Another example is *matrix/tensor recovery via rank-1 projections* [18, 28, 53], which arise by setting $\mathcal{X}_j = \mathbf{u}_j \circ \mathbf{v}_j \circ \mathbf{w}_j$, where $\mathbf{u}_j, \mathbf{v}_j, \mathbf{w}_j$ are random vectors and “ \circ ” represents the outer product, which includes phase retrieval [17, 21] as a special

case. The very popular matrix/tensor completion example [25, 72, 83, 119, 120, 125] arises by setting $\mathcal{X}_j = (\mathbf{e}_{a_j} \circ \mathbf{e}_{b_j} \circ \mathbf{e}_{c_j})$, where \mathbf{e}_j is the j th canonical vector and $\{a_j, b_j, c_j\}_{j=1}^n$ are randomly selected integers from $\{1, \dots, p_1\} \times \{1, \dots, p_2\} \times \{1, \dots, p_3\}$. Specific applications of this low-rank tensor regression model include neuroimaging analysis [50, 69, 132], longitudinal relational data analysis [56], 3D imaging processing [51], etc.

For convenience of presentation, we specialize the discussions on order-3 tensors later, while the results can be extended to the general order- d tensors. In the modern high-dimensional setting, a variety of matrix/tensor data satisfy intrinsic structural assumptions, such as low-rankness [114] or sparsity [132], which makes the accurate estimation of \mathcal{A} possible even if the sample size n is smaller than the number of coefficients in the target tensor \mathcal{A} . We thus focus on the low Tucker rank (r_1, r_2, r_3) tensor \mathcal{A} with the following Tucker decomposition [113]:

$$(1.2) \quad \mathcal{A} = [\![\mathcal{S}; \mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3]\!] := \mathcal{S} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3,$$

where \mathcal{S} is an r_1 -by- r_2 -by- r_3 core tensor and \mathbf{U}_k is a p_k -by- r_k matrix with orthonormal columns for $k = 1, 2, 3$. The rigorous definition of Tucker rank of a tensor and more discussions on tensor algebra are postponed to section 2.1. In addition, the canonical polyadic (CP) low-rank tensors have also been widely considered in recent literature [53, 54, 106, 132]. Since any CP-rank- r tensor $\mathcal{A} = \sum_{i=1}^r \lambda_i \mathbf{a}_i \circ \mathbf{b}_i \circ \mathbf{c}_i$ has the Tucker decomposition $\mathcal{A} = [\![\mathcal{L}; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$, where \mathcal{L} is the r -by- r -by- r diagonal tensor with diagonal entries $\lambda_1, \dots, \lambda_r$, $\mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_r]$, and likewise for \mathbf{B}, \mathbf{C} [63], our results naturally adapt to low CP-rank tensor regression. Also, with a slight abuse of notation, we will refer to low-rank and low Tucker rank interchangeably throughout the paper. Moreover, we also consider a sparse setting where there may exist a subset of modes, say $J_s \subseteq \{1, 2, 3\}$, such that \mathcal{A} is sparse along these modes, i.e.,

$$(1.3) \quad \mathcal{A} = [\![\mathcal{S}; \mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3]\!], \quad \|\mathbf{U}_k\|_0 = \sum_{i=1}^{p_k} 1_{\{(\mathbf{U}_k)_{[i,:]} \neq 0\}} \leq s_k, \quad k \in J_s.$$

1.2. Our contributions. We make the following major contributions to low-rank tensor regression in this article. First, we introduce the main algorithm—*Importance Sketching Low-rank Estimation for Tensors* (ISLET). Our algorithm has three steps: (i) first we use the tensor technique high-order orthogonal iteration (HOOI) [34] or sparse tensor alternating thresholding - singular value decomposition (STAT-SVD) [127] to determine the importance sketching directions. Here HOOI and STAT-SVD are regular and sparse tensor low-rank decomposition methods, respectively, whose explanations are postponed to sections 2.2 and 2.3; (ii) using the sketching directions from the first step, we perform importance sketching and then evaluate the dimension-reduced regression using the sketched tensors/matrices (to incorporate sparsity, we add a group-sparsity regularizer); (iii) we construct the final tensor estimator using the sketched components. Although the focus of this work is on low-rank tensor regression, we point out that our three-step procedure applies to general high-dimensional statistics problems with low-dimensional structure, provided that we can find a suitable projection operator in step (i) and inverse projection operator in step (iii).

One of the main advantages of ISLET is the scalability of the algorithm. The proposed procedure is computationally efficient due to the dimension reduction by importance sketchings. Most importantly, ISLET only require access to the full data twice, which significantly

saves run time for large-scale settings when it is not possible to store all samples into the core memory. We also show that our algorithm can be naturally distributed across multiple machines that can significantly reduce computation time.

Second, we prove a deterministic oracle inequality for the ISLET procedure under the low-Tucker-rank assumption and general noise and design (Theorems 2 and 3). We additionally show that ISLET achieves the optimal mean-squared error (with the optimal constant for nonsparse ISLET) under randomized Gaussian design (Theorems 4, 5, 6, and 7). The following informal statement summarizes two of the main results of the article.

Theorem 1 (ISLET for tensor regression: informal). *Consider the regular tensor regression problem with Gaussian ensemble design, where \mathcal{A} is Tucker rank- (r_1, r_2, r_3) , \mathcal{X}_j has i.i.d. standard normal entries, $\varepsilon_j \stackrel{iid}{\sim} N(0, \sigma^2)$, and $\varepsilon_j, \mathcal{X}_j$ are independent:*

- (a) *Under regularity conditions, ISLET achieves the following optimal rate of convergence with the matching constant:*

$$\mathbb{E} \left\| \widehat{\mathcal{A}} - \mathcal{A} \right\|_{\text{HS}}^2 = (1 + o(1)) \frac{m\sigma^2}{n},$$

where $m = r_1 r_2 r_3 + r_1(p_1 - r_1) + r_2(p_2 - r_2) + r_3(p_3 - r_3)$ is exactly the degree of freedom of all Tucker rank- (r_1, r_2, r_3) tensors in $\mathbb{R}^{p_1 \times p_2 \times p_3}$ and $\|\cdot\|_{\text{HS}}$ is the Hilbert–Schmidt norm to be defined in section 2.1.

- (b) *If, in addition, (1.3) holds with sparsity level s_k , then under regularity conditions, ISLET achieves the following optimal rate of convergence:*

$$\mathbb{E} \left\| \widehat{\mathcal{A}} - \mathcal{A} \right\|_{\text{HS}}^2 \asymp \frac{m_s \sigma^2}{n},$$

where $m_s = r_1 r_2 r_3 + \sum_{k \in J_s} s_k (r_k + \log(p_k/s_k)) + \sum_{k \notin J_s} p_k r_k$ and “ \asymp ” denotes the asymptotic equivalence between two number series (see a more formal definition in section 2.1).

To the best of our knowledge, we are the first to develop the matching-constant optimal rate results for regular tensor regression under randomized Gaussian ensemble design, even for the low-rank matrix recovery case since it is not clear whether prior approaches (e.g., nuclear norm minimization) achieve sharp constants. We are also the first to develop the optimal rate results for tensor regression with sparsity condition (1.3).

Third, proving the optimal mean-squared error bound presents a number of technical challenges and we introduce novel proof ideas to overcome these difficulties. In particular, one major difficulty lies in the analysis of reduced-dimensional regressions (see (2.4) in section 2) since we analyze sketched regression models. To this end, we introduce partial linear models for these reduced-dimensional regressions from which we develop estimation error upper bounds.

The final and most important computational contribution is to display through numerical studies the advantages of our ISLET algorithms. Compared to state-of-the-art tensor estimation algorithms including nonconvex projected gradient descent (PGD) [27], Tucker regression [132], and convex regularization [109], we show that our ISLET algorithm achieves comparable statistical performance with substantially faster computation. In particular, the run time is 1–3 orders of magnitude faster than existing methods. In the most prominent example, our

ISLET procedure can efficiently solve the ultrahigh-dimensional tensor regression with covariates of 7.68 terabytes. For the order-2 case, i.e., low-rank matrix regression, our simulation studies show that ISLET outperforms the classic nuclear norm minimization estimator. We also provide a real data application where we study the association between the attention deficit hyperactivity disorder disease and the high-dimensional MRI image tensors. We show that the proposed procedure provides significantly better prediction performance in much less time compared to state-of-the-art methods.

1.3. Related literature. Our work is related to a broad range of literature varying from a number of communities including scientific computing, computer science, signal processing, applied mathematics, and statistics. Here we make an attempt to discuss existing results from these various communities; however, we do not claim that our literature survey is exhaustive.

Large-scale linear systems where the solution admits a low-rank tensor structure commonly arise after discretizing high-dimensional partial differential equations [57, 58, 74], and various methods have been proposed. For example, the authors of [12] developed algebraic and Gauss–Newton methods to solve the linear system with a CP low-rank tensor solution. The authors of [7, 10] proposed iterative projection methods to solve large-scale linear systems with Kronecker-product-type design matrices. The authors of [46] introduced a greedy approach. The authors of [65, 66] considered Riemannian optimization methods and tensor Krylov subspace methods, respectively. The readers are referred to [49] for a recent survey. Different from these works, our proposed ISLET is a one-step procedure that only involves solving a simple least squares regression after performing dimension reduction on covariates by importance sketching (see Steps 1 and 2 in section 2.2). Moreover, many prior works mainly focused on computational aspects of their proposed methods [7, 13, 40, 46, 49], while we show that ISLET is not only computationally efficient (see more discussion and comparison on computation complexity in the *Computation and implementation part* of section 2.2) but also has optimal theoretical guarantees in terms of mean-squared error under the statistical setting.

In addition, sketching methods play an important role in computation acceleration and have been widely considered in previous literature. For example, the authors of [32, 82, 85] provided accurate approximation algorithms based on sketching with novel embedding matrices, where the run time is proportional to the number of the nonzero entries of the input matrix. Sketching methods have also been studied in robust ℓ_1 low-rank matrix approximation [79, 80, 81, 103, 131], general ℓ_p low-rank matrix approximation [8, 29], low-rank tensor approximation [104], etc. In the regression context, the sketching method has been considered for the least squares regression [32, 35, 85, 94, 95], ℓ_p regression [32, 82, 85], Kronecker product regression [35], ridge regression [3, 116], regularized kernel regression [20, 130], etc. Various types of random sketching matrices have been developed, including random sub-Gaussian [94], random sampling [37, 38], CountSketch [26, 31], Sparse Johnson–Lindenstrauss transformation [62], among many others. The readers are also referred to survey papers on sketching by Mahoney [76] and Woodruff [118]. The proposed method in this paper is different from these previous works in various aspects. First, many randomized sketching methods in the literature focus on relative approximation error [76, 118] and the sketching matrices are constructed only based on covariates [37, 38, 62, 94, 95]. In contrast, we explicitly construct “supervised”

sketching matrices based on both the response y_j and covariates \mathbf{X}_j and obtain optimal bounds in mean-squared error under the statistical setting. Second, essentially speaking, our proposed importance sketching scheme reduces the number of columns (parameters) instead of the number of rows (samples) in the linear equation system. Third, different from the sketching on an overdetermined system of least squares [32, 35, 85, 94, 95], we mainly focus on the high-dimensional setting where the number of samples can be significantly smaller than the number of coefficients.

1.4. Organization. In section 2.1, we introduce important notation; then we present our ISLET procedure under nonsparse and sparse settings in sections 2.2 and 2.3, respectively, and illustrate the procedure from a sketching perspective in section 2.4. In section 3, we provide general theoretical guarantees for our procedure which make no assumptions on the design or the noise distribution; in section 4, we specialize our bounds to tensor regression with low Tucker rank and assume the design is independent Gaussian; a simulation study showing the substantial computational benefits of our algorithm is provided in section 5. Additional notation, discussion on general-order ISLET, simulation results, an application to attention deficit hyperactivity disorder (ADHD) MRI imaging data analysis, and all technical proofs are provided in the supplementary materials [128], linked from the main article webpage.

2. Our procedure: ISLET. Here we introduce the general procedure of Importance Sketching Low-rank Estimation for Tensors (ISLET). Although for ease of presentation we will focus on order-3 tensors, the procedure for the general order- d case can also be treated. Details of matrices and tensors greater than order 3 are provided in section SM3 of the supplementary materials [128].

2.1. Notation and preliminaries. The following notation will be used throughout this article. Additional definitions can be found in section SM1 in the supplementary materials. Lowercase letters (e.g., a, b), lowercase boldface letters (e.g., \mathbf{u}, \mathbf{v}), uppercase boldface letters (e.g., \mathbf{U}, \mathbf{V}), and boldface calligraphic letters (e.g., \mathcal{A}, \mathcal{X}) are used to denote scalars, vectors, matrices, and order-3-or-higher tensors, respectively. For simplicity, we denote \mathcal{X}_j as the tensor indexed by j in a sequence of tensors $\{\mathcal{X}_j\}$. For any two series of numbers, say $\{a_i\}$ and $\{b_i\}$, denote $a \asymp b$ if there exist uniform constants $c, C > 0$ such that $ca_i \leq b_i \leq Ca_i$ for all i and $a = \Omega(b)$ if there exists uniform constant $c > 0$ such that $a_i \geq cb_i$ for all i . We use bracket subscripts to denote subvectors, submatrices, and subtensors. For example, $\mathbf{v}_{[2:r]}$ is the vector with the 2nd to r th entries of \mathbf{v} ; $\mathbf{D}_{[i_1, i_2]}$ is the entry of \mathbf{D} on the i_1 th row and i_2 th column; $\mathbf{D}_{[(r+1):p_1, :]}$ contains the $(r+1)$ th to the p_1 th rows of \mathbf{D} ; and $\mathcal{A}_{[1:s_1, 1:s_2, 1:s_3]}$ is the s_1 -by- s_2 -by- s_3 subtensor of \mathcal{A} with index set $\{(i_1, i_2, i_3) : 1 \leq i_1 \leq s_1, 1 \leq i_2 \leq s_2, 1 \leq i_3 \leq s_3\}$. For any vector $\mathbf{v} \in \mathbb{R}^{p_1}$, define its ℓ_q norm as $\|\mathbf{v}\|_q = (\sum_i |v_i|^q)^{1/q}$. For any matrix $\mathbf{D} \in \mathbb{R}^{p_1 \times p_2}$, let $\sigma_k(\mathbf{D})$ be the k th singular value of \mathbf{D} . In particular, the least nontrivial singular value of \mathbf{D} , defined as $\sigma_{\min}(\mathbf{D}) = \sigma_{p_1 \wedge p_2}(\mathbf{D})$, will be extensively used in later analysis. We also denote $\text{SVD}_r(\mathbf{D}) = [\mathbf{u}_1 \cdots \mathbf{u}_r]$ and $\text{QR}(\mathbf{D})$ as the subspace composed of the leading r left singular vectors and the Q part of the QR orthogonalization of \mathbf{D} , respectively. The matrix Frobenius and spectral norms are defined as $\|\mathbf{D}\|_F = (\sum_{i_1, i_2} \mathbf{D}_{[i_1, i_2]}^2)^{1/2} = (\sum_{i=1}^{p_1 \wedge p_2} \sigma_i^2(\mathbf{D}))^{1/2}$ and $\|\mathbf{D}\| = \max_{\mathbf{u} \in \mathbb{R}^{p_2}} \|\mathbf{D}\mathbf{u}\|_2 / \|\mathbf{u}\|_2 = \sigma_1(\mathbf{D})$. In addition, \mathbf{I}_r represents the r -by- r identity matrix. Let $\mathbb{O}_{p,r} = \{\mathbf{U} : \mathbf{U}^\top \mathbf{U} = \mathbf{I}_r\}$

be the set of all p -by- r matrices with orthonormal columns. For any $\mathbf{U} \in \mathbb{O}_{p,r}$, $P_{\mathbf{U}} = \mathbf{U}\mathbf{U}^\top$ represents the projection matrix onto the column space of \mathbf{U} ; we also use $\mathbf{U}_\perp \in \mathbb{O}_{p,p-r}$ to represent the orthonormal complement of \mathbf{U} . For any event A , let $\mathbb{P}(A)$ be the probability that A occurs.

For any matrix $\mathbf{D} \in \mathbb{R}^{p_1 \times p_2}$ and order- d tensor $\mathcal{A} \in \mathbb{R}^{p_1 \times \dots \times p_d}$, let $\text{vec}(\mathbf{D})$ and $\text{vec}(\mathcal{A})$ be the vectorization of \mathbf{D} and \mathcal{A} , respectively. The matricization $\mathcal{M}(\cdot)$ is the operation that unfolds or flattens the order- d tensor $\mathcal{A} \in \mathbb{R}^{p_1 \times \dots \times p_d}$ into the matrix $\mathcal{M}_k(\mathcal{A}) \in \mathbb{R}^{p_k \times \prod_{j \neq k} p_j}$ for $k = 1, \dots, d$. Since the formal entrywise definitions of matricization and vectorization are rather tedious, we leave them to section SM1 in the supplementary materials [128]. The Hilbert–Schmidt norm is defined as $\|\mathcal{A}\|_{\text{HS}} = (\sum_{i_1, \dots, i_d} \mathcal{A}_{[i_1, \dots, i_d]}^2)^{1/2}$. An order- d tensor is rank-one if it can be written as the outer product of d nonzero vectors. The CP rank of any tensor \mathcal{A} is defined as the minimal number r such that \mathcal{A} can be decomposed as $\mathcal{A} = \sum_{i=1}^r \mathcal{B}_i$ for rank-1 tensors \mathcal{B}_i . The Tucker rank (or multilinear rank) of a tensor \mathcal{A} is defined as a d -tuple (r_1, \dots, r_d) , where $r_k = \text{rank}(\mathcal{M}_k(\mathcal{A}))$. The k -mode product of $\mathcal{A} \in \mathbb{R}^{p_1 \times \dots \times p_d}$ with a matrix $\mathbf{U} \in \mathbb{R}^{p_k \times r_k}$ is denoted by $\mathcal{A} \times_k \mathbf{U}$ and is of size $p_1 \times \dots \times p_{k-1} \times r_k \times p_{k+1} \times \dots \times p_d$, such that

$$(\mathcal{A} \times_k \mathbf{U})_{[i_1, \dots, i_{k-1}, j, i_{k+1}, \dots, i_d]} = \sum_{i_k=1}^{p_k} \mathcal{A}_{[i_1, i_2, \dots, i_d]} \mathbf{U}_{[i_k, j]}.$$

For convenience of presentation, all mode indices $(\cdot)_k$ of an order-3 tensor are in the sense of modulo-3, e.g., $r_1 = r_4$, $s_2 = s_5$, $p_0 = p_3$, $\mathcal{X} \times_4 \mathbf{U}_4 = \mathcal{X} \times_1 \mathbf{U}_1$.

For any matrices $\mathbf{U} \in \mathbb{R}^{p_1 \times p_2}$ and $\mathbf{V} \in \mathbb{R}^{m_1 \times m_2}$, let

$$\mathbf{U} \otimes \mathbf{V} = \begin{bmatrix} \mathbf{U}_{[1,1]} \cdot \mathbf{V} & \dots & \mathbf{U}_{[1,p_2]} \cdot \mathbf{V} \\ \vdots & & \vdots \\ \mathbf{U}_{[p_1,1]} \cdot \mathbf{V} & \dots & \mathbf{U}_{[p_1,p_2]} \cdot \mathbf{V} \end{bmatrix} \in \mathbb{R}^{(p_1 m_1) \times (p_2 m_2)}$$

be the Kronecker product. Some intrinsic identities among Kronecker product, vectorization, and matricization, which will be used later in this paper, are summarized in Lemma 1 in the supplementary materials [128]. Readers can refer to [63] for a more comprehensive introduction to tensor algebra. Finally, we use C, C_1, C_2, c and other variations to represent the large and small constants, whose actual value may vary from line to line.

2.2. Regular low-rank tensor recovery. We first consider the tensor regression model (1.1), where \mathcal{A} is low-rank (1.2) without sparsity assumptions. The proposed algorithm of ISLET is divided into three steps, and a pictorial illustration is provided in Figures 1–3 for readers' better understanding. The pseudocode is provided in Algorithm 2.1.

Step 1 (Probing importance sketching directions) We first probe the importance sketching directions. When the covariates satisfy $\mathbb{E} \text{vec}(\mathcal{X}_j) \text{vec}(\mathcal{X}_j)^\top = \mathbf{I}_{p_1 p_2 p_3}$, we evaluate

$$(2.1) \quad \tilde{\mathcal{A}} = \frac{1}{n} \sum_{j=1}^n y_j \mathcal{X}_j.$$

$\tilde{\mathcal{A}}$ is essentially the covariance tensor between y and \mathcal{X} . Since $\mathcal{A} = [\mathcal{S}; \mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3]$ has low Tucker rank, we perform the high-order orthogonal iterations (HOOI) on

$\tilde{\mathcal{A}}$ to obtain $\tilde{\mathbf{U}}_k \in \mathbb{O}_{p_k, r_k}$, $k = 1, 2, 3$, as initial estimates for \mathbf{U}_k . Here HOOI is a classic method for tensor decomposition that can be traced back to De Lathauwer, De Moor, and Vandewalle [34]. The central idea of HOOI is the power iterated singular value thresholding. Then the outcome of HOOI $\{\tilde{\mathbf{U}}_k\}_{k=1}^3$ yields the following low-rank approximation for \mathcal{A} :

$$(2.2) \quad \mathcal{A} \approx [\tilde{\mathcal{S}}; \tilde{\mathbf{U}}_1, \tilde{\mathbf{U}}_2, \tilde{\mathbf{U}}_3], \quad \text{where } \tilde{\mathcal{S}} = [\tilde{\mathcal{A}}; \tilde{\mathbf{U}}_1^\top, \tilde{\mathbf{U}}_2^\top, \tilde{\mathbf{U}}_3^\top] \in \mathbb{R}^{r_1 \times r_2 \times r_3}.$$

We further evaluate

$$\tilde{\mathbf{V}}_k := \text{QR}(\mathcal{M}_k^\top(\tilde{\mathcal{S}})) \in \mathbb{O}_{r_{k+1}r_{k+2}, r_k}, \quad k = 1, 2, 3.$$

$\{\tilde{\mathbf{U}}_k, \tilde{\mathbf{V}}_k\}_{k=1}^3$ obtained here are regarded as the *importance sketching directions*. As we will further illustrate in section 3.1, the combinations of $\tilde{\mathbf{U}}_k$ and $\tilde{\mathbf{V}}_k$ provide approximations for singular subspaces of $\mathcal{M}_k(\mathcal{A})$.

Step 2 (Linear regression on sketched covariates) Next, we perform sketching to reduce the dimension of the original regression model (1.1). To be specific, we project the original high-dimensional covariates onto the dimension-reduced subspace that is important in the covariance between y and \mathcal{X} and construct the following *importance sketching covariates*:

$$(2.3) \quad \begin{aligned} \tilde{\mathbf{X}} &= [\tilde{\mathbf{X}}_{\mathcal{B}} \quad \tilde{\mathbf{X}}_{\mathbf{D}_1} \quad \tilde{\mathbf{X}}_{\mathbf{D}_2} \quad \tilde{\mathbf{X}}_{\mathbf{D}_3}] \in \mathbb{R}^{n \times m}, \\ \tilde{\mathbf{X}}_{\mathcal{B}} &\in \mathbb{R}^{n \times m_{\mathcal{B}}}, \quad (\tilde{\mathbf{X}}_{\mathcal{B}})_{[i,:]} = \text{vec}(\mathcal{X}_i \times_1 \tilde{\mathbf{U}}_1^\top \times_2 \tilde{\mathbf{U}}_2^\top \times_3 \tilde{\mathbf{U}}_3^\top), \\ \tilde{\mathbf{X}}_{\mathbf{D}_k} &\in \mathbb{R}^{n \times m_{\mathbf{D}_k}}, \quad (\tilde{\mathbf{X}}_{\mathbf{D}_k})_{[i,:]} = \text{vec}(\tilde{\mathbf{U}}_{k+1}^\top \mathcal{M}_k(\mathcal{X}_i \times_{k+1} \tilde{\mathbf{U}}_{k+1}^\top \times_{k+2} \tilde{\mathbf{U}}_{k+2}^\top) \tilde{\mathbf{V}}_k), \end{aligned}$$

where $m_{\mathcal{B}} = r_1 r_2 r_3$, $m_{\mathbf{D}_k} = (p_k - r_k) r_k$, $k = 1, 2, 3$, and $m = m_{\mathcal{B}} + m_{\mathbf{D}_1} + m_{\mathbf{D}_2} + m_{\mathbf{D}_3}$. Then we evaluate the least-squares estimator of the submodel with importance sketching covariates $\tilde{\mathbf{X}}$,

$$(2.4) \quad \hat{\boldsymbol{\gamma}} = \arg \min_{\boldsymbol{\gamma} \in \mathbb{R}^m} \|y - \tilde{\mathbf{X}}\boldsymbol{\gamma}\|_2^2.$$

The dimension of sketching covariate regression (2.4) is m , which is significantly smaller than the dimension of the original tensor regression model, $p_1 p_2 p_3$. Consequently, the computational cost can be significantly reduced.

Step 3 (Assembling the final estimate) Then $\hat{\boldsymbol{\gamma}}$ is divided into four segments according to the blockwise structure of $\tilde{\mathbf{X}} = [\tilde{\mathbf{X}}_{\mathcal{B}}, \tilde{\mathbf{X}}_{\mathbf{D}_1}, \tilde{\mathbf{X}}_{\mathbf{D}_2}, \tilde{\mathbf{X}}_{\mathbf{D}_3}]$,

$$(2.5) \quad \begin{aligned} \text{vec}(\hat{\mathcal{B}}) &= \hat{\boldsymbol{\gamma}}_{[1:m_{\mathcal{B}}]}, \\ \text{vec}(\hat{\mathbf{D}}_1) &= \hat{\boldsymbol{\gamma}}_{[(m_{\mathcal{B}}+1):(m_{\mathcal{B}}+m_{\mathbf{D}_1})]}, \\ \text{vec}(\hat{\mathbf{D}}_2) &= \hat{\boldsymbol{\gamma}}_{[(m_{\mathcal{B}}+m_{\mathbf{D}_1}+1):(m_{\mathcal{B}}+m_{\mathbf{D}_1}+m_{\mathbf{D}_2})]}, \\ \text{vec}(\hat{\mathbf{D}}_3) &= \hat{\boldsymbol{\gamma}}_{[(m_{\mathcal{B}}+m_{\mathbf{D}_1}+m_{\mathbf{D}_2}+1):(m_{\mathcal{B}}+m_{\mathbf{D}_1}+m_{\mathbf{D}_2}+m_{\mathbf{D}_3})]}. \end{aligned}$$

Finally, we construct the regression estimator $\widehat{\mathcal{A}}$ for the original problem (1.1) using the regression estimator $\widehat{\gamma}$ for the submodel (2.5): let $\widehat{\mathbf{B}}_k = \mathcal{M}_k(\widehat{\mathcal{B}})$, and calculate

$$(2.6) \quad \widehat{\mathbf{L}}_k = \left(\widetilde{\mathbf{U}}_k \widehat{\mathbf{B}}_k \widetilde{\mathbf{V}}_k + \widetilde{\mathbf{U}}_{k\perp} \widehat{\mathbf{D}}_k \right) \left(\widehat{\mathbf{B}}_k \widetilde{\mathbf{V}}_k \right)^{-1}, \quad k = 1, 2, 3, \quad \widehat{\mathcal{A}} = [\widehat{\mathcal{B}}; \widehat{\mathbf{L}}_1, \widehat{\mathbf{L}}_2, \widehat{\mathbf{L}}_3].$$

More interpretation of (2.6) is given in section 3.1.

Remark 1 (alternative construction of $\widetilde{\mathcal{A}}$ in Step 1). When $\mathbb{E}\text{vec}(\mathcal{X})\text{vec}(\mathcal{X})^\top \neq \mathbf{I}_{p_1 p_2 p_3}$, we could consider the following alternative ways to construct the initial estimate $\widetilde{\mathcal{A}}$. First, in some cases we could do construction depending on the covariance structure of \mathcal{X} . For example, in the framework of tensor recovery via rank-one sketching (discussed in the introduction), we have $\mathcal{X}_j = \mathbf{u}_j \circ \mathbf{u}_j \circ \mathbf{u}_j$ and $\mathbf{u}_j \in \mathbb{R}^p$ has i.i.d. entry $N(0, 1)$. By the high-order Stein identity [61], one can show that

$$\widetilde{\mathcal{A}} = \frac{1}{6} \left[\frac{1}{n} \sum_{j=1}^n y_j \mathbf{u}_j \circ \mathbf{u}_j \circ \mathbf{u}_j - \sum_{j=1}^p (\mathbf{w} \circ \mathbf{e}_j \circ \mathbf{e}_j + \mathbf{e}_j \circ \mathbf{w} \circ \mathbf{e}_j + \mathbf{e}_j \circ \mathbf{e}_j \circ \mathbf{w}) \right]$$

is a proper initial unbiased estimator for \mathcal{A} [53, Lemma 4]. Here $\mathbf{w} = \frac{1}{n} \sum_{j=1}^n y_j \mathbf{u}_j$, \mathbf{e}_j is the j th canonical basis in \mathbb{R}^p . Another commonly used setting in data analysis is the high-order Kronecker covariance structure: $\mathbb{E}(\text{vec}(\mathcal{X}_j)\text{vec}(\mathcal{X}_j)^\top) = \boldsymbol{\Sigma}_3 \otimes \boldsymbol{\Sigma}_2 \otimes \boldsymbol{\Sigma}_1$, where $\boldsymbol{\Sigma}_k \in \mathbb{R}^{p_k \times p_k}$, $k = 1, 2, 3$, are covariance matrices along three modes, respectively [55, 75, 78, 91, 133]. Under this assumption, we can first apply existing approaches to obtain estimators $\widehat{\boldsymbol{\Sigma}}_k$ for $\boldsymbol{\Sigma}_k$ and then whiten the covariates by replacing \mathcal{X}_j by $[\mathcal{X}_j; \widehat{\boldsymbol{\Sigma}}_1^{-1/2}, \widehat{\boldsymbol{\Sigma}}_2^{-1/2}, \widehat{\boldsymbol{\Sigma}}_3^{-1/2}]$. After this preprocessing step, the other steps of ISLET still follow. Moreover, it still remains an open question how to perform initialization if \mathcal{X} has the more general, unstructured, and unknown design.

Remark 2 (alternative methods to HOOI). In addition to HOOI, there are a variety of methods proposed in the literature to compute the low-rank tensor approximation, such as Newton-type optimization methods on manifolds [39, 59, 60, 99], black box approximation [9, 19, 77, 87, 88, 126], generalizations of Krylov subspace method [47, 98], greedy approximation method [46], among many others. Further, black box approximation methods [9, 19, 87, 88, 126] can be applied even if the initial estimator $\widetilde{\mathcal{A}}$ does not fit into the core memory. When the tensor is further approximately CP low-rank, we can also apply the randomized compressing method [101, 102] or randomized block sampling [115] to obtain the CP low-rank tensor approximation. Although the rest of our discussion will focus on the HOOI procedure for initialization, these alternative methods can also be applied to obtain an initialization for the ISLET algorithm.

$$\tilde{\mathcal{A}} = \frac{1}{n} (\textcolor{orange}{\square} \mathcal{X}_1 + \dots + \textcolor{yellow}{\square} \mathcal{X}_n)$$

(a) Construct the covariance tensor $\tilde{\mathcal{A}}$

$$\tilde{\mathcal{A}} \approx \tilde{S} \times_1 \tilde{U}_1 \times_2 \tilde{U}_2 \times_3 \tilde{U}_3$$

(b) Perform HOOI on $\tilde{\mathcal{A}}$ to obtain sketching directions

$$\begin{array}{c} \mathcal{M}_k(\tilde{\mathcal{A}}) \\ \approx \tilde{U}_k \tilde{\Sigma}_k \tilde{W}_k^T \\ \approx \tilde{U}_k \tilde{\Sigma}_k \tilde{W}_k^T \\ \approx \tilde{U}_k \tilde{\Sigma}_k \tilde{W}_k^T \end{array}$$

(c) The sketching directions yield low-rank approximations for $\mathcal{M}_k(\tilde{\mathcal{A}})$ **Figure 1.** Illustration for Step 1 of ISLET.

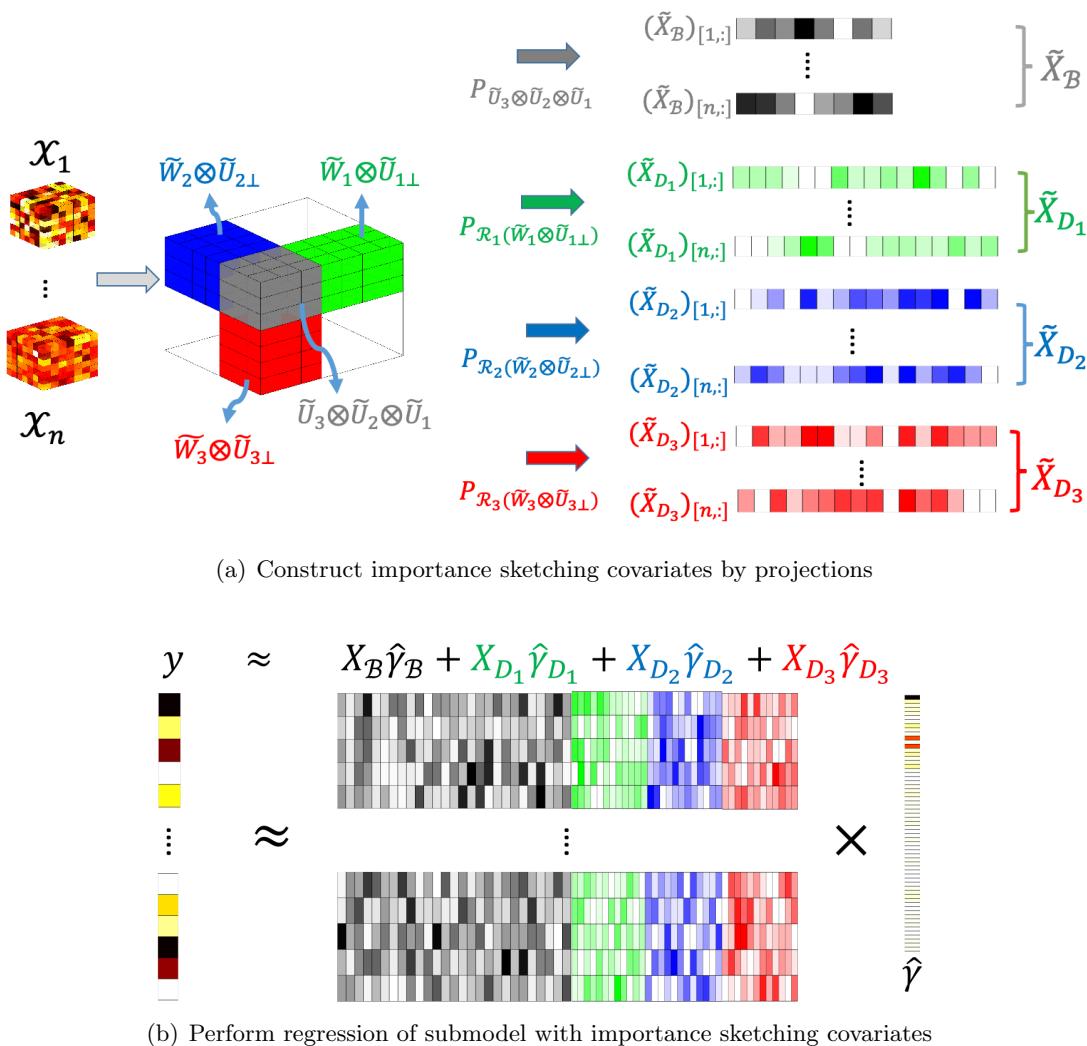


Figure 2. Illustration for Step 2 of ISLET.

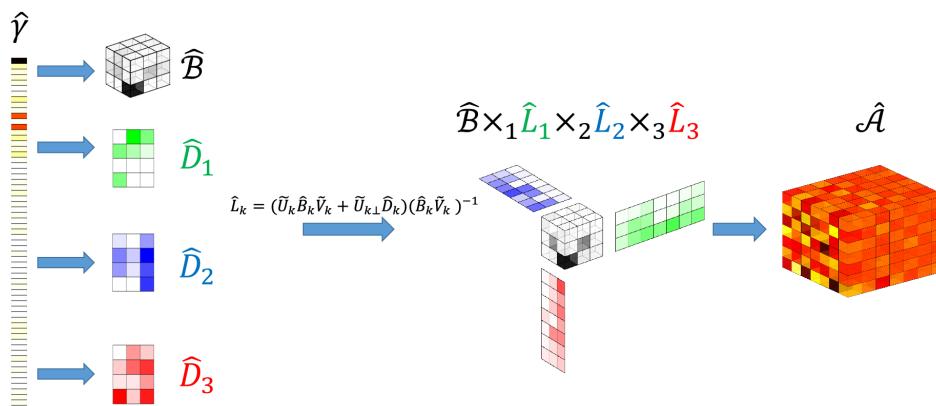


Figure 3. Illustration for Step 3 of ISLET.

Computation and implementation. We briefly discuss computational complexity and implementation aspects for the ISLET procedure here. It is noteworthy that ISLET accesses the sample only twice for constructing the covariance tensor (Step 1) and importance sketching covariates (Step 2), respectively. In large-scale cases where it is difficult to store the whole dataset into random-access memory (RAM), this advantage can highly save the computational costs.

In addition, in the order-3 tensor case, when each mode shares the same dimension $p_k = p$ and rank $r_k = r$, the total number of observable values is $O(np^3)$ and the time complexity of ISLET is $O(np^3r + nr^6 + Tp^4)$, where T is the number of HOOI iterations. For general order- d tensor regression, the time complexity of ISLET is $O(np^d r + nr^{2d} + Tp^{d+1})$. In contrast, the time complexity of the nonconvex PGD [27] is $O(T'(np^d + rp^{d+1}))$, where T' is the number of iterations of gradient descent; the authors of [13] introduced an optimization-based method with time complexity $O(T'dnp^d r)$, where T' is the number of iterations in the Gauss–Newton method. We can see that if $T' \geq r$, a typical situation in practice, ISLET is significantly faster than these previous methods.

It is worth pointing out that the computing time of ISLET is still high when the tensor parameter has a large order d . In fact, without any structural assumption on the design tensors \mathcal{X}_j , such a time cost may be unavoidable since reading in all data requires $O(np^d)$ operations. If there is extra structure on the design tensor, e.g., Kronecker product [7, 57, 58, 74] and low separation rank [10, 46], the computing time can be significantly reduced by applying methods in this body of literature. Here we mainly focus on the setting where \mathcal{X}_j does not satisfy a clear structural assumption since in many real data applications, e.g., the neuroimaging data example studied in this and many other works [1, 71, 106, 132], the design tensors \mathcal{X}_j may not have a clear known structure.

Moreover, in the order-3 tensor case, instead of storing all $\{\mathcal{X}_j\}_{j=1}^n$ in the memory which requires $O(np^3)$ RAM, ISLET only requires $O(p^3 + n(pr + r^3))$ RAM space if one chooses to access the samples from hard disks but not to store to RAM. This makes large-scale computing possible. We empirically investigate the computation cost by simulation studies in section 5.

The proposed ISLET procedure also allows convenient parallel computing. Suppose we distribute all n samples across B machines: $\{(\mathcal{X}_{bi}, y_{bi})\}_{i=1}^{B_b}$, $b = 1, \dots, B$, where $B_b \approx n/B$. To evaluate the covariance tensor in Step 1, we can calculate $\tilde{\mathcal{A}}_b = \sum_{i=1}^{B_b} y_{bi} \mathcal{X}_{bi}$ in each machine and then summarize them as $\tilde{\mathcal{A}} = \frac{1}{n} \sum_{b=1}^B \tilde{\mathcal{A}}_b$; to construct sketching covariates and perform partial regression in Step 2, we calculate

$$(2.7) \quad \mathbf{y}_b = (y_{b1}, \dots, y_{bB_b})^\top \in \mathbb{R}^{B_b},$$

$$(2.8) \quad \begin{aligned} \tilde{\mathbf{X}}_{bi} &= [\tilde{\mathbf{X}}_{\mathcal{B},bi} \quad \tilde{\mathbf{X}}_{\mathbf{D}_1,bi} \quad \tilde{\mathbf{X}}_{\mathbf{D}_2,bi} \quad \tilde{\mathbf{X}}_{\mathbf{D}_3,bi}] \in \mathbb{R}^m, \\ \tilde{\mathbf{X}}_{\mathcal{B},bi} &= \text{vec} \left(\mathcal{X}_{bi} \times_1 \tilde{\mathbf{U}}_1^\top \times_2 \tilde{\mathbf{U}}_2^\top \times_3 \tilde{\mathbf{U}}_3^\top \right), \\ \tilde{\mathbf{X}}_{\mathbf{D}_k,bi} &= \text{vec} \left(\tilde{\mathbf{U}}_{k+1}^\top \mathcal{M}_k \left(\mathcal{X}_{bi} \times_{k+1} \tilde{\mathbf{U}}_{k+1}^\top \times_{k+2} \tilde{\mathbf{U}}_{k+2}^\top \right) \tilde{\mathbf{V}}_k \right), \end{aligned}$$

$$(2.9) \quad \tilde{\mathbf{G}}_b = \sum_{i=1}^{B_b} \tilde{\mathbf{X}}_{bi}^\top \tilde{\mathbf{X}}_{bi}, \quad \tilde{\mathbf{z}}_b = \sum_{i=1}^{B_b} \tilde{\mathbf{X}}_{bi}^\top y_{bi}$$

in each machine. Then we combine the outcomes to

$$\hat{\gamma} = \left(\sum_{b=1}^B \tilde{\mathbf{G}}_b \right)^{-1} \left(\sum_{b=1}^B \tilde{\mathbf{z}}_b \right).$$

The computational complexity can be reduced to $O\left(\frac{np^3r+nr^6}{B} + Tp^4\right)$ via the parallel scheme. In the large-scale simulation we present in this article, we implement this parallel scheme for speed-up.

To implement the proposed procedure, the inputs of Tucker rank are required as tuning parameters. When they are unknown in practice, we can perform cross-validation or an adaptive rank selection scheme. A more detailed description and numerical results are postponed to section SM4 in the supplementary materials [128].

2.3. Sparse low-rank tensor recovery. When the target tensor \mathcal{A} is simultaneously low-rank and sparse, in the sense that (1.3) holds for a subset $J_s \subseteq \{1, 2, 3\}$ known a priori, we introduce the following sparse ISLET procedure. The pseudocode for sparse ISLET is summarized in Algorithm 2.2.

Step 1 (Probing sketching directions) When $\mathbb{E}\text{vec}(\mathcal{X})\text{vec}(\mathcal{X})^\top = \mathbf{I}_{p_1 p_2 p_3}$, we still evaluate the covariance tensor $\tilde{\mathcal{A}}$ as (2.1). Noting that $\mathcal{A} = [\mathcal{S}; \mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3]$ and $\{\mathbf{U}_k\}_{k \in J_s}$ are rowwise sparse, we apply the sparse tensor alternating thresholding singular value decomposition (STAT-SVD) [127] on $\tilde{\mathcal{A}}$ to obtain $\tilde{\mathbf{U}}_k \in \mathbb{O}_{p_k, r_k}$, $k = 1, 2, 3$, as initial estimates for \mathbf{U}_k . Here STAT-SVD is a sparse tensor decomposition method proposed by [127] with central ideas of the double projection and thresholding scheme as well as power iteration. Via STAT-SVD, we obtain the following sparse and low-rank approximation of \mathcal{A} :

$$\mathcal{A} \approx [\tilde{\mathcal{S}}; \tilde{\mathbf{U}}_1, \tilde{\mathbf{U}}_2, \tilde{\mathbf{U}}_3], \quad \tilde{\mathbf{U}}_k \in \mathbb{O}_{p_k, r_k}, \quad \tilde{\mathcal{S}} = [\tilde{\mathcal{A}}; \tilde{\mathbf{U}}_1^\top, \tilde{\mathbf{U}}_2^\top, \tilde{\mathbf{U}}_3^\top] \in \mathbb{R}^{r_1 \times r_2 \times r_3}.$$

We further evaluate

$$\tilde{\mathbf{V}}_k = \text{QR} \left(\mathcal{M}_k^\top (\tilde{\mathcal{S}}) \right) \in \mathbb{O}_{r_{k+1} r_{k+2}, r_k}.$$

Step 2 (Group Lasso on sketched covariates) We perform sketching and construct the following importance sketching covariates based on $\{\tilde{\mathbf{U}}_k, \tilde{\mathbf{V}}_k\}_{k=1}^3$:

$$(2.10) \quad \begin{aligned} \tilde{\mathbf{X}}_{\mathcal{B}} &\in \mathbb{R}^{n \times (r_1 r_2 r_3)}, \quad (\tilde{\mathbf{X}}_{\mathcal{B}})_{[i,:]} = \text{vec} \left(\mathcal{X}_i \times_1 \tilde{\mathbf{U}}_1^\top \times_2 \tilde{\mathbf{U}}_2^\top \times_3 \tilde{\mathbf{U}}_3^\top \right), \\ \tilde{\mathbf{X}}_{\mathbf{E}_k} &\in \mathbb{R}^{n \times p_k r_k}, \quad (\tilde{\mathbf{X}}_{\mathbf{E}_k})_{[i,:]} = \text{vec} \left(\mathcal{M}_k \left(\mathcal{X}_i \times_{k+1} \tilde{\mathbf{U}}_{k+1}^\top \times_{k+2} \tilde{\mathbf{U}}_{k+2}^\top \right) \tilde{\mathbf{V}}_k \right). \end{aligned}$$

Then we perform regression on submodels with these reduced-dimensional covariates $\tilde{\mathbf{X}}_{\mathcal{B}}$ and $\tilde{\mathbf{X}}_{\mathbf{E}_k}$, respectively, using least squares and group Lasso [44, 124]:

$$(2.11) \quad \widehat{\mathcal{B}} \in \mathbb{R}^{r_1 \times r_2 \times r_3}, \quad \text{vec}(\widehat{\mathcal{B}}) = \arg \min_{\gamma \in \mathbb{R}^{r_1 r_2 r_3}} \|y - \tilde{\mathbf{X}}_{\mathcal{B}} \gamma\|_2^2,$$

$$(2.12) \quad \widehat{\mathbf{E}}_k \in \mathbb{R}^{p_k \times r_k}, \quad \text{vec}(\widehat{\mathbf{E}}_k) = \begin{cases} \arg \min_{\gamma} \|y - \tilde{\mathbf{X}}_{\mathbf{E}_k} \gamma\|_2^2 & \text{if } k \notin J_s; \\ \arg \min_{\gamma} \|y - \tilde{\mathbf{X}}_{\mathbf{E}_k} \gamma\|_2^2 + \eta_k \sum_{j=1}^{p_k} \|\gamma_{G_j^k}\|_2 & \text{if } k \in J_s. \end{cases}$$

Here $\{\eta_k\}_{k \in J_s}$ are the penalization level and

$$(2.13) \quad G_j^k = \{j, j + p_k, \dots, j + p_k(r_k - 1)\}, \quad j = 1, \dots, p_k,$$

form a partition of $\{1, \dots, p_k r_k\}$ that is induced by the construction of $\tilde{\mathbf{X}}_{\mathbf{E}_k}$ (details for why to use group lasso can be found in section 3.2).

Step 3 (Constructing the final estimator) $\hat{\mathcal{A}}$ can be constructed using the regression coefficients $\hat{\mathcal{B}}$ and $\hat{\mathbf{E}}_k$'s in the submodels (2.11) and (2.12),

$$(2.14) \quad \hat{\mathcal{A}} = [\hat{\mathcal{B}}, (\hat{\mathbf{E}}_1(\tilde{\mathbf{U}}_1^\top \hat{\mathbf{E}}_1)^{-1}), (\hat{\mathbf{E}}_2(\tilde{\mathbf{U}}_2^\top \hat{\mathbf{E}}_2)^{-1}), (\hat{\mathbf{E}}_3(\tilde{\mathbf{U}}_3^\top \hat{\mathbf{E}}_3)^{-1})].$$

More interpretation of (2.14) can be found in section 3.2.

Algorithm 2.1 Importance Sketching Low-Rank Estimation for Tensors (ISLET): Order-3 Case.

- 1: Input: sample $\{y_j, \mathcal{X}_j\}_{j=1}^n$, Tucker rank $\mathbf{r} = (r_1, r_2, r_3)$.
- 2: Calculate $\tilde{\mathcal{A}} = \frac{1}{n} \sum_{j=1}^n y_j \mathcal{X}_j$.
- 3: Apply HOOI on $\tilde{\mathcal{A}}$ and obtain initial estimates $\tilde{\mathbf{U}}_1, \tilde{\mathbf{U}}_2, \tilde{\mathbf{U}}_3$.
- 4: Let $\tilde{\mathcal{S}} = [\tilde{\mathcal{A}}; \tilde{\mathbf{U}}_1^\top, \tilde{\mathbf{U}}_2^\top, \tilde{\mathbf{U}}_3^\top]$. Evaluate the sketching direction,

$$\tilde{\mathbf{V}}_k = \text{QR} [\mathcal{M}_k(\tilde{\mathcal{S}})^\top], \quad k = 1, 2, 3.$$

- 5: Construct $\tilde{\mathbf{X}} = [\tilde{\mathbf{X}}_{\mathcal{B}} \ \tilde{\mathbf{X}}_{\mathbf{D}_1} \ \tilde{\mathbf{X}}_{\mathbf{D}_2} \ \tilde{\mathbf{X}}_{\mathbf{D}_3}] \in \mathbb{R}^{n \times m}$, where

$$\begin{aligned} \tilde{\mathbf{X}}_{\mathcal{B}} &\in \mathbb{R}^{n \times m_{\mathcal{B}}}, (\tilde{\mathbf{X}}_{\mathcal{B}})_{[i,:]} = \text{vec} \left(\mathcal{X}_i \times_1 \tilde{\mathbf{U}}_1^\top \times_2 \tilde{\mathbf{U}}_2^\top \times_3 \tilde{\mathbf{U}}_3^\top \right), \\ \tilde{\mathbf{X}}_{\mathbf{D}_k} &\in \mathbb{R}^{n \times m_{\mathbf{D}_k}}, (\tilde{\mathbf{X}}_{\mathbf{D}_k})_{[i,:]} = \text{vec} \left(\tilde{\mathbf{U}}_{k\perp}^\top \mathcal{M}_k \left(\mathcal{X}_i \times_{k+1} \tilde{\mathbf{U}}_{k+1}^\top \times_{k+2} \tilde{\mathbf{U}}_{k+2}^\top \right) \tilde{\mathbf{V}}_k \right) \end{aligned}$$

for $m_{\mathcal{B}} = r_1 r_2 r_3, m_{\mathbf{D}_k} = (p_k - r_k) r_k$, and $k = 1, 2, 3$.

- 6: Solve $\hat{\gamma} = \arg \min_{\gamma \in \mathbb{R}^m} \|y - \tilde{\mathbf{X}}\gamma\|_2^2$.
- 7: Partition $\hat{\gamma}$ and assign each part to $\hat{\mathcal{B}}, \hat{\mathbf{D}}_1, \hat{\mathbf{D}}_2, \hat{\mathbf{D}}_3$, respectively,

$$\begin{aligned} \text{vec}(\hat{\mathcal{B}}) &:= \hat{\gamma}_{\mathcal{B}} = \hat{\gamma}_{[1:m_{\mathcal{B}}]}, \\ \text{vec}(\hat{\mathbf{D}}_k) &:= \hat{\gamma}_{\mathbf{D}_k} = \hat{\gamma}_{[(m_{\mathcal{B}} + \sum_{k'=1}^{k-1} m_{\mathbf{D}_{k'}} + 1):(m_{\mathcal{B}} + \sum_{k'=1}^k m_{\mathbf{D}_{k'}})]}, \quad k = 1, 2, 3. \end{aligned}$$

- 8: Let $\hat{\mathbf{B}}_k = \mathcal{M}_k(\hat{\mathcal{B}})$. Evaluate

$$\hat{\mathcal{A}} = [\hat{\mathcal{B}}; \hat{\mathbf{L}}_1, \hat{\mathbf{L}}_2, \hat{\mathbf{L}}_3], \quad \hat{\mathbf{L}}_k = \left(\tilde{\mathbf{U}}_k \hat{\mathbf{B}}_k \tilde{\mathbf{V}}_k + \tilde{\mathbf{U}}_{k\perp} \hat{\mathbf{D}}_k \right) \left(\hat{\mathbf{B}}_k \tilde{\mathbf{V}}_k \right)^{-1}, \quad k = 1, 2, 3.$$

Algorithm 2.2 Sparse Importance Sketching Low-Rank Estimation for Tensors (Sparse ISLET): Order-3 Case.

- 1: Input: sample $\{y_j, \mathbf{X}_j\}_{j=1}^n$, Tucker rank $\mathbf{r} = (r_1, r_2, r_3)$, sparsity index $J_s \subseteq \{1, 2, 3\}$.
- 2: Evaluate $\tilde{\mathcal{A}} = \frac{1}{n} \sum_{j=1}^n y_j \mathbf{X}_j$.
- 3: Apply STAT-SVD on $\tilde{\mathcal{A}}$ with sparsity index J_s . Let the outcome be $\tilde{\mathbf{U}}_1, \tilde{\mathbf{U}}_2, \tilde{\mathbf{U}}_3$.
- 4: Let $\tilde{\mathcal{S}} = [\tilde{\mathcal{A}}; \tilde{\mathbf{U}}_1^\top, \tilde{\mathbf{U}}_2^\top, \tilde{\mathbf{U}}_3^\top]$ and evaluate the probing direction,

$$\tilde{\mathbf{V}}_k = \text{QR} \left[\mathcal{M}_k(\tilde{\mathcal{S}})^\top \right], \quad k = 1, 2, 3.$$

- 5: Construct

$$\begin{aligned} \tilde{\mathbf{X}}_{\mathbf{B}} &\in \mathbb{R}^{n \times (r_1 r_2 r_3)}, \quad (\tilde{\mathbf{X}}_{\mathbf{B}})_{[i,:]} = \text{vec}(\mathbf{X}_i \times_1 \tilde{\mathbf{U}}_1^\top \times_2 \tilde{\mathbf{U}}_2^\top \times_3 \tilde{\mathbf{U}}_3^\top), \\ \tilde{\mathbf{X}}_{\mathbf{E}_k} &\in \mathbb{R}^{n \times (p_k r_k)}, \quad (\tilde{\mathbf{X}}_{\mathbf{E}_k})_{[i,:]} = \text{vec} \left(\mathcal{M}_k \left(\mathbf{X}_i \times_{k+1} \tilde{\mathbf{U}}_{k+1}^\top \times_{k+2} \tilde{\mathbf{U}}_{k+2}^\top \right) \tilde{\mathbf{V}}_k \right). \end{aligned}$$

- 6: Solve

$$\hat{\mathcal{B}} \in \mathbb{R}^{r_1 r_2 r_3}, \quad \text{vec}(\hat{\mathcal{B}}) = \arg \min_{\gamma \in \mathbb{R}^{r_1 r_2 r_3}} \|y - \tilde{\mathbf{X}}_{\mathbf{B}} \gamma\|_2^2;$$

$$\hat{\mathbf{E}}_k \in \mathbb{R}^{p_k \times r_k}, \quad \text{vec}(\hat{\mathbf{E}}_k) = \begin{cases} \arg \min_{\gamma} \|y - \tilde{\mathbf{X}}_{\mathbf{E}_k} \gamma\|_2^2 + \lambda_k \sum_{j=1}^{p_k} \|\gamma_{G_j^k}\|_2, & k \in J_s; \\ \arg \min_{\gamma} \|y - \tilde{\mathbf{X}}_{\mathbf{E}_k} \gamma\|_2^2, & k \notin J_s. \end{cases}$$

- 7: Evaluate

$$\hat{\mathcal{A}} = \left[\hat{\mathcal{B}}; (\hat{\mathbf{E}}_1 (\tilde{\mathbf{U}}_1^\top \hat{\mathbf{E}}_1)^{-1}), (\hat{\mathbf{E}}_2 (\tilde{\mathbf{U}}_2^\top \hat{\mathbf{E}}_2)^{-1}), (\hat{\mathbf{E}}_3 (\tilde{\mathbf{U}}_3^\top \hat{\mathbf{E}}_3)^{-1}) \right].$$

2.4. A sketching perspective of ISLET. While one of the main focuses of this article is on low-rank tensor regression, from a sketching perspective, ISLET can be seen as a special case of a more general algorithm that broadly applies to high-dimensional statistical problems with dimension-reduced structure. In fact, the three steps of the ISLET procedure are completely general and are summarized informally here:

Step 1 (Probing projection directions) For the tensor regression problem, we use the HOOI [34] or STAT-SVD [127] approach for finding the informative low-rank subspaces along which we project/sketch. More generally, if we let $\tilde{\mathcal{A}} = \frac{1}{n} \sum_{j=1}^n y_j \mathbf{X}_j$, where \mathbf{X}_j has ambient dimension p , we can define a general projection operator (with a slight abuse of notation) $\mathcal{P}_m(\cdot) : \mathbb{R}^p \rightarrow \mathbb{R}^m$ indexed by low dimension m and let $\mathcal{S}(\tilde{\mathcal{A}})$ be the m -dimensional subspace of \mathbb{R}^p determined by performing $\mathcal{P}_m(\tilde{\mathcal{A}})$.

Step 2 (Estimation in subspaces) The second step involves first projecting the data \mathbf{X} onto the subspace $\mathcal{S}(\tilde{\mathcal{A}})$, specifically $\tilde{\mathbf{X}} = \mathcal{P}_{\mathcal{S}(\tilde{\mathcal{A}})}(\mathbf{X}) \in \mathbb{R}^{n \times m}$. Then we perform regression or other procedures of choice using the sketched data $\tilde{\mathbf{X}}$ to determine the dimension-reduced parameter $\hat{\gamma} \in \mathbb{R}^m$.

Step 3 (Embedding to high-dimensional space) Finally, we need to project the estimator back

to the high-dimensional space \mathbb{R}^p by applying an equivalent to the inverse of the projection operator $\mathcal{P}_{\mathcal{S}(\tilde{\mathcal{A}})}^{-1} : \mathbb{R}^m \rightarrow \mathbb{R}^p$. For low-rank tensor regression, we require the formula (2.6).

The description above illustrates that the idea of ISLET is applicable to more general high-dimensional problems with dimension-reduced structure. In fact, the well-regarded *sure independence screening* in high-dimensional sparse linear regression [42, 121] can be seen as a special case of this idea. To be specific, consider the high-dimensional linear regression model,

$$y_i = X_{[i,:]} \boldsymbol{\beta} + \varepsilon_i, \quad i = 1, \dots, n,$$

where $\boldsymbol{\beta}$ is the m -sparse vector of interest and $y_i \in \mathbb{R}$ and $X_{[i,:]}^\top \in \mathbb{R}^p$ are the observable response and covariate. Then the m -dimensional subspace $\mathcal{S}(\tilde{\boldsymbol{\beta}})$ in Step 1 can be the coordinates corresponding to the m largest entries of $\tilde{\boldsymbol{\beta}} = \sum_{i=1}^n X_{[i,:]}^\top y_i$; Step 2 corresponds to the dimension reduced least squares in sure independence screening; and the inverse operator in Step 3 is simply filling in 0's in the coordinates that do not correspond to $\mathcal{S}(\tilde{\boldsymbol{\beta}})$. In addition, this idea applies more broadly to problems such as matrix and tensor completion. One of the novel contributions of this article is finding suitable projection and inverse operators for low-rank tensors.

We can also contrast this approach with prior approaches that involve randomized sketching [36, 93, 95]. These prior approaches showed that the randomized sketching may lose data substantially, increase the variance, and yield suboptimal results for many statistical problems. There are two key differences with how we exploit sketching in our context: (1) we sketch along the parameter directions of \mathcal{X} , reducing the data from $\mathbb{R}^{n \times p}$ to $\mathbb{R}^{n \times m}$; whereas approaches in [36, 93, 95] sketch along the sample directions, reducing the data from $\mathbb{R}^{n \times p}$ to $\mathbb{R}^{m \times p}$, which reduces the effective sample size from n to m ; (2) second, and most importantly, rather than using the randomized sketching that is *unsupervised* without the response y , our importance sketching is *supervised*, that is, obtained using both the response y and covariates \mathcal{X} . Then we sketch along the subspace $\mathcal{S}(\tilde{\mathcal{A}})$ which contains information on the low-dimensional structure of the parameter \mathcal{A} . This is why our general procedure has both desirable statistical and computational properties.

3. Oracle inequalities. In this section, we provide general oracle inequalities without focusing on specific design, which provides a general guideline for the theoretical analyses of our ISLET procedure. We first introduce a quantification of the errors in sketching directions obtained in the first step of ISLET. Let $\mathbf{V}_k \in \mathbb{O}_{r_{k+1}r_{k+2}, r_k}$ be the right singular subspace of $\mathcal{M}_k(\mathcal{S})$, where \mathcal{S} is the core tensor in the Tucker decomposition of \mathcal{A} : $\mathcal{A} = [\mathcal{S}; \mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3]$. By Lemma 1 in the supplementary materials [128],

$$(3.1) \quad \begin{aligned} \mathbf{W}_1 &:= (\mathbf{U}_3 \otimes \mathbf{U}_2) \mathbf{V}_1 \in \mathbb{O}_{p_2 p_3, r_1}, & \mathbf{W}_2 &:= (\mathbf{U}_3 \otimes \mathbf{U}_1) \mathbf{V}_2 \in \mathbb{O}_{p_1 p_3, r_2}, \\ &\text{and} & \mathbf{W}_3 &:= (\mathbf{U}_2 \otimes \mathbf{U}_1) \mathbf{V}_3 \in \mathbb{O}_{p_1 p_2, r_3} \end{aligned}$$

are the right singular subspaces of $\mathcal{M}_1(\mathcal{A})$, $\mathcal{M}_2(\mathcal{A})$, and $\mathcal{M}_3(\mathcal{A})$, respectively. Recall that we initially estimate \mathbf{U}_k and \mathbf{V}_k by $\tilde{\mathbf{U}}_k$ and $\tilde{\mathbf{V}}_k$, respectively, in Step 1 of ISLET. Define

$$\tilde{\mathbf{W}}_1 = (\tilde{\mathbf{U}}_3 \otimes \tilde{\mathbf{U}}_2) \tilde{\mathbf{V}}_1, \quad \tilde{\mathbf{W}}_2 = (\tilde{\mathbf{U}}_3 \otimes \tilde{\mathbf{U}}_1) \tilde{\mathbf{V}}_2, \quad \text{and} \quad \tilde{\mathbf{W}}_3 = (\tilde{\mathbf{U}}_2 \otimes \tilde{\mathbf{U}}_1) \tilde{\mathbf{V}}_3$$

in parallel to (3.1). Intuitively speaking, $\{\tilde{\mathbf{U}}_k, \tilde{\mathbf{W}}_k\}_{k=1}^3$ can be seen as the initial sample approximations for $\{\mathbf{U}_k, \mathbf{W}_k\}_{k=1}^3$. Therefore, we quantify the *sketching direction error* by

$$(3.2) \quad \theta := \max_{k=1,2,3} \left\{ \|\sin \Theta(\tilde{\mathbf{U}}_k, \mathbf{U}_k)\|, \|\sin \Theta(\tilde{\mathbf{W}}_k, \mathbf{W}_k)\| \right\}.$$

Next, we provide the oracle inequality via θ for ISLET under regular and sparse settings, respectively, in the next two subsections.

3.1. Regular tensor regression and oracle inequality. In order to study the theoretical properties of the proposed procedure, we need to introduce another representation of the original model (1.1). Decompose the vectorized parameter \mathcal{A} as follows:

$$(3.3) \quad \begin{aligned} \text{vec}(\mathcal{A}) &= P_{\tilde{\mathbf{U}}} \text{vec}(\mathcal{A}) + P_{\tilde{\mathbf{U}}_\perp} \text{vec}(\mathcal{A}) \\ &= P_{\tilde{\mathbf{U}}_3 \otimes \tilde{\mathbf{U}}_2 \otimes \tilde{\mathbf{U}}_1} \text{vec}(\mathcal{A}) + P_{\mathcal{R}_1(\tilde{\mathbf{W}}_1 \otimes \tilde{\mathbf{U}}_{1\perp})} \text{vec}(\mathcal{A}) + P_{\mathcal{R}_2(\tilde{\mathbf{W}}_2 \otimes \tilde{\mathbf{U}}_{2\perp})} \text{vec}(\mathcal{A}) \\ &\quad + P_{\mathcal{R}_3(\tilde{\mathbf{W}}_3 \otimes \tilde{\mathbf{U}}_{3\perp})} \text{vec}(\mathcal{A}) + P_{\tilde{\mathbf{U}}_\perp} \text{vec}(\mathcal{A}) \\ &= (\tilde{\mathbf{U}}_3 \otimes \tilde{\mathbf{U}}_2 \otimes \tilde{\mathbf{U}}_1) \text{vec}(\tilde{\mathcal{B}}) + \mathcal{R}_1(\tilde{\mathbf{W}}_1 \otimes \tilde{\mathbf{U}}_{1\perp}) \text{vec}(\tilde{\mathbf{D}}_1) + \mathcal{R}_2(\tilde{\mathbf{W}}_2 \otimes \tilde{\mathbf{U}}_{2\perp}) \text{vec}(\tilde{\mathbf{D}}_2) \\ &\quad + \mathcal{R}_3(\tilde{\mathbf{W}}_3 \otimes \tilde{\mathbf{U}}_{3\perp}) \text{vec}(\tilde{\mathbf{D}}_3) + P_{\tilde{\mathbf{U}}_\perp} \text{vec}(\mathcal{A}). \end{aligned}$$

(See the proof of Theorem 2 for a detailed derivation of (3.3).) Here

$$\tilde{\mathbf{U}} = \left[\tilde{\mathbf{U}}_3 \otimes \tilde{\mathbf{U}}_2 \otimes \tilde{\mathbf{U}}_1 \quad \mathcal{R}_1(\tilde{\mathbf{W}}_1 \otimes \tilde{\mathbf{U}}_{1\perp}) \quad \mathcal{R}_2(\tilde{\mathbf{W}}_2 \otimes \tilde{\mathbf{U}}_{2\perp}) \quad \mathcal{R}_3(\tilde{\mathbf{W}}_3 \otimes \tilde{\mathbf{U}}_{3\perp}) \right],$$

$$\tilde{\mathcal{B}} := \left[\mathcal{A}; \tilde{\mathbf{U}}_1^\top, \tilde{\mathbf{U}}_2^\top, \tilde{\mathbf{U}}_3^\top \right] \in \mathbb{R}^{r_1 r_2 r_3}, \quad \text{and} \quad \tilde{\mathbf{D}}_k := \tilde{\mathbf{U}}_{k\perp}^\top \mathcal{M}_k(\mathcal{A}) \tilde{\mathbf{W}}_k \in \mathbb{R}^{(p_k - r_k) \times r_k}$$

are the singular subspace of the “Cross structure” and the low-dimensional projections of \mathcal{A} onto the “body” and “arms” formed by sketching directions $\{\tilde{\mathbf{U}}_k, \tilde{\mathbf{V}}_k\}_{k=1}^3$, respectively. (See Figure 4 for an illustration of $\tilde{\mathbf{U}}$, $\tilde{\mathcal{B}}$, and $\tilde{\mathbf{V}}_k$.) Due to different alignments, the i th row of $\{\mathbf{W}_k \otimes \mathbf{U}_{k\perp}\}_{k=1}^3$ does not necessarily correspond to the i th entry of $\text{vec}(\mathcal{A})$ for all $1 \leq i \leq p_1 p_2 p_3$. We thus permute the rows of $\{\tilde{\mathbf{W}}_k \otimes \tilde{\mathbf{U}}_{k\perp}\}_{k=1}^3$ to match each row of $\mathcal{R}_k(\tilde{\mathbf{W}}_k \otimes \tilde{\mathbf{U}}_{k\perp})$ to the corresponding entry in $\text{vec}(\mathcal{A})$. The formal definition of the rowwise permutation operator \mathcal{R}_k is rather clunky and is postponed to section SM1 in the supplementary materials. Intuitively speaking, $P_{\tilde{\mathbf{U}}} \text{vec}(\mathcal{A})$ represents the projection of \mathcal{A} onto the Cross structure and $P_{\tilde{\mathbf{U}}_\perp} \text{vec}(\mathcal{A})$ can be seen as a residual. If the estimates $\{\tilde{\mathbf{U}}_k, \tilde{\mathbf{W}}_k\}_{k=1}^3$ are close enough to $\{\mathbf{U}_k, \mathbf{W}_k\}_{k=1}^3$, i.e., θ defined in (3.2) is small, we expect that the residual $P_{\tilde{\mathbf{U}}_\perp} \text{vec}(\mathcal{A})$ has small amplitude.

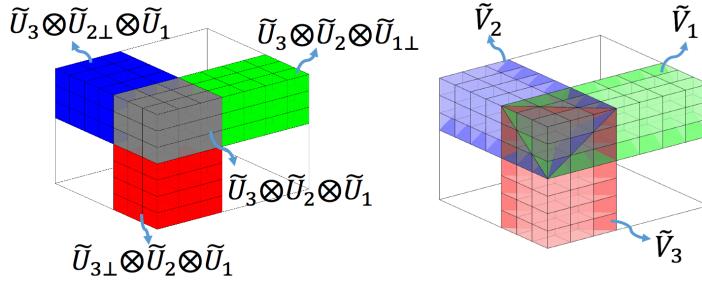


Figure 4. Illustration of decomposition (3.3). Here we assume $\tilde{\mathbf{U}}_k^\top = [\mathbf{I}_{r_k} \ \mathbf{0}_{r_k \times (p_k - r_k)}]$, $k = 1, 2, 3$, for a better visualization. The gray, green, blue, and red cubes represent the subspaces of $\tilde{\mathbf{U}}_3 \otimes \tilde{\mathbf{U}}_2 \otimes \tilde{\mathbf{U}}_1$, $\tilde{\mathbf{U}}_3 \otimes \tilde{\mathbf{U}}_2 \otimes \tilde{\mathbf{U}}_{1\perp}$, $\tilde{\mathbf{U}}_{3\perp} \otimes \tilde{\mathbf{U}}_2 \otimes \tilde{\mathbf{U}}_1$, and $\tilde{\mathbf{U}}_{3\perp} \otimes \tilde{\mathbf{U}}_2 \otimes \tilde{\mathbf{U}}_{1\perp}$. The gray cube also corresponds to the projected parameters $\tilde{\mathbf{B}}$; matricizations of green, blue, and red cubes correspond to the projected parameters $\tilde{\mathbf{U}}_{1\perp}^\top \mathcal{M}_1(\mathbf{A})(\tilde{\mathbf{U}}_3 \otimes \tilde{\mathbf{U}}_2)$, $\tilde{\mathbf{U}}_{2\perp}^\top \mathcal{M}_2(\mathbf{A})(\tilde{\mathbf{U}}_3 \otimes \tilde{\mathbf{U}}_1)$, and $\tilde{\mathbf{U}}_{3\perp}^\top \mathcal{M}_3(\mathbf{A})(\tilde{\mathbf{U}}_2 \otimes \tilde{\mathbf{U}}_1)$, respectively. The three planes in the right panel correspond to the subspaces of $\tilde{\mathbf{V}}_1$, $\tilde{\mathbf{V}}_2$, and $\tilde{\mathbf{V}}_3$, respectively.

Based on (3.3), we can rewrite the original regression model (1.1) into the following partial regression model:

$$(3.4) \quad \begin{aligned} y_j &= (\tilde{\mathbf{X}}_{\mathbf{B}})_{[j,:]} \text{vec}(\tilde{\mathbf{B}}) + \sum_{k=1}^3 (\tilde{\mathbf{X}}_{\mathbf{D}_k})_{[j,:]} \text{vec}(\tilde{\mathbf{D}}_k) + \text{vec}(\mathbf{X}_j)^\top P_{\tilde{\mathbf{U}}_\perp} \text{vec}(\mathbf{A}) + \varepsilon_j \\ &= \tilde{\mathbf{X}}_{[j,:]} \tilde{\gamma} + \tilde{\varepsilon}_j, \quad j = 1, \dots, n. \end{aligned}$$

(See the proof of Theorem 2 for a detailed derivation of (3.4).) Here we have the following:

- $\tilde{\varepsilon}_j = \text{vec}(\mathbf{X}_j)^\top P_{\tilde{\mathbf{U}}_\perp} \text{vec}(\mathbf{A}) + \varepsilon_j$ is the oracle noise; $\tilde{\varepsilon} = (\tilde{\varepsilon}_1, \dots, \tilde{\varepsilon}_n)^\top$;
- $\tilde{\mathbf{X}}_{\mathbf{B}}, \tilde{\mathbf{X}}_{\mathbf{D}_k}$ are sketching covariates introduced in (2.3);
- $\tilde{\gamma} = [\text{vec}(\tilde{\mathbf{B}})^\top, \text{vec}(\tilde{\mathbf{D}}_1)^\top, \text{vec}(\tilde{\mathbf{D}}_2)^\top, \text{vec}(\tilde{\mathbf{D}}_3)^\top]^\top = \tilde{\mathbf{U}}^\top \text{vec}(\mathbf{A}) \in \mathbb{R}^m$ is the dimension-reduced parameter.

Equation (3.4) reveals the essence of the least squares estimator (2.4) in the ISLET procedure—the outcomes of (2.4) and (2.5), i.e., $\tilde{\mathbf{B}}$ and $\tilde{\mathbf{D}}_k$, are sample-based estimates of \mathbf{B} and \mathbf{D}_k . Finally, based on the detailed algebraic calculation in Step 3 and the proof of Theorem 2,

$$(3.5) \quad \mathbf{A} = \left[\left[\tilde{\mathbf{B}}; \tilde{\mathbf{L}}_1, \tilde{\mathbf{L}}_2, \tilde{\mathbf{L}}_3 \right] \right], \quad \tilde{\mathbf{L}}_k = \left(\tilde{\mathbf{U}}_k \tilde{\mathbf{B}}_k \tilde{\mathbf{V}}_k + \tilde{\mathbf{U}}_{k\perp} \tilde{\mathbf{D}}_k \right) \left(\tilde{\mathbf{B}}_k \tilde{\mathbf{V}}_k \right)^{-1}.$$

Equation (3.5) is essentially a higher-order version of the Schur complement formula (also see [16]). Finally, we apply the plug-in estimator to obtain the final estimator $\hat{\mathbf{A}}$ (see (2.6) in Step 3 of the ISLET procedure).

Based on previous discussions, it can be seen that the estimation error of the original tensor regression is driven by the error of the least squares estimator $\hat{\gamma}$, i.e., $\|(\tilde{\mathbf{X}}^\top \tilde{\mathbf{X}})^{-1} \tilde{\mathbf{X}}^\top \tilde{\varepsilon}\|_2^2$. We have the following oracle inequality for the proposed ISLET procedure.

Theorem 2 (oracle inequality of regular tensor estimation: Order-3 case). Suppose $\mathbf{A} \in \mathbb{R}^{p_1 \times p_2 \times p_3}$ has Tucker rank- (r_1, r_2, r_3) tensor and $\hat{\mathbf{A}}$ is the outcome of Algorithm 2.1. Assume the sketching directions $\{\tilde{\mathbf{U}}_k, \tilde{\mathbf{V}}_k\}_{k=1}^3$ satisfy $\theta < 1/2$ (see (3.2) for the definition of θ) and

$\|\widehat{\mathbf{D}}_k(\widehat{\mathbf{B}}_k\widetilde{\mathbf{V}}_k)^{-1}\| \leq \rho$. We don't impose other specific assumptions on \mathcal{X}_i and ε_i . Then we have

$$\|\widehat{\mathcal{A}} - \mathcal{A}\|_{\text{HS}}^2 \leq (1 + C(\theta + \rho)) \left\| (\widetilde{\mathbf{X}}^\top \widetilde{\mathbf{X}})^{-1} \widetilde{\mathbf{X}}^\top \widetilde{\boldsymbol{\varepsilon}} \right\|_2^2$$

for uniform constant $C > 0$ that does not rely on any other parameters.

Proof. See Appendix SM6.1 for a complete proof. In particular, the proof contains three major steps. After introducing a number of notations, we first transform the original regression model to the partial regression model (3.4) and then rewrite the upper bound $\|(\widetilde{\mathbf{X}}^\top \widetilde{\mathbf{X}})^{-1} \widetilde{\mathbf{X}}^\top \widetilde{\boldsymbol{\varepsilon}}\|_2^2$ to $\|\widetilde{\mathcal{B}} - \widetilde{\mathcal{B}}\|_{\text{HS}}^2 + \sum_{k=1}^3 \|\widehat{\mathbf{D}}_k - \mathbf{D}_k\|_F^2$. Next, we introduce a factorization of \mathcal{A} in parallel with the one of $\widehat{\mathcal{A}}$, based on which the loss $\|\widehat{\mathcal{A}} - \mathcal{A}\|_{\text{HS}}$ is decomposed into eight terms. Finally, we introduce a novel deterministic error bound for the ‘‘Cross scheme’’ (Lemma 3 in the supplementary materials [128]; also see [126]), carefully analyze each term in the decomposition of $\|\widehat{\mathcal{A}} - \mathcal{A}\|_{\text{HS}}$, and finalize the proof. ■

Theorem 2 shows that once the sketching directions $\widetilde{\mathbf{U}}$ and $\widetilde{\mathbf{V}}$ are reasonably accurate, the estimation error for $\widehat{\mathcal{A}}$ will be close to the error of partial linear regression in (3.4). This bound is general and deterministic, which can be used as a key step in more specific settings of low-rank tensor regression.

3.2. Sparse tensor regression and oracle inequality. Next, we study the oracle performance of the proposed procedure for sparse tensor regression, where \mathcal{A} further satisfies the sparsity constraint (1.3). As in the previous section, we decompose the vectorized parameter as

$$(3.6) \quad \begin{aligned} \text{vec}(\mathcal{A}) &= P_{\widetilde{\mathbf{U}}_3 \otimes \widetilde{\mathbf{U}}_2 \otimes \widetilde{\mathbf{U}}_1} \text{vec}(\mathcal{A}) + P_{(\widetilde{\mathbf{U}}_3 \otimes \widetilde{\mathbf{U}}_2 \otimes \widetilde{\mathbf{U}}_1)^\perp} \text{vec}(\mathcal{A}) \\ &= (\widetilde{\mathbf{U}}_3 \otimes \widetilde{\mathbf{U}}_2 \otimes \widetilde{\mathbf{U}}_3) \text{vec}(\widetilde{\mathcal{B}}) + P_{(\widetilde{\mathbf{U}}_3 \otimes \widetilde{\mathbf{U}}_2 \otimes \widetilde{\mathbf{U}}_1)^\perp} \text{vec}(\mathcal{A}); \end{aligned}$$

$$(3.7) \quad \begin{aligned} \text{vec}(\mathcal{A}) &= P_{\mathcal{R}_k(\widetilde{\mathbf{W}}_k \otimes \mathbf{I}_{p_k})} \text{vec}(\mathcal{A}) + P_{\mathcal{R}_k(\widetilde{\mathbf{W}}_k \otimes \mathbf{I}_{p_k})^\perp} \text{vec}(\mathcal{A}) \\ &= \mathcal{R}_k(\widetilde{\mathbf{W}}_k \otimes \mathbf{I}_{p_k}) \text{vec}(\widetilde{\mathbf{E}}_k) + P_{\mathcal{R}_k(\widetilde{\mathbf{W}}_k \otimes \mathbf{I}_{p_k})^\perp} \text{vec}(\mathcal{A}), \quad k = 1, 2, 3. \end{aligned}$$

Here

$$(3.8) \quad \begin{aligned} \widetilde{\mathcal{B}} &:= [\mathcal{A}; \widetilde{\mathbf{U}}_1^\top, \widetilde{\mathbf{U}}_2^\top, \widetilde{\mathbf{U}}_3^\top] \in \mathbb{R}^{r_1 r_2 r_3}; \\ \widetilde{\mathbf{E}}_k &:= \mathcal{M}_k \left(\mathcal{A} \times_{(k+1)} \widetilde{\mathbf{U}}_{k+1}^\top \times_{(k+2)} \widetilde{\mathbf{U}}_{k+2}^\top \right) \widetilde{\mathbf{V}}_k \in \mathbb{R}^{p_k \times r_k}, \quad k = 1, 2, 3, \end{aligned}$$

are the low-dimensional projections of \mathcal{A} onto the importance sketching directions. Since $\{\mathbf{U}_k, \mathbf{W}_k\}$ are the left and right singular subspaces of $\mathcal{M}_k(\mathcal{A})$, we can demonstrate that $P_{(\mathbf{U}_3 \otimes \mathbf{U}_2 \otimes \mathbf{U}_1)^\perp} \text{vec}(\mathcal{A})$ and $P_{\mathcal{R}_k(\mathbf{W}_k \otimes \mathbf{I}_{p_k})^\perp} \text{vec}(\mathcal{A})$ are zeros. Thus if the estimates $\{\widetilde{\mathbf{U}}_k, \widetilde{\mathbf{W}}_k\}_{k=1}^3$ are sufficiently accurate, i.e., θ defined in (3.2) is small, we can expect that the residuals $P_{(\widetilde{\mathbf{U}}_3 \otimes \widetilde{\mathbf{U}}_2 \otimes \widetilde{\mathbf{U}}_1)^\perp} \text{vec}(\mathcal{A})$ and $P_{\mathcal{R}_k(\widetilde{\mathbf{W}}_k \otimes \mathbf{I}_{p_k})^\perp} \text{vec}(\mathcal{A})$ have small amplitudes. Then, based on a more detailed calculation in the proof of Theorem 3, the model of sparse and low-rank tensor regression $y_j = \langle \mathcal{X}_j, \mathcal{A} \rangle + \varepsilon_j$ can be rewritten as the following partial linear regression:

$$(3.9) \quad y_j = (\widetilde{\mathbf{X}}_{\mathcal{B}})_{[j,:]} \text{vec}(\widetilde{\mathcal{B}}) + (\widetilde{\boldsymbol{\varepsilon}}_{\mathcal{B}})_j,$$

$$(3.10) \quad y_j = (\tilde{\mathbf{X}}_{\mathbf{E}_k})_{[j,:]} \text{vec}(\tilde{\mathbf{E}}_k) + (\tilde{\boldsymbol{\varepsilon}}_{\mathbf{E}_k})_j, \quad k = 1, 2, 3.$$

Here $\tilde{\mathbf{X}}_{\mathbf{B}}$ and $\tilde{\mathbf{X}}_{\mathbf{E}_k}$ are the covariates defined in (2.10) and $\tilde{\boldsymbol{\varepsilon}}_{\mathbf{B}} = ((\tilde{\boldsymbol{\varepsilon}}_{\mathbf{B}})_1, \dots, (\tilde{\boldsymbol{\varepsilon}}_{\mathbf{B}})_n)^\top$, $\tilde{\boldsymbol{\varepsilon}}_{\mathbf{E}_k} = ((\tilde{\boldsymbol{\varepsilon}}_{\mathbf{E}_k})_1, \dots, (\tilde{\boldsymbol{\varepsilon}}_{\mathbf{E}_k})_n)^\top$ are oracle noises defined as

$$(3.11) \quad \begin{aligned} (\tilde{\boldsymbol{\varepsilon}}_{\mathbf{B}})_j &= \left\langle \text{vec}(\mathbf{X}_j), P_{(\tilde{\mathbf{U}}_3 \otimes \tilde{\mathbf{U}}_2 \otimes \tilde{\mathbf{U}}_1)^\perp} \text{vec}(\mathbf{A}) \right\rangle + \varepsilon_j \\ \text{and } (\tilde{\boldsymbol{\varepsilon}}_{\mathbf{E}_k})_j &= \left\langle \text{vec}(\mathbf{X}_j), P_{(\mathcal{R}_k(\tilde{\mathbf{W}}_k \otimes \mathbf{I}_{p_k}))^\perp} \text{vec}(\mathbf{A}) \right\rangle + \varepsilon_j. \end{aligned}$$

Therefore, Step 2 of sparse ISLET can be interpreted as the estimation of $\tilde{\mathbf{B}}$ and $\tilde{\mathbf{E}}_k$.

We apply regular least squares to estimate $\tilde{\mathbf{B}}$ and $\tilde{\mathbf{E}}_k$ for $k \notin J_s$. For any sparse mode $k \in J_s$, $\tilde{\mathbf{E}}_k$ are group sparse due to the definition (3.8) and the assumption that \mathbf{U}_k are rowwise sparse. Specifically, $\tilde{\mathbf{E}}_k$ satisfies

$$(3.12) \quad \left\| \text{vec}(\tilde{\mathbf{E}}_k) \right\|_{0,2} := \sum_{i=1}^{p_k} 1_{\{(\text{vec}(\tilde{\mathbf{E}}_k))_{G_i^k} \neq 0\}} \leq s_k,$$

where

$$G_i^k = \{i, i + p_k, \dots, i + p_k(r_k - 1)\}, \quad i = 1, \dots, p_k, \quad \forall k \in J_s,$$

is a partition of $\{1, \dots, p_k r_k\}$ (see the proof for Theorem 3 for a more detailed argument for (3.12)). By detailed calculations in Step 3 of the proof for Theorem 2, one can verify that

$$\mathbf{A} = \left[\tilde{\mathbf{B}}, (\tilde{\mathbf{E}}_1(\tilde{\mathbf{U}}_1^\top \tilde{\mathbf{E}}_1)^{-1}), (\tilde{\mathbf{E}}_2(\tilde{\mathbf{U}}_2^\top \tilde{\mathbf{E}}_2)^{-1}), (\tilde{\mathbf{E}}_3(\tilde{\mathbf{U}}_3^\top \tilde{\mathbf{E}}_3)^{-1}) \right].$$

Then the finally sparse ISLET estimator $\hat{\mathbf{A}}$ in (2.14) can be seen as the plug-in estimator.

To ensure that the group Lasso estimator in (2.12) provides a stable estimation for the proposed procedure, we introduce the following group restricted isometry condition, which can also be seen as an extension of the restricted isometry property (RIP), a commonly used condition in compressed sensing and high-dimensional linear regression literature [24].

Condition 1. We say a matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ satisfies the group restricted isometry property (GRIP) with respect to partition $G_1, \dots, G_m \subseteq \{1, \dots, p\}$ if there exists $\delta > 0$ such that

$$(3.13) \quad n(1 - \delta) \|\mathbf{v}\|_2^2 \leq \|\mathbf{X}\mathbf{v}\|_2^2 \leq n(1 + \delta) \|\mathbf{v}\|_2^2$$

for all groupwise sparse vectors \mathbf{v} satisfying $\sum_{k=1}^m 1_{\{\mathbf{v}_{G_k} \neq 0\}} \leq s$.

We still use θ defined in (3.2) to characterize the sketching direction errors. The following oracle inequality holds for sparse tensor regression with importance sketching.

Theorem 3 (oracle inequality for sparse tensor regression: Order-3 case). Consider the sparse low-rank tensor regression (1.1), (1.3). Suppose $\theta < 1/2$ and the importance sketching covariates $\tilde{\mathbf{X}}_{\mathbf{B}}$ and $\tilde{\mathbf{X}}_{\mathbf{E}_k}$ ($k \notin J_s$) are nonsingular. For any $k \in J_s$, $\tilde{\mathbf{X}}_{\mathbf{E}_k}$ satisfies the GRIP (Condition 1) with respect to partition $G_1^k, \dots, G_{p_k}^k$ in (2.13) and $\delta < 1/3$. We apply the proposed Algorithm 2.2 with group Lasso penalty

$$\eta_k = C_1 \max_{i=1, \dots, p_k} \left\| (\tilde{\mathbf{X}}_{\mathbf{E}_k, [:, G_i^k]})^\top \tilde{\boldsymbol{\varepsilon}}_{\mathbf{E}_k} \right\|_2$$

for $k \in J_s$ and some constant $C_1 \geq 3$. We also assume $\|\tilde{\mathbf{U}}_{k\perp}^\top \hat{\mathbf{E}}_k (\tilde{\mathbf{U}}_k^\top \hat{\mathbf{E}}_k)^{-1}\| \leq \rho$. Then

$$(3.14) \quad \begin{aligned} \|\hat{\mathcal{A}} - \mathcal{A}\|_{\text{HS}}^2 &\leq (1 + C_2 s(\theta + \rho)) \left(\left\| (\tilde{\mathbf{X}}_\mathcal{B}^\top \tilde{\mathbf{X}}_\mathcal{B})^{-1} \tilde{\mathbf{X}}_\mathcal{B}^\top \tilde{\boldsymbol{\varepsilon}}_\mathcal{B} \right\|_2^2 \right. \\ &\quad \left. + \sum_{k \notin J_s} \left\| (\tilde{\mathbf{X}}_{\mathbf{E}_k}^\top \tilde{\mathbf{X}}_{\mathbf{E}_k})^{-1} \tilde{\mathbf{X}}_{\mathbf{E}_k}^\top \tilde{\boldsymbol{\varepsilon}}_{\mathbf{E}_k} \right\|_2^2 + C_3 \sum_{k \in J_s} s_k \cdot \max_{i=1,\dots,p_k} \left\| (\tilde{\mathbf{X}}_{\mathbf{E}_k,[:,G_i^k]}^\top \tilde{\boldsymbol{\varepsilon}}_{\mathbf{E}_k}/n) \right\|_2^2 \right). \end{aligned}$$

Proof. See Appendix SM6.2. ■

Remark 3. In oracle error bound (3.14), $\|(\tilde{\mathbf{X}}_\mathcal{B}^\top \tilde{\mathbf{X}}_\mathcal{B})^{-1} \tilde{\mathbf{X}}_\mathcal{B}^\top \tilde{\boldsymbol{\varepsilon}}_\mathcal{B}\|_2^2$, $\|(\tilde{\mathbf{X}}_{\mathbf{E}_k}^\top \tilde{\mathbf{X}}_{\mathbf{E}_k})^{-1} \tilde{\mathbf{X}}_{\mathbf{E}_k}^\top \tilde{\boldsymbol{\varepsilon}}_{\mathbf{E}_k}\|_2^2$, and $s_k \max_{i=1,\dots,p_k} \left\| (\tilde{\mathbf{X}}_{\mathbf{E}_k,[:,G_i^k]}^\top \tilde{\boldsymbol{\varepsilon}}_{\mathbf{E}_k}/n) \right\|_2^2$ correspond to the estimation errors of $\hat{\mathcal{B}}$, $\hat{\mathbf{E}}_k$ of the nonsparse mode and $\hat{\mathbf{E}}_k$ of the sparse mode, respectively. When the GRIP (Condition 1) is replaced by the group restricted eigenvalue condition (see, e.g., [73]), a result similar to Theorem 3 can be derived.

4. Fast low-rank tensor regression via ISLET. We further study the low-rank tensor regression with Gaussian ensemble design; i.e., \mathcal{X}_i has i.i.d. standard normal entries. This has been considered a benchmark setting for low-rank tensor/matrix recovery literature [23, 27]. For convenience, we denote $\mathbf{p} = (p_1, p_2, p_3)$, $\mathbf{r} = (r_1, r_2, r_3)$, $p = \max\{p_1, p_2, p_3\}$, and $r = \max\{r_1, r_2, r_3\}$. We discuss the regular low-rank and sparse low-rank tensor regression in the next two subsections, respectively.

4.1. Regular low-rank tensor regression with ISLET. We have the following theoretical guarantee for ISLET under Gaussian ensemble design.

Theorem 4 (upper bound for tensor regression via ISLET). Consider the tensor regression model (1.1), where $\mathcal{A} \in \mathbb{R}^{p_1 \times p_2 \times p_3}$ is Tucker rank- (r_1, r_2, r_3) , \mathcal{X}_i has i.i.d. standard normal entries, and $\boldsymbol{\varepsilon} \stackrel{i.i.d.}{\sim} N(0, \sigma^2)$. Denote $\tilde{\sigma}^2 = \|\mathcal{A}\|_{\text{HS}}^2 + \sigma^2$, $\lambda_0 = \min_k \lambda_k$, $\lambda_k = \sigma_{r_k}(\mathcal{M}_k(\mathcal{A}))$, $\kappa = \max_k \|\mathcal{M}_k(\mathcal{A})\|/\sigma_{r_k}(\mathcal{M}_k(\mathcal{A}))$, and $m = r_1 r_2 r_3 + \sum_{k=1}^3 (p_k - r_k) r_k$. If $n_1 \wedge n_2 \geq \frac{C\tilde{\sigma}^2(p^{3/2} + \kappa pr)}{\lambda_0^2}$, then the sample-splitting ISLET estimator (see the forthcoming Remark 5) satisfies

$$\|\hat{\mathcal{A}} - \mathcal{A}\|_{\text{HS}}^2 \leq \frac{m}{n_2} \left(\sigma^2 + \frac{C_1 \tilde{\sigma}^4 mp}{n_1^2 \lambda_0^2} \right) \left(1 + C_2 \sqrt{\frac{\log p}{m}} + C_3 \sqrt{\frac{m \tilde{\sigma}^2}{(n_1 \wedge n_2) \lambda_0^2}} \right)$$

with probability at least $1 - p^{-C_4}$.

Proof. See section SM6.3 for details. Specifically, we first derive the estimation error upper bounds for sketching directions $\tilde{\mathbf{U}}_k$ via the deterministic error bound of HOOI [129]. Then we apply concentration inequalities to obtain upper bounds for $\|(\tilde{\mathbf{X}}^\top \tilde{\mathbf{X}})^{-1} \tilde{\mathbf{X}}^\top \tilde{\boldsymbol{\varepsilon}}\|_2^2$ and $\|\hat{\mathbf{D}}_k(\hat{\mathbf{B}}_k \tilde{\mathbf{V}}_k)^{-1}\|$ for $k = 1, 2, 3$. Finally, the oracle inequality of Theorem 2 leads to the desired upper bound. ■

Remark 4 (sample complexity). In Theorem 4, we show that as long as the sample size $n = \Omega(p^{3/2}r + pr^2)$, ISLET achieves consistent estimation under regularity conditions. This sample complexity outperforms many computationally feasible algorithms in previous literature, e.g.,

$n = \Omega(p^2 r \text{polylog}(p))$ in PGD [27], sum of nuclear norm minimization [110], and square norm minimization [84]. To the best of our knowledge, ISLET is the first computationally efficient algorithm that achieves this sample complexity result.

On the other hand, the authors of [84] showed that the direct nonconvex Tucker rank minimization, a computationally infeasible method, can do exact recovery with $O(pr + r^3)$ linear measurements in the noiseless setting. The authors of [13] showed that if tensor parameter \mathcal{A} is CP rank- r , the linear system $y_j = \langle \mathcal{A}, \mathcal{X}_j \rangle$, $j = 1, \dots, n$, has a unique solution with probability one if one has $O(pr)$ measurements. It remains an open question whether the sample complexity of $n = \Omega(p^{3/2}r + pr^2)$ is necessary for all computationally efficient procedures.

Remark 5 (sample splitting). The direct analysis for the proposed ISLET in Algorithm 2.1 is technically involved, among which one major difficulty is the dependency between the sketching directions $\tilde{\mathbf{U}}_k$ obtained in Step 1 and the regression noise $\tilde{\boldsymbol{\varepsilon}}$ in Step 2. To overcome this difficulty, we choose to analyze a modified procedure with the sample splitting scheme: we randomly split all n samples into two sets with cardinalities n_1 and n_2 , respectively. Then we use the first set of n_1 samples to construct the covariance tensor \mathcal{A} (Step 1) and use the second set of n_2 samples to evaluate the importance sketching covariates (Step 2). As illustrated by numerical studies in section 5, such a scheme is mainly for technical purposes and is not necessary in practice. Simulations suggest that it is preferable to use all samples $\{y_i, \mathcal{X}_i\}_{i=1}^n$ for both constructing the initial estimate $\tilde{\mathcal{A}}$ and performing linear regression on sketching covariates.

We further consider the statistical limits for low-rank tensor regression with Gaussian ensemble. Consider the following class of general low-rank tensors:

$$(4.1) \quad \mathcal{A}_{\mathbf{p}, \mathbf{r}} = \left\{ \mathcal{A} \in \mathbb{R}^{p_1 \times p_2 \times p_3} : \text{Tucker rank}(\mathcal{A}) \leq (r_1, r_2, r_3) \right\}.$$

The following minimax lower bound holds for all low-rank tensors in $\mathcal{A}_{\mathbf{p}, \mathbf{r}}$.

Theorem 5 (minimax lower bound). *If $n > m+1$, the following nonasymptotic lower bound in estimation error holds:*

$$(4.2) \quad \inf_{\hat{\mathcal{A}}} \sup_{\mathcal{A} \in \mathcal{A}_{\mathbf{p}, \mathbf{r}}} \mathbb{E} \left\| \hat{\mathcal{A}} - \mathcal{A} \right\|_{\text{HS}}^2 \geq \frac{m}{n-m-1} \cdot \sigma^2.$$

If $n \leq m+1$,

$$(4.3) \quad \inf_{\hat{\mathcal{A}}} \sup_{\mathcal{A} \in \mathcal{A}_{\mathbf{p}, \mathbf{r}}} \mathbb{E} \left\| \hat{\mathcal{A}} - \mathcal{A} \right\|_{\text{HS}}^2 = +\infty.$$

Proof. See Appendix SM6.4. ■

Combining Theorems 4 and 5, we can see that as long as the sample size satisfies $\frac{m\bar{\sigma}^2}{n_1\lambda_0^2} = o(1)$, $\frac{m(p_1+p_2+p_3)\bar{\sigma}^4}{n_1n_2\lambda_0^3} = o(\sigma^2)$, and $n_2 = (1+o(1))n$, the statistical loss of the proposed method is sharp with matching constant to the lower bound.

Remark 6 (matrix ISLET vs. previous matrix recovery methods). If the order of tensor reduces to two, the tensor regression becomes the well-regarded *low-rank matrix recovery* in the literature [23, 97]:

$$y_i = \langle \mathbf{X}_i, \mathbf{A} \rangle + \varepsilon_i, \quad i = 1, \dots, n.$$

Here $\mathbf{A} \in \mathbb{R}^{p_1 \times p_2}$ is the unknown rank- r target matrix, $\{\mathbf{X}_i\}_{i=1}^n$ are design matrices, and $\varepsilon_i \sim N(0, \sigma^2)$ are noises. The low-rank matrix recovery, including its instances, such as phase retrieval [21], has been widely considered in recent literature. Various methods, such as nuclear norm minimization [22, 97], PGD [108], singular value thresholding [15], Procrustes flow [112], etc., have been introduced, and both the theoretical and computational performances have been extensively studied. Similar to the proof of Theorem 4, the upper bound for matrix ISLET estimator $\hat{\mathbf{A}}$ (Algorithm SM3.2 in the supplementary materials [128])

$$\|\hat{\mathbf{A}} - \mathbf{A}\|_F^2 \leq \frac{m}{n_2} \left(\sigma^2 + \frac{C_1 \tilde{\sigma}^4 mp}{n_1^2 \lambda_0^2} \right) \left(1 + C_2 \sqrt{\frac{\log p}{m}} + C_3 \sqrt{\frac{m \tilde{\sigma}^2}{(n_1 \wedge n_2) \lambda_0^2}} \right)$$

can be established with high probability. Here $m = (p_1 + p_2 - r)r$, $\lambda_0 = \sigma_r(\mathbf{A})$, $\tilde{\sigma}^2 = \|\mathbf{A}\|_F^2 + \sigma^2$. The lower bound similarly to Theorem 5 also holds.

4.2. Sparse tensor regression with importance sketching. We further consider the simultaneously sparse and low-rank tensor regression with Gaussian ensemble design. We have the following theoretical guarantee for sparse ISLET. For the same reason as for regular ISLET (see Remark 5), the sample splitting scheme is introduced in our technical analysis.

Theorem 6 (upper bounds for sparse tensor regression via ISLET). Consider the tensor regression model (1.1), where \mathcal{A} is simultaneously low-rank and sparse (1.3), \mathcal{X}_i has i.i.d. standard Gaussian entries, and $\varepsilon_i \stackrel{i.i.d.}{\sim} N(0, \sigma^2)$. Denote $\lambda_0 = \min_k \sigma_{r_k}(\mathcal{M}_k(\mathcal{A}))$, $s_k = p_k$ if $k \notin J_s$, $m_s = r_1 r_2 r_3 + \sum_{k \in J_s} s_k (r_k + \log p_k) + \sum_{k \notin J_s} p_k r_k$, and $\kappa = \max_k \|\mathcal{M}_k(\mathcal{A})\| / \sigma_{r_k}(\mathcal{M}_k(\mathcal{A}))$. We apply the proposed Algorithm 2.2 with sample splitting scheme (see Remark 5) and group Lasso penalty $\eta_k = C_0 \tilde{\sigma} \sqrt{n_2(r_k + \log(p_k))}$. If $\log(p_1) \asymp \log(p_2) \asymp \log(p_3) \asymp \log(p)$,

$$n_1 \geq \frac{C_1 \kappa^2 \tilde{\sigma}^2}{\lambda_0^2} \left(s_1 s_2 s_3 \log(p) + \sum_{k=1}^3 (s_k^2 r_k^2 + r_{k+1}^2 r_{k+2}^2) \right), \quad n_2 \geq \frac{C_2 m_s \kappa^2 \tilde{\sigma}^2}{\lambda_0^2},$$

and the output $\hat{\mathcal{A}}$ of sparse ISLET satisfies

$$(4.4) \quad \|\hat{\mathcal{A}} - \mathcal{A}\|_{HS}^2 \leq \frac{C_3 m_s}{n_2} \left(\sigma^2 + \frac{C_4 m_s \kappa^2 \tilde{\sigma}^2}{n_1} \right)$$

with probability at least $1 - p^{-C}$.

Proof. See Appendix SM6.5. ■

We further consider the following class of simultaneously sparse and low-rank tensors:

$$(4.5) \quad \mathcal{A}_{p,r,s} = \{ \mathcal{A} = [\mathcal{S}; \mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3] : \mathbf{U}_k \in \mathbb{O}_{p_k, r_k}, \|\mathbf{U}_k\|_{0,2} \leq s_k, k \in J_s \}.$$

The following minimax lower bound of the estimation risk holds in this class.

Theorem 7 (lower bounds). There exists constant $C > 0$ such that whenever $m_s \geq C$, the following lower bound holds for any arbitrary estimator $\hat{\mathcal{A}}$ based on $\{\mathcal{X}_i, y_i\}_{i=1}^n$:

$$(4.6) \quad \inf_{\mathcal{A}} \sup_{\mathcal{A} \in \mathcal{A}_{p,r,s}} \mathbb{E} \|\hat{\mathcal{A}} - \mathcal{A}\|_{HS}^2 \geq \frac{cm_s}{n} \sigma^2.$$

Proof. See Appendix SM6.6. ■

Combining Theorems 6 and 7, we can see the proposed procedure achieves optimal rate of convergence if $\frac{m_s \|\mathcal{A}\|_{\text{HS}}^2}{n_1 \sigma^2} = O(1)$ and $n_2 \asymp n$.

5. Numerical analysis. In this section, we conduct a simulation study to investigate the numerical performance of ISLET. In each study, we construct sensing tensors $\mathcal{X}_j \in \mathbb{R}^{p \times p \times p}$ with independent standard normal entries. In the nonsparse settings, using the Tucker decomposition we generate the core tensor $\mathcal{S} \in \mathbb{R}^{r \times r \times r}$ and $\mathbf{E}_k \in \mathbb{R}_{p,r}$ with i.i.d. Gaussian entries, the coefficient tensor $\mathcal{A} = [\mathcal{S}; \mathbf{E}_1; \mathbf{E}_2; \mathbf{E}_3]$; in the sparse settings, we construct \mathcal{S} and \mathcal{A} in the same way and generate \mathbf{E}_k as

$$(\mathbf{E}_k)_{[i,:]} = \begin{cases} (\bar{\mathbf{E}}_k)_{[j,:]}, & i \in \Omega_k, \text{ and } i \text{ is the } j\text{th element of } \Omega_k; \\ 0, & i \notin \Omega_k, \end{cases}$$

where Ω_k is a uniform random subset of $\{1, \dots, p\}$ with cardinality s_k and $\bar{\mathbf{E}}_k$ has s_k -by- r i.i.d. Gaussian entries. Finally, let the response $y_j = \langle \mathcal{X}_j, \mathcal{A} \rangle + \varepsilon_j$, $j = 1, 2, \dots, n$, where $\varepsilon_j \stackrel{i.i.d.}{\sim} N(0, \sigma^2)$. We report both the average root mean-squared error (RMSE) $\|\hat{\mathcal{A}} - \mathcal{A}\|_{\text{HS}} / \|\mathcal{A}\|_{\text{HS}}$ and the run time for each setting. Unless otherwise noted, the reported results are based on the average of 100 repeats and on a computer with Intel Xeon E5-2680 2.50GHz CPU. Additional simulation results of tuning-free ISLET and approximate low-rank tensor regression are collected in sections SM4 and SM5 in the supplementary materials [128].

Since we proposed to evaluate sketching directions and dimension-reduced regression (Steps 1 and 2 of Algorithm 2.1) both using the complete sample, but introduced a sample splitting scheme (Remark 5) to prove Theorems 4 and 6, we investigate how the sample splitting scheme affects the numerical performance of ISLET in this simulation setting. Let n vary from 1000 to 4000, $p = 10$, $r = 3, 5$, $\sigma = 5$. In addition to the original ISLET without splitting, we also implement sample-splitting ISLET, where a random $n_1 \approx \{\frac{3}{10}n, \frac{4}{10}n, \frac{5}{10}n\}$ samples are allocated for importance direction estimation (Step 1 of ISLET) and $n - n_1$ are allocated for dimension-reduced regression (Step 2 of ISLET). The results plotted in Figure 5 clearly show that the no-sample-splitting scheme yields much smaller estimation error than all sample-splitting approaches. Although the sample-splitting scheme brings advantages for our theoretical analyses for ISLET, it is not necessary in practice. Therefore, we will only perform ISLET without sample splitting for the rest of the simulation studies.

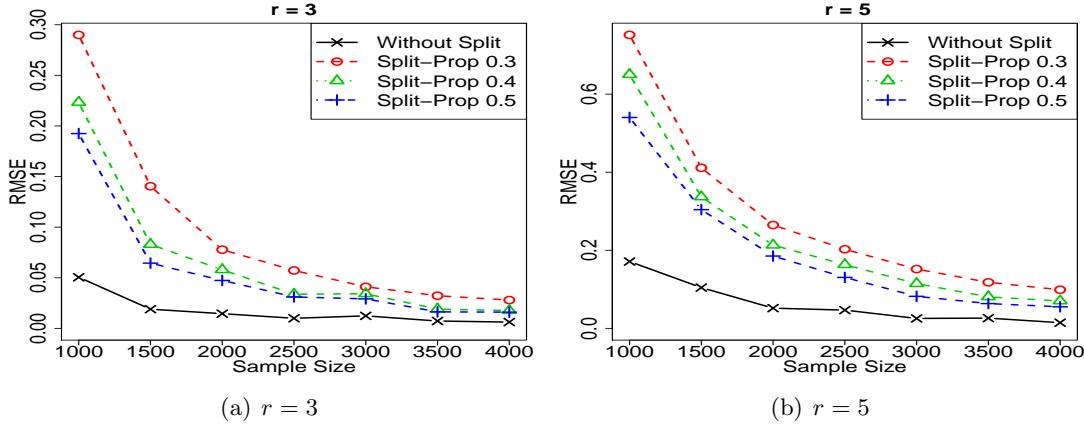


Figure 5. No-splitting vs. splitting ISLET: n varies from 1000 to 4000, $p = 10$, $r = 3, 5$, $\sigma = 5$.

We also compare the performance of nonsparse ISLET with a number of contemporary methods, including nonconvex projected gradient descent (nonconvex PGD) [27], Tucker low-rank regression via alternating gradient descent (Tucker regression)¹ [71, 132], and convex regularization low-rank tensor recovery (convex regularization)² [72, 96, 110]. We implement all four methods for $p = 10$, but only the ISLET and nonconvex projected PGD for $p = 50$, as the time cost of Tucker regression and convex regularization are beyond our computational limit if $p = 50$. Results for $p = 10$ and $p = 50$ are, respectively, plotted in panels (a) and (b) and panels (c) and (d) of Figure 6. Plots in Figures (a) and (c) show that the RMSEs of ISLET, tucker tensor regression, and nonconvex PGD are close, and all of them are slightly better than the convex regularization method; Figures 6(b) and (d) further indicate that ISLET is much faster than other methods—the advantage significantly increases as n and p grow. In particular, ISLET is about 10 times faster than nonconvex PGD when $p = 50$, $n = 12000$. In summary, the proposed ISLET achieves similar statistical performance within a significantly shorter time period comparing to the other state-of-the-art methods.

¹Software package downloaded at <https://hua-zhou.github.io/TensorReg/>

²The convex regularization aims to minimize the following objective function:

$$\sum_i^n \frac{1}{2n} (y_i - \langle \mathcal{X}_i, \mathcal{A} \rangle)^2 + \lambda \sum_{k=1}^3 \|\mathcal{M}_k(\mathcal{A})\|_*$$

Here, $\|\cdot\|_*$ is the matrix nuclear norm.

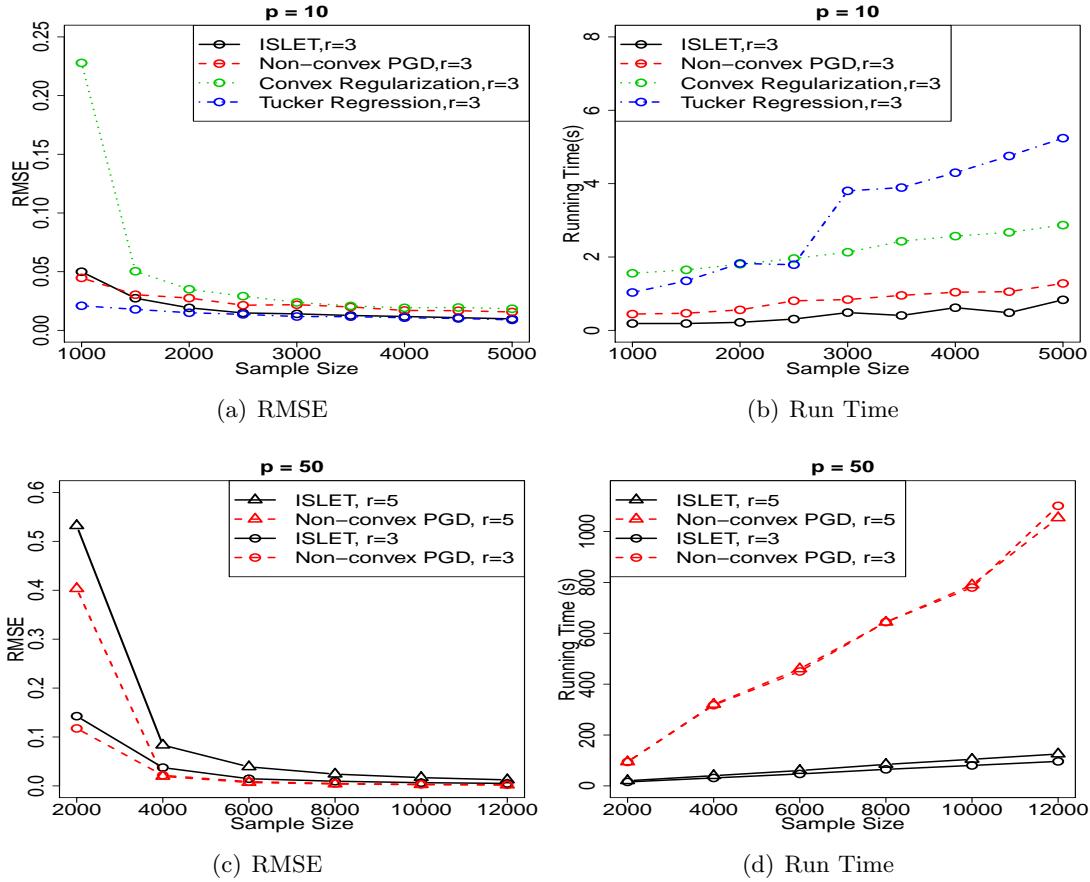


Figure 6. ISLET vs. nonconvex PGD, Tucker regression, and convex regularization. Here $\sigma = 5$; panels (a) and (b): $p = 10$; panels (c) and (d): $p = 50$.

Next, we investigate the performance of ISLET when p and n substantially grow. Let $p = 100, 150, 200$, $r = 3, 5$, $n \in [8000, 20000]$. The results in RMSE and run time are shown in Figures 7(a), (b), (c), and (d), respectively. We can see that the estimation error significantly decays as the sample size n grows, the dimension p decreases, or the Tucker rank r decreases.

We further fix $r = 2$, $n = 30000$ and let p grow to 400. Now the space cost for storing $\{\mathbf{X}_i\}_{i=1}^n$ reaches $400^3 \times 30000 \times 4\text{bytes} = 7.68$ terabytes, which is far beyond the volume of most personal computing devices. Since each sample is used only twice in ISLET, we perform this experiment in a parallel way. To be specific, in each machine $b = 1, \dots, 40$, we store the random seed, draw pseudorandom tensor \mathbf{X}_{bi} , evaluate y_{bi} and $\tilde{\mathbf{A}}_b$ by the procedure in section 2.2, and clean up the memory of \mathbf{X}_{bi} . After synchronizing the outcomes and obtaining the importance sketching directions, for each machine $b = 1, \dots, 40$, we generate pseudorandom covariates \mathbf{X}_{bi} again using the stored random seeds, evaluate $\tilde{\mathbf{G}}_b$ and $\tilde{\mathbf{X}}_{bi}$ by (2.8)–(2.9), and clean up the memory of \mathbf{X}_{bi} again. The rest of the procedure follows from section 2.2 and the original ISLET in Algorithm 2.1. The average RMSE and run time for five repeats are shown in Figure 8. We clearly see that ISLET yields good statistical performance within a

reasonable amount of time, while the other contemporary methods can hardly do so in such an ultrahigh-dimensional setting.

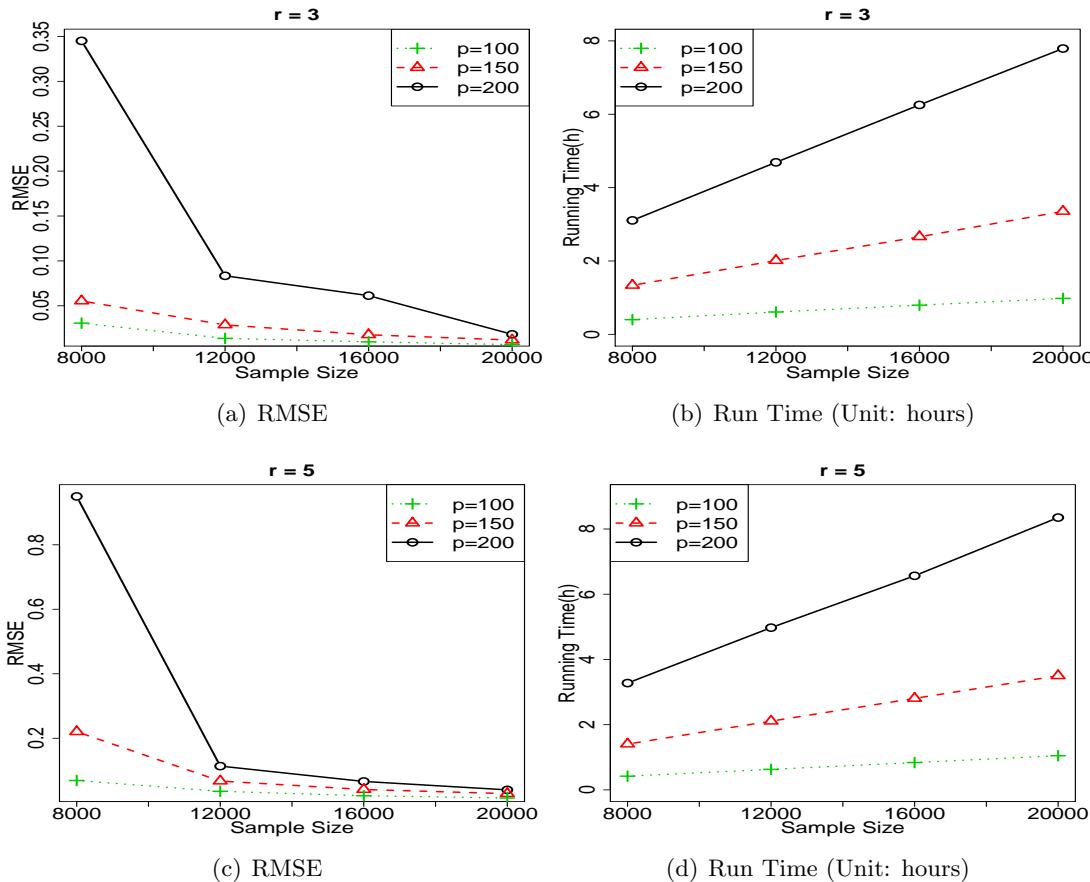


Figure 7. Performance of ISLET when p and n significantly grow.

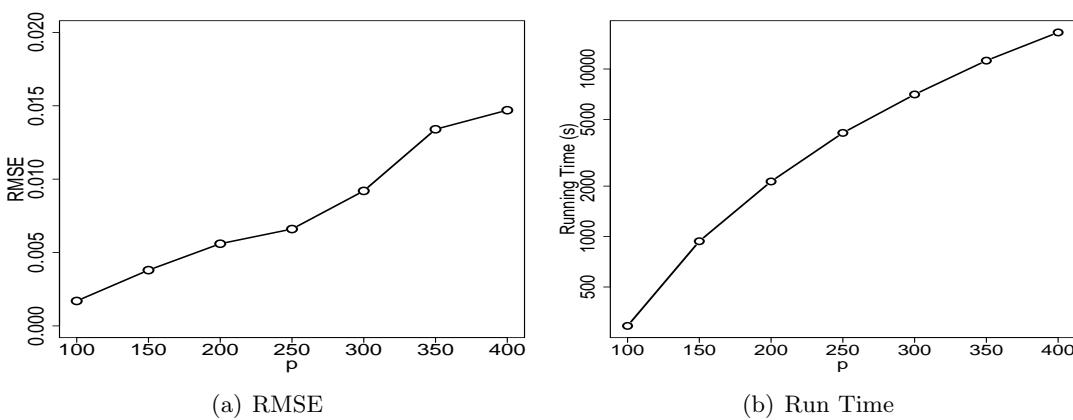


Figure 8. Performance of ISLET in ultrahigh-dimensional setting. p grows up to 400, $n = 30000$.

In addition, we explore the numerical performance of ISLET for simultaneously sparse and low-rank tensor regression. To perform sparse ISLET (Algorithm 2.2), we apply the *gglasso* package³ [122] for group Lasso and penalty level selection. Let n vary from 1500 to 4000, $p = 20, 25, 30$, $r = 3, 5$, $\sigma = 5$, $s = s_1 = s_2 = s_3 = 8$. The result is shown in Figure 9. Similar to the nonsparse ISLET, as sample size n increases or Tucker rank r decreases, the average estimation errors decrease.

We also compare sparse ISLET with slice-sparse nonconvex PGD proposed in [27]. Let $n \in [5000, 12000]$, $p = 50$, $r = 3, 5$, $\sigma = 5$, $s_1 = s_2 = s_3 = 15$. From Figure 10, we can see that ISLET yields much smaller estimation error with significantly shorter time than nonconvex PGD—the difference between two algorithms becomes more significant as n grows.

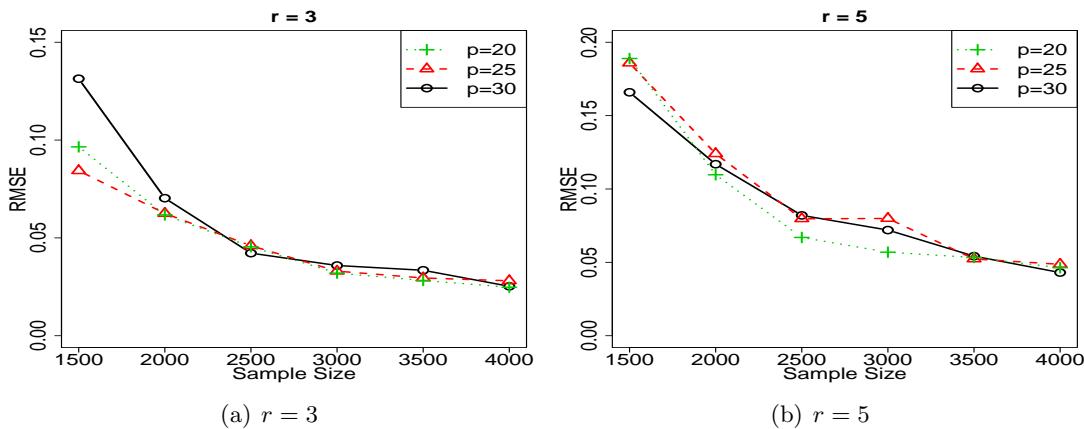


Figure 9. RMSE of ISLET for sparse and low-rank tensor recovery.

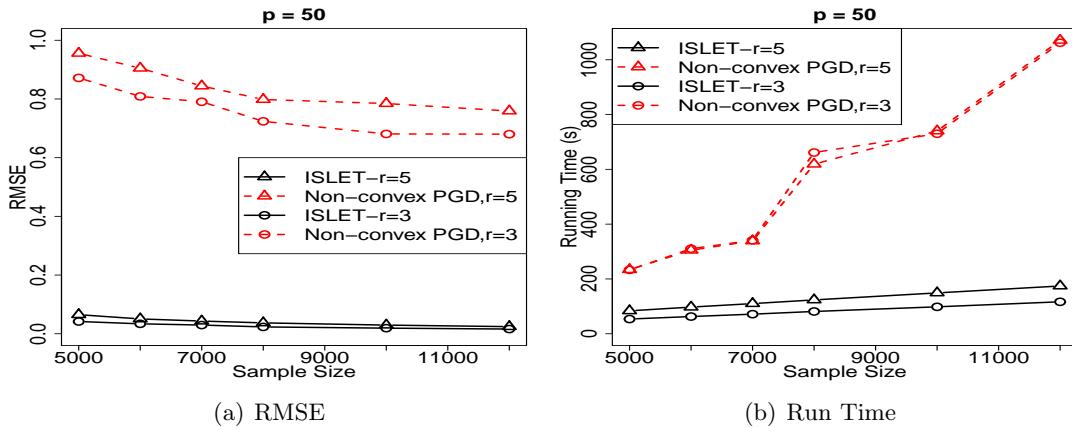


Figure 10. ISLET vs. nonconvex PGD for sparse tensor regression.

Finally, if the tensor is of order 2, tensor regression becomes the classic *low-rank matrix recovery* problem [23, 97]. Among existing approaches for low-rank matrix recovery, the

³Available online from <https://cran.r-project.org/web/packages/gglasso/index.html>.

nuclear norm minimization (NNM) has been proposed and extensively studied in recent literature. We compare the numerical performance of matrix ISLET (see Algorithm SM3.2 in section SM3 for implementation details) and NNM that aims to solve⁴

$$\sum_{i=1}^n (y_i - \langle \mathbf{X}_i, \mathbf{A} \rangle)^2 + \lambda \|\mathbf{A}\|_*,$$

where $\|\mathbf{A}\|_* = \sum_i \sigma_i(\mathbf{A})$ is the matrix nuclear norm. We consider two specific settings: (1) $p_1 = p_2 = 50$, $r = 2$, $\sigma = 10$, $n \in [2000, 16000]$; (2) $p_1 = p_2 = 100$, $r = 4$, $\sigma = 10$, $n \in [2000, 28000]$. From Figure 11, we find that ISLET has similar, or sometimes even better, performance than NNM in estimation error. On the other hand, the run time of ISLET is negligibly small compared to NNM.

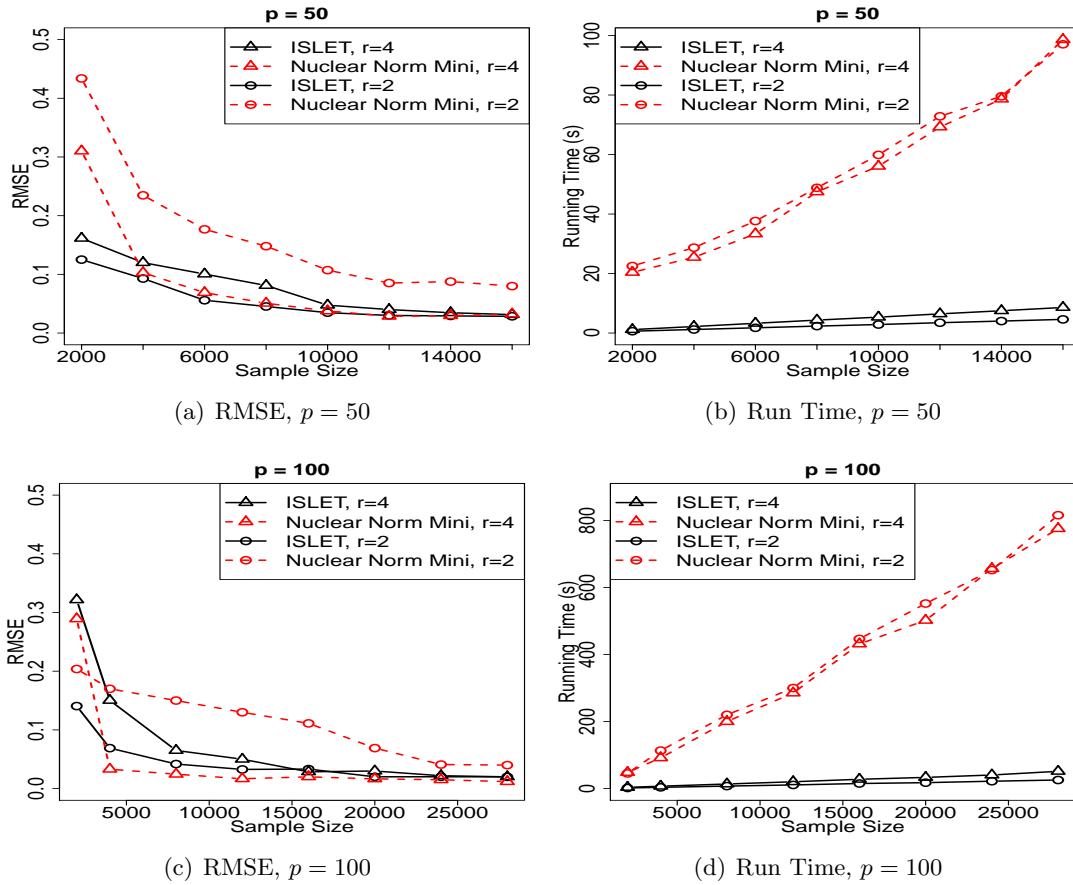


Figure 11. ISLET vs. NNM for low-rank matrix recovery.

6. Discussion. In this article, we develop a general importance sketching algorithm for high-dimensional low-rank tensor regression. In particular, to sufficiently reduce the dimension

⁴The optimization of NNM is implemented by accelerated proximal gradient method [108] using the software package available online from <https://blog.nus.edu.sg/mattohkc/softwares/nlsl/>.

of the higher-order structure, we propose a fast algorithm named *Importance Sketching Low-rank Estimation for Tensors* (ISLET). The proposed algorithm includes three major steps: we first apply tensor decomposition approaches, such as HOOI and STAT-SVD, to obtain importance sketching directions; then we perform regression using the sketched tensor/matrices (in the sparse case, we add group-sparsity regularizers); finally we assemble the final estimator. We establish deterministic oracle inequalities for the proposed procedure under general design and noise distributions. We also prove that ISLET achieves optimal mean-squared error rate under Gaussian ensemble design—regular ISLET can further achieve the optimal constant for mean-squared error. As illustrated in simulation studies, the proposed procedure is computationally efficient compared to contemporary methods. Although the presentation mainly focuses on order-3 tensors here, the method and theory for the general order- d tensors can be elaborated similarly.

It is also noteworthy that the storage cost for Tucker decomposition in the proposed procedure grows exponentially with the order d . Thus, if the target tensor has a large order, it is more desirable to consider other low-rank approximation methods than Tucker, such as the CP decomposition [12, 13], Hierarchical Tucker (HT) decomposition [7, 48, 52], Tensor Train (TT) decomposition [86, 89], etc. The ISLET framework can be adapted to these structures as long as there are two key components: there exist a sketching approach for dimension reduction and a computational inversion step for embedding the low-dimensional estimate back to the high-dimensional space (also see section 2.4). Whether these components hold for the previously described methods remains an interesting open question.

In addition to low-rank tensor regression, the idea of ISLET can be applied to various other high-dimensional problems. First, *high-order interaction pursuit* is an important topic in high-dimensional statistics that aims at the interaction among three or more variables in the regression setting. This problem can be transformed to the tensor estimation based on a number of rank-1 projections by the argument in [53]. Similarly to analysis on tensor regression in this paper, the idea of ISLET can be used to develop an optimal and efficient procedure for high-order interaction pursuit with provable advantages over other baseline methods.

In addition, *matrix/tensor completion* has attracted significant attention in recent literature [25, 72, 119, 120, 125]. The central task of matrix/tensor completion is to complete the low-rank matrix/tensor based on a limited number of observable entries. Since each observable entry in matrix/tensor completion can be seen as a special rank-one projection of the original matrix/tensor, the idea behind ISLET can be used to achieve a more efficient algorithm in matrix/tensor completion with theoretical guarantees. It will be an interesting future topic to further investigate the performance of ISLET on other high-dimensional problems.

Acknowledgments. The authors would like to thank the editors and anonymous referees for the useful suggestions that helped to improve the presentation of this paper.

REFERENCES

- [1] G. I. ALLEN, *Regularized Tensor Factorizations and Higher-Order Principal Components Analysis*, arXiv preprint, arXiv:1202.2476, 2012.
- [2] A. ANANDKUMAR, R. GE, D. HSU, S. M. KAKADE, AND M. TELGARSKY, *Tensor decompositions for learning latent variable models*, J. Mach. Learn. Res., 15 (2014), pp. 2773–2832.

- [3] H. AVRON, K. L. CLARKSON, AND D. P. WOODRUFF, *Sharper Bounds for Regression and Low-Rank Approximation with Regularization*, arXiv preprint, arXiv:1611.03225, 2016.
- [4] H. AVRON, H. NGUYEN, AND D. WOODRUFF, *Subspace embeddings for the polynomial kernel*, in Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA, 2014, pp. 2258–2266.
- [5] K. BALASUBRAMANIAN, J. FAN, AND Z. YANG, *Tensor Methods for Additive Index Models under Discordance and Heterogeneity*, arXiv preprint, arXiv:1807.06693, 2018.
- [6] N. BALDIN AND Q. BERTHET, *Optimal Link Prediction with Matrix Logistic Regression*, arXiv preprint, arXiv:1803.07054, 2018.
- [7] J. BALLANI AND L. GRASEDYCK, *A projection method to solve linear systems in tensor format*, Numer. Linear Algebra Appl., 20 (2013), pp. 27–43.
- [8] F. BAN, V. BHATTIPROLU, K. BRINGMANN, P. KOLEV, E. LEE, AND D. P. WOODRUFF, *A PTAs for ℓ_p -low rank approximation*, in Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, 2019, pp. 747–766, <https://doi.org/10.1137/1.9781611975482.47>.
- [9] M. BEBENDORF, *Adaptive cross approximation of multivariate functions*, Constr. Approx., 34 (2011), pp. 149–179.
- [10] G. BEYLKIN AND M. J. MOHLENKAMP, *Algorithms for numerical analysis in high dimensions*, SIAM J. Sci. Comput., 26 (2005), pp. 2133–2159, <https://doi.org/10.1137/040604959>.
- [11] X. BI, A. QU, AND X. SHEN, *Multilayer tensor factorization with applications to recommender systems*, Ann. Statist., 46 (2018), pp. 3308–3333.
- [12] M. BOUSSÉ, I. DOMANOV, AND L. DE LATHAUWER, *Linear Systems with a Multilinear Singular Value Decomposition Constrained Solution*, Tech. Rep., EESAT-STADIUS, KU Leuven, Leuven, Belgium, 2017.
- [13] M. BOUSSÉ, N. VERVLIET, I. DOMANOV, O. DEBALS, AND L. DE LATHAUWER, *Linear systems with a canonical polyadic decomposition constrained solution: Algorithms and applications*, Numer. Linear Algebra Appl., 25 (2018), e2190.
- [14] C. BOUTSIDIS AND D. P. WOODRUFF, *Optimal CUR matrix decompositions*, SIAM J. Comput., 46 (2017), pp. 543–589, <https://doi.org/10.1137/140977898>.
- [15] J.-F. CAI, E. J. CANDÈS, AND Z. SHEN, *A singular value thresholding algorithm for matrix completion*, SIAM J. Optim., 20 (2010), pp. 1956–1982, <https://doi.org/10.1137/080738970>.
- [16] T. CAI, T. T. CAI, AND A. ZHANG, *Structured matrix completion with applications to genomic data integration*, J. Amer. Statist. Assoc., 111 (2016), pp. 621–633.
- [17] T. T. CAI, X. LI, AND Z. MA, *Optimal rates of convergence for noisy sparse phase retrieval via thresholded Wirtinger flow*, Ann. Statist., 44 (2016), pp. 2221–2251.
- [18] T. T. CAI AND A. ZHANG, *ROP: Matrix recovery via rank-one projections*, Ann. Statist., 43 (2015), pp. 102–138.
- [19] C. F. CAIAFA AND A. CICHOCKI, *Generalizing the column–row matrix decomposition to multi-way arrays*, Linear Algebra Appl., 433 (2010), pp. 557–573.
- [20] R. CAMORIANO, T. ANGLES, A. RUDI, AND L. ROSASCO, *NYTRO: When subsampling meets early stopping*, in Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, 2016, pp. 1403–1411.
- [21] E. J. CANDÈS, X. LI, AND M. SOLTANOLKOTABI, *Phase retrieval via Wirtinger flow: Theory and algorithms*, IEEE Trans. Inform. Theory, 61 (2015), pp. 1985–2007.
- [22] E. J. CANDÈS AND Y. PLAN, *Matrix completion with noise*, Proc. IEEE, 98 (2010), pp. 925–936.
- [23] E. J. CANDÈS AND Y. PLAN, *Tight oracle inequalities for low-rank matrix recovery from a minimal number of noisy random measurements*, IEEE Trans. Inform. Theory, 57 (2011), pp. 2342–2359.
- [24] E. J. CANDÈS AND T. TAO, *Decoding by linear programming*, IEEE Trans. Inform. Theory, 51 (2005), pp. 4203–4215.
- [25] E. J. CANDÈS AND T. TAO, *The power of convex relaxation: Near-optimal matrix completion*, IEEE Trans. Inform. Theory, 56 (2010), pp. 2053–2080.
- [26] M. CHARIKAR, K. CHEN, AND M. FARACH-COLTON, *Finding frequent items in data streams*, Theoret. Comput. Sci., 312 (2004), pp. 3–15.
- [27] H. CHEN, G. RASKUTTI, AND M. YUAN, *Non-convex Projected Gradient Descent for Generalized Low-rank Tensor Regression*, arXiv preprint, arXiv:1611.10349, 2016.
- [28] Y. CHEN, Y. CHI, AND A. J. GOLDSMITH, *Exact and stable covariance estimation from quadratic*

- sampling via convex programming*, IEEE Trans. Inform. Theory, 61 (2015), pp. 4034–4059.
- [29] F. CHIERICHETTI, S. GOLLAPUDI, R. KUMAR, S. LATTANZI, R. PANIGRAHY, AND D. P. WOODRUFF, *Algorithms for ℓ_p low-rank approximation*, in Proceedings of the 34th International Conference on Machine Learning, Vol. 17, 2017, pp. 806–814.
- [30] 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*, IEEE Signal Process. Mag., 32 (2015), pp. 145–163.
- [31] K. L. CLARKSON AND D. P. WOODRUFF, *Input sparsity and hardness for robust subspace approximation*, in Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science, 2015, pp. 310–329.
- [32] K. L. CLARKSON AND D. P. WOODRUFF, *Low-rank approximation and regression in input sparsity time*, J. ACM, 63 (2017), 54.
- [33] G. DASARATHY, P. SHAH, B. N. BHASKAR, AND R. D. NOWAK, *Sketching sparse matrices, covariances, and graphs via tensor products*, IEEE Trans. Inform. Theory, 61 (2015), pp. 1373–1388.
- [34] L. DE LATHAUWER, B. DE MOOR, AND J. VANDEWALLE, *On the best rank-1 and rank- (r_1, r_2, \dots, r_n) approximation of higher-order tensors*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1324–1342, <https://doi.org/10.1137/S0895479898346995>.
- [35] H. DIAO, Z. SONG, W. SUN, AND D. WOODRUFF, *Sketching for Kronecker product regression and p-splines*, in Proceedings of the International Conference on Artificial Intelligence and Statistics, 2018, pp. 1299–1308.
- [36] E. DOBRIBAN AND S. LIU, *A New Theory for Sketching in Linear Regression*, arXiv preprint, arXiv:1810.06089, 2018.
- [37] P. DRINEAS, M. MAGDON-ISMAIL, M. W. MAHONEY, AND D. P. WOODRUFF, *Fast approximation of matrix coherence and statistical leverage*, J. Mach. Learn. Res., 13 (2012), pp. 3475–3506.
- [38] P. DRINEAS AND M. W. MAHONEY, *Effective Resistances, Statistical Leverage, and Applications to Linear Equation Solving*, arXiv preprint, arXiv:1005.3097, 2010.
- [39] L. ELDÉN AND B. SAVAS, *A Newton–Grassmann method for computing the best multilinear rank- (r_1, r_2, r_3) approximation of a tensor*, SIAM J. Matrix Anal. Appl., 31 (2009), pp. 248–271, <https://doi.org/10.1137/070688316>.
- [40] M. ESPIG, W. HACKBUSCH, T. ROHWEDDER, AND R. SCHNEIDER, *Variational calculus with sums of elementary tensors of fixed rank*, Numer. Math., 122 (2012), pp. 469–488.
- [41] J. FAN, W. GONG, AND Z. ZHU, *Generalized High-Dimensional Trace Regression via Nuclear Norm Regularization*, arXiv preprint, arXiv:1710.08083, 2017.
- [42] J. FAN AND J. LV, *Sure independence screening for ultrahigh dimensional feature space*, J. R. Stat. Soc. Ser. B. Stat. Methodol., 70 (2008), pp. 849–911.
- [43] J. FAN, W. WANG, AND Z. ZHU, *A Shrinkage Principle for Heavy-Tailed Data: High-Dimensional Robust Low-Rank Matrix Recovery*, arXiv preprint, arXiv:1603.08315, 2016.
- [44] J. FRIEDMAN, T. HASTIE, AND R. TIBSHIRANI, *A Note on the Goup Lasso and a Sparse Group Lasso*, arXiv preprint, arXiv:1001.0736, 2010.
- [45] R. GE, F. HUANG, C. JIN, AND Y. YUAN, *Escaping from saddle points–online stochastic gradient for tensor decomposition*, in Proceedings of the Conference on Learning Theory, 2015, pp. 797–842.
- [46] I. GEORGIEVA AND C. HOFREITHER, *Greedy low-rank approximation in Tucker format of solutions of tensor linear systems*, J. Comput. Appl. Math., 358 (2019), pp. 206–220.
- [47] S. A. GOREINOV, I. V. OSELEDETS, AND D. V. SAVOSTYANOV, *Wedderburn rank reduction and Krylov subspace method for tensor approximation. Part 1: Tucker case*, SIAM J. Sci. Comput., 34 (2012), pp. A1–A27, <https://doi.org/10.1137/100792056>.
- [48] L. GRASEDYCK, *Hierarchical singular value decomposition of tensors*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 2029–2054, <https://doi.org/10.1137/090764189>.
- [49] L. GRASEDYCK, D. KRESSNER, AND C. TOBLER, *A literature survey of low-rank tensor approximation techniques*, GAMM-Mitt., 36 (2013), pp. 53–78.
- [50] R. GUHANIYOGI, S. QAMAR, AND D. B. DUNSON, *Bayesian Tensor Regression*, arXiv preprint, arXiv:1509.06490, 2015.
- [51] W. GUO, I. KOTSIA, AND I. PATRAS, *Tensor learning for regression*, IEEE Trans. Image Process., 21 (2012), pp. 816–827.

- [52] W. HACKBUSCH AND S. KÜHN, *A new scheme for the tensor representation*, J. Fourier Anal. Appl., 15 (2009), pp. 706–722.
- [53] B. HAO, A. ZHANG, AND G. CHENG, *Sparse and Low-Rank Tensor Estimation via Cubic Sketchings*, arXiv preprint, arXiv:1801.09326, 2018.
- [54] J. HAUPPT, X. LI, AND D. P. WOODRUFF, *Near Optimal Sketching of Low-Rank Tensor Regression*, arXiv preprint, arXiv:1709.07093, 2017.
- [55] S. HE, J. YIN, H. LI, AND X. WANG, *Graphical model selection and estimation for high dimensional tensor data*, J. Multivariate Anal., 128 (2014), pp. 165–185.
- [56] P. D. HOFF, *Multilinear tensor regression for longitudinal relational data*, Ann. Appl. Stat., 9 (2015), pp. 1169–1193.
- [57] C. HOFREITHER, *A black-box low-rank approximation algorithm for fast matrix assembly in isogeometric analysis*, Comput. Methods Appl. Mech. Engrg., 333 (2018), pp. 311–330.
- [58] T. J. HUGHES, J. A. COTTRELL, AND Y. BAZILEVS, *Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement*, Comput. Methods Appl. Mech. Engrg., 194 (2005), pp. 4135–4195.
- [59] M. ISHTEVA, P.-A. ABSIL, S. VAN HUFFEL, AND L. DE LATHAUWER, *Best low multilinear rank approximation of higher-order tensors, based on the Riemannian trust-region scheme*, SIAM J. Matrix Anal. Appl., 32 (2011), pp. 115–135, <https://doi.org/10.1137/090764827>.
- [60] M. ISHTEVA, L. DE LATHAUWER, P.-A. ABSIL, AND S. VAN HUFFEL, *Differential-geometric Newton method for the best rank- (r_1, r_2, r_3) approximation of tensors*, Numer. Algorithms, 51 (2009), pp. 179–194.
- [61] M. JANZAMIN, H. SEDGHI, AND A. ANANDKUMAR, *Score Function Features for Discriminative Learning: Matrix and Tensor Framework*, arXiv preprint, arXiv:1412.2863, 2014.
- [62] D. M. KANE AND J. NELSON, *Sparser Johnson-Lindenstrauss transforms*, J. ACM, 61 (2014), 4.
- [63] T. G. KOLDA AND B. W. BADER, *Tensor decompositions and applications*, SIAM Rev., 51 (2009), pp. 455–500, <https://doi.org/10.1137/07070111X>.
- [64] V. KOLTCHINSKII, K. LOUNICI, AND A. B. TSYBAKOV, *Nuclear-norm penalization and optimal rates for noisy low-rank matrix completion*, Ann. Statist., 39 (2011), pp. 2302–2329.
- [65] D. KRESSNER, M. STEINLECHNER, AND B. VANDEREYCKEN, *Preconditioned low-rank Riemannian optimization for linear systems with tensor product structure*, SIAM J. Sci. Comput., 38 (2016), pp. A2018–A2044, <https://doi.org/10.1137/15M1032909>.
- [66] D. KRESSNER AND C. TOBLER, *Krylov subspace methods for linear systems with tensor product structure*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 1688–1714, <https://doi.org/10.1137/090756843>.
- [67] P. M. KROONENBERG, *Applied Multiway Data Analysis*, Wiley Ser. Probab. Stat. 702, John Wiley & Sons, New York, 2008.
- [68] J. D. LEE, B. RECHT, N. SREBRO, J. TROPP, AND R. R. SALAKHUTDINOV, *Practical large-scale optimization for max-norm regularization*, in Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA, 2010, pp. 1297–1305.
- [69] L. LI AND X. ZHANG, *Parsimonious tensor response regression*, J. Amer. Statist. Assoc. 112 (2017), pp. 1131–1146.
- [70] N. LI AND B. LI, *Tensor completion for on-board compression of hyperspectral images*, in Proceedings of the IEEE International Conference on Image Processing, 2010, pp. 517–520.
- [71] X. LI, D. XU, H. ZHOU, AND L. LI, *Tucker tensor regression and neuroimaging analysis*, Statist. Biosci., 10 (2018), pp. 520–545.
- [72] J. LIU, P. MUSIALSKI, P. WONKA, AND J. YE, *Tensor completion for estimating missing values in visual data*, IEEE Trans. Pattern Anal. Mach. Intell., 35 (2013), pp. 208–220.
- [73] K. LOUNICI, M. PONTIL, S. VAN DE GEER, AND A. B. TSYBAKOV, *Oracle inequalities and optimal inference under group sparsity*, Ann. Statist., 39 (2011), pp. 2164–2204.
- [74] R. LYNCH, J. R. RICE, AND D. H. THOMAS, *Tensor product analysis of partial difference equations*, Bull. Amer. Math. Soc., 70 (1964), pp. 378–384.
- [75] X. LYU, W. W. SUN, Z. WANG, H. LIU, J. YANG, AND G. CHENG, *Tensor graphical model: Non-convex optimization and statistical inference*, IEEE Trans. Pattern Anal. Mach. Intell., (2019), <https://doi.org/10.1109/tpami.2019.2907679>.
- [76] M. W. MAHONEY, *Randomized algorithms for matrices and data*, Found. Trends Mach. Learn., 3 (2011),

- pp. 123–224.
- [77] M. W. MAHONEY, M. MAGGIONI, AND P. DRINEAS, *Tensor-CUR decompositions for tensor-based data*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 957–987, <https://doi.org/10.1137/060665336>.
- [78] A. M. MANCEUR AND P. DUTILLEUL, *Maximum likelihood estimation for the tensor normal distribution: Algorithm, minimum sample size, and empirical bias and dispersion*, J. Comput. Appl. Math., 239 (2013), pp. 37–49.
- [79] P. P. MARKOPOULOS, G. N. KARYSTINOS, AND D. A. PADOS, *Optimal algorithms for $l_{\{1\}}$ -subspace signal processing*, IEEE Trans. Signal Process., 62 (2014), pp. 5046–5058.
- [80] P. P. MARKOPOULOS, S. KUNDU, S. CHAMADIA, AND D. A. PADOS, *Efficient l_1 -norm principal-component analysis via bit flipping*, IEEE Trans. Signal Process., 65 (2017), pp. 4252–4264.
- [81] D. MENG, Z. XU, L. ZHANG, AND J. ZHAO, *A cyclic weighted median method for l_1 low-rank matrix factorization with missing entries*, in Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, 2013.
- [82] X. MENG AND M. W. MAHONEY, *Low-distortion subspace embeddings in input-sparsity time and applications to robust linear regression*, in Proceedings of the 45th Annual ACM Symposium on Theory of Computing, 2013, pp. 91–100.
- [83] A. MONTANARI AND N. SUN, *Spectral Algorithms for Tensor Completion*, arXiv preprint, arXiv:1612.07866, 2016.
- [84] C. MU, B. HUANG, J. WRIGHT, AND D. GOLDFARB, *Square deal: Lower bounds and improved relaxations for tensor recovery*, in Proceedings of the International Conference on Machine Learning, 2014, pp. 73–81.
- [85] J. NELSON AND H. L. NGUYÉN, *OSNAP: Faster numerical linear algebra algorithms via sparser subspace embeddings*, in Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science, 2013, pp. 117–126.
- [86] I. V. OSELEDETS, *Tensor-train decomposition*, SIAM J. Sci. Comput., 33 (2011), pp. 2295–2317, <https://doi.org/10.1137/090752286>.
- [87] I. V. OSELEDETS, D. SAVOSTIANOV, AND E. E. TYRTYSHNIKOV, *Tucker dimensionality reduction of three-dimensional arrays in linear time*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 939–956, <https://doi.org/10.1137/060655894>.
- [88] I. V. OSELEDETS, D. V. SAVOSTYANOV, AND E. E. TYRTYSHNIKOV, *Cross approximation in tensor electron density computations*, Numer. Linear Algebra Appl., 17 (2010), pp. 935–952.
- [89] I. V. OSELEDETS AND E. E. TYRTYSHNIKOV, *Breaking the curse of dimensionality, or how to use SVD in many dimensions*, SIAM J. Sci. Comput., 31 (2009), pp. 3744–3759, <https://doi.org/10.1137/090748330>.
- [90] R. PAGH, *Compressed matrix multiplication*, ACM Trans. Comput. Theory, 5 (2013), 9.
- [91] Y. PAN, Q. MAI, AND X. ZHANG, *Covariate-adjusted tensor classification in high dimensions*, J. Amer. Statist. Assoc., 114 (2019), pp. 1305–1319.
- [92] N. PHAM AND R. PAGH, *Fast and scalable polynomial kernels via explicit feature maps*, in Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2013, pp. 239–247.
- [93] M. PILANCI AND M. J. WAINWRIGHT, *Randomized sketches of convex programs with sharp guarantees*, IEEE Trans. Inform. Theory, 61 (2015), pp. 5096–5115.
- [94] M. PILANCI AND M. J. WAINWRIGHT, *Iterative Hessian sketch: Fast and accurate solution approximation for constrained least-squares*, J. Mach. Learn. Res., 17 (2016), pp. 1842–1879.
- [95] G. RASKUTTI AND M. MAHONEY, *A Statistical Perspective on Randomized Sketching for Ordinary Least-squares*, arXiv preprint, arXiv:1406.5986, 2014.
- [96] G. RASKUTTI, M. YUAN, AND H. CHEN, *Convex Regularization for High-Dimensional Multi-response Tensor Regression*, arXiv preprint, arXiv:1512.01215, 2015.
- [97] B. RECHT, M. FAZEL, AND P. A. PARRILLO, *Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization*, SIAM Rev., 52 (2010), pp. 471–501, <https://doi.org/10.1137/070697835>.
- [98] B. SAVAS AND L. ELDÉN, *Krylov-type methods for tensor computations I*, Linear Algebra Appl., 438 (2013), pp. 891–918.
- [99] B. SAVAS AND L.-H. LIM, *Quasi-Newton methods on Grassmannians and multilinear approximations of*

- tensors*, SIAM J. Sci. Comput., 32 (2010), pp. 3352–3393, <https://doi.org/10.1137/090763172>.
- [100] N. D. SIDIROPOULOS, L. DE LATHAUWER, X. FU, K. HUANG, E. E. PAPALEXAKIS, AND C. FALOUTSOS, *Tensor decomposition for signal processing and machine learning*, IEEE Trans. Signal Process., 65 (2017), pp. 3551–3582.
- [101] N. D. SIDIROPOULOS AND A. KYRILLIDIS, *Multi-way compressed sensing for sparse low-rank tensors*, IEEE Signal Process. Lett., 19 (2012), pp. 757–760.
- [102] N. D. SIDIROPOULOS, E. E. PAPALEXAKIS, AND C. FALOUTSOS, *Parallel randomly compressed cubes: A scalable distributed architecture for big tensor decomposition*, IEEE Signal Process. Mag., 31 (2014), pp. 57–70.
- [103] Z. SONG, D. P. WOODRUFF, AND P. ZHONG, *Low rank approximation with entrywise l_1 -norm error*, in Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, 2017, pp. 688–701.
- [104] Z. SONG, D. P. WOODRUFF, AND P. ZHONG, *Relative error tensor low rank approximation*, in Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, 2019, pp. 2772–2789, <https://doi.org/10.1137/1.9781611975482.172>.
- [105] W. W. SUN AND L. LI, *Sparse Low-Rank Tensor Response Regression*, arXiv preprint, arXiv:1609.04523, 2016.
- [106] W. W. SUN AND L. LI, *Store: Sparse tensor response regression and neuroimaging analysis*, J. Mach. Learn. Res., 18 (2017), pp. 4908–4944.
- [107] Y. SUN, Y. GUO, C. LUO, J. TROPP, AND M. UDELL, *Low-Rank Tucker Approximation of a Tensor from Streaming Data*, arXiv preprint, arXiv:1904.10951, 2019.
- [108] K.-C. TOH AND S. YUN, *An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems*, Pac. J. Optim., 6 (2010), pp. 615–640.
- [109] R. TOMIOKA AND T. SUZUKI, *Convex tensor decomposition via structured Schatten norm regularization*, in Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA, 2013, pp. 1331–1339.
- [110] R. TOMIOKA, T. SUZUKI, K. HAYASHI, AND H. KASHIMA, *Statistical performance of convex tensor decomposition*, in Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA, 2011, pp. 972–980.
- [111] J. A. TROPP, A. YURTSEVER, M. UDELL, AND V. CEVHER, *Practical sketching algorithms for low-rank matrix approximation*, SIAM J. Matrix Anal. Appl., 38 (2017), pp. 1454–1485, <https://doi.org/10.1137/17M1111590>.
- [112] S. TU, R. BOČZAR, M. SIMCHOWITZ, M. SOLTANOLKOTABI, AND B. RECHT, *Low-rank solutions of linear matrix equations via procrustes flow*, in Proceedings of the International Conference on Machine Learning, 2016, pp. 964–973.
- [113] L. R. TUCKER, *Some mathematical notes on three-mode factor analysis*, Psychometrika, 31 (1966), pp. 279–311.
- [114] M. UDELL AND A. TOWNSEND, *Why are big data matrices approximately low rank?*, SIAM J. Math. Data Sci., 1 (2019), pp. 144–160, <https://doi.org/10.1137/18M1183480>.
- [115] N. VERVLIET AND L. DE LATHAUWER, *A randomized block sampling approach to canonical polyadic decomposition of large-scale tensors*, IEEE J. Sel. Topics Signal Process., 10 (2015), pp. 284–295.
- [116] J. WANG, J. D. LEE, M. MAHDavi, M. KOLAR, AND N. SREBRO, *Sketching meets random projection in the dual: A provable recovery algorithm for big and high-dimensional data*, Electron. J. Stat., 11 (2017), pp. 4896–4944.
- [117] Y. WANG, H.-Y. TUNG, A. J. SMOLA, AND A. ANANDKUMAR, *Fast and guaranteed tensor decomposition via sketching*, in Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA, 2015, pp. 991–999.
- [118] D. P. WOODRUFF, *Sketching as a tool for numerical linear algebra*, Found. Trends Theoret. Comput. Sci., 10 (2014), pp. 1–157.
- [119] D. XIA AND M. YUAN, *On Polynomial Time Methods for Exact Low Rank Tensor Completion*, arXiv preprint, arXiv:1702.06980, 2017.
- [120] D. XIA, M. YUAN, AND C.-H. ZHANG, *Statistically Optimal and Computationally Efficient Low Rank Tensor Completion from Noisy Entries*, arXiv preprint, arXiv:1711.04934, 2017.
- [121] L. XUE AND H. ZOU, *Sure independence screening and compressed random sensing*, Biometrika, 98

- (2011), pp. 371–380.
- [122] Y. YANG AND H. ZOU, *A fast unified algorithm for solving group-lasso penalize learning problems*, Stat. Comput., 25 (2015), pp. 1129–1141.
- [123] M. YU, Z. WANG, V. GUPTA, AND M. KOLAR, *Recovery of Simultaneous Low Rank and Two-Way Sparse Coefficient Matrices, a Nonconvex Approach*, arXiv preprint, arXiv:1802.06967, 2018.
- [124] M. YUAN AND Y. LIN, *Model selection and estimation in regression with grouped variables*, J. R. Stat. Soc. Ser. B. Stat. Methodol., 68 (2006), pp. 49–67.
- [125] M. YUAN AND C.-H. ZHANG, *On tensor completion via nuclear norm minimization*, Found. Comput. Math., 16 (2016), pp. 1031–1068.
- [126] A. ZHANG, *Cross: Efficient low-rank tensor completion*, Ann. Statist., 47 (2019), pp. 936–964.
- [127] A. ZHANG AND R. HAN, *Optimal sparse singular value decomposition for high-dimensional high-order data*, J. Amer. Statist. Assoc., 114 (2019), pp. 1708–1725.
- [128] A. ZHANG, Y. LUO, G. RASKUTTI, AND M. YUAN, *Supplement to “ISLET: Fast and optimal low-rank tensor regression via importance sketching,”* 2018.
- [129] A. ZHANG, Y. LUO, G. RASKUTTI, AND M. YUAN, *A Sharp Blockwise Tensor Perturbation Bound for Higher-order Orthogonal Iteration*, preprint, 2019.
- [130] L. ZHANG, M. MAHDAVI, R. JIN, T. YANG, AND S. ZHU, *Random projections for classification: A recovery approach*, IEEE Trans. Inform. Theory, 60 (2014), pp. 7300–7316.
- [131] Y. ZHENG, G. LIU, S. SUGIMOTO, S. YAN, AND M. OKUTOMI, *Practical low-rank matrix approximation under robust l_1 -norm*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 1410–1417.
- [132] H. ZHOU, L. LI, AND H. ZHU, *Tensor regression with applications in neuroimaging data analysis*, J. Amer. Statist. Assoc., 108 (2013), pp. 540–552.
- [133] S. ZHOU, *Gemini: Graph estimation with matrix variate normal instances*, Ann. Statist., 42 (2014), pp. 532–562.