Low-rank Tensor Estimation via Riemannian Gauss-Newton: Statistical Optimality and Second-Order Convergence

Yuetian Luo   YUETIAN@UCHICAGO.EDU
Data Science Institute
University of Chicago
Chicago, IL 60637, USA

Anru R. Zhang   ANRU.ZHANG@DUKE.EDU
Departments of Biostatistics & Bioinformatics and Computer Science
Duke University
Durham, NC 27710, USA

Editor: Suvrit Sra

Abstract

In this paper, we consider the estimation of a low Tucker rank tensor from a number of noisy linear measurements. The general problem covers many specific examples arising from applications, including tensor regression, tensor completion, and tensor PCA/SVD. We consider an efficient Riemannian Gauss-Newton (RGN) method for low Tucker rank tensor estimation. Different from the generic (super)linear convergence guarantee of RGN in the literature, we prove the first local quadratic convergence guarantee of RGN for low-rank tensor estimation in the noisy setting under some regularity conditions and provide the corresponding estimation error upper bounds. A deterministic estimation error lower bound, which matches the upper bound, is provided that demonstrates the statistical optimality of RGN. The merit of RGN is illustrated through two machine learning applications: tensor regression and tensor SVD. Finally, we provide the simulation results to corroborate our theoretical findings.

Keywords: Low-rank tensor estimation, quadratic convergence, Riemannian optimization, statistical optimality.

1 Introduction

The past decades have seen a large body of work on tensors or multiway arrays in applied mathematics, signal processing, machine learning, statistics, among many other fields. Tensors arise in numerous applications involving multiway data, such as brain imaging (Zhou et al., 2013; Zhang et al., 2019), electron microscopy imaging (Han et al., 2022b; Zhang et al., 2020b), recommender system design (Bi et al., 2018). 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 (Anandkumar et al., 2014a), additive index models (Balasubramanian et al., 2018), high-order interaction pursuit (Hao et al., 2020). In this paper, we focus on a prototypical model for tensor estimation:

$$\mathbf{y} = \mathscr{A}(\boldsymbol{\mathcal{X}}^*) + \boldsymbol{\varepsilon}. \tag{1}$$

Here, $\mathbf{y}, \boldsymbol{\varepsilon} \in \mathbb{R}^n$ are the observations and unknown noise and $\boldsymbol{\mathcal{X}}^* \in \mathbb{R}^{p_1 \times \cdots \times p_d}$ is an order-d parameter tensor of interest. $\mathscr{A} : \mathbb{R}^{p_1 \times \cdots \times p_d} \to \mathbb{R}^n$ is a known linear map, which can be explicitly expressed as

$$\mathscr{A}(\boldsymbol{\mathcal{X}}^*) = [\langle \boldsymbol{\mathcal{A}}_1, \boldsymbol{\mathcal{X}}^* \rangle, \ldots, \langle \boldsymbol{\mathcal{A}}_n, \boldsymbol{\mathcal{X}}^* \rangle]^\top, \langle \boldsymbol{\mathcal{A}}_i, \boldsymbol{\mathcal{X}}^* \rangle = \sum_{1 \leqslant i_k \leqslant p_k, 1 \leqslant k \leqslant d} (\boldsymbol{\mathcal{A}}_i)_{[i_1, \ldots, i_d]} \boldsymbol{\mathcal{X}}^*_{[i_1, \ldots, i_d]} \quad (2)$$

with the given measurement tensors $\{\boldsymbol{\mathcal{A}}_i\}_{i=1}^n \subseteq \mathbb{R}^{p_1 \times \cdots \times p_d}$. Our goal is to estimate $\boldsymbol{\mathcal{X}}^*$ based on $(\mathbf{y}, \mathscr{A})$. When $\boldsymbol{\varepsilon} = 0$, (1) becomes the low-rank tensor recovery problem (Rauhut et al., 2017) where the aim is to recover $\boldsymbol{\mathcal{X}}^*$ exactly.

In many applications, $\prod_{k=1}^d p_k$, i.e., the number of parameters in $\boldsymbol{\mathcal{X}}^*$, is much greater than the sample size n, so some structural conditions are often assumed to ensure the problem is well-posed. In the literature, the low-rank assumption was widely considered (Kolda and Bader, 2009; Zhou et al., 2013; Anandkumar et al., 2014b; Richard and Montanari, 2014; Montanari and Sun, 2018). In this work, we focus on the setting that the target parameter $\boldsymbol{\mathcal{X}}^*$ is low Tucker rank and admits the following Tucker (or multilinear) decomposition with Tucker rank $\mathbf{r} = (r_1, \ldots, r_d)$:

$$\boldsymbol{\mathcal{X}}^* = \boldsymbol{\mathcal{S}} \times_1 \mathbf{U}_1 \times \cdots \times_d \mathbf{U}_d. \quad (3)$$

Here, $\boldsymbol{\mathcal{S}} \in \mathbb{R}^{r_1 \times \cdots \times r_d}$ is the order-d core tensor; $\mathbf{U}_k$ is a p_k-by-r_k matrix with orthonormal columns, which represents the mode-k singular vectors of $\boldsymbol{\mathcal{X}}^*$; "$\times_k$" is the tensor-matrix product along mode k. The formal definitions of Tucker decomposition and tensor-matrix product are given in Section 1.4.

With different designs of $\mathscr{A}$, the general model (1) covers many specific settings arising from applications, such as recommender system (Bi et al., 2018), neuroimaging (Guhaniyogi et al., 2017; Li and Zhang, 2017), longitudinal relational data analysis (Hoff, 2015), imaging processing (Guo et al., 2012). The specific settings of model (1) include:

- **Tensor regression with general random or deterministic design** (Zhou et al., 2013; Raskutti et al., 2019), where $\boldsymbol{\mathcal{A}}_i$ are general tensors. Specifically, the Gaussian ensemble design ($\boldsymbol{\mathcal{A}}_i$ has i.i.d. Gaussian/sub-Gaussian entries) is widely studied in the literature.

- **Tensor completion** (Gandy et al., 2011; Liu et al., 2013; Yuan and Zhang, 2014): $\boldsymbol{\mathcal{A}}_i = \mathbf{e}_{a_1^{(i)}} \circ \cdots \circ \mathbf{e}_{a_d^{(i)}}$, $\mathbf{e}_{a_k^{(i)}}$ is the $a_k^{(i)}$th canonical vector and $\{a_1^{(i)}, \cdots, a_d^{(i)}\}_{i=1}^n$ are randomly selected integers from $[p_1] \times \cdots \times [p_d]$, "$\circ$" represents the outer product and $[p_k] = \{1, \ldots, p_k\}$;

- **Tensor estimation via rank-1 projections** (Hao et al., 2020): $\boldsymbol{\mathcal{A}}_i = \mathbf{a}_1^{(i)} \circ \cdots \circ \mathbf{a}_d^{(i)}$, where $\{\mathbf{a}_k^{(i)} \in \mathbb{R}^{p_k}\}_{k=1}^d$ are random vectors;

- **Tensor PCA/SVD** (Richard and Montanari, 2014; Hopkins et al., 2015; Zhang and Xia, 2018; Perry et al., 2020) is a special case of tensor completion where all entries are observable. In this particular setting, we can tensorize $\mathbf{y}, \boldsymbol{\varepsilon}$ and rewrite the model (1) equivalently to $\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{X}}^* + \boldsymbol{\mathcal{E}}$. Here $\boldsymbol{\mathcal{X}}^*$ is the low Tucker rank signal tensor and $\boldsymbol{\mathcal{E}}$ is the noise.

In view of model (1) and assumption (3), a natural estimator of $\boldsymbol{\mathcal{X}}^*$ is

$$\widehat{\boldsymbol{\mathcal{X}}} = \operatorname*{arg\,min}_{\boldsymbol{\mathcal{X}} \in \mathbb{R}^{p_1 \times \cdots \times p_d}} f(\boldsymbol{\mathcal{X}}) := \frac{1}{2} \left\| \mathbf{y} - \mathscr{A}(\boldsymbol{\mathcal{X}}) \right\|_2^2, \quad \text{subject to} \quad \text{Tucrank}(\boldsymbol{\mathcal{X}}) = \mathbf{r}. \qquad (4)$$

Here Tucrank$(\boldsymbol{\mathcal{X}})$ is the Tucker rank of $\boldsymbol{\mathcal{X}}$ (see definition in Section 1.4). However, the optimization problem in (4) is non-convex and NP-hard in general. To tame the non-convexity, a common scheme is the convex relaxation (Mu et al., 2014; Raskutti et al., 2019; Tomioka et al., 2011). However, this scheme may either obtain suboptimal statistical guarantees or require evaluating the tensor nuclear norm, which is NP-hard to compute in general (Hillar and Lim, 2013). Alternatively, a large body of literature turns to the non-convex formulation and focuses on developing computationally efficient two-stage procedures for estimating $\boldsymbol{\mathcal{X}}^*$: first, one obtains a warm initialization of $\boldsymbol{\mathcal{X}}^*$ and then runs local algorithms to refine the estimate. Provable guarantees on estimation or recovery of $\boldsymbol{\mathcal{X}}^*$ for such a two-stage paradigm have been developed in different scenarios (Rauhut et al., 2017; Chen et al., 2019a; Ahmed et al., 2020; Han et al., 2022b; Cai et al., 2019; Hao et al., 2020; Cai et al., 2020; Xia et al., 2021). In this work, we focus on the non-convex formulation and aim to develop a provable computationally efficient estimator for $\boldsymbol{\mathcal{X}}^*$. Departing from the existing literature that focuses on the first-order local methods, we consider a Riemannian Gauss-Newton (RGN) algorithm for iterative refinement and establish the first quadratic convergence guarantee on the estimation of $\boldsymbol{\mathcal{X}}^*$.

1.1 Our Contributions

In this paper, we develop a new Riemannian Gauss-Newton (RGN) algorithm for low-rank tensor estimation. The proposed algorithm is tuning-free and generally has the same per-iteration computational complexity as the alternating minimization (Zhou et al., 2013; Li et al., 2018) and comparable complexity to the other first-order methods including projected gradient descent (Chen et al., 2019a) and gradient descent (Han et al., 2022b).

Moreover, assuming $\mathscr{A}$ satisfies the tensor restricted isometry property (TRIP) (see Definition 2), we prove that with some proper initialization, the iterates generated by RGN converge quadratically to $\boldsymbol{\mathcal{X}}^*$ up to some statistical error. Especially in the noiseless setting, i.e., $\boldsymbol{\varepsilon} = 0$, RGN converges quadratically to the exact parameter $\boldsymbol{\mathcal{X}}^*$. Figure 1 shows the numerical performance of RGN in tensor regression (left panel) and tensor completion (right panel): in the noiseless case, RGN converges quadratically to $\boldsymbol{\mathcal{X}}^*$; in the noisy case, RGN converges quadratically to a neighborhood of $\boldsymbol{\mathcal{X}}^*$ up to some statistical error. More simulation results on tensor estimation via rank-1 projections and tensor SVD can be found in Section 6. Since RGN generally converges to a point with nonzero function value in the noisy setting, the generic theory on RGN can only guarantee a (super)linear convergence rate to a stationary point (Absil et al., 2009; Breiding and Vannieuwenhoven, 2018). Our result complements the classic theory of RGN: we show RGN converges quadratically to a neighborhood of the true parameter of interest, which achieves a statistically optimal estimation error rate. To our best knowledge, such a result is new and our RGN is the first algorithm with a provable guarantee of second-order convergence for the low-rank tensor estimation.
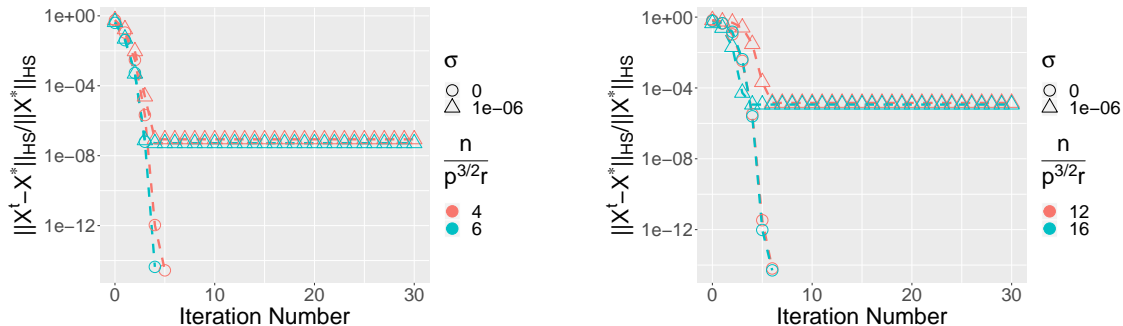
(a) Tensor regression. Here, n is the sample size, $\boldsymbol{\mathcal{X}}^* \in \mathbb{R}^{p \times p \times p}$ with $p = 30$, Tucker rank $\mathbf{r} = (3,3,3)$, $\boldsymbol{\varepsilon} \overset{i.i.d}{\sim} N(0, \sigma^2)$ with $\sigma \in \{0, 10^{-6}\}$ and $\boldsymbol{\mathcal{A}}_i$ has i.i.d. standard Gaussian entries.

(b) Tensor completion. Here, we observe partial uniformly-at-random sampled entries from the noisy tensor $\boldsymbol{\mathcal{Y}}$ index by Ω, where $\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{X}}^* + \boldsymbol{\mathcal{E}} \in \mathbb{R}^{p \times p \times p}$ with $p = 50$. Tucker rank of $\boldsymbol{\mathcal{X}}^*$ is $(3,3,3)$, $n = |\Omega|$ and $\boldsymbol{\mathcal{E}}$ has i.i.d. $N(0, \sigma^2)$ entries with $\sigma \in \{0, 10^{-6}\}$.

Figure 1: RGN achieves a quadratic rate of convergence in low-rank tensor estimation. More details of the simulation setting are given in Section 6.

Furthermore, we provide a deterministic minimax lower bound for the estimation error under model (1). The lower bound matches the estimation error upper bound, which demonstrates the statistical rate-optimality of RGN.

Next, we apply RGN to two problems arising from applications in machine learning and statistics: tensor regression and tensor SVD. In both problems, we prove the iterates of RGN converge quadratically to a neighborhood of $\boldsymbol{\mathcal{X}}^*$ that achieves the minimax optimal estimation error. A comparison of RGN and prior algorithms on tensor regression and tensor SVD is given in Table 1. We can see for fixed r, RGN achieves the best estimation error and signal-to-noise ratio requirement, i.e., sample complexity in tensor regression and least singular value in tensor SVD, compared to the state of the art while maintaining a relatively low computational cost. Moreover, RGN is the only algorithm with guaranteed quadratic convergence in both applications. Finally, we conduct numerical studies to support our theoretical findings in Section 6. The simulation studies show RGN offers much faster convergence compared to the existing approaches in the literature.

1.2 Related Literature

Our work is related to a broad range of literature from a number of communities. Here we make an attempt to discuss existing results without claiming that the survey is exhaustive.

First, the low-rank tensor estimation has attracted much recent attention from machine learning and statistics communities. Various methods were proposed, including the convex relaxation (Mu et al., 2014; Raskutti et al., 2019; Tomioka et al., 2011), projected gradient descent (Rauhut et al., 2017; Chen et al., 2019a; Ahmed et al., 2020; Yu and Liu, 2016), gradient descent on the factorized model (Han et al., 2022b; Cai et al., 2019; Hao et al., 2020), alternating minimization (Zhou et al., 2013; Jain and Oh, 2014; Liu and Moitra,

	Tensor Regression			
Algorithm	required sample size	convergence rate	estimation error	per-iteration cost
RGN (this work)	$p^{d/2}r^{3/2}$	quadratic	$\sigma\sqrt{\frac{pr}{n}}$	$np^d r$
GD (Han et al., 2022b)	$p^{d/2}r^{3/2}$	linear	$\sigma\sqrt{\frac{pr}{n}}$	np^d
Nonconvex-PGD (Chen et al., 2019a)	$p^{d-1}r$	linear	$\sigma\sqrt{\frac{p^{d-1}}{n}}$	np^d
Alter Mini (Zhou et al., 2013)	N.A.	linear	N.A.	$np^d r$
	Tensor SVD			
Algorithm	least singular value	convergence rate	estimation error	per-iteration cost
RGN (this work)	$p^{d/4}r^{1/4}\sigma$	quadratic	$\sigma\sqrt{pr}$	$p^d r$
GD (Han et al., 2022b)	$p^{d/4}r^{1/4}\sigma$	linear	$\sigma\sqrt{pr}$	$p^d r$
Alter Mini (Zhang and Xia, 2018)	$p^{d/4}\sigma$	linear	$\sigma\sqrt{pr}$	$p^d r$

Table 1: Comparison of RGN with gradient descent (GD), nonconvex projected gradient descent (Nonconvex-PGD) and alternating minimization (Alter Mini) in the literature in aspects of signal-to-noise ratio requirement (i.e., overall sample complexity in tensor regression and least singular value in tensor SVD), convergence rate, estimation error, and per-iteration computational cost. Here the convergence rate is global for nonconvex-PGD and is local for other algorithms. We assume the Tucker rank satisfies $r_1 = \cdots = r_d = r$, the tensor dimension satisfies $p_1 = \cdots = p_d$, $r, d \ll n, p$, the tensor parameter of interest is well-conditioned and σ is the standard deviation of the Gaussian noise in tensor regression and tensor SVD.

2020; Xia et al., 2021), and importance sketching (Zhang et al., 2020a). A scaled GD was proposed in the concurrent work Tong et al. (2022). Moreover, when the target tensor has order two, our problem reduces to the widely studied low-rank matrix recovery/estimation (Recht et al., 2010; Li et al., 2019; Ma et al., 2019; Sun and Luo, 2015; Tu et al., 2016; Wang et al., 2017; Zhao et al., 2015; Zheng and Lafferty, 2015; Charisopoulos et al., 2021; Luo et al., 2023; Bauch et al., 2021). The readers are referred to a recent survey in Chi et al. (2019).

Second, Riemannian manifold optimization methods have been powerful in solving optimization problems with geometric constraints (Absil et al., 2009; Boumal, 2020). Many progresses in this topic were made for the low-rank matrix estimation (Keshavan et al., 2009; Boumal and Absil, 2011, 2015; Wei et al., 2016; Meyer et al., 2011; Mishra et al., 2014; Vandereycken, 2013; Huang and Hand, 2018; Cherian and Sra, 2016). In particular,

Luo et al. (2023) proposed a recursive importance sketching algorithm for solving rank-constrained least-squares problem and showed that it is closely related to the Riemannian Gauss-Newton method. See also the recent survey on this line of work at Cai and Wei (2018); Uschmajew and Vandereycken (2020). Moreover, the Riemannian manifold optimization method has been applied for various problems on low-rank tensor estimation, such as tensor regression (Kressner et al., 2016), tensor completion (Rauhut et al., 2015; Kasai and Mishra, 2016; Dong et al., 2022; Kressner et al., 2014; Heidel and Schulz, 2018; Xia and Yuan, 2017; Steinlechner, 2016; Da Silva and Herrmann, 2015), and robust tensor PCA (Cai et al., 2022). These papers mostly focus on the first-order Riemannian optimization methods, possibly due to the hardness of deriving the exact expressions of the Riemannian Hessian. A few exceptions also appear: Heidel and Schulz (2018); Kasai and Mishra (2016) and Psenka and Boumal (2020) developed Riemannian trust-region method for tensor completion under Tucker and tensor-train formats, respectively; a Riemannian Gauss-Newton algorithm was also considered in Heidel and Schulz (2018); Kressner et al. (2016) proposed approximate Riemannian Newton methods for tensor regression in tensor-train and Tucker formats under the setting that the linear map has additive and Kronecker-product-type structures. Departing from these results that focus on the geometric objects and numerical implementations, in this paper we not only develop an efficient implementation of RGN under the Tucker format but also prove the quadratic convergence of the iterates and the optimal estimation error rate for the estimation of $\mathcal{X}^*$.

Finally, the low-rank tensor estimation model (1) is related to various problems in different contexts. In the tensor-based scientific computing community, large-scale linear systems where the solution admits a low-rank tensor structure commonly arise after discretizing high-dimensional partial differential equations (PDEs) (Lynch et al., 1964; Hofreither, 2018), which exactly become the central problem (1) in this paper. In the literature, various methods have been proposed there to solve (1). For example, Boussé et al. (2018) developed the algebraic method and Gauss-Newton method to solve the linear system with a CP low-rank tensor solution. Georgieva and Hofreither (2019) and Kressner et al. (2016) respectively introduced a greedy approach and an approximate Riemannian Newton method to approximate the linear system by a low Tucker rank tensor. The readers are also referred to Grasedyck et al. (2013) for a recent survey. There are some key differences from this line of work to ours: first, their goal is often to find a low-rank tensor that approximately solves a linear system with small *approximation error*, while we aim to develop an estimator with small *estimation error*; second, the design matrix in linear systems from discretized PDEs often has Kronecker-product-type structure, we do not assume such structure in this paper. On the other hand, the structures of the design assumed here are application dependent, e.g., sub-Gaussian design in tensor regression and "one-hot" design in tensor completion as we mentioned in the introduction; finally, their work mainly focuses on computational aspects of the proposed methods (Grasedyck et al., 2013), while this paper develops Riemannian Gauss-Newton for solving the low-rank tensor estimation problem and gives theoretical guarantees for the quadratic convergence of the algorithm and for the optimal estimation error bound of the final estimator.

1.3 Organization of the Paper

After a brief introduction of notation and preliminaries in Section 1.4, we introduce our main algorithm RGN and its geometric ingredients in Section 2. The theoretical results of RGN and its applications in tensor regression and tensor SVD are discussed in Sections 3 and 4, respectively. The computational complexity of RGN and numerical studies are presented in Sections 5 and 6, respectively. Conclusion and future work are given in Section 7. Additional algorithms and all technical proofs are presented in Appendices A-C.

1.4 Notation and Preliminaries

The following notation will be used throughout this article. Lowercase letters (e.g., a), lowercase boldface letters (e.g., $\mathbf{u}$), uppercase boldface letters (e.g., $\mathbf{U}$), and boldface calligraphic letters (e.g., $\boldsymbol{\mathcal{A}}$) are used to denote scalars, vectors, matrices, and order-3-or-higher tensors, respectively. For simplicity, we denote $\boldsymbol{\mathcal{A}}_j$ as the tensor indexed by j in a sequence of tensors $\{\boldsymbol{\mathcal{A}}_j\}$. We use bracket subscripts to denote sub-vectors, sub-matrices, and sub-tensors. For any vector $\mathbf{v}$, define its ℓ_2 norm as $\|\mathbf{v}\|_2 = \left(\sum_i |v_i|^2\right)^{1/2}$. For any matrix $\mathbf{D} \in \mathbb{R}^{p_1 \times p_2}$, let $\sigma_k(\mathbf{D})$ be the kth largest singular value of $\mathbf{D}$. We also denote $\mathrm{SVD}_r(\mathbf{D}) = [\mathbf{u}_1 \cdots \mathbf{u}_r]$ and $\mathrm{QR}(\mathbf{D})$ as the subspace composed of the leading r left singular vectors and the Q part of the QR orthogonalization of $\mathbf{D}$, respectively. $\mathbf{I}_r$ 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 use $\mathbf{U}_\perp \in \mathbb{O}_{p,p-r}$ to represent the orthonormal complement of $\mathbf{U}$.

The matricization $\mathcal{M}_k(\cdot)$ is the operation that unfolds the order-d tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{p_1 \times \cdots \times p_d}$ along mode k into the matrix $\mathcal{M}_k(\boldsymbol{\mathcal{A}}) \in \mathbb{R}^{p_k \times p_{-k}}$ where $p_{-k} = \prod_{j \neq k} p_j$. Specifically, the mode-k matricization of $\boldsymbol{\mathcal{A}}$ is formally defined as

$$\mathcal{M}_k(\boldsymbol{\mathcal{A}}) \in \mathbb{R}^{p_k \times p_{-k}}, \quad (\mathcal{M}_k(\boldsymbol{\mathcal{A}}))_{[i_k,j]} = \boldsymbol{\mathcal{A}}_{[i_1,\ldots,i_d]}, \quad j = 1 + \sum_{\substack{l=1 \\ l \neq k}}^{d} \left\{ (i_l - 1) \prod_{\substack{m=1 \\ m \neq k}}^{l-1} p_m \right\} \quad (5)$$

for any $1 \leqslant i_l \leqslant p_l, l = 1, \ldots, d$. We also use notation $\mathcal{T}_k(\cdot)$ to denote the mode-k tensorization or reverse operator of $\mathcal{M}_k(\cdot)$. Throughout the paper, $\mathcal{T}_k$, as a reversed operation of $\mathcal{M}_k(\cdot)$, maps a $\mathbb{R}^{p_k \times p_{-k}}$ matrix back to a $\mathbb{R}^{p_1 \times \cdots \times p_d}$ tensor. The Hilbert-Schmidt norm of $\boldsymbol{\mathcal{A}}$ is defined as $\|\boldsymbol{\mathcal{A}}\|_{\mathrm{HS}} = (\langle \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{A}}\rangle)^{1/2}$. The Tucker rank of a tensor $\boldsymbol{\mathcal{A}}$ is denoted by $\mathrm{Tucrank}(\boldsymbol{\mathcal{A}})$ and defined as a d-tuple $\mathbf{r} := (r_1, \ldots, r_d)$, where $r_k = \mathrm{rank}(\mathcal{M}_k(\boldsymbol{\mathcal{A}}))$. For any Tucker rank-$(r_1, \ldots, r_d)$ tensor $\boldsymbol{\mathcal{A}}$, it has Tucker decomposition (Tucker, 1966):

$$\boldsymbol{\mathcal{A}} = [\![\boldsymbol{\mathcal{S}}; \mathbf{U}_1, \ldots, \mathbf{U}_d]\!] := \boldsymbol{\mathcal{S}} \times_1 \mathbf{U}_1 \times \cdots \times_d \mathbf{U}_d, \quad (6)$$

where $\boldsymbol{\mathcal{S}} \in \mathbb{R}^{r_1 \times \cdots \times r_d}$ is the core tensor and $\mathbf{U}_k = \mathrm{SVD}_{r_k}(\mathcal{M}_k(\boldsymbol{\mathcal{A}}))$ is the mode-k singular vectors. Here, the mode-k product of $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{p_1 \times \cdots \times p_d}$ with a matrix $\mathbf{B} \in \mathbb{R}^{r_k \times p_k}$ is denoted by $\boldsymbol{\mathcal{A}} \times_k \mathbf{B}$ and is of size $p_1 \times \cdots \times p_{k-1} \times r_k \times p_{k+1} \times \cdots \times p_d$, and its formal definition is given below

$$(\boldsymbol{\mathcal{A}} \times_k \mathbf{B})_{[i_1,\ldots,i_{k-1},j,i_{k+1},\ldots,i_d]} = \sum_{i_k=1}^{p_k} \boldsymbol{\mathcal{A}}_{[i_1,i_2,\ldots,i_d]}\mathbf{B}_{[j,i_k]}. \quad (7)$$

It is convenient to introduce the following abbreviations to denote the tensor-matrix product along multiple modes: $\boldsymbol{\mathcal{A}} \times_{k=1}^d \mathbf{U}_k := \boldsymbol{\mathcal{A}} \times_1 \mathbf{U}_1 \times \cdots \times_d \mathbf{U}_d$; $\boldsymbol{\mathcal{A}} \times_{l \neq k} \mathbf{U}_l := \boldsymbol{\mathcal{A}} \times_1 \mathbf{U}_1 \times \cdots \times_{k-1} \mathbf{U}_{k-1} \times_{k+1} \mathbf{U}_{k+1} \times \cdots \times_d \mathbf{U}_d$. The following property about tensor matricization will be used (Kolda, 2001, Section 4):

$$\mathcal{M}_k \left(\boldsymbol{\mathcal{S}} \times_1 \mathbf{U}_1 \times \cdots \times_d \mathbf{U}_d \right) = \mathbf{U}_k \mathcal{M}_k(\boldsymbol{\mathcal{S}})(\mathbf{U}_d^\top \otimes \cdots \otimes \mathbf{U}_{k+1}^\top \otimes \mathbf{U}_{k-1}^\top \otimes \cdots \otimes \mathbf{U}_1^\top), \quad (8)$$

where "$\otimes$" is the matrix Kronecker product. For any tensor $\boldsymbol{\mathcal{Z}} \in \mathbb{R}^{p_1 \times \cdots \times p_d}$, we define $\boldsymbol{\mathcal{Z}}_{\max(\mathbf{r})} := \boldsymbol{\mathcal{Z}} \times_{k=1}^d P_{\widehat{\mathbf{U}}_k}$ as the best Tucker rank $\mathbf{r}$ approximation of $\boldsymbol{\mathcal{Z}}$ in terms of Hilbert-Schmidt norm, where $(\widehat{\mathbf{U}}_1, \ldots, \widehat{\mathbf{U}}_d) = \arg\max_{\mathbf{U}_k \in \mathbb{O}_{p_k, r_k}, k=1,\ldots,d} \| \boldsymbol{\mathcal{Z}} \times_{k=1}^d P_{\mathbf{U}_k} \|_{\mathrm{HS}}$. Finally, for any linear operator $\mathcal{L}$, we denote $\mathcal{L}^*$ as its adjoint operator.

2 Algorithm

We introduce the geometry of low Tucker rank tensor Riemannian manifolds in Section 2.1 and present the procedure of RGN in Section 2.2.

2.1 The Geometry for Low Tucker Rank Tensor Manifolds

Denote the collection of $(p_1, \ldots, p_d)$-dimensional tensors of Tucker rank $\mathbf{r}$ by $\mathbb{M}_{\mathbf{r}} = \{ \boldsymbol{\mathcal{X}} \in \mathbb{R}^{p_1 \times \cdots \times p_d}, \text{Tucrank}(\boldsymbol{\mathcal{X}}) = \mathbf{r} \}$. Then $\mathbb{M}_{\mathbf{r}}$ forms a smooth submanifold embedded in $\mathbb{R}^{p_1 \times \cdots \times p_d}$ with dimension $\prod_{j=1}^d r_j + \sum_{j=1}^d r_j(p_j - r_j)$ (Uschmajew and Vandereycken, 2013; Kressner et al., 2014). Throughout the paper, we use the natural Euclidean inner product as the Riemannian metric. Suppose $\boldsymbol{\mathcal{X}} \in \mathbb{M}_{\mathbf{r}}$ has Tucker decomposition $[\![\boldsymbol{\mathcal{S}}; \mathbf{U}_1, \ldots, \mathbf{U}_d]\!]$; Koch and Lubich (2010) showed that the tangent space of $\mathbb{M}_{\mathbf{r}}$ at $\boldsymbol{\mathcal{X}}$, $T_{\boldsymbol{\mathcal{X}}}\mathbb{M}_{\mathbf{r}}$, can be represented as

$$T_{\boldsymbol{\mathcal{X}}}\mathbb{M}_{\mathbf{r}} = \left\{ \boldsymbol{\mathcal{B}} \times_{k=1}^d \mathbf{U}_k + \sum_{k=1}^d \boldsymbol{\mathcal{S}} \times_k \bar{\mathbf{D}}_k \times_{j \neq k} \mathbf{U}_j : \begin{array}{l} \boldsymbol{\mathcal{B}} \in \mathbb{R}^{r_1 \times \cdots \times r_d}, \bar{\mathbf{D}}_k \in \mathbb{R}^{p_k \times r_k}, \\ \bar{\mathbf{D}}_k^\top \mathbf{U}_k = \mathbf{0}, k = 1, \ldots, d \end{array} \right\}. \quad (9)$$

In the representation above, $\bar{\mathbf{D}}_k$s are not free parameters due to the constraints $\bar{\mathbf{D}}_k^\top \mathbf{U}_k = 0$, $k = 1, \ldots, d$. In the following Lemma 1, we introduce another representation of $T_{\boldsymbol{\mathcal{X}}}\mathbb{M}_{\mathbf{r}}$ with a minimal parameterization, which matches the degree of freedom of the tangent space ($\prod_{j=1}^d r_j + \sum_{j=1}^d r_j(p_j - r_j)$). For $\boldsymbol{\mathcal{X}} = [\![\boldsymbol{\mathcal{S}}; \mathbf{U}_1, \ldots, \mathbf{U}_d]\!]$, we let $\mathbf{V}_k = \mathrm{QR}(\mathcal{M}_k(\boldsymbol{\mathcal{S}})^\top)$, which corresponds to the row space of $\mathcal{M}_k(\boldsymbol{\mathcal{S}})$, and define

$$\mathbf{W}_k := (\mathbf{U}_d \otimes \cdots \otimes \mathbf{U}_{k+1} \otimes \mathbf{U}_{k-1} \otimes \mathbf{U}_1)\mathbf{V}_k \in \mathbb{O}_{p_{-k}, r_k}, \quad k = 1, \ldots, d, \quad (10)$$

where $p_{-k} = \prod_{j \neq k} p_j$. By (8), $\mathbf{U}_k, \mathbf{W}_k$ correspond to the subspaces of the column and row spans of $\mathcal{M}_k(\boldsymbol{\mathcal{X}})$, respectively.

Lemma 1 *The tangent space of $\mathbb{M}_{\mathbf{r}}$ at $\boldsymbol{\mathcal{X}} = [\![\boldsymbol{\mathcal{S}}; \mathbf{U}_1, \ldots, \mathbf{U}_d]\!]$ in (9) can be written as*

$$T_{\boldsymbol{\mathcal{X}}}\mathbb{M}_{\mathbf{r}} = \left\{ \boldsymbol{\mathcal{B}} \times_{k=1}^d \mathbf{U}_k + \sum_{k=1}^d \mathcal{T}_k(\mathbf{U}_{k\perp}\mathbf{D}_k\mathbf{W}_k^\top) : \boldsymbol{\mathcal{B}} \in \mathbb{R}^{r_1 \times \cdots \times r_d}, \mathbf{D}_k \in \mathbb{R}^{(p_k - r_k) \times r_k}, k = 1, \ldots, d \right\},$$

where $\mathcal{T}_k(\cdot)$ is the mode-k tensorization operator and $\mathbf{W}_k$ is given in (10).

We can also show that any tensor in $T_{\boldsymbol{\mathcal{X}}}\mathbb{M}_{\mathbf{r}}$ is at most Tucker rank $2\mathbf{r}$. This fact will facilitate the efficient computation of RGN to be discussed in Section 5.

Lemma 2 *Any tensor $\boldsymbol{\mathcal{Z}} \in T_{\boldsymbol{\mathcal{X}}}\mathbb{M}_{\mathbf{r}}$ is at most Tucker rank $2\mathbf{r}$.*

Lemma 3.1 of Koch and Lubich (2010) and the tangent space representation in Lemma 1 yield the following projection operator $P_{T_{\boldsymbol{\mathcal{X}}}}$ that projects any tensor $\boldsymbol{\mathcal{Z}}$ onto the tangent space of $\mathbb{M}_{\mathbf{r}}$ at $\boldsymbol{\mathcal{X}}$:

$$P_{T_{\boldsymbol{\mathcal{X}}}}(\boldsymbol{\mathcal{Z}}) := \mathcal{L}\mathcal{L}^*(\boldsymbol{\mathcal{Z}}) = \boldsymbol{\mathcal{Z}} \times_{k=1}^{d} P_{\mathbf{U}_k} + \sum_{k=1}^{d} \mathcal{T}_k(P_{\mathbf{U}_{k\perp}}\mathcal{M}_k(\boldsymbol{\mathcal{Z}})P_{\mathbf{W}_k}), \quad \forall \boldsymbol{\mathcal{Z}} \in \mathbb{R}^{p_1 \times \cdots \times p_d}, \quad (11)$$

where $\mathcal{L}^*$ and $\mathcal{L}$ are respectively the contraction map and extension map defined as follows:

$$\mathcal{L} : \mathbb{R}^{r_1 \times \cdots \times r_d} \times \prod_{k=1}^{d} \mathbb{R}^{(p_k - r_k) \times r_k} \to T_{\boldsymbol{\mathcal{X}}}\mathbb{M}_{\mathbf{r}}, \quad (\boldsymbol{\mathcal{B}}, \{\mathbf{D}_k\}_{k=1}^{d}) \mapsto \boldsymbol{\mathcal{B}} \times_{k=1}^{d} \mathbf{U}_k + \sum_{k=1}^{d} \mathcal{T}_k(\mathbf{U}_{k\perp}\mathbf{D}_k\mathbf{W}_k^{\top}),$$

$$\mathcal{L}^* : \mathbb{R}^{p_1 \times \cdots \times p_d} \to \mathbb{R}^{r_1 \times \cdots \times r_d} \times \prod_{k=1}^{d} \mathbb{R}^{(p_k - r_k) \times r_k}, \quad \boldsymbol{\mathcal{Z}} \mapsto (\boldsymbol{\mathcal{Z}} \times_{k=1}^{d} \mathbf{U}_k^{\top}, \{\mathbf{U}_{k\perp}^{\top}\mathcal{M}_k(\boldsymbol{\mathcal{Z}})\mathbf{W}_k\}_{k=1}^{d}).$$

$$(12)$$

In particular, $\mathcal{L}^*$ is the adjoint operator of $\mathcal{L}$. We will see in Section 2.2 that the representation in (11) helps the efficient implementation of RGN.

2.2 Riemannian Optimization and Riemannian Gauss-Newton

In this subsection, we first give a preliminary for Riemannian optimization and then introduce the procedure of RGN for low-rank tensor estimation.

Overall three-step procedure of Riemannian optimization. Riemannian optimization concerns optimizing a real-valued function f defined on a Riemannian manifold $\mathbb{M}$, for which the readers are referred to Absil et al. (2009) and Boumal (2020) for an introduction. Due to the common non-linearity, the continuous optimization on the Riemannian manifold often requires calculations on the tangent space. A typical procedure of a Riemannian optimization method contains three steps per iteration: Step 1. find the tangent space; Step 2. update the point on the tangent space; Step 3. map the point from the tangent space back to the manifold.

Low-rank tensor Riemannian manifold (Step 1). We have already discussed the tangent space of low Tucker rank tensor manifolds in Section 2.1, i.e., Step 1 above.

Update on tangent space (Step 2). Next, we describe the procedure of RGN in the tangent space. We begin by introducing a few more preliminaries for Riemannian manifold optimization. The Riemannian gradient of a smooth function $f : \mathbb{M}_{\mathbf{r}} \to \mathbb{R}$ at $\boldsymbol{\mathcal{X}} \in \mathbb{M}_{\mathbf{r}}$ is defined as the unique tangent vector $\operatorname{grad} f(\boldsymbol{\mathcal{X}}) \in T_{\boldsymbol{\mathcal{X}}}\mathbb{M}_{\mathbf{r}}$ such that $\langle \operatorname{grad} f(\boldsymbol{\mathcal{X}}), \boldsymbol{\mathcal{Z}} \rangle = \mathrm{D} f(\boldsymbol{\mathcal{X}})[\boldsymbol{\mathcal{Z}}], \forall \boldsymbol{\mathcal{Z}} \in T_{\boldsymbol{\mathcal{X}}}\mathbb{M}_{\mathbf{r}}$, where $\mathrm{D}f(\boldsymbol{\mathcal{X}})[\boldsymbol{\mathcal{Z}}]$ denotes the directional derivative of f at point $\boldsymbol{\mathcal{X}}$ along direction $\boldsymbol{\mathcal{Z}}$. Specifically for the embedded submanifold $\mathbb{M}_{\mathbf{r}}$, we have:

9

Lemma 3 *For $f(\boldsymbol{\mathcal{X}})$ in (4), $\operatorname{grad} f(\boldsymbol{\mathcal{X}}) = P_{T_{\boldsymbol{\mathcal{X}}}}(\mathscr{A}^*(\mathscr{A}(\boldsymbol{\mathcal{X}}) - \mathbf{y}))$, where $P_{T_{\boldsymbol{\mathcal{X}}}}(\cdot)$ is the projection operator onto the tangent space of $\mathbb{M}_\mathbf{r}$ at $\boldsymbol{\mathcal{X}}$ defined in (11).*

A common way to derive RGN update in the literature is to first write down the Riemannian Newton equation, then replace the Riemannian Hessian by its Gauss-Newton approximation (Absil et al., 2009, Chapter 8.4.1), and finally solve the modified Riemannian Newton equation, i.e., the Riemannian Gauss-Newton equation. In our low-rank tensor estimation problem with the objective function (4), suppose the current iterate is $\boldsymbol{\mathcal{X}}^t$, the RGN update $\eta^{RGN} \in T_{\boldsymbol{\mathcal{X}}^t}\mathbb{M}_\mathbf{r}$ should solve the following RGN equation (Absil et al., 2009, Chapter 8.4),

$$-\operatorname{grad} f(\boldsymbol{\mathcal{X}}^t) = P_{T_{\boldsymbol{\mathcal{X}}^t}}\left(\mathscr{A}^*(\mathscr{A}(\eta^{RGN}))\right). \tag{13}$$

However, it is not clear how to solve this equation directly in practice.

Inspired by the classical Gauss-Newton (GN) algorithm, we instead introduce another scheme to derive RGN. Recall in solving the nonlinear least squares problem in the Euclidean space $\min_x \frac{1}{2}\|h(x)\|_2^2$, the classic Gauss-Newton can be viewed as a modified Newton method, and can also be derived by replacing the non-linear function $h(x)$ by its local linear approximation at the current iterate x_k (Nocedal and Wright, 2006, Chapter 10.3). These two ways of interpretation are equivalent. A similar local linearization idea can be extended to the manifold setting except that the linearization needs to be taken in the tangent space in each iterate. Specifically, consider the objective function $f(\boldsymbol{\mathcal{X}})$ in (4), the linearization of $\mathbf{y} - \mathscr{A}(\boldsymbol{\mathcal{X}})$ at $\boldsymbol{\mathcal{X}}^t$ in $T_{\boldsymbol{\mathcal{X}}^t}\mathbb{M}_\mathbf{r}$ is $\mathbf{y} - \mathscr{A}(\boldsymbol{\mathcal{X}}^t) - \mathscr{A}P_{T_{\boldsymbol{\mathcal{X}}^t}}(\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{X}}^t)$, which can be simplified to $\mathbf{y} - \mathscr{A}P_{T_{\boldsymbol{\mathcal{X}}^t}}(\boldsymbol{\mathcal{X}})$. After we further constraint the update direction to $T_{\boldsymbol{\mathcal{X}}^t}\mathbb{M}_\mathbf{r}$, we have

$$\boldsymbol{\mathcal{Z}}^{t+1} = \operatorname*{arg\,min}_{\boldsymbol{\mathcal{Z}} \in T_{\boldsymbol{\mathcal{X}}^t}\mathbb{M}_\mathbf{r}} \frac{1}{2}\|\mathbf{y} - \mathscr{A}P_{T_{\boldsymbol{\mathcal{X}}^t}}(\boldsymbol{\mathcal{Z}})\|_2^2. \tag{14}$$

By mapping $\boldsymbol{\mathcal{Z}}^{t+1}$ back to the manifold, we get the new iterate $\boldsymbol{\mathcal{X}}^{t+1}$.

Next, we show the proposed update derived in (14) actually matches the standard RGN update (13).

Proposition 1 *Let $\boldsymbol{\mathcal{Z}}^{t+1}$ be the update computed in (14). Then, $\boldsymbol{\mathcal{Z}}^{t+1} - \boldsymbol{\mathcal{X}}^t$ is the Riemannian Gauss-Newton update, i.e., it solves the Riemannian Gauss-Newton equation (13).*

Proposition 1 shows that (14) yields the RGN update, which directly provides a simple implementation of RGN. To see this, recall $P_{T_{\boldsymbol{\mathcal{X}}^t}} = \mathcal{L}_t\mathcal{L}_t^*$, where $\mathcal{L}_t$ and $\mathcal{L}_t^*$ are defined in the similar way as in (12) except evaluated on $\boldsymbol{\mathcal{X}}^t = [\![\boldsymbol{\mathcal{S}}^t; \mathbf{U}_1^t, \ldots, \mathbf{U}_d^t]\!]$; then the objective function in (14) can be rewritten as follows,

$$
\begin{aligned}
\frac{1}{2}\|\mathbf{y} - \mathscr{A}P_{T_{\boldsymbol{\mathcal{X}}^t}}(\boldsymbol{\mathcal{Z}})\|_2^2 &= \frac{1}{2}\sum_{i=1}^n (\mathbf{y}_i - \langle \boldsymbol{\mathcal{A}}_i, \mathcal{L}_t\mathcal{L}_t^*\boldsymbol{\mathcal{Z}}\rangle)^2 = \frac{1}{2}\sum_{i=1}^n (\mathbf{y}_i - \langle \mathcal{L}_t^*\boldsymbol{\mathcal{A}}_i, \mathcal{L}_t^*\boldsymbol{\mathcal{Z}}\rangle)^2 \\
&= \frac{1}{2}\sum_{i=1}^n \left(\mathbf{y}_i - \langle \boldsymbol{\mathcal{A}}_i \times_{k=1}^d \mathbf{U}_k^{t\top}, \boldsymbol{\mathcal{B}}\rangle - \sum_{k=1}^d \langle \mathbf{U}_{k\perp}^{t\top}\mathcal{M}_k(\boldsymbol{\mathcal{A}}_i)\mathbf{W}_k^t, \mathbf{D}_k\rangle\right)^2,
\end{aligned}
\tag{15}
$$

where $(\boldsymbol{\mathcal{B}}, \{\mathbf{D}_k\}_{k=1}^d) := \mathcal{L}_t^*\boldsymbol{\mathcal{Z}}$. Based on the calculation in (15), we define the following covariates maps $\mathscr{A}_{\boldsymbol{\mathcal{B}}} : \mathbb{R}^{r_1 \times \cdots \times r_d} \to \mathbb{R}^n, \mathscr{A}_{\mathbf{D}_k} : \mathbb{R}^{(p_k - r_k)r_k} \to \mathbb{R}^n, k = 1, \cdots, d$, where for

$1 \leqslant i \leqslant n$,

$$(\mathscr{A}_{\mathcal{B}})_i = \mathcal{A}_i \times_{k=1}^d \mathbf{U}_k^{t\top}, \quad (\mathscr{A}_{\mathbf{D}_k})_i = \mathbf{U}_{k\perp}^{t\top} \mathcal{M}_k (\mathcal{A}_i) \mathbf{W}_k^t.$$

Here, $(\mathscr{A}_{\mathcal{B}})_i$ satisfies $[\mathscr{A}_{\mathcal{B}}(\cdot)]_i = \langle \cdot, (\mathscr{A}_{\mathcal{B}})_i \rangle$ and similarly for $(\mathscr{A}_{\mathbf{D}_k})_i$. Then, by (15) and the fact that $\mathcal{Z} \in T_{\mathcal{X}^t} \mathbb{M}_{\mathbf{r}}$, (14) can be equivalently solved by

$$\mathcal{Z}^{t+1} = \mathcal{L}_t(\mathbf{B}^{t+1}, \mathbf{D}_1^{t+1}, \cdots, \mathbf{D}_d^{t+1}),$$

where

$$(\mathcal{B}^{t+1}, \mathbf{D}_1^{t+1}, \cdots, \mathbf{D}_d^{t+1}) = \underset{\substack{\mathcal{B} \in \mathbb{R}^{r_1 \times \cdots \times r_d}, \\ \mathbf{D}_k \in \mathbb{R}^{(p_k - r_k) \times r_k}, k=1,\ldots,d}}{\arg\min} \left\| \mathbf{y} - \mathscr{A}_{\mathcal{B}}(\mathcal{B}) - \sum_{k=1}^d \mathscr{A}_{\mathbf{D}_k}(\mathbf{D}_k) \right\|_2^2. \qquad (16)$$

Note that (16) is an unconstrained least squares with the number of parameters equal to $\sum_{k=1}^d (p_k - r_k)r_k + \prod_{k=1}^d r_k$.

Retraction (Step 3). Finally, we discuss how to map the point from the tangent space back to the manifold, i.e., Step 3 above. An ideal method is via the *exponential map*, which moves a point on the manifold along the geodesic. However, computing the exponential map is prohibitively expensive in most situations, and a more practical choice is the so-called *retraction*. Retraction is in general a first-order approximation of the exponential map. In tensor manifold $\mathbb{M}_{\mathbf{r}}$, the retraction map, denoted by R, should be a smooth map from $T\mathbb{M}_{\mathbf{r}}$ to $\mathbb{M}_{\mathbf{r}}$ that satisfies i) $R(\mathcal{X}, 0) = \mathcal{X}$ and ii) $\frac{d}{dt} R(\mathcal{X}, t\eta)|_{t=0} = \eta$ for all $\mathcal{X} \in \mathbb{M}_{\mathbf{r}}$ and $\eta \in T_{\mathcal{X}} \mathbb{M}_{\mathbf{r}}$ (Absil et al., 2009, Chapter 4). Here, $T\mathbb{M}_{\mathbf{r}} = \{(\mathcal{X}, T_{\mathcal{X}} \mathbb{M}_{\mathbf{r}}) : \mathcal{X} \in \mathbb{M}_{\mathbf{r}}\}$ is the tangent bundle of $\mathbb{M}_{\mathbf{r}}$.

In the low Tucker rank tensor manifolds, Proposition 2.3 of Kressner et al. (2014) showed that the truncated high-order singular value decomposition (T-HOSVD) (De Lathauwer et al., 2000) is a retraction. We further show in the following Lemma 4 that the sequentially truncated HOSVD (ST-HOSVD) (Vannieuwenhoven et al., 2012), a computationally more efficient procedure than T-HOSVD, also satisfies the retraction properties. The detailed procedures of T-HOSVD and ST-HOSVD are given in Appendix A.

Lemma 4 (Retraction of Sequentially Truncated HOSVD) *For ST-HOSVD defined in Appendix A, the map*

$$R : T\mathbb{M}_{\mathbf{r}} \to \mathbb{M}_{\mathbf{r}}, (\mathcal{X}, \eta) \to \text{ST-HOSVD}(\mathcal{X} + \eta)$$

is a retraction on $\mathbb{M}_{\mathbf{r}}$ around $\mathcal{X}$.

Although ST-HOSVD has been widely used in practice, the retraction property of ST-HOSVD we established in Lemma 4 is new.

Summary of RGN. We give the complete RGN algorithm for low-rank tensor estimation in Algorithm 1.

Algorithm 1 Riemannian Gauss-Newton for Low-rank Tensor Estimation

---

Input: $\mathbf{y} \in \mathbb{R}^n, \mathcal{A}_1, \ldots, \mathcal{A}_n \in \mathbb{R}^{p_1 \times \cdots \times p_d}, t_{\max}$, Tucker rank $\mathbf{r}$, initialization $\mathcal{X}^0$ with Tucker decomposition $[\![\mathcal{S}^0; \mathbf{U}_1^0, \ldots, \mathbf{U}_d^0]\!]$, and $\mathbf{W}_k^0$ defined as (10).

1: **for** $t = 0, 1, \ldots, t_{\max} - 1$ **do**

2:    Construct the covariates maps $\mathscr{A}_{\mathcal{B}} : \mathbb{R}^{r_1 \times \cdots \times r_d} \to \mathbb{R}^n, \mathscr{A}_{\mathbf{D}_k} : \mathbb{R}^{(p_k - r_k) r_k} \to \mathbb{R}^n, k = 1, \cdots, d$, where for $1 \leqslant i \leqslant n$

$$(\mathscr{A}_{\mathcal{B}})_i = \mathcal{A}_i \times_{k=1}^d \mathbf{U}_k^{t\top}, \quad (\mathscr{A}_{\mathbf{D}_k})_i = \mathbf{U}_{k\perp}^{t\top} \mathcal{M}_k(\mathcal{A}_i) \mathbf{W}_k^t. \tag{17}$$

3:    Solve the unconstrained least squares problem

$$(\mathcal{B}^{t+1}, \mathbf{D}_1^{t+1}, \cdots, \mathbf{D}_d^{t+1}) = \underset{\substack{\mathcal{B} \in \mathbb{R}^{r_1 \times \cdots \times r_d}, \\ \mathbf{D}_k \in \mathbb{R}^{(p_k - r_k) \times r_k}, k=1,\ldots,d}}{\arg\min} \left\| \mathbf{y} - \mathscr{A}_{\mathcal{B}}(\mathcal{B}) - \sum_{k=1}^d \mathscr{A}_{\mathbf{D}_k}(\mathbf{D}_k) \right\|_2^2. \tag{18}$$

4:    Update

$$\mathcal{X}^{t+1} = [\![\mathcal{S}^{t+1}; \mathbf{U}_1^{t+1}, \ldots, \mathbf{U}_d^{t+1}]\!] = \mathcal{H}_{\mathbf{r}}\left(\mathcal{B}^{t+1} \times_{k=1}^d \mathbf{U}_k^t + \sum_{k=1}^d \mathcal{T}_k(\mathbf{U}_{k\perp}^t \mathbf{D}_k^{t+1} \mathbf{W}_k^{t\top}) \right) \tag{19}$$

   and $\mathbf{W}_k^{t+1}$ via (10). Here $\mathcal{H}_{\mathbf{r}}(\cdot)$ is the retraction map onto $\mathbb{M}_{\mathbf{r}}$ (two choices are ST-HOSVD and T-HOSVD).

5: **end for**

Output: $\mathcal{X}^{t_{\max}}$.

Remark 1 (Operator $\mathcal{H}_{\mathbf{r}}$) *In (19), $\mathcal{H}_{\mathbf{r}}$ plays the role of retraction that maps the iterate from the tangent space of $\mathbb{M}_{\mathbf{r}}$ at $\mathcal{X}^t$ back onto the manifold. Since $\mathcal{H}_{\mathbf{r}}$ directly operates on the updated tensor, to distinguish with the canonical notation $R(\cdot, \cdot)$ for retraction, we use a simplified notation $\mathcal{H}_{\mathbf{r}}$ to represent this map here. As we mentioned before, T-HOSVD (De Lathauwer et al., 2000) and ST-HOSVD (Vannieuwenhoven et al., 2012) are two choices of retractions.*

3 Theoretical Analysis

We analyze the convergence rate of RGN in this section.

3.1 Quasi-projection Property and Tensor Restricted Isometry Property

We begin by introducing the quasi-projection property of T-HOSVD and ST-HOSVD and the assumption on the linear map $\mathscr{A}$. Different from the low-rank matrix projection, which can be efficiently and exactly computed via truncated SVD, performing low-rank tensor projection exactly, even for $\mathbf{r} = 1$, can be NP-hard in general (Hillar and Lim, 2013). We thus introduce the following quasi-projection property and the approximation constant $\delta(d)$.

Definition 1 (Quasi-projection of $\mathcal{H}_\mathbf{r}$ and Approximation Constant $\delta(d)$) *Let $P_{\mathbb{M}_\mathbf{r}}(\cdot)$ be the projection map from $\mathbb{R}^{p_1 \times \cdots \times p_d}$ to the tensor space of Tucker rank at most $\mathbf{r}$, i.e., for any $\mathbf{\mathcal{Z}} \in \mathbb{R}^{p_1 \times \cdots \times p_d}$ and $\widehat{\mathbf{\mathcal{Z}}}$ of Tucker rank at most $\mathbf{r}$, one always has $\|\mathbf{\mathcal{Z}} - \widehat{\mathbf{\mathcal{Z}}}\|_{\mathrm{HS}} \geqslant \|\mathbf{\mathcal{Z}} - P_{\mathbb{M}_\mathbf{r}}(\mathbf{\mathcal{Z}})\|_{\mathrm{HS}}$.*

We say $\mathcal{H}_\mathbf{r}$ satisfies the quasi-projection property with approximation constant $\delta(d)$ if $\|\mathbf{\mathcal{Z}} - \mathcal{H}_\mathbf{r}(\mathbf{\mathcal{Z}})\|_{\mathrm{HS}} \leqslant \delta(d)\|\mathbf{\mathcal{Z}} - P_{\mathbb{M}_\mathbf{r}}(\mathbf{\mathcal{Z}})\|_{\mathrm{HS}}$ for any $\mathbf{\mathcal{Z}} \in \mathbb{R}^{p_1 \times \cdots \times p_d}$.

It is known that T-HOSVD and ST-HOSVD satisfy the *quasi-projection property* (see Chapter 10 in (Hackbusch, 2012)).

Proposition 2 (Quasi-projection property of T-HOSVD and ST-HOSVD) *T-HOSVD and ST-HOSVD described in Appendix A satisfy the quasi-projection property with approximation constant $\delta(d) = \sqrt{d}$.*

For technical convenience, we also assume $\mathscr{A}$ satisfies the following Tensor Restricted Isometry Property (TRIP) (Rauhut et al., 2017). One major reason we need this assumption is to control the spectrum of the operator $\mathcal{L}_t^* \mathscr{A}^* \mathscr{A} \mathcal{L}_t$ presented in Lemma 6 in Appendix B of the paper. The TRIP condition can be seen as a tensor generalization of the restricted isometry property (RIP). In the compressed-sensing and low-rank matrix recovery literature, the RIP condition has been widely used as one standard assumption (Candès and Plan, 2011; Cai and Zhang, 2013).

Definition 2 (Tensor Restricted Isometry Property (TRIP)) *Let $\mathscr{A} : \mathbb{R}^{p_1 \times \cdots \times p_d} \rightarrow \mathbb{R}^n$ be a linear map. For a fixed d-tuple $\mathbf{r} = (r_1, \ldots, r_d)$ with $1 \leqslant r_k \leqslant p_k$ for $k = 1, \ldots, d$, define the $\mathbf{r}$-tensor restricted isometry constant to be the smallest number $R_\mathbf{r}$ such that $(1 - R_\mathbf{r})\|\mathbf{\mathcal{Z}}\|_{\mathrm{HS}}^2 \leqslant \|\mathscr{A}(\mathbf{\mathcal{Z}})\|_2^2 \leqslant (1 + R_\mathbf{r})\|\mathbf{\mathcal{Z}}\|_{\mathrm{HS}}^2$ holds for all $\mathbf{\mathcal{Z}}$ of Tucker rank at most $\mathbf{r}$. If $0 \leqslant R_\mathbf{r} < 1$, we say $\mathscr{A}$ satisfies $\mathbf{r}$-tensor restricted isometry property ($\mathbf{r}-TRIP$).*

In Rauhut et al. (2017), the authors showed that TRIP can be satisfied in a number of different scenarios. For example, if sensing tensors $\mathbf{\mathcal{A}}_i$ are composed of i.i.d. sub-Gaussian entries, then with high probability, the TRIP condition can be satisfied with TRIP constant $R_\mathbf{r}$ as long as $n \geqslant C(\sum_{k=1}^d p_k r_k + \prod_{k=1}^d r_k) \log d / R_\mathbf{r}^2$ for some constant $C > 0$. In addition, TRIP also holds for more structured measurement ensembles such as the random Fourier mapping (Rauhut et al., 2017).

3.2 Main Convergence Results

In this subsection, we establish the deterministic convergence theory for RGN.

Theorem 1 (Convergence of RGN) *Suppose $\mathcal{H}_\mathbf{r}$ is either T-HOSVD or ST-HOSVD, $\mathscr{A}$ satisfies the $3\mathbf{r}$-TRIP, and the initialization $\mathbf{\mathcal{X}}^0$ satisfies $\|\mathbf{\mathcal{X}}^0 - \mathbf{\mathcal{X}}^*\|_{\mathrm{HS}} \leqslant \frac{\underline{\lambda}}{4d(\sqrt{d}+1)(R_{3\mathbf{r}}/(1-R_{2\mathbf{r}})+1)}$, where $\underline{\lambda} := \min_{k=1,\ldots,d} \sigma_{r_k}(\mathcal{M}_k(\mathbf{\mathcal{X}}^*))$ is the minimum of least singular values at each matricization of $\mathbf{\mathcal{X}}^*$. Then for all $t \geqslant 0$,*

$$\|\mathbf{\mathcal{X}}^{t+1} - \mathbf{\mathcal{X}}^*\|_{\mathrm{HS}} \leqslant d(\sqrt{d}+1)\left(\frac{R_{3\mathbf{r}}}{1-R_{2\mathbf{r}}} + 1\right)\frac{\|\mathbf{\mathcal{X}}^t - \mathbf{\mathcal{X}}^*\|_{\mathrm{HS}}^2}{\underline{\lambda}} + \frac{\sqrt{d}+1}{1-R_{2\mathbf{r}}}\|(\mathscr{A}^*(\boldsymbol{\varepsilon}))_{\max(2\mathbf{r})}\|_{\mathrm{HS}}.$$

Recall, $(\cdot)_{\max(\mathbf{r})}$ denotes the best Tucker rank $\mathbf{r}$ approximation of the tensor "$\cdot$".

13

In particular, if $\boldsymbol{\varepsilon} = 0$, then $\{\boldsymbol{\mathcal{X}}^t\}$ converges quadratically to $\boldsymbol{\mathcal{X}}^*$ as

$$\|\boldsymbol{\mathcal{X}}^{t+1} - \boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}} \leqslant d(\sqrt{d} + 1)\left(\frac{R_{3\mathbf{r}}}{1 - R_{2\mathbf{r}}} + 1\right)\frac{\|\boldsymbol{\mathcal{X}}^t - \boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}}^2}{\underline{\lambda}}, \quad \forall\, t \geqslant 0.$$

Theorem 1 shows with some proper assumptions on $\mathscr{A}$ and initialization, the iterates of RGN converge quadratically to the ball centered at $\boldsymbol{\mathcal{X}}^*$ and of radius $O(\|(\mathscr{A}^*(\boldsymbol{\varepsilon}))_{\max(2\mathbf{r})}\|_{\mathrm{HS}})$. Especially if $\boldsymbol{\varepsilon} = 0$, i.e., the observations are noiseless, $\boldsymbol{\mathcal{X}}^t$ converges quadratically to the exact $\boldsymbol{\mathcal{X}}^*$. To the best of our knowledge, this is the first provable quadratic convergence guarantee for both low-rank tensor estimation and recovery.

We note that, in the noisy setting, RGN in general converges to a stationary point with nonzero function value, so the classical optimization theory can only yield local (super)linear convergence guarantee for RGN (Absil et al., 2009; Breiding and Vannieuwenhoven, 2018). Our result complements the classic theory of RGN: Absil et al. (2009); Breiding and Vannieuwenhoven (2018) studied the limiting convergence rate of RGN to a stationary point, while we show RGN converges quadratically to $\boldsymbol{\mathcal{X}}^*$, the true parameter of interest, up to some optimal statistical error (see the forthcoming Theorem 2). This also suggests that to achieve quadratic convergence performance in low-rank tensor estimation, the more sophisticated Riemannian Newton algorithm may be unnecessary as simple RGN already enjoys the quadratic convergence for estimating $\boldsymbol{\mathcal{X}}^*$ with theoretical guarantees.

Remark 2 (Initialization) *The convergence theory in Theorem 1 requires an initialization condition. Our condition says that $\|\boldsymbol{\mathcal{X}}^0 - \boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}}$ needs to be on the order of $\underline{\lambda}$. In the matrix setting, i.e., $d = 2$, this condition matches the initialization condition in the literature for using two-stage nonconvex optimization methods to solve the low-rank matrix recovery problems (Charisopoulos et al., 2021; Ma et al., 2019; Sun and Luo, 2015; Tu et al., 2016; Wang et al., 2017; Zhao et al., 2015; Zheng and Lafferty, 2015). To this point of view, our initialization condition is an extension of that. Moreover, in practice, the SVD-based methods often provide a sufficiently good initialization that meets the requirement in many statistical applications. We will further illustrate this point in Section 4. The numerical studies in Section 6.1 show that RGN can still work well under random initialization. We leave future work to provide convergence guarantees of RGN under random initialization.*

Moreover, there is also a factor $1/d^{3/2}$ in the initialization condition. In typical applications, such as brain MRI images or fMRI images, d will be a moderately large value, say 3 or 4. If d is large, then the $1/d^{3/2}$ factor will be important and the power of d also comes for reasons. The first reason is due to the fact that performing exact low-rank tensor projection is computationally intractable and efficient procedures such as T-HOSVD and ST-HOSVD can only achieve quasi-projection property with approximation constant $\sqrt{d}$ as we discussed in Proposition 2. The second reason is from the result in Lemma 9 that it has a factor d in the numerator of the bound. This can also be seen from the proof that the effective dimension of the orthogonal complement of $T_{\boldsymbol{\mathcal{X}}^t}\mathbb{M}_{\mathbf{r}}$, which controls the error, scales linearly in d. These two factors together yield the $1/d^{3/2}$ factor.

Remark 3 (Convergence under Restricted Isometry Property) *In the literature, the RIP-type assumptions are widely used to establish linear convergence guarantees for various first-order algorithms in low-rank matrix/tensor recovery. A common strategy to*

establish such results is to first show the linear convergence of the empirical loss and then transfer to the convergence of the iterates (Jain et al., 2010). To our best knowledge, we are the first to use RIP to establish the second-order convergence of RGN by directly showing the contraction of the iterates. The key lemmas in our theoretical analysis are Lemma 8, which bounds the per-iteration least squares estimation error, and Lemma 9, which bounds the projection of $\boldsymbol{\mathcal{X}}^*$ on the orthogonal complement of $T_{\boldsymbol{\mathcal{X}}^t}\mathbb{M}_{\mathbf{r}}$.

Corollary 1 (Two Phases of Convergence of RGN) *Suppose the conditions in Theorem 1 hold. Define* $\Delta := \sqrt{\frac{\underline{\lambda}\|(\mathscr{A}^*(\boldsymbol{\varepsilon}))_{\max(2\mathbf{r})}\|_{\mathrm{HS}}}{d(1+R_{3\mathbf{r}}-R_{2\mathbf{r}})}}$. *At iteration* t,

(Phase I) If $\|\boldsymbol{\mathcal{X}}^t - \boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}} \geqslant \Delta$, *then* $\|\boldsymbol{\mathcal{X}}^{t+1} - \boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}} \leqslant 2d(\sqrt{d}+1)(\frac{R_{3\mathbf{r}}}{1-R_{2\mathbf{r}}}+1)\frac{\|\boldsymbol{\mathcal{X}}^t - \boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}}^2}{\underline{\lambda}}$;

(Phase II) If $\|\boldsymbol{\mathcal{X}}^t - \boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}} \leqslant \Delta$, *then* $\|\boldsymbol{\mathcal{X}}^{t+1} - \boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}} \leqslant \frac{2(\sqrt{d}+1)}{1-R_{2\mathbf{r}}}\|(\mathscr{A}^*(\boldsymbol{\varepsilon}))_{\max(2\mathbf{r})}\|_{\mathrm{HS}}$.

In summary, we have

$$\|\boldsymbol{\mathcal{X}}^t - \boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}} \leqslant 2^{-2^t}\|\boldsymbol{\mathcal{X}}^0 - \boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}} + \frac{2(\sqrt{d}+1)}{1-R_{2\mathbf{r}}}\|(\mathscr{A}^*(\boldsymbol{\varepsilon}))_{\max(2\mathbf{r})}\|_{\mathrm{HS}}, \quad \forall t \geqslant 0.$$

In addition, as long as

$$t_{\max} \geqslant T_{\max} := \left\lceil \log\left(1 \vee \frac{1}{2}\log\left(\frac{d(1+R_{3\mathbf{r}}-R_{2\mathbf{r}})\|\boldsymbol{\mathcal{X}}^0 - \boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}}^2}{\underline{\lambda}\|(\mathscr{A}^*(\boldsymbol{\varepsilon}))_{\max(2\mathbf{r})}\|_{\mathrm{HS}}}\right)\right)\right\rceil + 1, \qquad (20)$$

we have $\|\boldsymbol{\mathcal{X}}^{t_{\max}} - \boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}} \leqslant \frac{2(\sqrt{d}+1)}{1-R_{2\mathbf{r}}}\|(\mathscr{A}^*(\boldsymbol{\varepsilon}))_{\max(2\mathbf{r})}\|_{\mathrm{HS}}$.

Notice the initialization error required in Theorem 1 is of order $\underline{\lambda}$, which is in general larger than $\|(\mathscr{A}^*(\boldsymbol{\varepsilon}))_{\max(2\mathbf{r})}\|_{\mathrm{HS}}$, so the magnitude of Δ in Corollary 1 can be much bigger than $\|(\mathscr{A}^*(\boldsymbol{\varepsilon}))_{\max(2\mathbf{r})}\|_{\mathrm{HS}}$. Corollary 1 shows that the convergence of RGN has two different phases: in Phase I that $\|\boldsymbol{\mathcal{X}}^t - \boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}}$ is bigger than the threshold Δ, $\boldsymbol{\mathcal{X}}^t$ converges quadratically to $\boldsymbol{\mathcal{X}}^*$; in Phase II that $\|\boldsymbol{\mathcal{X}}^t - \boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}}$ is smaller than the threshold, with one extra step, the estimation error of $\boldsymbol{\mathcal{X}}^{t+1}$ becomes at most $\frac{2(\sqrt{d}+1)}{1-R_{2\mathbf{r}}}\|(\mathscr{A}^*(\boldsymbol{\varepsilon}))_{\max(2\mathbf{r})}\|_{\mathrm{HS}}$.

3.3 Optimality of RGN

Next, we further introduce a lower bound to show $\xi := \|(\mathscr{A}^*(\boldsymbol{\varepsilon}))_{\max(2\mathbf{r})}\|_{\mathrm{HS}}$ is essential in the estimation error upper bounds of Theorem 1 and Corollary 1.

Theorem 2 (Minimax Lower Bound for Tensor Estimation) *Consider the following class of* $(\widetilde{\mathscr{A}}, \widetilde{\boldsymbol{\mathcal{X}}}, \widetilde{\boldsymbol{\varepsilon}})$:

$$\mathcal{F}_{\mathbf{r}}(\xi) = \left\{ \left(\widetilde{\mathscr{A}}, \widetilde{\boldsymbol{\mathcal{X}}}, \widetilde{\boldsymbol{\varepsilon}}\right) : \begin{array}{l} \widetilde{\mathscr{A}} \text{ satisfies } 3\mathbf{r}\text{-TRIP}, \widetilde{\boldsymbol{\mathcal{X}}} \text{ is of Tucker rank at most } \mathbf{r}, \\ \|(\widetilde{\mathscr{A}}^*(\widetilde{\boldsymbol{\varepsilon}}))_{\max(2\mathbf{r})}\|_{\mathrm{HS}} \leqslant \xi \end{array} \right\}.$$

Under the low-rank tensor estimation model (1), *we have*

$$\inf_{\widehat{\boldsymbol{\mathcal{X}}}} \sup_{\left(\widetilde{\mathscr{A}}, \widetilde{\boldsymbol{\mathcal{X}}}, \widetilde{\boldsymbol{\varepsilon}}\right) \in \mathcal{F}_{\mathbf{r}}(\xi)} \|\widehat{\boldsymbol{\mathcal{X}}} - \widetilde{\boldsymbol{\mathcal{X}}}\|_{\mathrm{HS}} \geqslant 2^{-1/2}\xi.$$

Remark 4 ((One-step) Optimality of RGN for Low-rank Tensor Estimation)
Theorem 2 and Corollary 1 together show that with a fixed tensor order d and a proper initialization, RGN achieves rate-optimal estimation error in class $\mathcal{F}_{\mathbf{r}}(\xi)$ after at most double-logarithmic, i.e., $T_{\max}$ defined in (20), number iterations.

Corollary 1 also shows in Phase II convergence of RGN, with one extra step, the estimation error $\boldsymbol{\mathcal{X}}^{t+1}$ becomes statistical rate-optimal. Such the one-step optimality shares the same spirit as the "one-step MLE" property for Newton algorithm in the literature on statistical inference (see Bickel (1975) and (Shao, 2006, Chapter 4.5)). To our best knowledge, this one-step optimality phenomenon is new for RGN.

4 Implications in Statistics and Machine Learning

In this section, we study the performance of RGN in two specific problems in machine learning: tensor regression and tensor SVD. In addition, the algorithm is applicable to a broader range of settings discussed in the introduction. Throughout this section, we denote $\bar{p} := \max_k p_k, \underline{p} := \min_k p_k, \bar{r} = \max_k r_k, \underline{\lambda} := \min_k \sigma_{r_k}(\mathcal{M}_k(\boldsymbol{\mathcal{X}}^*)), \bar{\lambda} := \max_k \sigma_1(\mathcal{M}_k(\boldsymbol{\mathcal{X}}^*))$ and $\kappa := \bar{\lambda}/\underline{\lambda}$.

4.1 Tensor Regression

Tensor Regression is a basic problem for supervised tensor learning, for which the readers are referred to Section 1.2 for a literature review. Specifically, we assume $\{\boldsymbol{\mathcal{A}}_i\}_{i=1}^n$ are independent and have i.i.d. $N(0, 1/n)$ entries; $\boldsymbol{\varepsilon}_i \overset{i.i.d.}{\sim} N(0, \sigma^2/n)$ in model (1). Suppose the initialization is obtained by T-HOSVD:

$$\boldsymbol{\mathcal{X}}^0 = \mathscr{A}^*(\mathbf{y}) \times_{k=1}^d P_{\mathbf{U}_k^0}, \tag{21}$$

where $\mathbf{U}_k^0 = \text{SVD}_{r_k}(\mathcal{M}_k(\mathscr{A}^*(\mathbf{y})))$. Then we have the following theoretical guarantee for the outcome of RGN for tensor regression.

Theorem 3 (RGN for Tensor Regression) *Consider RGN for tensor regression. Suppose $\bar{r} \leqslant \underline{p}^{1/2}$, $\mathcal{H}_{\mathbf{r}}$ is either T-HOSVD or ST-HOSVD. If $n \geqslant c(d)(\|\boldsymbol{\mathcal{X}}^*\|_{\text{HS}}^2 + \sigma^2)\kappa^2\sqrt{\bar{r}}\bar{p}^{d/2}/\underline{\lambda}^2$, and $t_{\max} \geqslant C(d) \log \log \left(\dfrac{\underline{\lambda}\sqrt{n}}{\sigma\sqrt{\sum_{k=1}^d r_k p_k + \prod_{k=1}^d r_k}} \right)$, then*

$$\|\boldsymbol{\mathcal{X}}^{t_{\max}} - \boldsymbol{\mathcal{X}}^*\|_{\text{HS}} \leqslant c(\sqrt{d} + 1)\sigma\sqrt{\left(\sum_{k=1}^d r_k p_k + \prod_{k=1}^d r_k \right)/n} \tag{22}$$

holds with probability at least $1 - \underline{p}^{-C}$. Here c, C are some universal positive constants, and $c(d), C(d)$ are some constants that depend on d only.

Suppose $d, \bar{r}$ are fixed. Note that $O(d\bar{p}\bar{r})$ samples is enough to guarantee the $\mathbf{r}$-TRIP and indeed this is the information-theoretic limit to make the problem solvable (Zhang et al., 2020a, Theorem 5). However, we find to achieve the initialization assumption in Theorem 1 via spectral method, a significantly larger sample complexity $O(c(d)\sqrt{\bar{r}}\bar{p}^{d/2})$ is

needed. Such a gap originates from the difficulty of computing the best Tucker rank $\mathbf{r}$ approximation of $\mathscr{A}^*(\mathbf{y})$ efficiently in initialization (Hillar and Lim, 2013). This difficulty is also a common reason that causes the so-called "statistical and computational gaps" in various tensor problems (Richard and Montanari, 2014; Zhang and Xia, 2018; Barak and Moitra, 2016; Luo and Zhang, 2020, 2022a; Brennan and Bresler, 2020; Han et al., 2022a). See more discussions in Section 4.3.

4.2 Tensor SVD

Tensor SVD is a specific model covered by the prototypical model (1), which can be equivalently written as

$$\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{X}}^* + \boldsymbol{\mathcal{E}},$$

where $\boldsymbol{\mathcal{X}}^*$ has Tucker decomposition as (3) and $\boldsymbol{\mathcal{E}}$ has i.i.d. $N(0, \sigma^2)$ entries. The goal is to estimate $\boldsymbol{\mathcal{X}}^*$ based on $\boldsymbol{\mathcal{Y}}$. As illustrated by the following Lemma 5, the RGN algorithm for tensor SVD can be significantly simplified from the original Algorithm 1.

Lemma 5 (Least Squares Solution of RGN in tensor SVD) *Consider the tensor SVD model* $\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{X}}^* + \boldsymbol{\mathcal{E}}$. *Suppose at the t-th iteration of RGN, the current iterate is* $\boldsymbol{\mathcal{X}}^t$ *with Tucker decomposition* $[\![\boldsymbol{\mathcal{S}}^t; \mathbf{U}_1^t, \dots, \mathbf{U}_d^t]\!]$. *Then the least squares in (18) can be solved by*

$$\boldsymbol{\mathcal{B}}^{t+1} = \boldsymbol{\mathcal{Y}} \times_{k=1}^d \mathbf{U}_k^{t\top}, \quad \mathbf{D}_k^{t+1} = \mathbf{U}_{k\perp}^{t\top} \mathcal{M}_k(\boldsymbol{\mathcal{Y}}) \mathbf{W}_k^t, \quad k = 1, \cdots, d.$$

Here $\mathbf{W}_k^t$ *is given in (10).*

Consequently, we simplify RGN for tensor SVD to the following Algorithm 2.

Algorithm 2 Riemannian Gauss-Newton for Tensor SVD

Input: $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{p_1 \times \cdots \times p_d}$, $t_{\max}$, a Tucker rank $\mathbf{r}$ initialization $\boldsymbol{\mathcal{X}}^0$ with Tucker decomposition $[\![\boldsymbol{\mathcal{S}}^0; \mathbf{U}_1^0, \dots, \mathbf{U}_d^0]\!]$.

1: **for** $t = 0, 1, \dots, t_{\max} - 1$ **do**
2:      Update
$$\boldsymbol{\mathcal{X}}^{t+1} = [\![\boldsymbol{\mathcal{S}}^{t+1}; \mathbf{U}_1^{t+1}, \dots, \mathbf{U}_d^{t+1}]\!] = \mathcal{H}_{\mathbf{r}}\left(P_{T_{\boldsymbol{\mathcal{X}}^t}}(\boldsymbol{\mathcal{Y}})\right).$$

     Here $P_{T_{\boldsymbol{\mathcal{X}}^t}}(\cdot)$ is the projection operator that project $\boldsymbol{\mathcal{Y}}$ onto the tangent space of $\boldsymbol{\mathcal{X}}^t$ defined in (11), $\mathcal{H}_{\mathbf{r}}(\cdot)$ maps the input tensor to be a Tucker rank $\mathbf{r}$ tensor.
3: **end for**
Output: $\boldsymbol{\mathcal{X}}^{t_{\max}}$.

The following Theorem 4 gives the theoretical guarantee of RGN initialized with T-HOSVD for the tensor SVD.

Theorem 4 (RGN for Tensor SVD) *Consider RGN for tensor SVD. Suppose* $\bar{r} \leqslant p^{1/2}$, $\mathcal{H}_{\mathbf{r}}$ *is either T-HOSVD or ST-HOSVD, and the algorithm is initialized by T-HOSVD, i.e.,* $\boldsymbol{\mathcal{X}}^0 = \boldsymbol{\mathcal{Y}} \times_{k=1}^d P_{\mathbf{U}_k^0}$ *where* $\mathbf{U}_k^0 = \mathrm{SVD}_{r_k}(\mathcal{M}_k(\boldsymbol{\mathcal{Y}}))$. *If the least singular value* $\underline{\lambda} \geqslant$

$$c(d)\kappa\bar{p}^{d/4}\bar{r}^{1/4}\sigma, \text{ and } t_{\max} \geqslant C(d)\log\log\left\{\underline{\lambda}\Big/\left(\sigma\sqrt{\sum_{k=1}^{d}r_k p_k + \prod_{k=1}^{d}r_k}\right)\right\},$$

$$\|\boldsymbol{\mathcal{X}}^{t_{\max}} - \boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}} \leqslant c \cdot (\sqrt{d}+1)\sigma\sqrt{\sum_{k=1}^{d}r_k p_k + \prod_{k=1}^{d}r_k} \tag{23}$$

holds with probability at least $1 - \exp(-C\underline{p})$.

Remark 5 (RGN Versus Existing Algorithms in Tucker Tensor Decomposition)
We note that Algorithm 2 can also be viewed as a Riemannian Gauss-Newton algorithm for Tucker tensor decomposition (Kolda and Bader, 2009). In the literature, a few other second-order methods have been proposed for Tucker tensor decomposition under Grassmann or quotient manifold structures, such as (quasi-)Newton-Grassmann method (Eldén and Savas, 2009; Savas and Lim, 2010), geometric Newton method (Ishteva et al., 2009), and Riemannian trust region method (Ishteva et al., 2011). To the best of our knowledge, this is the first Riemannian Gauss-Newton algorithm for Tucker tensor decomposition developed using the embedded manifold structure on $\mathbb{M}_{\mathbf{r}}$. At the same time, we comment that different from the common goal in tensor decomposition which aims to find a good low-rank approximation of $\boldsymbol{\mathcal{Y}}$, our goal here is to find a good estimator for $\boldsymbol{\mathcal{X}}^*$. So the convergence results established in Eldén and Savas (2009); Savas and Lim (2010); Ishteva et al. (2009, 2011) are not directly comparable to ours. Moreover, due to the difference in the targets, the quasi-projection property of T-HOSVD and ST-HOSVD in Tucker tensor decomposition (see Definition 1) does not imply they are good estimators for $\boldsymbol{\mathcal{X}}^*$. In fact, it has been shown in Zhang and Xia (2018) that T-HOSVD is strictly suboptimal in estimating $\boldsymbol{\mathcal{X}}^*$ in tensor SVD, see more discussions in the second point of Section 4.3.

4.3 A Few More Remarks for Tensor Regression and Tensor SVD

In this section, we provide a few remarks regarding our results in Theorems 3 and 4.

- **(Estimation Error, Convergence Rate, and Signal Strength)** In both tensor regression and SVD, the estimation upper bounds in Theorems 3 and 4 match the lower bounds in the literature, (Zhang and Xia, 2018, Theorem 3) and (Zhang et al., 2020a, Theorem 5), which shows that RGN achieves the minimax optimal rate of estimation error. Compared to existing algorithms in the literature on tensor regression and SVD (Ahmed et al., 2020; Chen et al., 2019a; Han et al., 2022b; Zhang and Xia, 2018), RGN is the first to achieve the minimax rate-optimal estimation error with only a *double-logarithmic number of iterations* attributed to its second-order convergence.

  Suppose $d, \bar{r}$ are fixed and the condition number κ is of order $O(1)$, we note that the sample size requirement ($n \geqslant O(\sqrt{\bar{r}}\bar{p}^{d/2})$) in tensor regression and least singular value requirement ($\underline{\lambda}/\sigma \geqslant O(\bar{p}^{d/4}\bar{r}^{1/4})$) in tensor SVD match the start-of-the-arts in literature (see Table 1 for a comparison). Rigorous evidence has been established to show that the least singular value lower bounds in tensor SVD are essential for any polynomial-time algorithm to succeed (Zhang and Xia, 2018; Brennan and Bresler, 2020; Luo and Zhang, 2022a). Recent studies in Luo and Zhang (2022b); Diakonicolas et al. (2023) have

presented similar evidence for tensor regression, suggesting that a sample complexity of $O(\bar{p}^{d/2})$ is essential for the success of any polynomial-time algorithm.

- (**Guarantees of the Spectral Initializations in Tensor Regression and Tensor SVD**.) From the proof of Theorem 3 (46) and Theorem 4 (50), we show the spectral initializations in tensor regression and tensor SVD have the following guarantees:

  - tensor regression:

$$\|\boldsymbol{\mathcal{X}}^0 - \boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}} \leqslant C(d) \left(\kappa\tilde{\sigma}\sqrt{\frac{\sum_{k=1}^d p_k r_k + \prod_{k=1}^d r_k}{n}} + \frac{(\bar{r}\prod_{k=1}^d p_k)^{1/2}\kappa\tilde{\sigma}^2}{\underline{\lambda}n} \right), \quad (24)$$

  where $C(d)$ is some constant depending on d only and $\tilde{\sigma} := \sqrt{\|\boldsymbol{\mathcal{A}}\|_{\mathrm{HS}}^2 + \sigma^2} > \sigma$.

  - tensor SVD:

$$\|\boldsymbol{\mathcal{X}}^0 - \boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}} \leqslant C(d) \left(\kappa\sigma\sqrt{\sum_{k=1}^d p_k r_k + \prod_{k=1}^d r_k} + \frac{(\bar{r}\prod_{k=1}^d p_k)^{1/2}\kappa\sigma}{\underline{\lambda}} \right). \quad (25)$$

Comparing (24) and (25) with (22) and (23), we can see the estimation error guarantees of the spectral initializations are strictly suboptimal comparing to the ones with iterative refinement.

5 Computational Complexity of RGN

Next, we investigate the computational complexity of RGN. First, the per-iteration computational cost for RGN with a general linear map $\mathscr{A}$ is $O(np^d r + n(r^d + dpr)^2)$ if $p_1 = \cdots = p_d = p$ and $r_1 = \cdots = r_d = r$. Here, $O(np^d r)$ and $O(n(r^d + dpr)^2)$ are due to the costs for constructing the covariate maps (17) and for solving the least squares problem (18), respectively. If $r, d \ll n, p$ (which is a typical case in practice), the cost of constructing the covariates maps dominates and the per-iteration cost of RGN is $O(np^d r)$. Performing T-HOSVD or ST-HOSVD in $\mathcal{H}_{\mathbf{r}}$ can be expensive in general. Since the tensor we apply $\mathcal{H}_{\mathbf{r}}$ on lies in the tangent space of the current iterate and is at most Tucker rank $2\mathbf{r}$ (Lemma 2), the retraction via T-HOSVD and ST-HOSVD can be performed efficiently (Cai et al., 2020).

A comparison of the per-iteration computational complexity of RGN and several classic algorithms, including alternating minimization (Alter Mini), projected gradient descent (PGD), and gradient descent (GD), in tensor regression and tensor SVD examples is provided in Table 1 in the introduction section. The main complexity of alternating minimization (Alter Mini) (Zhou et al., 2013; Li et al., 2018) is from constructing the covariates in solving the least squares and the main complexity of the projected gradient descent (PGD) (Chen et al., 2019a) and gradient descent (GD) (Han et al., 2022b) are from computing the gradient. We can see RGN has the same per-iteration complexity as Alter Mini and comparable complexity with PGD and GD when $r \ll n, p$. In addition, RGN and Alter Mini are tuning-free, while a proper step size is crucial for PGD and GD to have fast convergence. Finally, RGN enjoys a second-order convergence as shown in Section 3, while the convergence rates

of all other algorithms are at most linear. We will further provide a numerical comparison of these algorithms in Section 6.2.

Furthermore, in specific scenarios where the covariates $\boldsymbol{\mathcal{A}}_i$ have more structures, the procedure of RGN can be simplified and the computational complexity can be reduced. In Section 4.2, we have already seen RGN can be significantly simplified in tensor SVD. Here we discuss two additional scenarios: tensor estimation via rank-1 projections and tensor completion.

- **Tensor estimation via rank-1 projections.** In this application, $\boldsymbol{\mathcal{A}}_i = \mathbf{a}_1^{(i)} \circ \cdots \circ \mathbf{a}_d^{(i)}$ and the construction of covariates maps in (17) can be simplified as follows

$$(\mathscr{A}_{\boldsymbol{\mathcal{B}}})_i = \mathbf{U}_1^{t\top}\mathbf{a}_1^{(i)} \circ \cdots \circ \mathbf{U}_d^{t\top}\mathbf{a}_d^{(i)}, \quad (\mathscr{A}_{\mathbf{D}_k})_i = \mathbf{U}_{k\perp}^{t\top}\mathbf{a}_k^{(i)}(\otimes_{j \neq k}\mathbf{U}_j^{t\top}\mathbf{a}_j^{(i)})^\top \mathbf{V}_k^t, \ k = 1, \ldots, d.$$

  Here $\mathbf{V}_k^t = \mathrm{QR}(\mathcal{M}_k(\boldsymbol{\mathcal{S}}^t)^\top)$ and $\boldsymbol{\mathcal{S}}^t = \boldsymbol{\mathcal{X}}^t \times_{k=1}^d \mathbf{U}_k^{t\top}$. The computational cost for constructing the covariates maps by such a scheme is $O(n(p^2 + r^d))$, which is much cheaper than $O(np^d r)$, the computational cost in the general setting, when $d \geqslant 3$.

- **Tensor completion.** We observe a fraction of entries indexed by Ω from the target tensor. For $(i_1, \ldots, i_d) \in \Omega$, the corresponding covariate is $\mathbf{e}_{i_1} \circ \cdots \circ \mathbf{e}_{i_d}$. Then, simple calculation yields the covariates maps can be calculated as

$$(\mathscr{A}_{\boldsymbol{\mathcal{B}}})_i = (\mathbf{U}_1^{t\top})_{[:,i_1]} \circ \cdots \circ (\mathbf{U}_d^{t\top})_{[:,i_d]}, \quad (\mathscr{A}_{\mathbf{D}_k})_i = (\mathbf{U}_{k\perp}^{t\top})_{[:,i_k]}(\otimes_{j \neq k}(\mathbf{U}_j^t)_{[i_j,:]})\mathbf{V}_k^t, \ k = 1, \ldots, d.$$

  The per-iteration cost for constructing the covariates maps above is $O(n(r^d + pr))$.

6 Numerical Studies on RGN

We consider four specific numerical settings: 1. tensor regression under random design; 2. tensor estimation via rank-1 projections; 3. tensor completion; 4. tensor SVD. In tensor regression and tensor estimation via rank-1 projections, we generate the covariates $\boldsymbol{\mathcal{A}}_i$ and $\mathbf{a}_k^{(i)}$ with i.i.d. $N(0,1)$ entries. In tensor completion, partial observations indexed by Ω are sampled uniformly at random from the noisy tensor $\boldsymbol{\mathcal{Y}}$. In each simulation setting, we generate $\varepsilon_i \overset{i.i.d.}{\sim} N(0, \sigma^2)$, $\{\mathbf{U}_k\}_{k=1}^3$ uniformly at random from $\mathbb{O}_{p,r}$, and $\boldsymbol{\mathcal{S}} \in \mathbb{R}^{r \times r \times r}$ with i.i.d. $N(0,1)$ entries; then we calculate $\boldsymbol{\mathcal{X}}^* = \boldsymbol{\mathcal{S}} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3$. In tensor SVD, we also rescale $\boldsymbol{\mathcal{S}}$ so that $\min_{k=1,2,3} \sigma_{r_k}(\mathcal{M}_k(\boldsymbol{\mathcal{X}}^*))$ is equal to a pre-specified value $\underline{\lambda}$. The implementation details of RGN under each setting have been discussed in Sections 4 and 5. We apply the following initialization schemes respectively for each problem.

- Tensor regression/Tensor estimation via rank-1 projections: $\boldsymbol{\mathcal{X}}^0 = \mathscr{A}^*(\mathbf{y}) \times_{k=1}^d P_{\mathbf{U}_k^0}$, where $\mathbf{U}_k^0 = \mathrm{SVD}_{r_k}(\mathcal{M}_k(\mathscr{A}^*(\mathbf{y})))$.

- Tensor completion: Suppose $\rho = |\Omega|/(\prod_{j=1}^d p_j)$ is the sampling ratio. Denote

$$\boldsymbol{\mathcal{Y}}_\Omega = \begin{cases} \boldsymbol{\mathcal{Y}}_{[i_1,\ldots,i_d]}, & \text{if } (i_1, \ldots, i_d) \in \Omega \\ 0, & \text{otherwise.} \end{cases}$$

  Calculate $\mathcal{M}_k(\boldsymbol{\mathcal{Y}}_\Omega)\mathcal{M}_k(\boldsymbol{\mathcal{Y}}_\Omega)^\top$, zero out the diagonal entries of $\mathcal{M}_k(\boldsymbol{\mathcal{Y}}_\Omega)\mathcal{M}_k(\boldsymbol{\mathcal{Y}}_\Omega)^\top$, let $\mathbf{U}_k^0$ be the leading r_k singular vectors of the diagonal-zero-out matrix $\mathcal{M}_k(\boldsymbol{\mathcal{Y}}_\Omega)\mathcal{M}_k(\boldsymbol{\mathcal{Y}}_\Omega)^\top$, and initialize $\boldsymbol{\mathcal{X}}^0 = (\boldsymbol{\mathcal{Y}}_\Omega/\rho) \times_{k=1}^d P_{\mathbf{U}_k^0}$ (Xia et al., 2021).

- Tensor SVD: Initialize $\boldsymbol{\mathcal{X}}^0 = \boldsymbol{\mathcal{Y}} \times_{k=1}^d P_{\mathbf{U}_k^0}$, where $\mathbf{U}_k^0 = \text{SVD}_{r_k}(\mathcal{M}_k(\boldsymbol{\mathcal{Y}}))$.

Throughout the simulation studies, the error metric we consider is the relative root-mean-squared error (Relative RMSE) $\|\boldsymbol{\mathcal{X}}^t - \boldsymbol{\mathcal{X}}^*\|_{\text{HS}}/\|\boldsymbol{\mathcal{X}}^*\|_{\text{HS}}$. The algorithm is terminated when it reaches the maximum number of iterations $t_{\max} = 300$ or the corresponding error metric is less than 10^{-14}. Unless otherwise noted, the reported results are based on the averages of 50 simulations and on a computer with Intel Xeon E5-2680 2.5GHz CPU. The code of our algorithm can be found at https://github.com/yuetianluo/RGN-for-Tensor-Estimation.

6.1 Numerical Performance of RGN

We first examine the convergence rate of RGN in each of the above-mentioned problems. We set $\sigma = 1$ for tensor SVD and $\sigma \in \{0, 10^{-6}\}$ in the other three problems. We skip the noiseless setting of tensor SVD since the initialization via T-HOSVD already achieves exact recovery. The convergence performance of RGN in tensor regression and tensor completion is presented in Figure 1 and the performance in tensor estimation via rank-1 projections and tensor SVD is presented in Figure 2. In tensor regression under random design/rank-1 projections and tensor completion, the estimation error converges quadratically to the minimum precision in the noiseless setting and converges quadratically to a limit determined by the noise level in the noisy setting. In tensor SVD, we observe RGN initialized with T-HOSVD converges with almost one iteration. We tried several other simulation settings and observe a similar phenomenon. This suggests that in tensor SVD, RGN may achieve one-step optimality directly after initialization in estimating $\boldsymbol{\mathcal{X}}^*$ as discussed in Remark 4. We leave it as future work to further investigate this phenomenon.

(a) Tensor estimation via rank-1 projections: $p = 30, r = 3$.

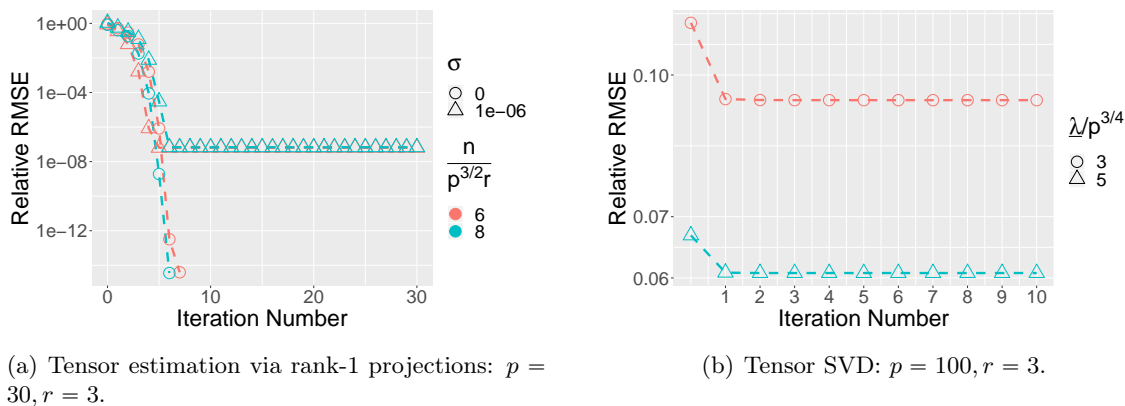(b) Tensor SVD: $p = 100, r = 3$.

Figure 2: Convergence performance of RGN in tensor estimation via rank-1 projections and tensor SVD under spectral initialization.

We note that in many problems, spectral initialization may not be obtainable, and another common initialization choice for iterative algorithms in practice is random initialization. Next, we illustrate the performance of RGN with random initialization in two examples: tensor regression and tensor estimation via rank-1 projections. Here we simply initialize $\boldsymbol{\mathcal{X}}^0$ by i.i.d. standard Gaussian entries and the simulation results are given in Fig-

ure 3. We can see that RGN can still converge and estimate/recover $\mathcal{X}^*$ well under random initialization. However, there are two main differences compared to the convergence of RGN under spectral initialization: first, we find RGN generally requires a slightly larger sample size to convergence under random initialization; second, the iterate tends to fluctuate at the beginning stage before it enters the attraction region and larger sample size seems to make the algorithm more stable.
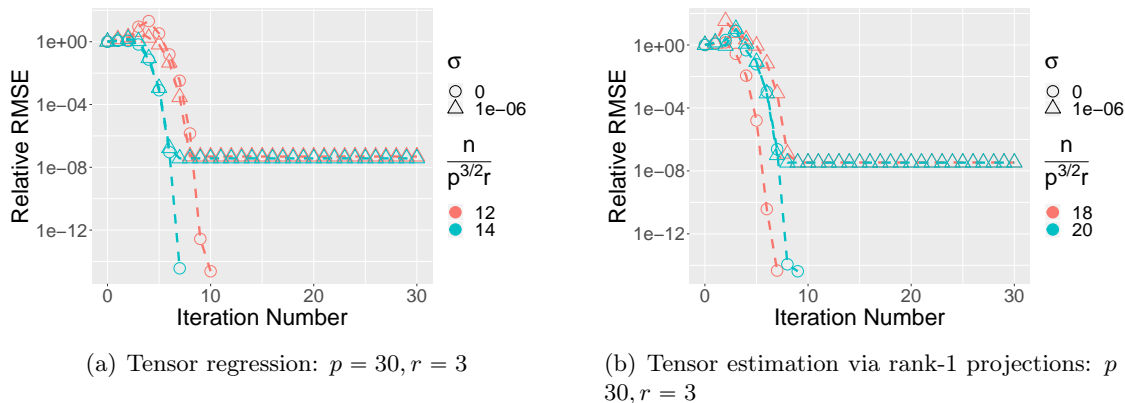

(a) Tensor regression: $p = 30, r = 3$

(b) Tensor estimation via rank-1 projections: $p = 30, r = 3$

Figure 3: Convergence performance of RGN in tensor regression and tensor estimation via rank-1 projections under random initialization

6.2 Comparison of RGN with Previous Algorithms

In this subsection, we compare RGN with other existing algorithms, including the Riemannian trust region method (RTR) (Heidel and Schulz, 2018), alternating minimization (Alter Mini) (Zhou et al., 2013; Li et al., 2018) [1], projected gradient descent (PGD)(Chen et al., 2019a) and gradient descent (GD) (Han et al., 2022b), in tensor regression. Since the approximate Riemannian Newton in Kressner et al. (2016) is developed under the setting where the linear map has additive and Kronecker-product-type structures, we choose not to compare it here. While implementing GD and PGD, we evaluate three choices of step size, $\frac{1}{n} * \{0.1, 0.5, 1\}$, then choose the best one following Zheng and Lafferty (2015). We set $p = 30, r = 3, n = 5 * p^{3/2}r$ and consider both the noiseless case ($\sigma = 0$) and the noisy case ($\sigma = 10^{-6}$).

We plot the relative RMSE versus iteration number/runtime in both the noiseless and noisy tensor regression settings in Figures 4 and 5, respectively. In both settings, RGN converges quadratically, and the Riemannian trust region method is slightly slower than our method but also has superlinear or quadratic convergence performance. All the other baseline algorithms converge in a much slower linear rate. To achieve an accuracy of 10^{-14} in the noiseless setting or the statistical error in the noisy setting, RGN requires a much smaller runtime than PGD, Alter Mini, and GD.

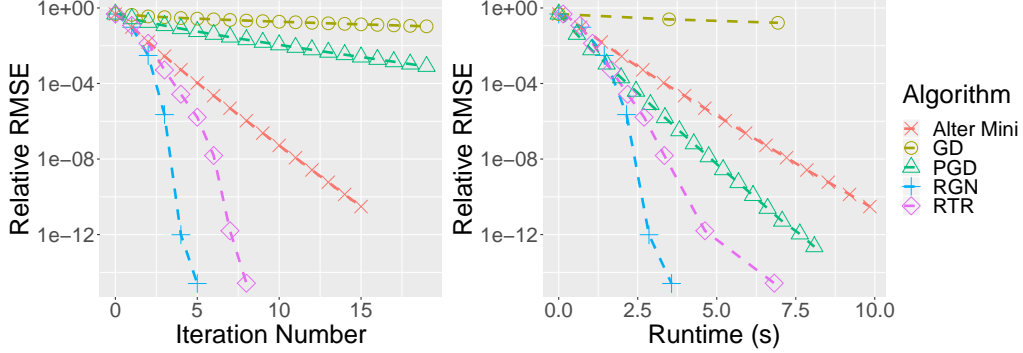1. Software package available at Zhou (2017)

Figure 4: Relative RMSE of RGN (this work), Riemannian trust region (RTR), alternating minimization (Alter Mini), projected gradient descent (PGD), and gradient descent (GD) in noiseless tensor regression. Here, $p = 30, r = 3, n = 5 * p^{3/2}r, \sigma = 0$.
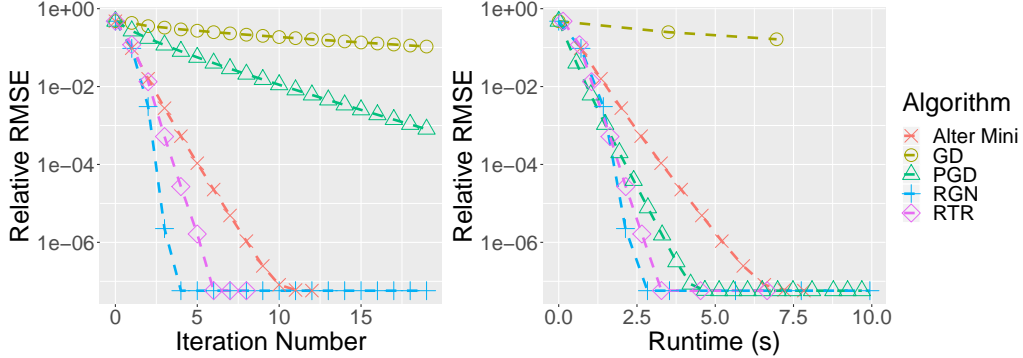

Figure 5: Relative RMSE of RGN (this work), Riemannian trust region (RTR), alternating minimization (Alter Mini), projected gradient descent (PGD) and gradient descent (GD) in noisy tensor regression. Here, $p = 30, r = 3, n = 5 * p^{3/2}r, \sigma = 10^{-6}$.

6.3 A Real Data Example

In this section, we demonstrate the advantages of our algorithm in the application of high-order image compression via rank-1 projection (Hao et al., 2020; Cai and Zhang, 2015). We consider the ADHD-200 dataset that contains magnetic resonance imaging (MRI) data from both the attention deficit hyperactivity disorder (ADHD) patients and the control group[2]. The dataset includes 973 subjects and each subject is associated with a 121-by-145-by-121 MRI image denoted by $\mathcal{X}$. The total storage space for these data through naive format is $121 \times 145 \times 121 \times 973 \times 4B \approx 7.48GB$, which is expensive for both storage and computation. Our goal is to compress the high-order image data via rank-1 projections and allow for efficient retrieval from the compressed rank-1 projections. Here for each image, we choose

---

2. Available at `http://neurobureau.projects.nitrc.org/ADHD200/Data.html`

23

to let $\boldsymbol{\mathcal{X}}^*$ be the subtensor $\boldsymbol{\mathcal{X}}_{[1:40,1:50,1:40]}$ with size $40 \times 50 \times 40$. There are two reasons for this choice: first, this experimental scale is already challenging for other algorithms as it is unclear how to leverage the rank-1 projection structure efficiently there; second, since the boundary of an MRI image often contains many zero entries, the selected subtensor has a low-rank structure. An illustration of singular value decay of three matricizations of one MRI image is given in Figure 6.
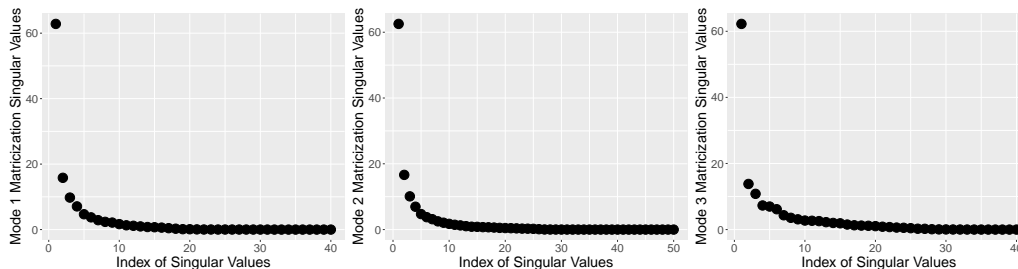

Figure 6: Illustration of low-rankness of $\boldsymbol{\mathcal{X}}^*$. Three plots represent the singular values for $\mathcal{M}_i(\boldsymbol{\mathcal{X}}^*)$, $i = 1, 2, 3$.

Specifically, we generate random vectors $\{\mathbf{a}_1^{(i)}, \mathbf{a}_2^{(i)}, \mathbf{a}_3^{(i)}\}_{i=1}^n$ with i.i.d. standard Gaussian entries and compute

$$\mathbf{y}_i = \langle \boldsymbol{\mathcal{X}}^*, \mathbf{a}_1^{(i)} \circ \mathbf{a}_2^{(i)} \circ \mathbf{a}_3^{(i)} \rangle, \quad \forall i = 1, \ldots, n.$$

We can store $\mathbf{y}$ and $\{\mathbf{a}_1^{(i)}, \mathbf{a}_2^{(i)}, \mathbf{a}_3^{(i)}\}_{i=1}^n$ in instead of the whole tensor $\boldsymbol{\mathcal{X}}^*$. If $n \ll p_1 p_2 p_3 / (\max p_i)$, we can reduce the memory cost from $O(p_1 p_2 p_3)$ to $O(n(p_1 + p_2 + p_3))$. Furthermore, We can apply our algorithm with inputs $\mathbf{y}$ and $\{\mathbf{a}_1^{(i)}, \mathbf{a}_2^{(i)}, \mathbf{a}_3^{(i)}\}_{i=1}^n$ to recover $\boldsymbol{\mathcal{X}}^*$.

We compare the relative RMSE of our algorithm with the Riemannian trust region method and the PGD method as these two have the best performance from the last simulation study. We stop the algorithm when the decrease of relative RMSE per iteration is less than 10^{-3}. The recovery performance of these three algorithms with spectral initialization for a randomly drawn MRI image is shown in Figure 7. We can see that in this image all three methods achieve roughly the same relative RMSE, but our algorithm requires significantly less number of iterations and runtime.

Moreover, we repeat the experiment for randomly drawn 100 MRI images, and then compare the averaged recovery performance and runtime of these algorithms. The results are given in Table 2. We can see that on average, our method achieves similar recovery guarantees as the Riemannian trust region method but in less runtime. Both these algorithms achieve better recovery performance than PGD.

7 Conclusion and Discussions

In this paper, we propose a new algorithm, Riemannian Gauss-Newton (RGN), for low Tucker rank tensor estimation. Under some reasonable assumptions, we show RGN achieves
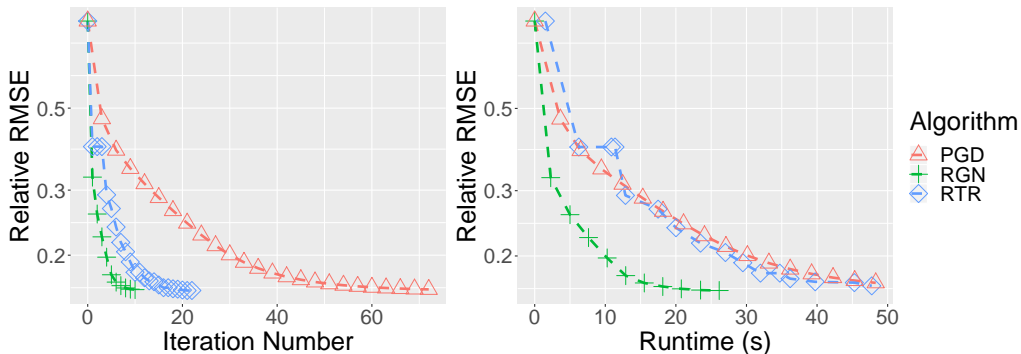
Figure 7: Relative RMSE of RGN (this work), Riemannian trust region (RTR), and projected gradient descent (PGD) over iteration number and runtime in MRI image recovery.

Algorithm	Relative RMSE	Runtime
RGN	0.155(0.0137)	**39.2** (6.49)
RTR	**0.153**(0.0137)	104.9 (17.17)
PGD	0.300(0.2598)	79.5 (48.79)

Table 2: Average relative RMSE and runtime (in second) of RGN (this work), Riemannian trust region (RTR), and projected gradient descent (PGD) in MRI image recovery. Mean values with standard deviation in the parenthesis are reported.

a local quadratic convergence and an optimal statistical error rate for low-rank tensor estimation.

There are a number of directions worth exploring in the future. First, our current convergence theory relies on the TRIP assumption, which may not hold in scenarios, such as tensor estimation via rank-1 projections and tensor completion. In Section 6.1, we show via simulation that RGN still works well without TRIP. It is an interesting future work to establish the "TRIP-less" theoretical guarantees of RGN. Another future direction is to study the convergence of RGN under random initialization. Some progress has been made on the convergence of randomly initialized (Riemannian) gradient descent in low-rank matrix recovery problems (Chen et al., 2019b; Hou et al., 2020). However, it can be much harder to establish similar results for RGN in low-rank tensor recovery problems.

Second, throughout the applications, we assume the noise is Gaussian distributed. In the scenarios that the noise is heavy-tailed or the data have outliers (Cai et al., 2022), we would like to consider using the robust loss (e.g., l_1 loss or Huber loss) in (18) instead of the l_2 loss or consider quantile tensor regression (Lu et al., 2020). It is interesting to see whether RGN work in those settings and can we give some theoretical guarantees there.

Third, this paper mainly focuses on the scalar response and tensor predictor model (1). In the literature, several papers have also studied the tensor response model (Sun and Li,

2017; Li and Zhang, 2017), it is interesting to see whether the RGN method can be applied to that setting.

Finally, we focus on low Tucker rank tensors in this paper. Although the Tucker format has many advantages, in ultra higher-order tensor problems, the storage cost of the core tensor in the Tucker format scales exponentially with respect to the tensor order and it is more desirable to consider other low-rank tensor decomposition formats, such as the hierarchical Tucker decomposition (Grasedyck, 2010; Hackbusch and Kühn, 2009) and tensor-train decomposition (Oseledets, 2011; Zhou et al., 2022). It is known that the set of fixed hierarchical rank or tensor-train rank tensors forms a smooth manifold (Uschmajew and Vandereycken, 2013; Hackbusch, 2012; Holtz et al., 2012), so it is interesting to see whether the RGN algorithms can be established in these settings.

Acknowledgments and Disclosure of Funding

Appendix A. T-HOSVD and ST-HOSVD

In this section, we present the procedures of truncated HOSVD (T-HOSVD) (De Lathauwer et al., 2000) and sequentially truncated HOSVD (ST-HOSVD) (Vannieuwenhoven et al., 2012). For simplicity, we present the sequentially truncated HOSVD with the truncation order from mode 1 to mode d.

Algorithm 3 Truncated High-order Singular Value Decomposition (T-HOSVD)

Input: $\mathcal{Y} \in \mathbb{R}^{p_1 \times \cdots \times p_d}$, Tucker rank $\mathbf{r} = (r_1, \ldots, r_d)$.

1: Compute $\mathbf{U}_k^0 = \mathrm{SVD}_{r_k}(\mathcal{M}_k(\mathcal{Y}))$ for $k = 1, \ldots, d$.

Output: $\widehat{\mathcal{Y}} = \mathcal{Y} \times_{k=1}^d P_{\mathbf{U}_k^0}$.

Algorithm 4 Sequentially Truncated High-order Singular Value Decomposition (ST-HOSVD)

Input: $\mathcal{Y} \in \mathbb{R}^{p_1 \times \cdots \times p_d}$, Tucker rank $\mathbf{r} = (r_1, \ldots, r_d)$.

1: Compute $\mathbf{U}_1^0 = \mathrm{SVD}_{r_1}(\mathcal{M}_1(\mathcal{Y}))$.

2: **for** $k = 2, \ldots, d$ **do**

3:     Compute $\mathbf{U}_k^0 = \mathrm{SVD}_{r_k}(\mathcal{M}_k(\mathcal{Y} \times_{l=1}^{k-1} P_{\mathbf{U}_l^0}))$.

4: **end for**

Output: $\widehat{\mathcal{Y}} = \mathcal{Y} \times_{k=1}^d P_{\mathbf{U}_k^0}$.

Appendix B. Proofs

We collect all proofs for the main results in this section. We begin by introducing a few preliminary results and then give the proof for all theorems/corollaries/lemmas in subsections.

First, by $\mathcal{L}_t$ defined in (12), we can write the least squares in (18) in the following compact way

$$(\mathcal{B}^{t+1}, \{\mathbf{D}_k^{t+1}\}_{k=1}^d) = \underset{\substack{\mathcal{B} \in \mathbb{R}^{r_1 \times \cdots \times r_d}, \\ \mathbf{D}_k \in \mathbb{R}^{(p_k - r_k) \times r_k}, k=1,\ldots,d}}{\arg\min} \left\| \mathbf{y} - \mathscr{A}\mathcal{L}_t(\mathcal{B}, \{\mathbf{D}_k\}_{k=1}^d) \right\|_2^2. \tag{26}$$

Also for $\mathcal{X} \in \mathbb{M}_{\mathbf{r}}$ and the projector $P_{T_{\mathcal{X}}}(\cdot)$ in (11), we let $P_{(T_{\mathcal{X}})_\perp}(\mathcal{Z}) := \mathcal{Z} - P_{T_{\mathcal{X}}}(\mathcal{Z})$ be the orthogonal complement of the projector $P_{T_{\mathcal{X}}}$.

Next, we introduce a tensorized view of $(\mathcal{Z} \times_{k=1}^d \mathbf{U}_k^{t\top}, \{\mathbf{U}_{k\perp}^{t\top}\mathcal{M}_k(\mathcal{Z})\mathbf{W}_k^t\}_{k=1}^d)$ generated from $\mathcal{L}_t^*(\mathcal{Z})$ (12). For simplicity, denote $\mathcal{B}_{\mathcal{Z}} = \mathcal{Z} \times_{k=1}^d \mathbf{U}_k^{t\top}$, $\mathbf{D}_{k\mathcal{Z}} = \mathbf{U}_{k\perp}^{t\top}\mathcal{M}_k(\mathcal{Z})\mathbf{W}_k^t$. By construction, it is convenient to view $(\mathcal{B}_{\mathcal{Z}}, \{\mathbf{D}_{k\mathcal{Z}}\}_{k=1}^d)$ lie in a Tucker rank $2\mathbf{r}$ tensor space in $\mathbb{R}^{p_1 \times \cdots \times p_d}$ and this fact is useful in the proof. In Figure 8, we draw a pictorial illustration to illustrate how does $\mathcal{L}_t^*(\mathcal{Z})$ look like in a special setting.

By the explanation above, throughout the proof section, we will use the notation $(\mathcal{B}^{t+1}, \{\mathbf{D}_k^{t+1}\}_{k=1}^d)$ to represent a tensor where $\mathcal{B}^{t+1}$ and $\{\mathbf{D}_k^{t+1}\}_{k=1}^d$ are located in the same places as $\mathcal{B}_{\mathcal{Z}}$ and $\{\mathbf{D}_{k\mathcal{Z}}\}_{k=1}^d$ in $\mathbb{R}^{p_1 \times \cdots \times p_d}$, and $(\mathcal{B}^{t+1}, \{\mathbf{D}_k^{t+1}\}_{k=1}^d) - \mathcal{L}_t^*(\mathcal{Z})$ denotes the difference of these two tensors.
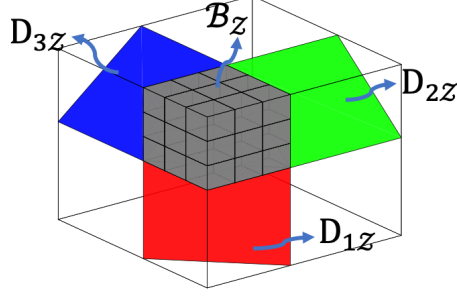
Figure 8: Illustration of $(\mathcal{B}_{\mathcal{Z}}, \{\mathbf{D}_{k\mathcal{Z}}\}_{k=1}^d)$. Here, we assume $\mathbf{U}_k^\top = [\mathbf{I}_{r_k} \ \mathbf{0}_{r_k \times (p_k - r_k)}]$, $k = 1, 2, 3$, for a better visualization. The gray core tensor represents $\mathcal{B}_{\mathcal{Z}}$ and red, green, blue blocks represent $\mathbf{D}_{1\mathcal{Z}}, \mathbf{D}_{2\mathcal{Z}}$ and $\mathbf{D}_{3\mathcal{Z}}$, respectively.

For any given linear operator $\mathcal{L}$, we use $\mathcal{R}(\mathcal{L})$ to denote its range space. The first Lemma gives the bounds on the spectrum of $\mathcal{L}_t^* \mathscr{A}^* \mathscr{A} \mathcal{L}_t$.

Lemma 6 (Bounds for Spectrum of $\mathcal{L}_t^* \mathscr{A}^* \mathscr{A} \mathcal{L}_t$) *Recall the definition of $\mathcal{L}_t$ in (12). It holds that*

$$\|\mathcal{L}_t(\mathcal{Z})\|_{\mathrm{HS}} = \|\mathcal{Z}\|_{\mathrm{HS}}, \quad \forall \, \mathcal{Z} \in \mathcal{R}(\mathcal{L}_t^*). \tag{27}$$

Suppose the linear map $\mathscr{A}$ satisfies the $2\mathbf{r}$-TRIP. Then, it holds that for any tensor $\mathcal{Z} \in \mathcal{R}(\mathcal{L}_t^*)$,

$$(1 - R_{2\mathbf{r}})\|\mathcal{Z}\|_{\mathrm{HS}} \leqslant \|\mathcal{L}_t^* \mathscr{A}^* \mathscr{A} \mathcal{L}_t(\mathcal{Z})\|_{\mathrm{HS}} \leqslant (1 + R_{2\mathbf{r}})\|\mathcal{Z}\|_{\mathrm{HS}}. \tag{28}$$

and

$$\frac{\|\mathcal{Z}\|_{\mathrm{HS}}}{1 + R_{2\mathbf{r}}} \leqslant \|(\mathcal{L}_t^* \mathscr{A}^* \mathscr{A} \mathcal{L}_t)^{-1}(\mathcal{Z})\|_{\mathrm{HS}} \leqslant \frac{\|\mathcal{Z}\|_{\mathrm{HS}}}{1 - R_{2\mathbf{r}}}. \tag{29}$$

Proof. Equation (27) can be directly verified from definitions of $\mathcal{L}_t$ and $\mathcal{L}_t^*$ in (12) and the orthogonality for each component in $\mathcal{L}_t^*$. (29) follows from (28) by the relationship of the spectrum of an operator and its inverse, so we just need to show (28).

The second claim is equivalent to say the spectrum of $\mathcal{L}_t^* \mathscr{A}^* \mathscr{A} \mathcal{L}_t$ is lower and upper bounded by $1 - R_{2\mathbf{r}}$ and $1 + R_{2\mathbf{r}}$, respectively, for $\mathcal{Z} \in \mathcal{R}(\mathcal{L}_t^*)$. Since $\mathcal{L}_t^* \mathscr{A}^* \mathscr{A} \mathcal{L}_t$ is a symmetric operator, its spectrum can be upper bounded by $\sup_{\mathcal{Z} \in \mathcal{R}(\mathcal{L}_t^*), \|\mathcal{Z}\|_{\mathrm{HS}}=1} \langle \mathcal{Z}, \mathcal{L}_t^* \mathscr{A}^* \mathscr{A} \mathcal{L}_t(\mathcal{Z}) \rangle$ and lower bounded by $\inf_{\mathcal{Z} \in \mathcal{R}(\mathcal{L}_t^*), \|\mathcal{Z}\|_{\mathrm{HS}}=1} \langle \mathcal{Z}, \mathcal{L}_t^* \mathscr{A}^* \mathscr{A} \mathcal{L}_t(\mathcal{Z}) \rangle$. Also

$$\sup_{\mathcal{Z} \in \mathcal{R}(\mathcal{L}_t^*), \|\mathcal{Z}\|_{\mathrm{HS}}=1} \langle \mathcal{Z}, \mathcal{L}_t^* \mathscr{A}^* \mathscr{A} \mathcal{L}_t(\mathcal{Z}) \rangle = \sup_{\mathcal{Z} \in \mathcal{R}(\mathcal{L}_t^*), \|\mathcal{Z}\|_{\mathrm{HS}}=1} \|\mathscr{A} \mathcal{L}_t(\mathcal{Z})\|_{\mathrm{HS}}^2 \overset{(a)}{\leqslant} 1 + R_{2\mathbf{r}}$$

$$\inf_{\mathcal{Z} \in \mathcal{R}(\mathcal{L}_t^*), \|\mathcal{Z}\|_{\mathrm{HS}}=1} \langle \mathcal{Z}, \mathcal{L}_t^* \mathscr{A}^* \mathscr{A} \mathcal{L}_t(\mathcal{Z}) \rangle = \inf_{\mathcal{Z} \in \mathcal{R}(\mathcal{L}_t^*), \|\mathcal{Z}\|_{\mathrm{HS}}=1} \|\mathscr{A} \mathcal{L}_t(\mathcal{Z})\|_{\mathrm{HS}}^2 \overset{(a)}{\geqslant} 1 - R_{2\mathbf{r}}.$$

Here (a) is by the TRIP condition for $\mathscr{A}$, $\mathcal{L}_t(\mathcal{Z})$ is at most Tucker rank $2\mathbf{r}$ and (27). ∎

By assuming TRIP for $\mathscr{A}$, Lemma 6 shows the linear operator $\mathcal{L}_t^* \mathscr{A}^* \mathscr{A} \mathcal{L}_t$ is always invertible over $\mathcal{R}(\mathcal{L}_t^*)$ (i.e. the least squares (18) has the unique solution).

Next let us take a detailed look at the error in the least squares in (18). To have a better understanding of this least squares, let us rewrite $\mathbf{y}_i$ in the following way

$$
\begin{aligned}
\mathbf{y}_i &= \langle \boldsymbol{\mathcal{A}}_i, \boldsymbol{\mathcal{X}}^* \rangle + \varepsilon_i \\
&= \langle \boldsymbol{\mathcal{A}}_i, P_{T_{\boldsymbol{\mathcal{X}}^t}} \boldsymbol{\mathcal{X}}^* \rangle + \langle \boldsymbol{\mathcal{A}}_i, P_{(T_{\boldsymbol{\mathcal{X}}^t})_\perp} \boldsymbol{\mathcal{X}}^* \rangle + \varepsilon_i \\
&= \underbrace{\langle \mathcal{L}_t^* \boldsymbol{\mathcal{A}}_i, \mathcal{L}_t^* \boldsymbol{\mathcal{X}}^* \rangle}_{(a)} + \underbrace{\langle \boldsymbol{\mathcal{A}}_i, P_{(T_{\boldsymbol{\mathcal{X}}^t})_\perp} \boldsymbol{\mathcal{X}}^* \rangle + \varepsilon_i}_{(b)} \\
&= \langle \mathcal{L}_t^* \boldsymbol{\mathcal{A}}_i, \mathcal{L}_t^* \boldsymbol{\mathcal{X}}^* \rangle + \varepsilon_i^t.
\end{aligned}
\tag{30}
$$

Here $\boldsymbol{\varepsilon}^t := \mathscr{A}(P_{(T_{\boldsymbol{\mathcal{X}}^t})_\perp} \boldsymbol{\mathcal{X}}^*) + \boldsymbol{\varepsilon}$. (30) can be viewed as the partial linear regression model we considered in performing the least squares in (18). In the first expression (a) on the right-hand side of (30), we have the covariates $\mathcal{L}_t^*(\boldsymbol{\mathcal{A}}_i)$ and (b) is the residual in the new partial linear model. Thus, we can see that the estimating target of $(\boldsymbol{\mathcal{B}}^{t+1}, \{\mathbf{D}_k^{t+1}\}_{k=1}^d)$ is $\mathcal{L}_t^* \boldsymbol{\mathcal{X}}^*$ and its estimation error is given in the following Lemma.

Lemma 7 (Least Squares Error in RGN) *Recall the definition of $\boldsymbol{\varepsilon}^t = \mathscr{A}(P_{(T_{\boldsymbol{\mathcal{X}}^t})_\perp} \boldsymbol{\mathcal{X}}^*) + \boldsymbol{\varepsilon}$ from (30). If the operator $\mathcal{L}_t^* \mathscr{A}^* \mathscr{A} \mathcal{L}_t$ is invertible over $\mathcal{R}(\mathcal{L}_t^*)$, then $(\boldsymbol{\mathcal{B}}^{t+1}, \{\mathbf{D}_k^{t+1}\}_{k=1}^d)$ in (18) satisfy*

$$
(\boldsymbol{\mathcal{B}}^{t+1}, \{\mathbf{D}_k^{t+1}\}_{k=1}^d) - \mathcal{L}_t^* \boldsymbol{\mathcal{X}}^* = (\mathcal{L}_t^* \mathscr{A}^* \mathscr{A} \mathcal{L}_t)^{-1} \mathcal{L}_t^* \mathscr{A}^* \boldsymbol{\varepsilon}^t,
\tag{31}
$$

and

$$
\|(\boldsymbol{\mathcal{B}}^{t+1}, \{\mathbf{D}_k^{t+1}\}_{k=1}^d) - \mathcal{L}_t^* \boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}} = \|(\mathcal{L}_t^* \mathscr{A}^* \mathscr{A} \mathcal{L}_t)^{-1} \mathcal{L}_t^* \mathscr{A}^* \boldsymbol{\varepsilon}^t\|_{\mathrm{HS}}.
\tag{32}
$$

Proof. First by the decomposition of (30), we have

$$
\mathbf{y} = \mathscr{A} \mathcal{L}_t \mathcal{L}_t^*(\boldsymbol{\mathcal{X}}^*) + \boldsymbol{\varepsilon}^t.
\tag{33}
$$

In view of (26), if the operator $\mathcal{L}_t^* \mathscr{A}^* \mathscr{A} \mathcal{L}_t$ is invertible, the output of least squares in (18) satisfies

$$
(\boldsymbol{\mathcal{B}}^{t+1}, \{\mathbf{D}_k^{t+1}\}_{k=1}^d) = (\mathcal{L}_t^* \mathscr{A}^* \mathscr{A} \mathcal{L}_t)^{-1} \mathcal{L}_t^* \mathscr{A}^* \mathbf{y} = \mathcal{L}_t^*(\boldsymbol{\mathcal{X}}^*) + (\mathcal{L}_t^* \mathscr{A}^* \mathscr{A} \mathcal{L}_t)^{-1} \mathcal{L}_t^* \mathscr{A}^* \boldsymbol{\varepsilon}^t,
$$

where the second equality is due to (33). This finishes the proof. ∎

Next, we begin the proof for the main results in the paper one by one.

B.1 Proof of Lemma 1

To show the tangent space representation in Lemma 1 is equivalent to the tangent space representation in literature (9), we just need to find a one-to-one correspondence between $\bar{\mathbf{D}}_k$ and $\mathbf{D}_k$ in two representations as the $\boldsymbol{\mathcal{B}}$ is the same in both representations. Given $\bar{\mathbf{D}}_k \in \mathbb{R}^{p_k \times r_k}$ and $\bar{\mathbf{D}}_k^\top \mathbf{U}_k = \mathbf{0}$, we have $\bar{\mathbf{D}}_k = \mathbf{U}_{k\perp} \mathbf{M}$ for some $\mathbf{M} \in \mathbb{R}^{(p_k - r_k) \times r_k}$. So

$$
\mathcal{M}_k(\boldsymbol{\mathcal{S}} \times_k \bar{\mathbf{D}}_k \times_{i \neq k} \mathbf{U}_i) \overset{(8)}{=} \mathbf{U}_{k\perp} \mathbf{M} \mathcal{M}_k(\boldsymbol{\mathcal{S}}) (\otimes_{i=d, i \neq k}^1 \mathbf{U}_i)^\top \overset{(a)}{=} \mathbf{U}_{k\perp} \mathbf{M} \mathbf{R}^\top \mathbf{W}_k^\top.
$$

Here (a) is because $\mathbf{V}_k = \mathrm{QR}(\mathcal{M}_k(\boldsymbol{\mathcal{S}})^\top)$ and $\mathcal{M}_k(\boldsymbol{\mathcal{S}}) = \mathbf{V}_k \mathbf{R}$ for some invertible $\mathbf{R} \in \mathbb{R}^{r_k \times r_k}$ matrix. Thus, we can see that $\mathbf{D}_k = \mathbf{M} \mathbf{R}^\top$ in the new representation. Similarly, given $\mathbf{D}_k \in \mathbb{R}^{(p_k - r_k) \times r_k}$,

$$
\mathcal{T}_k(\mathbf{U}_{k\perp} \mathbf{D}_k \mathbf{W}_k^\top) = \mathcal{T}_k(\mathbf{U}_{k\perp} \mathbf{D}_k (\mathbf{R}^\top)^{-1} \mathbf{R}^\top \mathbf{V}_k^\top (\otimes_{i=d, i \neq k}^1 \mathbf{U}_i)^\top) \overset{(8)}{=} \boldsymbol{\mathcal{S}} \times_k \mathbf{U}_{k\perp} \mathbf{D}_k (\mathbf{R}^\top)^{-1} \times_{i \neq k} \mathbf{U}_i.
$$

So $\bar{\mathbf{D}}_k = \mathbf{U}_{k\perp} \mathbf{D}_k (\mathbf{R}^\top)^{-1}$ and this finishes the proof. ∎

B.2 Proof of Lemma 2

Suppose $\mathbf{\mathcal{Z}} \in T_{\mathbf{\mathcal{X}}}\mathbb{M}_{\mathbf{r}}$ and it has representation $\mathbf{\mathcal{B}} \times_{k=1}^{d} \mathbf{U}_k + \sum_{k=1}^{d} \mathcal{T}_k(\mathbf{U}_{k\perp}\mathbf{D}_k\mathbf{W}_k^{\top})$ for some $\mathbf{\mathcal{B}}$ and $\{\mathbf{D}_k\}_{k=1}^{d}$. To prove the result, it is enough to show $\mathrm{rank}(\mathcal{M}_k(\mathbf{\mathcal{Z}})) \leqslant 2r_k$ for $k = 1, \ldots, d$. Let us show this is true for $k = 1$ and the proof for other modes is similar.

Let $\mathbf{\mathcal{V}}_k$ be the tensorzied $\mathbf{V}_k^{\top}$ such that $\mathcal{M}_k(\mathbf{\mathcal{V}}_k) = \mathbf{V}_k^{\top}$. Recall the definition of $\mathbf{W}_k$ in (10), we have $\mathcal{T}_k(\mathbf{U}_{k\perp}\mathbf{D}_k\mathbf{W}_k^{\top}) = \mathbf{\mathcal{V}}_k \times_{l\neq k} \mathbf{U}_l \times_k \mathbf{U}_{k\perp}\mathbf{D}_k$. So

$$\mathcal{M}_1(\mathbf{\mathcal{Z}}) = \mathbf{U}_1 \left(\mathcal{M}_1(\mathbf{\mathcal{B}} \times_{k\neq 1} \mathbf{U}_k) + \sum_{k=2}^{d} \mathcal{M}_1(\mathbf{\mathcal{V}}_k \times \mathbf{U}_{k\perp}\mathbf{D}_k \times_{l\neq 1,k} \mathbf{U}_l) \right) + \mathbf{U}_{1\perp}\mathbf{D}_1\mathcal{M}_1(\mathbf{\mathcal{V}}_1 \times_{k\neq 1} \mathbf{U}_k). \tag{34}$$

Since $\mathbf{\mathcal{B}} \in \mathbb{R}^{r_1 \times \cdots \times r_d}, \mathbf{D}_1 \in \mathbb{R}^{(p_1-r_1)\times r_1}$, each matrix on the right hand side of (34) is of rank at most r_1. Thus $\mathcal{M}_1(\mathbf{\mathcal{Z}})$ is at most rank $2r_1$. This finishes the proof. ∎

B.3 Proof of Proposition 1

In view of the Riemannian Gauss-Newton equation in (13) and the Riemannian gradient in Lemma 3, to prove the claim, we only need to show

$$P_{T_{\mathbf{\mathcal{X}}^t}}(\mathscr{A}^*(\mathscr{A}(\mathbf{\mathcal{Z}}^{t+1}) - \mathbf{y})) = 0. \tag{35}$$

Here we replace η^{RNG} by $\mathbf{\mathcal{Z}}^{t+1} - \mathbf{\mathcal{X}}^t$.

From the optimality condition of the least squares problem (14), we know that the least squares solution $\mathbf{\mathcal{Z}}^{t+1}$ satisfies

$$P_{T_{\mathbf{\mathcal{X}}^t}}\mathscr{A}^* \left(\mathscr{A}P_{T_{\mathbf{\mathcal{X}}^t}}(\mathbf{\mathcal{Z}}^{t+1}) - \mathbf{y} \right) = 0. \tag{36}$$

Since $\mathbf{\mathcal{Z}}^{t+1}$ lies in $T_{\mathbf{\mathcal{X}}^t}\mathbb{M}_{\mathbf{r}}$, $P_{T_{\mathbf{\mathcal{X}}^t}}(\mathbf{\mathcal{Z}}^{t+1}) = \mathbf{\mathcal{Z}}^{t+1}$. Hence, (36) implies (35). ∎

B.4 Proof of Theorem 1

In this section, we prove our main theorem. This section is divided into two subsections. In the first subsection, we introduce two key Lemmas in proving the results. In the second subsection, we prove the main results.

B.4.1 Key Lemmas

The first Lemma gives an upper bound for the distance of $(\mathbf{\mathcal{B}}^{t+1}, \{\mathbf{D}_k^{t+1}\}_{k=1}^{d})$ to their target $\mathcal{L}_t^* \mathbf{\mathcal{X}}^*$ in (32).

Lemma 8 (Upper Bound for the Least Squares Estimation Error) *Let* $\varepsilon^t = \mathscr{A}\left(P_{(T_{\mathbf{\mathcal{X}}^t})\perp}\mathbf{\mathcal{X}}^*\right) + \varepsilon$. *Suppose that* $\mathscr{A}$ *satisfies the* $3\mathbf{r}$*-TRIP. Then at tth iteration of RGN, the approximation error* (32) *has the following upper bound:*

$$\|(\mathcal{L}_t^*\mathscr{A}^*\mathscr{A}\mathcal{L}_t)^{-1}\mathcal{L}_t^*\mathscr{A}^*\varepsilon^t\|_{\mathrm{HS}} \leqslant \frac{R_{3\mathbf{r}}\|P_{(T_{\mathbf{\mathcal{X}}^t})\perp}\mathbf{\mathcal{X}}^*\|_{\mathrm{HS}}}{1 - R_{2\mathbf{r}}} + \frac{\|(\mathscr{A}^*(\varepsilon))_{\max(2\mathbf{r})}\|_{\mathrm{HS}}}{1 - R_{2\mathbf{r}}}. \tag{37}$$

Proof. Since $\mathscr{A}$ satisfies 3r-TRIP, $R_{2\mathbf{r}} \leqslant R_{3\mathbf{r}} < 1$. Then, Lemma 6's assumption holds and $\mathcal{L}_t^* \mathscr{A}^* \mathscr{A} \mathcal{L}_t$ is invertible over $\mathcal{R}(\mathcal{L}_t^*)$.

$$
\begin{aligned}
\|(\mathcal{L}_t^* \mathscr{A}^* \mathscr{A} \mathcal{L}_t)^{-1} \mathcal{L}_t^* \mathscr{A}^* \boldsymbol{\varepsilon}^t\|_{\mathrm{HS}} &\overset{Lemma\,6}{\leqslant} \frac{1}{1 - R_{2\mathbf{r}}} \|\mathcal{L}_t^* \mathscr{A}^* \boldsymbol{\varepsilon}^t\|_{\mathrm{HS}} \\
&\overset{(a)}{=} \frac{1}{1 - R_{2\mathbf{r}}} \|\mathcal{L}_t^* \mathscr{A}^* (\mathscr{A}(P_{(T_{\boldsymbol{\mathcal{X}}^t})_\perp} \boldsymbol{\mathcal{X}}^*) + \boldsymbol{\varepsilon})\|_{\mathrm{HS}} \\
&\overset{(b)}{\leqslant} \frac{1}{1 - R_{2\mathbf{r}}} \left(\|\mathcal{L}_t^* \mathscr{A}^* \mathscr{A}(P_{(T_{\boldsymbol{\mathcal{X}}^t})_\perp} \boldsymbol{\mathcal{X}}^*)\|_{\mathrm{HS}} + \|(\mathscr{A}^*(\boldsymbol{\varepsilon}))_{\max(2\mathbf{r})}\|_{\mathrm{HS}} \right) \\
&\overset{(c)}{\leqslant} \frac{R_{3\mathbf{r}} \|P_{(T_{\boldsymbol{\mathcal{X}}^t})_\perp} \boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}}}{1 - R_{2\mathbf{r}}} + \frac{\|(\mathscr{A}^*(\boldsymbol{\varepsilon}))_{\max(2\mathbf{r})}\|_{\mathrm{HS}}}{1 - R_{2\mathbf{r}}},
\end{aligned}
$$

here (a) is by the definition of $\boldsymbol{\varepsilon}^t$; (b) is by triangle inequality and $\mathcal{L}_t^*(\mathscr{A}^*(\boldsymbol{\varepsilon}))$ is of at most Tucker rank $2\mathbf{r}$ as we discussed before; (c) is because

$$
\begin{aligned}
\|\mathcal{L}_t^* \mathscr{A}^* \mathscr{A}(P_{(T_{\boldsymbol{\mathcal{X}}^t})_\perp} \boldsymbol{\mathcal{X}}^*)\|_{\mathrm{HS}} &= \sup_{\boldsymbol{\mathcal{Z}} : \|\boldsymbol{\mathcal{Z}}\|_{\mathrm{HS}} \leqslant 1} \langle \mathcal{L}_t^* \mathscr{A}^* \mathscr{A}(P_{(T_{\boldsymbol{\mathcal{X}}^t})_\perp} \boldsymbol{\mathcal{X}}^*), \boldsymbol{\mathcal{Z}} \rangle \\
&= \sup_{\boldsymbol{\mathcal{Z}} : \|\boldsymbol{\mathcal{Z}}\|_{\mathrm{HS}} \leqslant 1} \langle \mathscr{A}(P_{(T_{\boldsymbol{\mathcal{X}}^t})_\perp} \boldsymbol{\mathcal{X}}^*), \mathscr{A} \mathcal{L}_t(\boldsymbol{\mathcal{Z}}) \rangle \\
&\overset{(a)}{\leqslant} \sup_{\boldsymbol{\mathcal{Z}} : \|\boldsymbol{\mathcal{Z}}\|_{\mathrm{HS}} \leqslant 1} R_{3\mathbf{r}} \|P_{(T_{\boldsymbol{\mathcal{X}}^t})_\perp} \boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}} \|\mathcal{L}_t(\boldsymbol{\mathcal{Z}})\|_{\mathrm{HS}} \\
&\overset{(b)}{\leqslant} R_{3\mathbf{r}} \|P_{(T_{\boldsymbol{\mathcal{X}}^t})_\perp} \boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}}.
\end{aligned}
$$

Here (a) is due to Lemma 10, $\langle P_{(T_{\boldsymbol{\mathcal{X}}^t})_\perp} \boldsymbol{\mathcal{X}}^*, \mathcal{L}_t(\boldsymbol{\mathcal{Z}}) \rangle = 0$, $P_{(T_{\boldsymbol{\mathcal{X}}^t})_\perp} \boldsymbol{\mathcal{X}}^*$ and $\mathcal{L}_t(\boldsymbol{\mathcal{Z}})$ are of Tucker rank at most $\mathbf{r}$ and $2\mathbf{r}$, respectively; (b) is because $\|\mathcal{L}_t(\boldsymbol{\mathcal{Z}})\|_{\mathrm{HS}} \leqslant \|\boldsymbol{\mathcal{Z}}\|_{\mathrm{HS}} \leqslant 1$. $\blacksquare$

The following Lemma plays a key role in showing the quadratic convergence of RGN.

Lemma 9 (Projection of $\boldsymbol{\mathcal{X}}^*$ on $P_{(T_{\boldsymbol{\mathcal{X}}^t})_\perp}$) *For any two order-d Tucker rank* $\mathbf{r} := (r_1, \ldots, r_d)$ *tensors* $\boldsymbol{\mathcal{X}}^*, \boldsymbol{\mathcal{X}}^t \in \mathbb{R}^{p_1 \times \cdots \times p_d}$, *we have*

$$
\|P_{(T_{\boldsymbol{\mathcal{X}}^t})_\perp} \boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}} \leqslant \frac{d \|\boldsymbol{\mathcal{X}}^t - \boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}}^2}{\underline{\lambda}},
$$

where $\underline{\lambda} := \min_{k=1,\ldots,d} \sigma_{r_k}(\mathcal{M}_k(\boldsymbol{\mathcal{X}}^*))$.

Proof. Suppose $\boldsymbol{\mathcal{X}}^t$ and $\boldsymbol{\mathcal{X}}^*$ have Tucker rank $\mathbf{r}$ decomposition $[\![\boldsymbol{\mathcal{S}}^t; \mathbf{U}_1^t, \ldots, \mathbf{U}_d^t]\!]$ and $[\![\boldsymbol{\mathcal{S}}; \mathbf{U}_1, \ldots, \mathbf{U}_d]\!]$, respectively. Recall

$$
\mathbf{W}_k := (\mathbf{U}_d \otimes \cdots \otimes \mathbf{U}_{k+1} \otimes \mathbf{U}_{k-1} \otimes \ldots \otimes \mathbf{U}_1) \mathbf{V}_k \in \mathbb{O}_{p_{-k}, r_k}
$$

in (10), where $\mathbf{V}_k = \mathrm{QR}(\mathcal{M}_k(\boldsymbol{\mathcal{S}})^\top)$. Similarly we have $\mathbf{W}_k^t$ for $\boldsymbol{\mathcal{X}}^t$. For $\boldsymbol{\mathcal{X}}^*$, it can be decomposed in the following way

$$
\begin{aligned}
\boldsymbol{\mathcal{X}}^* =& \boldsymbol{\mathcal{X}}^* \times_1 P_{\mathbf{U}_{1\perp}^t} + \boldsymbol{\mathcal{X}}^* \times_1 P_{\mathbf{U}_1^t} \times_2 P_{\mathbf{U}_{2\perp}^t} + \cdots + \boldsymbol{\mathcal{X}}^* \times_{l=1}^{k-1} P_{\mathbf{U}_l^t} \times_k P_{\mathbf{U}_{k\perp}^t} + \cdots + \boldsymbol{\mathcal{X}}^* \times_{l=1}^{d} P_{\mathbf{U}_l^t} \\
=& \sum_{k=1}^{d} \boldsymbol{\mathcal{X}}^* \times_{l=1}^{k-1} P_{\mathbf{U}_l^t} \times_k P_{\mathbf{U}_{k\perp}^t} + \boldsymbol{\mathcal{X}}^* \times_{l=1}^{d} P_{\mathbf{U}_l^t}
\end{aligned}
$$

$$(38)$$

Then

$$
\begin{aligned}
P_{(T_{\boldsymbol{\mathcal{X}}^t})_\perp}\boldsymbol{\mathcal{X}}^* =&\boldsymbol{\mathcal{X}}^* - P_{T_{\boldsymbol{\mathcal{X}}^t}}\boldsymbol{\mathcal{X}}^* \\
&\overset{(11)}{=}\boldsymbol{\mathcal{X}}^* - (\boldsymbol{\mathcal{X}}^* \times_{k=1}^d P_{\mathbf{U}_k^t} + \sum_{k=1}^d \mathcal{T}_k(P_{\mathbf{U}_{k\perp}^t}\mathcal{M}_k(\boldsymbol{\mathcal{X}}^*)P_{\mathbf{W}_k^t})) \\
&\overset{(38)}{=}\sum_{k=1}^d \left(\boldsymbol{\mathcal{X}}^* \times_{l=1}^{k-1} P_{\mathbf{U}_l^t} \times_k P_{\mathbf{U}_{k\perp}^t} - \mathcal{T}_k(P_{\mathbf{U}_{k\perp}^t}\mathcal{M}_k(\boldsymbol{\mathcal{X}}^*)P_{\mathbf{W}_k^t})\right) \\
&\overset{(8)}{=}\sum_{k=1}^d \left(\mathcal{T}_k\left(P_{\mathbf{U}_{k\perp}^t}\mathcal{M}_k(\boldsymbol{\mathcal{X}}^*)(\otimes_{l=d}^{k+1}\mathbf{I}_{p_l}\otimes_{l=k-1}^1 P_{\mathbf{U}_l^t} - P_{\mathbf{W}_k^t})\right)\right) \\
&\overset{(a)}{=}\sum_{k=1}^d \left(\mathcal{T}_k\left((P_{\mathbf{U}_k} - P_{\mathbf{U}_k^t})\mathcal{M}_k(\boldsymbol{\mathcal{X}}^*)(\otimes_{l=d}^{k+1}\mathbf{I}_{p_l}\otimes_{l=k-1}^1 P_{\mathbf{U}_l^t} - P_{\mathbf{W}_k^t})\right)\right) \\
&\overset{(b)}{=}\sum_{k=1}^d \left(\mathcal{T}_k\left((P_{\mathbf{U}_k} - P_{\mathbf{U}_k^t})\mathcal{M}_k(\boldsymbol{\mathcal{X}}^* - \boldsymbol{\mathcal{X}}^t)(\otimes_{l=d}^{k+1}\mathbf{I}_{p_l}\otimes_{l=k-1}^1 P_{\mathbf{U}_l^t} - P_{\mathbf{W}_k^t})\right)\right),
\end{aligned}
\tag{39}
$$

here (a) is because the $\mathbf{U}_k$ spans the column space of $\mathcal{M}_k(\boldsymbol{\mathcal{X}}^*)$, (b) is because $\mathcal{M}_k(\boldsymbol{\mathcal{X}}^t)(\otimes_{l=d}^{k+1}\mathbf{I}_{p_l}\otimes_{l=k-1}^1 P_{\mathbf{U}_l^t} - P_{\mathbf{W}_k^t}) = 0$.

It is easy to check $\otimes_{l=d}^{k+1}\mathbf{I}_{p_l}\otimes_{l=k-1}^1 P_{\mathbf{U}_l^t} - P_{\mathbf{W}_k^t}$ is a projection matrix. So from (39), we have

$$
\begin{aligned}
\|P_{(T_{\boldsymbol{\mathcal{X}}^t})_\perp}\boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}} &\leqslant \sum_{k=1}^d \|\mathcal{T}_k\left((P_{\mathbf{U}_k} - P_{\mathbf{U}_k^t})\mathcal{M}_k(\boldsymbol{\mathcal{X}}^* - \boldsymbol{\mathcal{X}}^t)(\otimes_{l=d}^{k+1}\mathbf{I}_{p_l}\otimes_{l=k-1}^1 P_{\mathbf{U}_l^t} - P_{\mathbf{W}_k^t})\right)\|_{\mathrm{HS}} \\
&\leqslant \sum_{k=1}^d \|(\boldsymbol{\mathcal{X}}^* - \boldsymbol{\mathcal{X}}^t) \times_k (P_{\mathbf{U}_k} - P_{\mathbf{U}_k^t})\|_{\mathrm{HS}} \\
&\leqslant d\|\boldsymbol{\mathcal{X}}^* - \boldsymbol{\mathcal{X}}^t\|_{\mathrm{HS}} \max_{k=1,\dots,d}\|P_{\mathbf{U}_k} - P_{\mathbf{U}_k^t}\| \\
&\overset{(a)}{\leqslant} \frac{d\|\boldsymbol{\mathcal{X}}^t - \boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}}^2}{\underline{\lambda}},
\end{aligned}
$$

here (a) is due to the matrix subspace perturbation bound $\|P_{\mathbf{U}_k} - P_{\mathbf{U}_k^t}\| \leqslant \frac{\|\mathcal{M}_k(\boldsymbol{\mathcal{X}}^*)-\mathcal{M}_k(\boldsymbol{\mathcal{X}}^t)\|}{\sigma_{r_k}(\mathcal{M}_k(\boldsymbol{\mathcal{X}}^*))} \leqslant \frac{\|\mathcal{M}_k(\boldsymbol{\mathcal{X}}^*)-\mathcal{M}_k(\boldsymbol{\mathcal{X}}^t)\|_F}{\sigma_{r_k}(\mathcal{M}_k(\boldsymbol{\mathcal{X}}^*))} = \frac{\|\boldsymbol{\mathcal{X}}^*-\boldsymbol{\mathcal{X}}^t\|_{\mathrm{HS}}}{\sigma_{r_k}(\mathcal{M}_k(\boldsymbol{\mathcal{X}}^*))}$ (for example see Lemma 4.2 of Wei et al. (2016)). This finishes the proof of this Lemma. ∎

We note a similar result to Lemma 9 appears in Lemma 5.2 of Cai et al. (2020) and here we have exponential improvement on the dependence of d.

B.4.2 Proof of Theorem 1

Now we prove the main results. First, notice the convergence result in the noiseless setting follows easily from the noisy setting by setting $\boldsymbol{\varepsilon} = 0$. Suppose $\mathcal{H}_{\mathbf{r}}$ satisfies the quasi-projection property with approximation constant $\delta(d)$. For notation simplicity, let us denote

$$\boldsymbol{\mathcal{X}}^{t+0.5} := \boldsymbol{\mathcal{Z}}^{t+1} = \mathcal{L}_t(\boldsymbol{\mathcal{B}}^{t+1}, \{\mathbf{D}_k^{t+1}\}_{k=1}^d) = \boldsymbol{\mathcal{B}}^{t+1} \times_{k=1}^d \mathbf{U}_k^t + \sum_{k=1}^d \mathcal{T}_k(\mathbf{U}_{k\perp}^t \mathbf{D}_k^{t+1} \mathbf{W}_k^{t\top}).$$

$$
\begin{aligned}
\|\boldsymbol{\mathcal{X}}^{t+1} - \boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}} =& \|\mathcal{H}_{\mathbf{r}}(\boldsymbol{\mathcal{X}}^{t+0.5}) - \boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}} \\
\leqslant& \|\mathcal{H}_{\mathbf{r}}(\boldsymbol{\mathcal{X}}^{t+0.5}) - \boldsymbol{\mathcal{X}}^{t+0.5}\|_{\mathrm{HS}} + \|\boldsymbol{\mathcal{X}}^{t+0.5} - \boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}} \\
\overset{(a)}{\leqslant}& \delta(d)\|P_{\mathbb{M}_{\mathbf{r}}}(\boldsymbol{\mathcal{X}}^{t+0.5}) - \boldsymbol{\mathcal{X}}^{t+0.5}\|_{\mathrm{HS}} + \|\boldsymbol{\mathcal{X}}^{t+0.5} - \boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}} \\
\overset{(b)}{\leqslant}& (\delta(d)+1)\|\boldsymbol{\mathcal{X}}^{t+0.5} - \boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}} \\
=& (\delta(d)+1)\|\mathcal{L}_t(\boldsymbol{\mathcal{B}}^{t+1}, \{\mathbf{D}_k^{t+1}\}_{k=1}^d) - P_{T_{\boldsymbol{\mathcal{X}}^t}}\boldsymbol{\mathcal{X}}^* - P_{(T_{\boldsymbol{\mathcal{X}}^t})_\perp}\boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}} \\
\overset{(c)}{\leqslant}& (\delta(d)+1)\left(\|\mathcal{L}_t(\boldsymbol{\mathcal{B}}^{t+1}, \{\mathbf{D}_k^{t+1}\}_{k=1}^d) - P_{T_{\boldsymbol{\mathcal{X}}^t}}\boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}} + \|P_{(T_{\boldsymbol{\mathcal{X}}^t})_\perp}\boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}}\right) \\
\overset{Lemma\,6}{=}& (\delta(d)+1)\left(\|(\boldsymbol{\mathcal{B}}^{t+1}, \{\mathbf{D}_k^{t+1}\}_{k=1}^d) - \mathcal{L}_t^*(\boldsymbol{\mathcal{X}}^*)\|_{\mathrm{HS}} + \|P_{(T_{\boldsymbol{\mathcal{X}}^t})_\perp}\boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}}\right)
\end{aligned}
\tag{40}
$$

here (a) is by the quasi-projection property of $\mathcal{H}_{\mathbf{r}}$; (b) is by the projection property of $P_{\mathbb{M}_{\mathbf{r}}}(\cdot)$; (c) is by triangle inequality.

Notice that by the $3\mathbf{r}$-TRIP assumption of $\mathscr{A}$, the assumptions in Lemma 7 and 8 are satisfied. By Lemma 7, 8 and 9, the right hand side of (40) can be bounded as follows

$$
\begin{aligned}
\|\boldsymbol{\mathcal{X}}^{t+1} - \boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}} \overset{Lemma\,7,8}{\leqslant}& (\delta(d)+1)\left((1 + \frac{R_{3\mathbf{r}}}{1 - R_{2\mathbf{r}}})\|P_{(T_{\boldsymbol{\mathcal{X}}^t})_\perp}\boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}} + \frac{\|(\mathscr{A}^*(\boldsymbol{\varepsilon}))_{\max(2\mathbf{r})}\|_{\mathrm{HS}}}{1 - R_{2\mathbf{r}}}\right) \\
\overset{Lemma\,9}{\leqslant}& d(\delta(d)+1)(\frac{R_{3\mathbf{r}}}{1 - R_{2\mathbf{r}}} + 1)\frac{\|\boldsymbol{\mathcal{X}}^t - \boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}}^2}{\underline{\lambda}} + \frac{\delta(d)+1}{1 - R_{2\mathbf{r}}}\|(\mathscr{A}^*(\boldsymbol{\varepsilon}))_{\max(2\mathbf{r})}\|_{\mathrm{HS}}.
\end{aligned}
$$

Finally, the convergence is guaranteed under the initialization condition. By taking $\delta(d) = \sqrt{d}$, we finish the proof of this Theorem. ∎

B.5 Proof of Corollary 1

Recall from Theorem 1, we have

$$
\|\boldsymbol{\mathcal{X}}^{t+1} - \boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}} \leqslant \underbrace{d(\delta(d)+1)(\frac{R_{3\mathbf{r}}}{1 - R_{2\mathbf{r}}} + 1)\frac{\|\boldsymbol{\mathcal{X}}^t - \boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}}^2}{\underline{\lambda}}}_{(A1)} + \underbrace{\frac{\delta(d)+1}{1 - R_{2\mathbf{r}}}\|(\mathscr{A}^*(\boldsymbol{\varepsilon}))_{\max(2\mathbf{r})}\|_{\mathrm{HS}}}_{(A2)}.
$$

Notice, in the first phase, (A1) dominates (A2) and in the second phase (A2) dominates (A1) and the two-phase convergence results follow.

By induction, it is easy to show under the initialization condition, in the first phase we have

$$\|\boldsymbol{\mathcal{X}}^t - \boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}} \leqslant 2^{-2^t}\|\boldsymbol{\mathcal{X}}^0 - \boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}}.$$

When $t \geqslant T_{\max}$ indicated in the Theorem, the algorithm enters the second phase and $\|\boldsymbol{\mathcal{X}}^t - \boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}} \leqslant \frac{2(\delta(d)+1)}{1-R_{2\mathbf{r}}}\|(\mathscr{A}^*(\boldsymbol{\varepsilon}))_{\max(2\mathbf{r})}\|_{\mathrm{HS}}$. Combining phases 1 and 2, we have

$$\|\boldsymbol{\mathcal{X}}^t - \boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}} \leqslant 2^{-2^t}\|\boldsymbol{\mathcal{X}}^0 - \boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}} + \frac{2(\delta(d)+1)}{1 - R_{2\mathbf{r}}}\|(\mathscr{A}^*(\boldsymbol{\varepsilon}))_{\max(2\mathbf{r})}\|_{\mathrm{HS}}$$

for all t. By taking $\delta(d) = \sqrt{d}$, we finish the proof. ∎

B.6 Proof of Theorem 2

The proof is done by construction. We consider a very special setting where $\mathscr{A}(\boldsymbol{\mathcal{Z}}) = \text{vec}(\boldsymbol{\mathcal{Z}})$, and here $\text{vec}(\boldsymbol{\mathcal{Z}})$ denotes the vectorization of $\boldsymbol{\mathcal{Z}}$. This can be viewed as the tensor decomposition setting and we can tensorize model (1) and get $\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{X}}^* + \boldsymbol{\mathcal{E}}$. It is easy to verify $\mathscr{A}$ satisfies the TRIP condition and here $\xi = \|\boldsymbol{\mathcal{E}}_{\max(2\mathbf{r})}\|_{\text{HS}}$. Let us denote $r = \min_{k=1,\dots,d} r_k$ and $\boldsymbol{\mathcal{I}}_r \in \mathbb{R}^{r \times r \times \dots \times r}$ as the order-d identity tensor with entries $(i, i, \dots, i)$ to be 1 and others are 0. We construct

$$\boldsymbol{\mathcal{E}}_1 = \frac{\xi}{\sqrt{r}} \boldsymbol{\mathcal{I}}_r \times_1 \begin{pmatrix} \mathbf{0}_{r \times r} \\ \mathbf{I}_r \\ \mathbf{0}_{(p_1 - 2r) \times r} \end{pmatrix} \times \cdots \times_d \begin{pmatrix} \mathbf{0}_{r \times r} \\ \mathbf{I}_r \\ \mathbf{0}_{(p_d - 2r) \times r} \end{pmatrix},$$

where $\mathbf{0}_{m \times n}$ denotes a $m \times n$ matrix with all entries to be 0.

It is easy to check that $\|(\boldsymbol{\mathcal{E}}_1)_{\max(2\mathbf{r})}\|_{\text{HS}} = \xi$. Similarly, we construct

$$\boldsymbol{\mathcal{X}}_1 = \frac{\xi}{\sqrt{r}} \boldsymbol{\mathcal{I}}_r \times_1 \begin{pmatrix} \mathbf{I}_r \\ \mathbf{0}_{r \times r} \\ \mathbf{0}_{(p_1 - 2r) \times r} \end{pmatrix} \times \cdots \times_d \begin{pmatrix} \mathbf{I}_r \\ \mathbf{0}_{r \times r} \\ \mathbf{0}_{(p_d - 2r) \times r} \end{pmatrix}.$$

Also we let $\boldsymbol{\mathcal{E}}_2 = \boldsymbol{\mathcal{X}}_1$ and $\boldsymbol{\mathcal{X}}_2 = \boldsymbol{\mathcal{E}}_1$, and it is easy to check $(\mathscr{A}, \text{vec}(\boldsymbol{\mathcal{X}}_1), \text{vec}(\boldsymbol{\mathcal{E}}_1)) \in \mathcal{F}_{\mathbf{r}}(\xi)$ and $(\mathscr{A}, \text{vec}(\boldsymbol{\mathcal{X}}_2), \text{vec}(\boldsymbol{\mathcal{E}}_2)) \in \mathcal{F}_{\mathbf{r}}(\xi)$. At the same time, we have $\boldsymbol{\mathcal{E}}_1 + \boldsymbol{\mathcal{X}}_1 = \boldsymbol{\mathcal{X}}_2 + \boldsymbol{\mathcal{E}}_2$. Thus

$$
\begin{aligned}
\inf_{\widehat{\boldsymbol{\mathcal{X}}}} \sup_{(\widetilde{\mathscr{A}}, \widetilde{\boldsymbol{\mathcal{X}}}, \widetilde{\varepsilon}) \in \mathcal{F}_{\mathbf{r}}(\xi)} \|\widehat{\boldsymbol{\mathcal{X}}} - \widetilde{\boldsymbol{\mathcal{X}}}\|_{\text{HS}} &\geq \inf_{\widehat{\boldsymbol{\mathcal{X}}}} \max \left\{ \|\widehat{\boldsymbol{\mathcal{X}}} - \boldsymbol{\mathcal{X}}_1\|_{\text{HS}}, \|\widehat{\boldsymbol{\mathcal{X}}} - \boldsymbol{\mathcal{X}}_2\|_{\text{HS}} \right\} \\
&\geq \frac{1}{2} \left(\|\widehat{\boldsymbol{\mathcal{X}}} - \boldsymbol{\mathcal{X}}_1\|_{\text{HS}} + \|\widehat{\boldsymbol{\mathcal{X}}} - \boldsymbol{\mathcal{X}}_2\|_{\text{HS}} \right) \\
&\geq \frac{1}{2} \|\boldsymbol{\mathcal{X}}_1 - \boldsymbol{\mathcal{X}}_2\|_{\text{HS}} = \frac{\sqrt{2}}{2} \xi.
\end{aligned}
$$

∎

B.7 Proof of Lemma 5

Note that in the tensor decomposition model, the map $\mathscr{A}$ can be viewed as an identity map, and we have the tensorized model $\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{X}}^* + \boldsymbol{\mathcal{E}}$. The objective function of the least squares in (18) can be written in the following way

$$
\begin{aligned}
&\|\boldsymbol{\mathcal{Y}} - \boldsymbol{\mathcal{B}} \times_{k=1}^d \mathbf{U}_k^t - \sum_{k=1}^d \mathcal{T}_k(\mathbf{U}_{k\perp}^t \mathbf{D}_k \mathbf{W}_k^{t\top})\|_{\text{HS}}^2 \\
=&\|\boldsymbol{\mathcal{Y}} - \mathcal{L}_t(\boldsymbol{\mathcal{B}}, \{\mathbf{D}_k\}_{k=1}^d)\|_{\text{HS}}^2 \\
\overset{(a)}{=}&\|P_{T_{\boldsymbol{\mathcal{X}}^t}} \boldsymbol{\mathcal{Y}} - \mathcal{L}_t(\boldsymbol{\mathcal{B}}, \{\mathbf{D}_k\}_{k=1}^d)\|_{\text{HS}}^2 + \|P_{(T_{\boldsymbol{\mathcal{X}}^t})\perp} \boldsymbol{\mathcal{Y}}\|_{\text{HS}}^2 \\
=&\|\mathcal{L}_t(\mathcal{L}_t^*(\boldsymbol{\mathcal{Y}}) - (\boldsymbol{\mathcal{B}}, \{\mathbf{D}_k\}_{k=1}^d))\|_{\text{HS}}^2 + \|P_{(T_{\boldsymbol{\mathcal{X}}^t})\perp} \boldsymbol{\mathcal{Y}}\|_{\text{HS}}^2
\end{aligned}
\tag{41}
$$

here (a) is because $P_{T_{\boldsymbol{\mathcal{X}}^t}} \boldsymbol{\mathcal{Y}}$ and $\mathcal{L}_t(\boldsymbol{\mathcal{B}}, \{\mathbf{D}_k\}_{k=1}^d)$ lie in the tangent space of $\boldsymbol{\mathcal{X}}^t$, while $P_{(T_{\boldsymbol{\mathcal{X}}^t})\perp} \boldsymbol{\mathcal{Y}}$ lies in the orthogonal complement of the tangent space of $\boldsymbol{\mathcal{X}}^t$.

Thus, to minimize the loss function, we just need to minimize the first term on the right-hand side of (41) and it is easy to see the minimizer is obtained when $(\boldsymbol{\mathcal{B}}^{t+1}, \{\mathbf{D}_k^{t+1}\}_{k=1}^d)$ is equal to $\mathcal{L}_t^*(\boldsymbol{\mathcal{Y}})$. The results follow by the definition of $\mathcal{L}_t^*$ in (12). ∎

B.8 Proof of Theorem 3

First the $3\mathbf{r}$-TRIP condition for $\mathscr{A}$ holds when $n \geqslant C(\bar{r}^d + d\bar{p}\bar{r})/R_{3\mathbf{r}}^2$ by (Rauhut et al., 2017, Theorem 2). Denote $\tilde{\sigma}^2 = \|\mathcal{A}\|_{\mathrm{HS}}^2 + \sigma^2$, we claim with probability at least $1 - cc\underline{p}^{-C}$, the following inequalities hold:

$$\left\|\sin\Theta(\mathbf{U}_k^0, \mathbf{U}_k)\right\| = \|\mathbf{U}_{k\perp}^{0\top}\mathbf{U}_k\| \leqslant \frac{\sqrt{p_k/n}\tilde{\sigma}\underline{\lambda} + (\prod_{k=1}^d p_k)^{1/2}\tilde{\sigma}^2/n}{\underline{\lambda}^2}$$

$$\left\|(\mathscr{A}^*(\boldsymbol{\varepsilon}))_{\max(2\mathbf{r})}\right\|_{\mathrm{HS}} \leqslant C\sigma\sqrt{\frac{\sum_{k=1}^d r_k p_k + \prod_{k=1}^d r_k}{n}}. \tag{42}$$

Here the first result holds when $n \geqslant c(d)(\|\boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}}^2 + \sigma^2)\frac{\bar{p}^{d/2}}{\underline{\lambda}^2}$ by the proof of Theorem 4 in Zhang et al. (2020a) and the second result is by Lemma 12.

Since $\boldsymbol{\mathcal{X}}^0 = (\mathscr{A}^*(\mathscr{A}(\boldsymbol{\mathcal{X}}^*) + \boldsymbol{\varepsilon})) \times_{k=1}^d P_{\mathbf{U}_k^0}$, by Lemma 11, we have

$$\left\|\boldsymbol{\mathcal{X}}^0 - \boldsymbol{\mathcal{X}}^*\right\|_{\mathrm{HS}} \leqslant \left\|(\mathscr{A}^*\mathscr{A}(\boldsymbol{\mathcal{X}}^*) - \boldsymbol{\mathcal{X}}^* + \mathscr{A}^*(\boldsymbol{\varepsilon})) \times_{k=1}^d P_{\mathbf{U}_k^0}\right\|_{\mathrm{HS}} + \sum_{k=1}^d \left\|\mathbf{U}_{k\perp}^{0\top}\mathcal{M}_k(\boldsymbol{\mathcal{X}}^*)\right\|_F.$$

For $k = 1, \ldots, d$

$$\begin{aligned}
\left\|\mathbf{U}_k^{0\top}\mathcal{M}_k(\boldsymbol{\mathcal{X}}^*)\right\|_F &= \|\mathbf{U}_{k\perp}^{0\top}\mathbf{U}_k\mathbf{U}_k^\top\mathcal{M}_k(\boldsymbol{\mathcal{X}}^*)\|_F \\
&\leqslant \|\mathbf{U}_{k\perp}^{0\top}\mathbf{U}_k\|_F\|\mathbf{U}_k^\top\mathcal{M}_k(\boldsymbol{\mathcal{X}}^*)\| \\
&\overset{(47)}{\leqslant} \frac{\sqrt{p_k r_k/n}\tilde{\sigma}\underline{\lambda} + (r_k\prod_{k=1}^d p_k)^{1/2}\tilde{\sigma}^2/n}{\underline{\lambda}^2}\bar{\lambda}.
\end{aligned}$$

Moreover,

$$\begin{aligned}
&\left\|(\mathscr{A}^*\mathscr{A}(\boldsymbol{\mathcal{X}}^*) - \boldsymbol{\mathcal{X}}^* + \mathscr{A}^*(\boldsymbol{\varepsilon})) \times_{k=1}^d P_{\mathbf{U}_k^0}\right\|_{\mathrm{HS}} \\
&\leqslant \|(\mathscr{A}^*\mathscr{A}(\boldsymbol{\mathcal{X}}^*) - \boldsymbol{\mathcal{X}}^*) \times_{k=1}^d P_{\mathbf{U}_k^0}\|_{\mathrm{HS}} + \|\mathscr{A}^*(\boldsymbol{\varepsilon}) \times_{k=1}^d P_{\mathbf{U}_k^0}\|_{\mathrm{HS}}.
\end{aligned} \tag{43}$$

Notice that by Lemma 12, with probability at least $1 - \exp(-C\underline{p})$, we have

$$\|\mathscr{A}^*(\boldsymbol{\varepsilon}) \times_{k=1}^d P_{\mathbf{U}_k^0}\|_{\mathrm{HS}} \leqslant \left\|(\mathscr{A}^*(\boldsymbol{\varepsilon}))_{\max(2\mathbf{r})}\right\|_{\mathrm{HS}} \leqslant C'\sigma\sqrt{\frac{\sum_{k=1}^d r_k p_k + \prod_{k=1}^d r_k}{n}}. \tag{44}$$

In addition

$$\begin{aligned}
\|(\mathscr{A}^*\mathscr{A}(\boldsymbol{\mathcal{X}}^*) - \boldsymbol{\mathcal{X}}^*) \times_{k=1}^d P_{\mathbf{U}_k^0}\|_{\mathrm{HS}} &= \sup_{\boldsymbol{\mathcal{Z}}:\|\boldsymbol{\mathcal{Z}}\|_{\mathrm{HS}}=1} \left\langle(\mathscr{A}^*\mathscr{A}(\boldsymbol{\mathcal{X}}^*) - \boldsymbol{\mathcal{X}}^*) \times_{k=1}^d P_{\mathbf{U}_k^0}, \boldsymbol{\mathcal{Z}}\right\rangle \\
&= \sup_{\boldsymbol{\mathcal{Z}}:\|\boldsymbol{\mathcal{Z}}\|_{\mathrm{HS}}=1} \left\langle(\mathscr{A}^*\mathscr{A}(\boldsymbol{\mathcal{X}}^*) - \boldsymbol{\mathcal{X}}^*), \boldsymbol{\mathcal{Z}} \times_{k=1}^d P_{\mathbf{U}_k^0}\right\rangle \\
&\leqslant \sup_{\boldsymbol{\mathcal{Z}}:\|\boldsymbol{\mathcal{Z}}\|_{\mathrm{HS}}=1} \left|\langle\mathscr{A}(\boldsymbol{\mathcal{X}}^*), \mathscr{A}(\boldsymbol{\mathcal{Z}} \times_{k=1}^d P_{\mathbf{U}_k^0})\rangle - \langle\boldsymbol{\mathcal{X}}^*, \boldsymbol{\mathcal{Z}} \times_{k=1}^d P_{\mathbf{U}_k^0}\rangle\right| \\
&\overset{\text{Lemma 10}}{\leqslant} \sup_{\boldsymbol{\mathcal{Z}}:\|\boldsymbol{\mathcal{Z}}\|_{\mathrm{HS}}=1} R_{2\mathbf{r}}\|\boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}}\|\boldsymbol{\mathcal{Z}} \times_{k=1}^d P_{\mathbf{U}_k^0}\|_{\mathrm{HS}} \\
&\leqslant R_{2\mathbf{r}}\|\boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}} \leqslant C\sqrt{\frac{\sum_{k=1}^d r_k p_k + \prod_{k=1}^d r_k}{n}}\|\boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}},
\end{aligned} \tag{45}$$

where the last inequality uses the fact $R_{2\mathbf{r}}$ is of order $\sqrt{\frac{\sum_{k=1}^{d} r_k p_k + \prod_{k=1}^{d} r_k}{n}}$ by (Rauhut et al., 2017, Theorem 2).

By plugging (44), (45) into (43), we have

$$
\begin{aligned}
\|\boldsymbol{\mathcal{X}}^0 - \boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}} &\leqslant C(d) \left(\kappa \sum_{k=1}^{d} \sqrt{\frac{p_k r_k}{n}} \widetilde{\sigma} + \sqrt{\frac{\prod_{k=1}^{d} r_k}{n}} (\sigma + \|\boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}}) + \frac{(\bar{r} \prod_{k=1}^{d} p_k)^{1/2} \kappa \widetilde{\sigma}^2}{\underline{\lambda} n} \right) \\
&\leqslant C(d) \left(\kappa \widetilde{\sigma} \sqrt{\frac{\sum_{k=1}^{d} p_k r_k + \prod_{k=1}^{d} r_k}{n}} + \frac{(\bar{r} \prod_{k=1}^{d} p_k)^{1/2} \kappa \widetilde{\sigma}^2}{\underline{\lambda} n} \right).
\end{aligned}
\tag{46}
$$

Since the sample complexity in satisfying TRIP is smaller than the sample complexity needed in initialization (42), overall when $n \geqslant c(d)(\|\boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}}^2 + \sigma^2) \frac{\kappa^2 \sqrt{\bar{r}} \bar{p}^{d/2}}{\underline{\lambda}^2}$ and $\bar{r} \leqslant \underline{p}^{1/2}$, we have the initialization condition in Theorem 1 is satisfied. Then by Corollary 1 and the upper bound of $\|(\mathscr{A}^*(\boldsymbol{\varepsilon}))_{\max(2\mathbf{r})}\|_{\mathrm{HS}}$ in (42), when

$$
t_{\max} \geqslant C(d) \log\log \left(\frac{\underline{\lambda} \sqrt{n}}{\sigma \sqrt{\sum_{k=1}^{d} r_k p_k + \prod_{k=1}^{d} r_k}} \right),
$$

we have

$$
\|\boldsymbol{\mathcal{X}}^{t_{\max}} - \boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}} \leqslant c(\sqrt{d} + 1)\sigma \sqrt{\left(\sum_{k=1}^{d} r_k p_k + \prod_{k=1}^{d} r_k \right) / n}.
$$

∎

B.9 Proof of Theorem 4

As we have mentioned in this setting, $\mathscr{A}$ satisfies TRIP with $R_{2\mathbf{r}} = 0, R_{3\mathbf{r}} = 0$ and here $\|(\mathscr{A}^*(\boldsymbol{\varepsilon}))_{\max(2\mathbf{r})}\|_{\mathrm{HS}} = \|\boldsymbol{\mathcal{E}}_{\max(2\mathbf{r})}\|_{\mathrm{HS}}$ where $\boldsymbol{\mathcal{E}}$ is the noise in the tensorized model. Without loss of generality, we assume $\sigma = 1$.

First, we claim with probability at least $1 - C\exp(-c\underline{p})$, the following inequalities hold:

$$
\begin{aligned}
\left\|\sin\Theta(\mathbf{U}_k^0, \mathbf{U}_k)\right\| &= \|\mathbf{U}_{k\perp}^{0\top} \mathbf{U}_k\| \leqslant \frac{\sqrt{p_k}\underline{\lambda} + (\prod_{k=1}^{d} p_k)^{1/2}}{\underline{\lambda}^2} \\
\|\boldsymbol{\mathcal{E}}_{\max(2\mathbf{r})}\|_{\mathrm{HS}} &\leqslant C \sqrt{\sum_{k=1}^{d} r_k p_k + \prod_{k=1}^{d} r_k}.
\end{aligned}
\tag{47}
$$

Here the first result holds when $\underline{\lambda} \geqslant c(d)\kappa \bar{p}^{d/4} \bar{r}^{1/4}$ by the proof of Theorem 1 in Zhang and Xia (2018) and the second result is by Lemma 12.

Since $\boldsymbol{\mathcal{X}}^0 = \boldsymbol{\mathcal{Y}} \times_{k=1}^{d} P_{\mathbf{U}_k^0}$, by Lemma 11, we have

$$
\|\boldsymbol{\mathcal{X}}^0 - \boldsymbol{\mathcal{X}}^*\|_{\mathrm{HS}} \leqslant \left\|\boldsymbol{\mathcal{E}} \times_{k=1}^{d} P_{\mathbf{U}_k^0}\right\|_{\mathrm{HS}} + \sum_{k=1}^{d} \|\mathbf{U}_{k\perp}^{0\top} \mathcal{M}_k(\boldsymbol{\mathcal{X}}^*)\|_F.
\tag{48}
$$

For $k = 1, \ldots, d$

$$
\begin{aligned}
\left\| \mathbf{U}_k^{0\top} \mathcal{M}_k(\boldsymbol{\mathcal{X}}^*) \right\|_F &= \| \mathbf{U}_{k\perp}^{0\top} \mathbf{U}_k \mathbf{U}_k^\top \mathcal{M}_k(\boldsymbol{\mathcal{X}}^*) \|_F \\
&\leqslant \| \mathbf{U}_{k\perp}^{0\top} \mathbf{U}_k \|_F \| \mathbf{U}_k^\top \mathcal{M}_k(\boldsymbol{\mathcal{X}}^*) \| \\
&\overset{(47)}{\leqslant} \frac{\sqrt{p_k r_k}\underline{\lambda} + (r_k \prod_{k=1}^d p_k)^{1/2}}{\underline{\lambda}^2} \bar{\lambda}.
\end{aligned}
\tag{49}
$$

Notice $\left\| \boldsymbol{\mathcal{E}} \times_{k=1}^d P_{\mathbf{U}_k^0} \right\|_{\mathrm{HS}} \leqslant \| \boldsymbol{\mathcal{E}}_{\max(2\mathbf{r})} \|_{\mathrm{HS}}$ by definition of $\| (\cdot)_{\max(2\mathbf{r})} \|_{\mathrm{HS}}$, combining (47) and (49), from (48), we have

$$
\| \boldsymbol{\mathcal{X}}^0 - \boldsymbol{\mathcal{X}}^* \|_{\mathrm{HS}} \leqslant C(d) \left(\kappa \sum_{k=1}^d \sqrt{p_k r_k} + (\prod_{k=1}^d r_k)^{1/2} + \frac{(\bar{r} \prod_{k=1}^d p_k)^{1/2} \kappa}{\underline{\lambda}} \right).
\tag{50}
$$

Notice, when $\underline{\lambda} \geqslant c(d) \kappa \bar{p}^{d/4} \bar{r}^{1/4}$ and $\bar{r} \leqslant \underline{p}^{1/2}$, we have the initialization condition in Theorem 1 is satisfied. Then by Corollary 1 and the upper bound of $\| \boldsymbol{\mathcal{E}}_{\max(2\mathbf{r})} \|_{\mathrm{HS}}$ in (47), when

$$
t_{\max} \geqslant C(d) \log \log \left(\frac{\underline{\lambda}}{\sqrt{\sum_{k=1}^d r_k p_k + \prod_{k=1}^d r_k}} \right),
$$

we have

$$
\| \boldsymbol{\mathcal{X}}^{t_{\max}} - \boldsymbol{\mathcal{X}}^* \|_{\mathrm{HS}} \leqslant 2(\sqrt{d} + 1) \sqrt{\sum_{k=1}^d r_k p_k + \prod_{k=1}^d r_k}.
$$

∎

Appendix C. Additional Proofs and Lemmas

Proof of Lemma 4. The proof is similar to the proof of (Kressner et al., 2014, Proposition 2.3). Recall retraction R is a smooth map from $T\mathbb{M}_{\mathbf{r}}$ to $\mathbb{M}_{\mathbf{r}}$ that satisfies

- Property i) $R(\boldsymbol{\mathcal{X}}, 0) = \boldsymbol{\mathcal{X}}$;

- Property ii) $\frac{d}{dt} R(\boldsymbol{\mathcal{X}}, t\eta)|_{t=0} = \eta$ for all $\boldsymbol{\mathcal{X}} \in \mathbb{M}_{\mathbf{r}}$ and $\eta \in T_{\boldsymbol{\mathcal{X}}} \mathbb{M}_{\mathbf{r}}$.

We begin by checking the smoothness of ST-HOSVD. For any tensor $\boldsymbol{\mathcal{Y}}$, let $\{\mathbf{U}_k^0\}_{k=1}^d$ be the mode-k singular vectors computed in the ST-HOSVD algorithm. Let D_1 denotes the collection of tensors whose mode-1 matricization has a nonzero singular gap between the r_1th and $(r_1 + 1)$th singular values, and let D_k be collection of tensors $\boldsymbol{\mathcal{Y}}$ such that $\mathcal{M}_{r_k}(\boldsymbol{\mathcal{Y}} \times_{l=1}^{k-1} P_{\mathbf{U}_l^0})$ has a nonzero singular gap between the r_kth and $(r_k + 1)$th singular values. Let $\mathcal{P}_{\mathbf{U}_k^0}$ be a projection operator in the tensor space such that $\mathcal{P}_{\mathbf{U}_k^0} \boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{Y}} \times_k P_{\mathbf{U}_k^0}$. From standard results in matrix perturbation theory (Chern and Dieci, 2001), $\mathcal{P}_{\mathbf{U}_k^0}$ is smooth and well-defined on D_k. Since $\boldsymbol{\mathcal{X}}$ is contained in all D_k for $k = 1, \ldots, d$ and is a fixed point of every $\mathcal{P}_{\mathbf{U}_k^0}$, we can construct an open neighborhood D of $\boldsymbol{\mathcal{X}}$ such that $\mathcal{P}_{\mathbf{U}_1^0} \circ \cdots \circ \mathcal{P}_{\mathbf{U}_d^0} D \subseteq D_k$ for all k (here "$\circ$" is the composition of the projection operator). The smoothness of ST-HOSVD is implied by the chain rule.

Next, we check the two properties of retraction. Property i) is clear as $\boldsymbol{\mathcal{X}} \in \mathbb{M}_{\mathbf{r}}$. For Property ii), because the tangent space $T_{\boldsymbol{\mathcal{X}}}\mathbb{M}_{\mathbf{r}}$ is a first-order approximation of $\mathbb{M}_{\mathbf{r}}$ around η, we have $\|(\boldsymbol{\mathcal{X}} + t\eta) - P_{\mathbb{M}_{\mathbf{r}}}(\boldsymbol{\mathcal{X}} + t\eta)\| = O(t^2)$ as $t \to 0$. Thus, by the quasi-projection property of ST-HOSVD (Hackbusch, 2012, Chapter 10), we have

$$\|(\boldsymbol{\mathcal{X}} + t\eta) - \text{ST-HOSVD}(\boldsymbol{\mathcal{X}} + t\eta)\| \leqslant \sqrt{d}\|(\boldsymbol{\mathcal{X}} + t\eta) - P_{\mathbb{M}_{\mathbf{r}}}(\boldsymbol{\mathcal{X}} + t\eta)\| = O(t^2).$$

Hence, $R(\boldsymbol{\mathcal{X}} + t\eta) = (\boldsymbol{\mathcal{X}} + t\eta) + O(t^2)$ and $\frac{d}{dt}R(\boldsymbol{\mathcal{X}}, t\eta)|_{t=0} = \eta$. This has finished the proof. ∎

Proof of Lemma 3. Since $\mathbb{M}_{\mathbf{r}}$ is an embedded submanifold of $\mathbb{R}^{p_1 \times \cdots \times p_d}$, from (Absil et al., 2009, (3.37)), we have the result. ∎

Lemma 10 (Tensor Restricted Orthogonal Property) *Let $\boldsymbol{\mathcal{Z}}_1, \boldsymbol{\mathcal{Z}}_2 \in \mathbb{R}^{p_1 \times \cdots \times p_d}$ be two low Tucker rank tensors with $\text{Tucrank}(\boldsymbol{\mathcal{Z}}_1) = \mathbf{r}_1 := (r_1, \ldots, r_d)$, $\text{Tucrank}(\boldsymbol{\mathcal{Z}}_2) = \mathbf{r}_2 := (r'_1, \ldots, r'_d)$. If $\mathscr{A}$ satisfies the $(\mathbf{r}_1 + \mathbf{r}_2)$-TRIP condition, hen we have*

$$|\langle \mathscr{A}(\boldsymbol{\mathcal{Z}}_1), \mathscr{A}(\boldsymbol{\mathcal{Z}}_2)\rangle - \langle \boldsymbol{\mathcal{Z}}_1, \boldsymbol{\mathcal{Z}}_2\rangle| \leqslant R_{\mathbf{r}_1 + \mathbf{r}_2}\|\boldsymbol{\mathcal{Z}}_1\|_{\text{HS}}\|\boldsymbol{\mathcal{Z}}_2\|_{\text{HS}}. \tag{51}$$

In particular, if $\langle \boldsymbol{\mathcal{Z}}_1, \boldsymbol{\mathcal{Z}}_2\rangle = 0$, we have

$$|\langle \mathscr{A}(\boldsymbol{\mathcal{Z}}_1), \mathscr{A}(\boldsymbol{\mathcal{Z}}_2)\rangle| \leqslant R_{\mathbf{r}_1 + \mathbf{r}_2}\|\boldsymbol{\mathcal{Z}}_1\|_{\text{HS}}\|\boldsymbol{\mathcal{Z}}_2\|_{\text{HS}}.$$

Proof. The proof for the matrix version of this result can be found in (Candès and Plan, 2011, Lemma 3.3) and here we present the proof in the tensor setting. Without loss of generality, assume $\|\boldsymbol{\mathcal{Z}}_1\|_{\text{HS}} = 1$, $\|\boldsymbol{\mathcal{Z}}_2\|_{\text{HS}} = 1$. Notice that $\boldsymbol{\mathcal{Z}}_1 + \boldsymbol{\mathcal{Z}}_2$ is of at most Tucker rank $\mathbf{r}_1 + \mathbf{r}_2$ as the matricization of $\boldsymbol{\mathcal{Z}}_1 + \boldsymbol{\mathcal{Z}}_2$ on each mode k is of at most rank $r_k + r'_k$. Similarly, $\boldsymbol{\mathcal{Z}}_1 - \boldsymbol{\mathcal{Z}}_2$ is also at most Tucker rank $\mathbf{r}_1 + \mathbf{r}_2$. By the TRIP of $\mathscr{A}$, we have

$$2(1 - R_{\mathbf{r}_1 + \mathbf{r}_2}) \pm 2(1 - R_{\mathbf{r}_1 + \mathbf{r}_2})\langle \boldsymbol{\mathcal{Z}}_1, \boldsymbol{\mathcal{Z}}_2\rangle = (1 - R_{\mathbf{r}_1 + \mathbf{r}_2})\|\boldsymbol{\mathcal{Z}}_1 \pm \boldsymbol{\mathcal{Z}}_2\|_{\text{HS}}^2 \leqslant \|\mathscr{A}(\boldsymbol{\mathcal{Z}}_1 \pm \boldsymbol{\mathcal{Z}}_2)\|_{\text{HS}}^2$$
$$2(1 + R_{\mathbf{r}_1 + \mathbf{r}_2}) \pm 2(1 + R_{\mathbf{r}_1 + \mathbf{r}_2})\langle \boldsymbol{\mathcal{Z}}_1, \boldsymbol{\mathcal{Z}}_2\rangle = (1 + R_{\mathbf{r}_1 + \mathbf{r}_2})\|\boldsymbol{\mathcal{Z}}_1 \pm \boldsymbol{\mathcal{Z}}_2\|_{\text{HS}}^2 \geqslant \|\mathscr{A}(\boldsymbol{\mathcal{Z}}_1 \pm \boldsymbol{\mathcal{Z}}_2)\|_{\text{HS}}^2.$$

Then we have

$$\langle \mathscr{A}(\boldsymbol{\mathcal{Z}}_1), \mathscr{A}(\boldsymbol{\mathcal{Z}}_2)\rangle = \frac{1}{4}\left(\|\mathscr{A}(\boldsymbol{\mathcal{Z}}_1 + \boldsymbol{\mathcal{Z}}_2)\|_{\text{HS}}^2 - \mathscr{A}(\boldsymbol{\mathcal{Z}}_1 - \boldsymbol{\mathcal{Z}}_2)\|_{\text{HS}}^2\right) \leqslant R_{\mathbf{r}_1 + \mathbf{r}_2} + \langle \boldsymbol{\mathcal{Z}}_1, \boldsymbol{\mathcal{Z}}_2\rangle$$
$$\langle \mathscr{A}(\boldsymbol{\mathcal{Z}}_1), \mathscr{A}(\boldsymbol{\mathcal{Z}}_2)\rangle = \frac{1}{4}\left(\|\mathscr{A}(\boldsymbol{\mathcal{Z}}_1 + \boldsymbol{\mathcal{Z}}_2)\|_{\text{HS}}^2 - \mathscr{A}(\boldsymbol{\mathcal{Z}}_1 - \boldsymbol{\mathcal{Z}}_2)\|_{\text{HS}}^2\right) \geqslant -(R_{\mathbf{r}_1 + \mathbf{r}_2} + \langle \boldsymbol{\mathcal{Z}}_1, \boldsymbol{\mathcal{Z}}_2\rangle).$$

Finally, we have

$$|\langle \mathscr{A}(\boldsymbol{\mathcal{Z}}_1), \mathscr{A}(\boldsymbol{\mathcal{Z}}_2)\rangle - \langle \boldsymbol{\mathcal{Z}}_1, \boldsymbol{\mathcal{Z}}_2\rangle| \leqslant R_{\mathbf{r}_1 + \mathbf{r}_2}.$$

∎

Lemma 11 (Tensor Estimation from Projection) *Given two order-d tensors $\boldsymbol{\mathcal{Y}}, \boldsymbol{\mathcal{X}} \in \mathbb{R}^{p_1 \times \cdots \times p_d}$. Suppose $\mathbf{U}_k^0 \in \mathbb{O}_{p_k, r_k}$, then*

$$\left\|\boldsymbol{\mathcal{Y}} \times_{k=1}^d P_{\mathbf{U}_k^0} - \boldsymbol{\mathcal{X}}\right\|_{\text{HS}} \leqslant \left\|(\boldsymbol{\mathcal{Y}} - \boldsymbol{\mathcal{X}}) \times_1 P_{\mathbf{U}_1^0} \times \cdots \times_d P_{\mathbf{U}_d^0}\right\|_{\text{HS}} + \sum_{k=1}^d \left\|\mathbf{U}_{k\perp}^{0\top}\mathcal{M}_k(\boldsymbol{\mathcal{X}})\right\|_F.$$

Proof. First, notice the following decomposition

$$
\begin{aligned}
\boldsymbol{\mathcal{X}} =& \boldsymbol{\mathcal{X}} \times_1 P_{\mathbf{U}_1^0} \times \cdots \times_d P_{\mathbf{U}_d^0} + \boldsymbol{\mathcal{X}} \times_1 P_{\mathbf{U}_{1\perp}^0} \times_2 P_{\mathbf{U}_2^0} \times \cdots \times_d P_{\mathbf{U}_d^0} \\
&+ \boldsymbol{\mathcal{X}} \times_1 \mathbf{I}_{p_1} \times_2 P_{\mathbf{U}_{2\perp}^0} \times \cdots \times_d P_{\mathbf{U}_d^0} + \boldsymbol{\mathcal{X}} \times_1 \mathbf{I}_{p_1} \times_2 \mathbf{I}_{p_2} \times_3 P_{\mathbf{U}_{3\perp}^0} \cdots \times_d P_{\mathbf{U}_d^0} \\
&+ \ldots + \boldsymbol{\mathcal{X}} \times_{i=1}^{d-1} \mathbf{I}_{p_i} \times_d P_{\mathbf{U}_{d\perp}^0} \\
=& \boldsymbol{\mathcal{X}} \times_1 P_{\mathbf{U}_1^0} \times \cdots \times_d P_{\mathbf{U}_d^0} + \sum_{k=1}^d \boldsymbol{\mathcal{X}} \times_{i=1}^{k-1} \mathbf{I}_{p_i} \times_k P_{\mathbf{U}_{k\perp}^0} \times_{i=k+1}^d P_{\mathbf{U}_i^0}.
\end{aligned}
$$

Thus,

$$
\begin{aligned}
&\left\| \boldsymbol{\mathcal{Y}} \times_1 P_{\mathbf{U}_1^0} \times \cdots \times_d P_{\mathbf{U}_d^0} - \boldsymbol{\mathcal{X}} \right\|_{\mathrm{HS}} \\
&\leqslant \left\| (\boldsymbol{\mathcal{Y}} - \boldsymbol{\mathcal{X}}) \times_1 P_{\mathbf{U}_1^0} \times \cdots \times_d P_{\mathbf{U}_d^0} \right\|_{\mathrm{HS}} + \sum_{k=1}^d \left\| \boldsymbol{\mathcal{X}} \times_{i=1}^{k-1} \mathbf{I}_{p_i} \times_k P_{\mathbf{U}_{k\perp}^0} \times_{i=k+1}^d P_{\mathbf{U}_i^0} \right\|_{\mathrm{HS}} \\
&\leqslant \left\| (\boldsymbol{\mathcal{Y}} - \boldsymbol{\mathcal{X}}) \times_1 P_{\mathbf{U}_1^0} \times \cdots \times_d P_{\mathbf{U}_d^0} \right\|_{\mathrm{HS}} + \sum_{k=1}^d \left\| \mathbf{U}_{k\perp}^{0\top} \mathcal{M}_k(\boldsymbol{\mathcal{X}}) \right\|_F.
\end{aligned}
$$

∎

Lemma 12 (Concentration of the Noise) *Suppose* $\boldsymbol{\mathcal{E}} \in \mathbb{R}^{p_1 \times \cdots \times p_d}$ *and it has i.i.d.* $N(0,1)$ *entries. Then with probability at least* $1 - \exp(-C\underline{p})$ *(* $\underline{p} := \min_{k=1,\ldots,d} p_k$ *), we have*

$$
\left\| \boldsymbol{\mathcal{E}}_{\max(2\mathbf{r})} \right\|_{\mathrm{HS}} \leqslant C' \sqrt{\sum_{k=1}^d r_k p_k + \prod_{k=1}^d r_k},
$$

for some $C' > 0$.

Suppose $\mathscr{A} \in \mathbb{R}^{p_1 \times \cdots \times p_d} \to \mathbb{R}^n$ *is a linear map in (2) and its covariates* $\boldsymbol{\mathcal{A}}_i$ *are independent and has i.i.d.* $N(0, \frac{1}{n})$ *entries, and* $\varepsilon_i \overset{i.i.d.}{\sim} N(0, \frac{\sigma^2}{n})$. *Then with probability at least* $1 - \exp(-C\underline{p})$, *we have*

$$
\left\| (\mathscr{A}^*(\boldsymbol{\varepsilon}))_{\max(2\mathbf{r})} \right\|_{\mathrm{HS}} \leqslant C' \sigma \sqrt{\frac{\sum_{k=1}^d r_k p_k + \prod_{k=1}^d r_k}{n}},
$$

for some $C' > 0$.

Proof. The first result is proved in Lemma 5 of Zhang and Xia (2018) in $d = 3$ case and it can be easily generalized to order-d setting.

For the second result, recall

$$
\begin{aligned}
\left\| (\mathscr{A}^*(\boldsymbol{\varepsilon}))_{\max(2\mathbf{r})} \right\|_{\mathrm{HS}} &= \sup_{\mathbf{U}_k \in \mathbb{O}_{p_k, 2r_k}, k=1,\ldots,d} \left\| \mathscr{A}^*(\boldsymbol{\varepsilon}) \times_{k=1}^d P_{\mathbf{U}_k} \right\|_{\mathrm{HS}} \\
&= \sup_{\mathbf{U}_k \in \mathbb{O}_{p_k, 2r_k}, k=1,\ldots,d} \sup_{\boldsymbol{\mathcal{Z}}: \|\boldsymbol{\mathcal{Z}}\|_{\mathrm{HS}} \leqslant 1} \left\langle \boldsymbol{\mathcal{Z}}, \mathscr{A}^*(\boldsymbol{\varepsilon}) \times_{k=1}^d P_{\mathbf{U}_k} \right\rangle \\
&= \sup_{\mathbf{U}_k \in \mathbb{O}_{p_k, 2r_k}, k=1,\ldots,d} \sup_{\boldsymbol{\mathcal{Z}}: \|\boldsymbol{\mathcal{Z}}\|_{\mathrm{HS}} \leqslant 1} \left\langle \boldsymbol{\mathcal{Z}} \times_{k=1}^d P_{\mathbf{U}_k}, \mathscr{A}^*(\boldsymbol{\varepsilon}) \right\rangle.
\end{aligned}
$$

It was proved in Theorem 4.2 of Han et al. (2022b) that

$$\sup_{\mathbf{U}_k \in \mathbb{O}_{p_k, 2r_k}, k=1, \ldots, d} \sup_{\boldsymbol{\mathcal{Z}}: \|\boldsymbol{\mathcal{Z}}\|_{\mathrm{HS}} \leqslant 1} \langle \boldsymbol{\mathcal{Z}} \times_{k=1}^{d} P_{\mathbf{U}_k}, \mathscr{A}^*(\boldsymbol{\varepsilon}) \rangle \leqslant 4\sigma \sqrt{\frac{\sum_{k=1}^{d} r_k p_k + \prod_{k=1}^{d} r_k}{n}}$$

holds with probability at least $1 - \exp(-Cp)$ when $d = 3$. It is straightforward to extend to the order-d case. For simplicity, we omit the proof here. ∎

References

P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds.* Princeton University Press, 2009.

Talal Ahmed, Haroon Raja, and Waheed U Bajwa. Tensor regression using low-rank and sparse tucker decompositions. *SIAM Journal on Mathematics of Data Science*, 2(4):944–966, 2020.

Animashree Anandkumar, Rong Ge, Daniel Hsu, Sham M Kakade, and Matus Telgarsky. Tensor decompositions for learning latent variable models. *The Journal of Machine Learning Research*, 15(1):2773–2832, 2014a.

Animashree Anandkumar, Rong Ge, and Majid Janzamin. Guaranteed non-orthogonal tensor decomposition via alternating rank-1 updates. *arXiv preprint arXiv:1402.5180*, 2014b.

Krishnakumar Balasubramanian, Jianqing Fan, and Zhuoran Yang. Tensor methods for additive index models under discordance and heterogeneity. *arXiv preprint arXiv:1807.06693*, 2018.

Boaz Barak and Ankur Moitra. Noisy tensor completion via the sum-of-squares hierarchy. In *Conference on Learning Theory*, pages 417–445, 2016.

Jonathan Bauch, Boaz Nadler, and Pini Zilber. Rank 2r iterative least squares: efficient recovery of ill-conditioned low rank matrices from few entries. *SIAM Journal on Mathematics of Data Science*, 3(1):439–465, 2021.

Xuan Bi, Annie Qu, and Xiaotong Shen. Multilayer tensor factorization with applications to recommender systems. *The Annals of Statistics*, 46(6B):3308–3333, 2018.

Peter J Bickel. One-step huber estimates in the linear model. *Journal of the American Statistical Association*, 70(350):428–434, 1975.

Nicolas Boumal. An introduction to optimization on smooth manifolds. *http://sma.epfl.ch/ nboumal/#book*, 2020.

Nicolas Boumal and P-A Absil. Low-rank matrix completion via preconditioned optimization on the Grassmann manifold. *Linear Algebra and its Applications*, 475:200–239, 2015.

Nicolas Boumal and Pierre-antoine Absil. Rtrmc: A Riemannian trust-region method for low-rank matrix completion. In *Advances in neural information processing systems*, pages 406–414, 2011.

Martijn Boussé, Nico Vervliet, Ignat Domanov, Otto Debals, and Lieven De Lathauwer. Linear systems with a canonical polyadic decomposition constrained solution: Algorithms and applications. *Numerical Linear Algebra with Applications*, 25(6):e2190, 2018.

Paul Breiding and Nick Vannieuwenhoven. Convergence analysis of Riemannian Gauss–Newton methods and its connection with the geometric condition number. *Applied Mathematics Letters*, 78:42–50, 2018.

Matthew Brennan and Guy Bresler. Reducibility and statistical-computational gaps from secret leakage. In *Conference on Learning Theory*, pages 648–847. PMLR, 2020.

Changxiao Cai, Gen Li, H. Vincent Poor, and Yuxin Chen. Nonconvex low-rank tensor completion from noisy data. In *Advances in Neural Information Processing Systems*, volume 32, pages 1863–1874, 2019.

Jian-Feng Cai and Ke Wei. Exploiting the structure effectively and efficiently in low-rank matrix recovery. In *Handbook of Numerical Analysis*, volume 19, pages 21–51. Elsevier, 2018.

Jian-Feng Cai, Lizhang Miao, Yang Wang, and Yin Xian. Provable near-optimal low-multilinear-rank tensor recovery. *arXiv preprint arXiv:2007.08904*, 2020.

Jian-Feng Cai, Jingyang Li, and Dong Xia. Generalized low-rank plus sparse tensor estimation by fast Riemannian optimization. *Journal of the American Statistical Association*, pages 1–17, 2022.

T Tony Cai and Anru Zhang. Sharp RIP bound for sparse signal and low-rank matrix recovery. *Applied and Computational Harmonic Analysis*, 35(1):74–93, 2013.

T Tony Cai and Anru Zhang. Rop: Matrix recovery via rank-one projections. *The Annals of Statistics*, 43(1):102–138, 2015.

Emmanuel J Candès and Yaniv Plan. Tight oracle inequalities for low-rank matrix recovery from a minimal number of noisy random measurements. *IEEE Transactions on Information Theory*, 57(4):2342–2359, 2011.

Vasileios Charisopoulos, Yudong Chen, Damek Davis, Mateo Díaz, Lijun Ding, and Dmitriy Drusvyatskiy. Low-rank matrix recovery with composite optimization: good conditioning and rapid convergence. *Foundations of Computational Mathematics*, pages 1–89, 2021.

Han Chen, Garvesh Raskutti, and Ming Yuan. Non-convex projected gradient descent for generalized low-rank tensor regression. *The Journal of Machine Learning Research*, 20(1):172–208, 2019a.

Yuxin Chen, Yuejie Chi, Jianqing Fan, and Cong Ma. Gradient descent with random initialization: Fast global convergence for nonconvex phase retrieval. *Mathematical Programming*, 176(1):5–37, 2019b.

Anoop Cherian and Suvrit Sra. Riemannian dictionary learning and sparse coding for positive definite matrices. *IEEE transactions on neural networks and learning systems*, 28 (12):2859–2871, 2016.

Jann-Long Chern and Luca Dieci. Smoothness and periodicity of some matrix decompositions. *SIAM Journal on Matrix Analysis and Applications*, 22(3):772–792, 2001.

Yuejie Chi, Yue M Lu, and Yuxin Chen. Nonconvex optimization meets low-rank matrix factorization: An overview. *IEEE Transactions on Signal Processing*, 67(20):5239–5269, 2019.

Curt Da Silva and Felix J Herrmann. Optimization on the hierarchical tucker manifold–applications to tensor completion. *Linear Algebra and its Applications*, 481:131–173, 2015.

Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

Ilias Diakonicolas, Daniel M. Kane, Yuetian Luo, and Anru R Zhang. Statistical and computational limits for tensor-on-tensor association detection. In *Conference on Learning Theory*. PMLR, 2023.

Shuyu Dong, Bin Gao, Yu Guan, and François Glineur. New Riemannian preconditioned algorithms for tensor completion via polyadic decomposition. *SIAM Journal on Matrix Analysis and Applications*, 43(2):840–866, 2022.

Lars Eldén and Berkant Savas. A Newton–Grassmann method for computing the best multilinear rank-(r_1, r_2, r_3) approximation of a tensor. *SIAM Journal on Matrix Analysis and applications*, 31(2):248–271, 2009.

Silvia Gandy, Benjamin Recht, and Isao Yamada. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems*, 27(2):025010, 2011.

Irina Georgieva and Clemens Hofreither. Greedy low-rank approximation in tucker format of solutions of tensor linear systems. *Journal of Computational and Applied Mathematics*, 358:206–220, 2019.

Lars Grasedyck. Hierarchical singular value decomposition of tensors. *SIAM Journal on Matrix Analysis and Applications*, 31(4):2029–2054, 2010.

Lars Grasedyck, Daniel Kressner, and Christine Tobler. A literature survey of low-rank tensor approximation techniques. *GAMM-Mitteilungen*, 36(1):53–78, 2013.

Rajarshi Guhaniyogi, Shaan Qamar, and David B Dunson. Bayesian tensor regression. *The Journal of Machine Learning Research*, 18(1):2733–2763, 2017.

Weiwei Guo, Irene Kotsia, and Ioannis Patras. Tensor learning for regression. *IEEE Transactions on Image Processing*, 21(2):816–827, 2012.

Wolfgang Hackbusch. *Tensor spaces and numerical tensor calculus*, volume 42. Springer, 2012.

Wolfgang Hackbusch and Stefan Kühn. A new scheme for the tensor representation. *Journal of Fourier analysis and applications*, 15(5):706–722, 2009.

Rungang Han, Yuetian Luo, Miaoyan Wang, and Anru R Zhang. Exact clustering in tensor block model: Statistical optimality and computational limit. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(5):1666–1698, 2022a.

Rungang Han, Rebecca Willett, and Anru R Zhang. An optimal statistical and computational framework for generalized tensor estimation. *The Annals of Statistics*, 50(1):1–29, 2022b.

Botao Hao, Anru Zhang, and Guang Cheng. Sparse and low-rank tensor estimation via cubic sketchings. *IEEE Transactions on Information Theory*, 2020.

Gennadij Heidel and Volker Schulz. A Riemannian trust-region method for low-rank tensor completion. *Numerical Linear Algebra with Applications*, 25(6):e2175, 2018.

Christopher J Hillar and Lek-Heng Lim. Most tensor problems are np-hard. *Journal of the ACM (JACM)*, 60(6):1–39, 2013.

Peter D Hoff. Multilinear tensor regression for longitudinal relational data. *The Annals of Applied Statistics*, 9(3):1169, 2015.

Clemens Hofreither. A black-box low-rank approximation algorithm for fast matrix assembly in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 333: 311–330, 2018.

Sebastian Holtz, Thorsten Rohwedder, and Reinhold Schneider. On manifolds of tensors of fixed tt-rank. *Numerische Mathematik*, 120(4):701–731, 2012.

Samuel B Hopkins, Jonathan Shi, and David Steurer. Tensor principal component analysis via sum-of-square proofs. In *Proceedings of The 28th Conference on Learning Theory, COLT*, pages 3–6, 2015.

Thomas Y Hou, Zhenzhen Li, and Ziyun Zhang. Fast global convergence for low-rank matrix recovery via Riemannian gradient descent with random initialization. *arXiv preprint arXiv:2012.15467*, 2020.

Wen Huang and Paul Hand. Blind deconvolution by a steepest descent algorithm on a quotient manifold. *SIAM Journal on Imaging Sciences*, 11(4):2757–2785, 2018.

Mariya Ishteva, Lieven De Lathauwer, P-A Absil, and Sabine Van Huffel. Differential-geometric Newton method for the best rank-(r_1, r_2, r_3) approximation of tensors. *Numerical Algorithms*, 51(2):179–194, 2009.

Mariya Ishteva, P-A Absil, Sabine Van Huffel, and Lieven De Lathauwer. Best low multilinear rank approximation of higher-order tensors, based on the Riemannian trust-region scheme. *SIAM Journal on Matrix Analysis and Applications*, 32(1):115–135, 2011.

Prateek Jain and Sewoong Oh. Provable tensor factorization with missing data. In *Advances in Neural Information Processing Systems*, volume 27, 2014.

Prateek Jain, Raghu Meka, and Inderjit S Dhillon. Guaranteed rank minimization via singular value projection. In *Advances in Neural Information Processing Systems*, pages 937–945, 2010.

Hiroyuki Kasai and Bamdev Mishra. Low-rank tensor completion: a Riemannian manifold preconditioning approach. In *International Conference on Machine Learning*, pages 1012–1021. PMLR, 2016.

Raghunandan H Keshavan, Sewoong Oh, and Andrea Montanari. Matrix completion from a few entries. In *2009 IEEE International Symposium on Information Theory*, pages 324–328. IEEE, 2009.

Othmar Koch and Christian Lubich. Dynamical tensor approximation. *SIAM Journal on Matrix Analysis and Applications*, 31(5):2360–2375, 2010.

Tamara G Kolda. Orthogonal tensor decompositions. *SIAM Journal on Matrix Analysis and Applications*, 23(1):243–255, 2001.

Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

Daniel Kressner, Michael Steinlechner, and Bart Vandereycken. Low-rank tensor completion by Riemannian optimization. *BIT Numerical Mathematics*, 54(2):447–468, 2014.

Daniel Kressner, Michael Steinlechner, and Bart Vandereycken. Preconditioned low-rank Riemannian optimization for linear systems with tensor product structure. *SIAM Journal on Scientific Computing*, 38(4):A2018–A2044, 2016.

Lexin Li and Xin Zhang. Parsimonious tensor response regression. *Journal of the American Statistical Association*, pages 1–16, 2017.

Xiaodong Li, Shuyang Ling, Thomas Strohmer, and Ke Wei. Rapid, robust, and reliable blind deconvolution via nonconvex optimization. *Applied and computational harmonic analysis*, 47(3):893–934, 2019.

Xiaoshan Li, Da Xu, Hua Zhou, and Lexin Li. Tucker tensor regression and neuroimaging analysis. *Statistics in Biosciences*, pages 1–26, 2018.

Allen Liu and Ankur Moitra. Tensor completion made practical. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Ji Liu, Przemyslaw Musialski, Peter Wonka, and Jieping Ye. Tensor completion for estimating missing values in visual data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):208–220, 2013.

Wenqi Lu, Zhongyi Zhu, and Heng Lian. High-dimensional quantile tensor regression. *Journal of Machine Learning Research*, 21(250):1–31, 2020.

Yuetian Luo and Anru R Zhang. Open problem: Average-case hardness of hypergraphic planted clique detection. In *Conference on Learning Theory*, pages 3852–3856. PMLR, 2020.

Yuetian Luo and Anru R Zhang. Tensor clustering with planted structures: Statistical optimality and computational limits. *The Annals of Statistics*, 50(1):584–613, 2022a.

Yuetian Luo and Anru R Zhang. Tensor-on-tensor regression: Riemannian optimization, over-parameterization, statistical-computational gap, and their interplay. *arXiv preprint arXiv:2206.08756*, 2022b.

Yuetian Luo, Wen Huang, Xudong Li, and Anru Zhang. Recursive importance sketching for rank constrained least squares: Algorithms and high-order convergence. *Operations Research*, 2023.

Re Lynch, John R Rice, and Donald H Thomas. Tensor product analysis of partial difference equations. *Bulletin of the American Mathematical Society*, 70(3):378–384, 1964.

Cong Ma, Kaizheng Wang, Yuejie Chi, and Yuxin Chen. Implicit regularization in nonconvex statistical estimation: Gradient descent converges linearly for phase retrieval, matrix completion, and blind deconvolution. *Foundations of Computational Mathematics*, pages 1–182, 2019.

Gilles Meyer, Silvere Bonnabel, and Rodolphe Sepulchre. Linear regression under fixed-rank constraints: a Riemannian approach. In *Proceedings of the 28th international conference on machine learning*, 2011.

Bamdev Mishra, Gilles Meyer, Silvere Bonnabel, and Rodolphe Sepulchre. Fixed-rank matrix factorizations and Riemannian low-rank optimization. *Computational Statistics*, 29 (3-4):591–621, 2014.

Andrea Montanari and Nike Sun. Spectral algorithms for tensor completion. *Communications on Pure and Applied Mathematics*, 71(11):2381–2425, 2018.

Cun Mu, Bo Huang, John Wright, and Donald Goldfarb. Square deal: Lower bounds and improved relaxations for tensor recovery. In *ICML*, pages 73–81, 2014.

Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

Ivan V Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33 (5):2295–2317, 2011.

Amelia Perry, Alexander S Wein, Afonso S Bandeira, et al. Statistical limits of spiked tensor models. *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, 56(1):230–264, 2020.

Michael Psenka and Nicolas Boumal. Second-order optimization for tensors with fixed tensor-train rank. *NeurIPS workshop, OPT2020*, 2020.

Garvesh Raskutti, Ming Yuan, and Han Chen. Convex regularization for high-dimensional multiresponse tensor regression. *The Annals of Statistics*, 47(3):1554–1584, 2019.

Holger Rauhut, Reinhold Schneider, and Zeljka Stojanac. Tensor completion in hierarchical tensor representations. In *Compressed sensing and its applications*, pages 419–450. Springer, 2015.

Holger Rauhut, Reinhold Schneider, and Zeljka Stojanac. Low rank tensor recovery via iterative hard thresholding. *Linear Algebra and its Applications*, 523:220–262, 2017.

Benjamin Recht, Maryam Fazel, and Pablo A Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review*, 52(3):471–501, 2010.

Emile Richard and Andrea Montanari. A statistical model for tensor pca. In *Advances in Neural Information Processing Systems*, pages 2897–2905, 2014.

Berkant Savas and Lek-Heng Lim. Quasi-Newton methods on Grassmannians and multilinear approximations of tensors. *SIAM Journal on Scientific Computing*, 32(6):3352–3393, 2010.

Jun Shao. *Mathematical statistics*. Springer Science & Business Media, 2006.

Michael Steinlechner. Riemannian optimization for high-dimensional tensor completion. *SIAM Journal on Scientific Computing*, 38(5):S461–S484, 2016.

Ruoyu Sun and Zhi-Quan Luo. Guaranteed matrix completion via nonconvex factorization. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 270–289. IEEE, 2015.

Will Wei Sun and Lexin Li. Store: sparse tensor response regression and neuroimaging analysis. *The Journal of Machine Learning Research*, 18(1):4908–4944, 2017.

Ryota Tomioka, Taiji Suzuki, Kohei Hayashi, and Hisashi Kashima. Statistical performance of convex tensor decomposition. In *Advances in Neural Information Processing Systems*, pages 972–980, 2011.

Tian Tong, Cong Ma, Ashley Prater-Bennette, Erin Tripp, and Yuejie Chi. Scaling and scalability: Provable nonconvex low-rank tensor estimation from incomplete measurements. *Journal of Machine Learning Research*, 23(163):1–77, 2022.

Stephen Tu, Ross Boczar, Max Simchowitz, Mahdi Soltanolkotabi, and Ben Recht. Low-rank solutions of linear matrix equations via Procrustes flow. In *International Conference on Machine Learning*, pages 964–973, 2016.

Ledyard R Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

André Uschmajew and Bart Vandereycken. The geometry of algorithms using hierarchical tensors. *Linear Algebra and its Applications*, 439(1):133–166, 2013.

André Uschmajew and Bart Vandereycken. Geometric methods on low-rank matrix and tensor manifolds. In *Handbook of Variational Methods for Nonlinear Geometric Data*, pages 261–313. Springer, 2020.

Bart Vandereycken. Low-rank matrix completion by Riemannian optimization. *SIAM Journal on Optimization*, 23(2):1214–1236, 2013.

Nick Vannieuwenhoven, Raf Vandebril, and Karl Meerbergen. A new truncation strategy for the higher-order singular value decomposition. *SIAM Journal on Scientific Computing*, 34(2):A1027–A1052, 2012.

Lingxiao Wang, Xiao Zhang, and Quanquan Gu. A unified computational and statistical framework for nonconvex low-rank matrix estimation. In *Artificial Intelligence and Statistics*, pages 981–990, 2017.

Ke Wei, Jian-Feng Cai, Tony F Chan, and Shingyu Leung. Guarantees of Riemannian optimization for low rank matrix recovery. *SIAM Journal on Matrix Analysis and Applications*, 37(3):1198–1222, 2016.

Dong Xia and Ming Yuan. On polynomial time methods for exact low-rank tensor completion. *Foundations of Computational Mathematics*, pages 1–49, 2017.

Dong Xia, Ming Yuan, and Cun-Hui Zhang. Statistically optimal and computationally efficient low rank tensor completion from noisy entries. *The Annals of Statistics*, 49(1), 2021.

Rose Yu and Yan Liu. Learning from multiway data: Simple and efficient tensor regression. In *International Conference on Machine Learning*, pages 373–381. PMLR, 2016.

Ming Yuan and Cun-Hui Zhang. On tensor completion via nuclear norm minimization. *Foundations of Computational Mathematics*, pages 1–38, 2014.

Anru Zhang and Dong Xia. Tensor svd: Statistical and computational limits. *IEEE Transactions on Information Theory*, 64(11):7311–7338, 2018.

Anru R Zhang, Yuetian Luo, Garvesh Raskutti, and Ming Yuan. ISLET: Fast and optimal low-rank tensor regression via importance sketching. *SIAM Journal on Mathematics of Data Science*, 2(2):444–479, 2020a.

Chenyu Zhang, Rungang Han, Anru R Zhang, and Paul M Voyles. Denoising atomic resolution 4d scanning transmission electron microscopy data with tensor singular value decomposition. *Ultramicroscopy*, 219:113123, 2020b.

Zhengwu Zhang, Genevera I Allen, Hongtu Zhu, and David Dunson. Tensor network factorizations: Relationships between brain structural connectomes and traits. *Neuroimage*, 197:330–343, 2019.

Tuo Zhao, Zhaoran Wang, and Han Liu. A nonconvex optimization framework for low rank matrix estimation. In *Advances in Neural Information Processing Systems*, pages 559–567, 2015.

Qinqing Zheng and John Lafferty. A convergent gradient descent algorithm for rank minimization and semidefinite programming from random linear measurements. In *Advances in Neural Information Processing Systems*, pages 109–117, 2015.

Hua Zhou. *Matlab TensorReg Toolbox Version 1.0*. Available online at https://hua-zhou.github.io/TensorReg/, 2017.

Hua Zhou, Lexin Li, and Hongtu Zhu. Tensor regression with applications in neuroimaging data analysis. *Journal of the American Statistical Association*, 108(502):540–552, 2013.

Yuchen Zhou, Anru R Zhang, Lili Zheng, and Yazhen Wang. Optimal high-order tensor svd via tensor-train orthogonal iteration. *IEEE Transactions on Information Theory*, 68(6): 3991–4019, 2022.