Scaling and Scalability: Provable Nonconvex Low-Rank Tensor Completion

Tian Tong
CMU

Cong Ma
UChicago

Ashley Prater-Bennette
AFRL

Erin Tripp
AFRL

Yuejie Chi
CMU

Abstract

Tensors, which provide a powerful and flexible model for representing multi-attribute data and multi-way interactions, play an indispensable role in modern data science across various fields in science and engineering. A fundamental task is tensor completion, which aims to faithfully recover the tensor from a small subset of its entries in a statistically and computationally efficient manner. Harnessing the low-rank structure of tensors in the Tucker decomposition, this paper develops a scaled gradient descent (ScaledGD) algorithm to directly recover the tensor factors with tailored spectral initializations, and shows that it provably converges at a linear rate independent of the condition number of the ground truth tensor for tensor completion as soon as the sample size is above the order of $n^{3/2}$ ignoring other parameter dependencies, where n is the dimension of the tensor. To the best of our knowledge, ScaledGD is the first algorithm that achieves near-optimal statistical and computational complexities simultaneously for low-rank tensor completion with the Tucker decomposition. Our algorithm highlights the power of appropriate preconditioning in accelerating nonconvex statistical estimation, where the iteration-varying preconditioners promote desirable invariance properties of the trajectory with respect to the underlying symmetry in low-rank tensor factorization.

Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151. Copyright 2022 by the author(s).

1 INTRODUCTION

Tensors (Kolda and Bader, 2009; Sidiropoulos et al., 2017), which provide a powerful and flexible model for representing multi-attribute data and multi-way interactions across various fields, play an indispensable role in modern data science with ubiquitous applications in image inpainting (Liu et al., 2012), hyperspectral imaging (Dian et al., 2017), collaborative filtering (Xiong et al., 2010), topic modeling (Anandkumar et al., 2014), network analysis (Papalexakis et al., 2016), and many more.

1.1 Low-rank tensor completion

In many problems across science and engineering, the central task can be regarded as tensor completion, where the goal is to estimate an order-3 tensor[1] $\boldsymbol{\mathcal{X}}_\star \in \mathbb{R}^{n_1\times n_2\times n_3}$ from only a small subset of its revealed entries in some index set Ω:

$$\boldsymbol{\mathcal{X}}_\star(i_1,i_2,i_3), \qquad (i_1,i_2,i_3)\in\Omega,$$

where $(i_1,i_2,i_3)\in\Omega$ if and only if that entry is observed. The goal is then to recover the tensor $\boldsymbol{\mathcal{X}}_\star$ from the observed entries in Ω. A celebrated application is collaborative filtering, where one aims to predict the users' evolving preferences from partial observations of a tensor composed of ratings for any triplet of *user, item, time* (Karatzoglou et al., 2010). Importantly, the number $|\Omega|$ of observations is often much smaller than the ambient dimension $n_1n_2n_3$ of the tensor due to resource or physical constraints, necessitating the need of exploiting low-dimensional structures to allow for meaningful recovery.

One of the most widely adopted low-dimensional structures—which is the focus of this paper—is the low-rank structure under the *Tucker* decomposition (Tucker, 1966). Specifically, we assume that the ground

[1] For ease of presentation, we focus on 3-way tensors; our algorithm and theory can be generalized to higher-order tensors in a straightforward manner.

truth tensor $\boldsymbol{\mathcal{X}}_\star$ admits the following Tucker decomposition[2]

$$\boldsymbol{\mathcal{X}}_\star = (\boldsymbol{U}_\star, \boldsymbol{V}_\star, \boldsymbol{W}_\star) \boldsymbol{\cdot} \boldsymbol{\mathcal{S}}_\star,$$

where $\boldsymbol{\mathcal{S}}_\star \in \mathbb{R}^{r_1 \times r_2 \times r_3}$ is the core tensor, and $\boldsymbol{U}_\star \in \mathbb{R}^{n_1 \times r_1}$, $\boldsymbol{V}_\star \in \mathbb{R}^{n_2 \times r_2}$, $\boldsymbol{W}_\star \in \mathbb{R}^{n_3 \times r_3}$ are orthonormal matrices corresponding to the factors of each mode. The tensor $\boldsymbol{\mathcal{X}}_\star$ is said to be low-multilinear-rank, or simply low-rank, when its multilinear rank $\boldsymbol{r} = (r_1, r_2, r_3)$ satisfies $r_k \ll n_k$, for all $k = 1, 2, 3$. Compared with other tensor decompositions such as the CP decomposition (Kolda and Bader, 2009) and tensor-SVD (Zhang et al., 2014), the Tucker decomposition offers several advantages: it allows flexible modeling of low-rank tensor factors with a small number of parameters, fully exploits the multi-dimensional algebraic structure of a tensor, and admits efficient and stable computation without suffering from degeneracy (Paatero, 2000).

1.2 A gradient descent approach?

Recent years remarkable successes have emerged in developing a plethora of provably efficient algorithms for low-rank *matrix* estimation (i.e. the special case of order-2 tensors) via both convex and nonconvex optimization. However, unique challenges arise when dealing with tensors, since they have more sophisticated algebraic structures (Hackbusch, 2012). For instance, while nuclear norm minimization achieves near-optimal statistical guarantees for low-rank matrix estimation (Candès and Tao, 2010) within a polynomial run time, computing the nuclear norm of a tensor turns out to be NP-hard (Friedland and Lim, 2018). Therefore, there have been a number of efforts to develop polynomial-time algorithms for tensor recovery, including but not limited to the sum-of-squares hierarchy (Barak and Moitra, 2016; Potechin and Steurer, 2017), nuclear norm minimization with unfolding (Gandy et al., 2011; Mu et al., 2014), regularized gradient descent (Han et al., 2020), to name a few; see Tong et al. (2021c) for further discussions.

In view of the low-rank Tucker decomposition, a natural approach is to seek to recover the factor quadruple $\boldsymbol{F}_\star \coloneqq (\boldsymbol{U}_\star, \boldsymbol{V}_\star, \boldsymbol{W}_\star, \boldsymbol{\mathcal{S}}_\star)$ directly by optimizing the unconstrained least-squares loss:

$$\min_{\boldsymbol{F}} \quad \mathcal{L}(\boldsymbol{F}) \coloneqq \frac{1}{2} \left\| \mathcal{P}_\Omega \left((\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}) \boldsymbol{\cdot} \boldsymbol{\mathcal{S}} - \boldsymbol{\mathcal{X}}_\star \right) \right\|_{\mathsf{F}}^2, \quad (1)$$

where $\mathcal{P}_\Omega : \mathbb{R}^{n_1 \times n_2 \times n_3} \mapsto \mathbb{R}^{n_1 \times n_2 \times n_3}$ is a projection operator that keeps the observed entries, $\boldsymbol{F} \coloneqq (\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}, \boldsymbol{\mathcal{S}})$ consists of $\boldsymbol{U} \in \mathbb{R}^{n_1 \times r_1}$, $\boldsymbol{V} \in \mathbb{R}^{n_2 \times r_2}$, $\boldsymbol{W} \in \mathbb{R}^{n_3 \times r_3}$, and $\boldsymbol{\mathcal{S}} \in \mathbb{R}^{r_1 \times r_2 \times r_3}$. Since the factors have a much lower complexity than the tensor itself due to the low-rank structure, it is expected that manipulating the factors results in more scalable algorithms in terms of both computation and storage. This optimization problem is however, highly nonconvex, since the factors are not uniquely determined.[3] Nonetheless, one might be tempted to solve the problem (1) via gradient descent (GD) and its variants due to their simplicity and amenability to parallel computing. Despite a flurry of activities for understanding factored gradient descent in the matrix setting (Chi et al., 2019), this line of algorithmic thinkings has been severely under-explored for the tensor setting, especially when it comes to provable guarantees for both sample and computational complexities. To the best of our knowledge, there is *no* provably linearly-convergent algorithm that accommodates low-rank tensor completion under the Tucker decomposition. The question is thus:

> *Can we develop a factored gradient-based algorithm that converges fast even for highly ill-conditioned tensors with near-optimal sample complexities for tensor completion under the Tucker decomposition?*

In this paper, we provide an affirmative answer to the above question.

1.3 A new algorithm: scaled gradient descent

We propose a novel algorithm—dubbed scaled gradient descent (ScaledGD)—to solve the tensor completion problem. More specifically, at the core it performs the following iterative updates[4] to minimize the loss function (1), with an additional low-complexity projection step:

$$\begin{aligned}
\boldsymbol{U}_{t+1} &= \boldsymbol{U}_t - \eta \nabla_{\boldsymbol{U}} \mathcal{L}(\boldsymbol{F}_t) \big(\breve{\boldsymbol{U}}_t^\top \breve{\boldsymbol{U}}_t \big)^{-1}, \\
\boldsymbol{V}_{t+1} &= \boldsymbol{V}_t - \eta \nabla_{\boldsymbol{V}} \mathcal{L}(\boldsymbol{F}_t) \big(\breve{\boldsymbol{V}}_t^\top \breve{\boldsymbol{V}}_t \big)^{-1}, \\
\boldsymbol{W}_{t+1} &= \boldsymbol{W}_t - \eta \nabla_{\boldsymbol{W}} \mathcal{L}(\boldsymbol{F}_t) \big(\breve{\boldsymbol{W}}_t^\top \breve{\boldsymbol{W}}_t \big)^{-1}, \\
\boldsymbol{\mathcal{S}}_{t+1} &= \boldsymbol{\mathcal{S}}_t - \eta \Big((\boldsymbol{U}_t^\top \boldsymbol{U}_t)^{-1}, \\
&\qquad (\boldsymbol{V}_t^\top \boldsymbol{V}_t)^{-1}, (\boldsymbol{W}_t^\top \boldsymbol{W}_t)^{-1} \Big) \boldsymbol{\cdot} \nabla_{\boldsymbol{\mathcal{S}}} \mathcal{L}(\boldsymbol{F}_t),
\end{aligned} \qquad (2)$$

where $\nabla_{\boldsymbol{U}} \mathcal{L}(\boldsymbol{F})$, $\nabla_{\boldsymbol{V}} \mathcal{L}(\boldsymbol{F})$, $\nabla_{\boldsymbol{W}} \mathcal{L}(\boldsymbol{F})$, and $\nabla_{\boldsymbol{\mathcal{S}}} \mathcal{L}(\boldsymbol{F})$ are the partial derivatives of $\mathcal{L}(\boldsymbol{F})$ with respect to the

[2] Other popular notation for Tucker decomposition in the literature includes $[\![\boldsymbol{\mathcal{S}}_\star; \boldsymbol{U}_\star, \boldsymbol{V}_\star, \boldsymbol{W}_\star]\!]$ and $\boldsymbol{\mathcal{S}}_\star \times_1 \boldsymbol{U}_\star \times_2 \boldsymbol{V}_\star \times_3 \boldsymbol{W}_\star$. In this work, we adopt the same notation $(\boldsymbol{U}_\star, \boldsymbol{V}_\star, \boldsymbol{W}_\star) \boldsymbol{\cdot} \boldsymbol{\mathcal{S}}_\star$ as in Xia and Yuan (2019) for convenience of our theoretical developments.

[3] $(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}) \boldsymbol{\cdot} \boldsymbol{\mathcal{S}} = (\boldsymbol{U}\boldsymbol{Q}_1, \boldsymbol{V}\boldsymbol{Q}_2, \boldsymbol{W}\boldsymbol{Q}_3) \boldsymbol{\cdot} ((\boldsymbol{Q}_1^{-1}, \boldsymbol{Q}_2^{-1}, \boldsymbol{Q}_3^{-1}) \boldsymbol{\cdot} \boldsymbol{\mathcal{S}})$, for any invertible matrices $\boldsymbol{Q}_k \in \mathbb{R}^{r_k \times r_k}$, $k = 1, 2, 3$.

[4] The matrix inverses in ScaledGD always exist under the assumptions of our theory.

corresponding variables, and

$$\begin{aligned}\breve{\boldsymbol{U}}_t &:= (\boldsymbol{W}_t \otimes \boldsymbol{V}_t)\mathcal{M}_1(\boldsymbol{\mathcal{S}}_t)^\top,\\ \breve{\boldsymbol{V}}_t &:= (\boldsymbol{W}_t \otimes \boldsymbol{U}_t)\mathcal{M}_2(\boldsymbol{\mathcal{S}}_t)^\top,\\ \breve{\boldsymbol{W}}_t &:= (\boldsymbol{V}_t \otimes \boldsymbol{U}_t)\mathcal{M}_3(\boldsymbol{\mathcal{S}}_t)^\top.\end{aligned} \tag{3}$$

Here, $\mathcal{M}_k(\boldsymbol{\mathcal{S}})$ is the matricization of the tensor $\boldsymbol{\mathcal{S}}$ along the k-th mode (see Tong et al. (2021c) for details), for $k = 1, 2, 3$, and $\otimes$ denotes the Kronecker product. ScaledGD exploits the structures of Tucker decomposition and possesses many desirable properties:

- *Low per-iteration cost:* as a preconditioned GD or quasi-Newton algorithm, ScaledGD updates the factors along the descent direction of a scaled gradient, where the preconditioners can be viewed as the inverse of the diagonal blocks of the Hessian for the population loss (i.e. tensor factorization). As the sizes of the preconditioners are proportional to the multilinear rank, the matrix inverses are cheap to compute with a minimal overhead and the overall per-iteration cost is still low and linear in the time it takes to read the input data.

- *Equivariance to parameterization:* one crucial property of ScaledGD is that if we reparameterize the factors by some invertible transforms (i.e. replacing $(\boldsymbol{U}_t, \boldsymbol{V}_t, \boldsymbol{W}_t, \boldsymbol{\mathcal{S}}_t)$ by $(\boldsymbol{U}_t\boldsymbol{Q}_1, \boldsymbol{V}_t\boldsymbol{Q}_2, \boldsymbol{W}_t\boldsymbol{Q}_3, (\boldsymbol{Q}_1^{-1}, \boldsymbol{Q}_2^{-1}, \boldsymbol{Q}_3^{-1}) \cdot \boldsymbol{\mathcal{S}}_t)$ for some invertible matrices $\{\boldsymbol{Q}_k\}_{k=1}^3$), the entire trajectory will go through the same reparameterization, leading to an *invariant* sequence of low-rank tensor updates $\boldsymbol{\mathcal{X}}_t = (\boldsymbol{U}_t, \boldsymbol{V}_t, \boldsymbol{W}_t) \cdot \boldsymbol{\mathcal{S}}_t$ regardless of the parameterization being adopted.

- *Implicit balancing:* ScaledGD optimizes the natural loss function (1) in an *unconstrained* manner without requiring additional regularizations or orthogonalizations used in prior literature (Han et al., 2020; Frandsen and Ge, 2020; Kasai and Mishra, 2016), and the factors stay balanced in an automatic manner—a feature sometimes referred to as implicit regularization (Ma et al., 2021).

Theoretical guarantees. We investigate the theoretical properties of ScaledGD for tensor completion, which are notably more challenging than the matrix variant (Tong et al., 2021a). It is demonstrated that ScaledGD—when initialized properly using appropriate spectral methods —achieves linear convergence at a rate *independent* of the condition number of the ground truth tensor with near-optimal sample complexities. In other words, ScaledGD needs no more than $O(\log(1/\varepsilon))$ iterations to reach ε-accuracy; together with its low computational and memory costs by operating in the factor space, this makes ScaledGD a highly scalable method. More specifically, under the Bernoulli sampling model, ScaledGD succeeds with high probability as long as the sample complexity is above the order of $n^{3/2}r^{5/2}\kappa^3 \log^3 n$, where $n = \max_{k=1,2,3} n_k$, $r = \max_{k=1,2,3} r_k$, and κ is a sort of condition number of $\boldsymbol{\mathcal{X}}_\star$. Connected to some well-reckoned conjecture on computational barriers, it is widely believed that no polynomial-time algorithm will be successful if the sample complexity is less than the order of $n^{3/2}$ for tensor completion (Barak and Moitra, 2016), which suggests the near-optimality of the sample complexity of ScaledGD. Compared with existing approaches (cf. Table 1), ScaledGD provides the first computationally efficient algorithm with a near-linear run time at the near-optimal sample complexity.

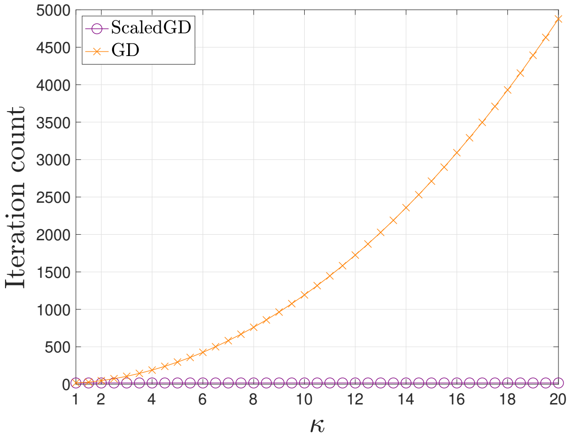

Figure 1: The iteration complexities of ScaledGD (this paper) and GD to achieve $\|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{X}}_\star\|_\mathsf{F} \le 10^{-3}\|\boldsymbol{\mathcal{X}}_\star\|_\mathsf{F}$ with respect to different condition numbers for low-rank tensor completion with $n_1 = n_2 = n_3 = 100$, $r_1 = r_2 = r_3 = 5$, and the probability of observation $p = 0.1$.

It is worth highlighting that, the scaling in ScaledGD plays a crucial role to achieve a fast linear convergence rate which is insensitive to the condition number, by contrasting with the GD algorithm (Han et al., 2020). Figure 1 illustrates the number of iterations needed to achieve a relative error $\|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{X}}_\star\|_\mathsf{F} \le 10^{-3}\|\boldsymbol{\mathcal{X}}_\star\|_\mathsf{F}$ for ScaledGD in comparison to the GD algorithm under different condition numbers for tensor completion under the Bernoulli sampling model to illustrate the benefit of scaling. Clearly, the iteration complexity of GD deteriorates at a super linear rate with respect to the condition number κ, while ScaledGD enjoys an iteration complexity that is independent of κ as predicted by our theory. Indeed, with a seemingly small modification, ScaledGD takes merely 17 iterations to achieve the desired accuracy over the entire range of κ, while GD takes thousands of iterations even with a moderate condition number!

Algorithms	Sample complexity	Iteration complexity	Parameter space
Unfolding + nuclear norm min. (Huang et al., 2015)	$n^2 r \log^2 n$	polynomial	tensor
Tensor nuclear norm min. (Yuan and Zhang, 2016)	$n^{3/2} r^{1/2} \log^{3/2} n$	NP-hard	tensor
Grassmannian GD (Xia and Yuan, 2019)	$n^{3/2} r^{7/2} \kappa^4 \log^{7/2} n$	N/A	factor
ScaledGD (this paper)	$n^{3/2} r^{5/2} \kappa^3 \log^3 n$	$\log \frac{1}{\varepsilon}$	factor

Table 1: Comparisons of ScaledGD with existing algorithms for tensor completion when the tensor is incoherent and low-rank under the Tucker decomposition. Here, we say that the output $\boldsymbol{\mathcal{X}}$ of an algorithm reaches ε-accuracy, if it satisfies $\|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{X}}_\star\|_{\mathsf{F}} \le \varepsilon \sigma_{\min}(\boldsymbol{\mathcal{X}}_\star)$. Here, κ and $\sigma_{\min}(\boldsymbol{\mathcal{X}}_\star)$ are the condition number and the minimum singular value of $\boldsymbol{\mathcal{X}}_\star$ (defined in Section 2.1). For simplicity, we let $n = \max_{k=1,2,3} n_k$ and $r = \max_{k=1,2,3} r_k$, and assume $r \vee \kappa \ll n^\delta$ for some small constant δ to keep only terms with dominating orders of n.

1.4 Comparison with Tong et al. (2021a)

While the proposed ScaledGD algorithm is inspired by its matrix variant in Tong et al. (2021a) by utilizing the same principle of preconditioning, the exact form of preconditioning for tensor factorization needs to be designed carefully and is not trivially obtainable. There are many technical novelty in our analysis compared to Tong et al. (2021a). In the matrix case, the low-rank matrix is factorized as $\boldsymbol{L}\boldsymbol{R}^\top$, and only two factors are needed to be estimated. In contrast, in the tensor case, the low-rank tensor is factorized as $(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}) \boldsymbol{\cdot} \boldsymbol{\mathcal{S}}$, and four factors are needed to be estimated, leading to a much more complicated nonconvex landscape than the matrix case. In fact, when specialized to matrix completion, our ScaledGD algorithm does not degenerate to the same matrix variant in Tong et al. (2021a), due to overparamterization and estimating four factors at once, but still maintains the near-optimal performance guarantees. In addition, the tensor algebra possesses unique algebraic properties that requires much more delicate treatments in the analysis. For the local convergence, we establish new concentration properties regarding tensors, which are more challenging compared to the matrix counterparts; for spectral initialization, we establish the effectiveness of a second-order spectral method in the Tucker setting for the first time.

Due to space limits, we refer detailed related work and tensor algebra as well as notation to Tong et al. (2021c).

2 MAIN RESULTS

2.1 Models and assumptions

We assume the ground truth tensor $\boldsymbol{\mathcal{X}}_\star = [\boldsymbol{\mathcal{X}}_\star(i_1, i_2, i_3)] \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ admits the following Tucker decomposition

$$\boldsymbol{\mathcal{X}}_\star(i_1, i_2, i_3) = \sum_{j_1=1}^{r_1} \sum_{j_2=1}^{r_2} \sum_{j_3=1}^{r_3} \boldsymbol{U}_\star(i_1, j_1) \boldsymbol{V}_\star(i_2, j_2) \boldsymbol{W}_\star(i_3, j_3) \boldsymbol{\mathcal{S}}_\star(j_1, j_2, j_3), \quad 1 \le i_k \le n_k, \quad (4)$$

or more compactly,

$$\boldsymbol{\mathcal{X}}_\star = (\boldsymbol{U}_\star, \boldsymbol{V}_\star, \boldsymbol{W}_\star) \boldsymbol{\cdot} \boldsymbol{\mathcal{S}}_\star, \quad (5)$$

where $\boldsymbol{\mathcal{S}}_\star = [\boldsymbol{\mathcal{S}}_\star(j_1, j_2, j_3)] \in \mathbb{R}^{r_1 \times r_2 \times r_3}$ is the core tensor of multilinear rank $\boldsymbol{r} = (r_1, r_2, r_3)$, and $\boldsymbol{U}_\star = [\boldsymbol{U}_\star(i_1, j_1)] \in \mathbb{R}^{n_1 \times r_1}$, $\boldsymbol{V}_\star = [\boldsymbol{V}_\star(i_2, j_2)] \in \mathbb{R}^{n_2 \times r_2}$, $\boldsymbol{W}_\star = [\boldsymbol{W}_\star(i_3, j_3)] \in \mathbb{R}^{n_3 \times r_3}$ are the factor matrices of each mode. Letting $\mathcal{M}_k(\boldsymbol{\mathcal{X}}_\star)$ be the mode-k matricization of $\boldsymbol{\mathcal{X}}_\star$, we have

$$\mathcal{M}_1(\boldsymbol{\mathcal{X}}_\star) = \boldsymbol{U}_\star \mathcal{M}_1(\boldsymbol{\mathcal{S}}_\star)(\boldsymbol{W}_\star \otimes \boldsymbol{V}_\star)^\top, \quad (6a)$$

$$\mathcal{M}_2(\boldsymbol{\mathcal{X}}_\star) = \boldsymbol{V}_\star \mathcal{M}_2(\boldsymbol{\mathcal{S}}_\star)(\boldsymbol{W}_\star \otimes \boldsymbol{U}_\star)^\top, \quad (6b)$$

$$\mathcal{M}_3(\boldsymbol{\mathcal{X}}_\star) = \boldsymbol{W}_\star \mathcal{M}_3(\boldsymbol{\mathcal{S}}_\star)(\boldsymbol{V}_\star \otimes \boldsymbol{U}_\star)^\top. \quad (6c)$$

It is straightforward to see that the Tucker decomposition is not uniquely specified: for any invertible matrices $\boldsymbol{Q}_k \in \mathbb{R}^{r_k \times r_k}$, $k = 1, 2, 3$, one has $(\boldsymbol{U}_\star, \boldsymbol{V}_\star, \boldsymbol{W}_\star) \boldsymbol{\cdot} \boldsymbol{\mathcal{S}}_\star = (\boldsymbol{U}_\star \boldsymbol{Q}_1, \boldsymbol{V}_\star \boldsymbol{Q}_2, \boldsymbol{W}_\star \boldsymbol{Q}_3) \boldsymbol{\cdot} ((\boldsymbol{Q}_1^{-1}, \boldsymbol{Q}_2^{-1}, \boldsymbol{Q}_3^{-1}) \boldsymbol{\cdot} \boldsymbol{\mathcal{S}}_\star)$. We shall fix the ground truth factor such that $\boldsymbol{U}_\star$, $\boldsymbol{V}_\star$ and $\boldsymbol{W}_\star$ are orthonormal matrices consisting of left singular vectors of each mode. Furthermore, the core tensor $\boldsymbol{\mathcal{S}}_\star$ is related to the singular values in each mode as

$$\mathcal{M}_k(\boldsymbol{\mathcal{S}}_\star) \mathcal{M}_k(\boldsymbol{\mathcal{S}}_\star)^\top = \boldsymbol{\Sigma}_{\star,k}^2, \qquad k = 1, 2, 3, \quad (7)$$

where $\boldsymbol{\Sigma}_{\star,k} := \mathrm{diag}[\sigma_1(\mathcal{M}_k(\boldsymbol{\mathcal{X}}_\star)), \ldots, \sigma_{r_k}(\mathcal{M}_k(\boldsymbol{\mathcal{X}}_\star))]$ is a diagonal matrix whose diagonal elements are the nonzero singular values of $\mathcal{M}_k(\boldsymbol{\mathcal{X}}_\star)$ and $r_k = \mathrm{rank}(\mathcal{M}_k(\boldsymbol{\mathcal{X}}_\star))$ for $k = 1, 2, 3$.

Key parameters. Of particular interest is a sort of condition number of $\boldsymbol{\mathcal{X}}_\star$, which plays an important role in governing the computational efficiency of first-order algorithms.

Definition 1 (Condition number)**.** The condition number of $\boldsymbol{\mathcal{X}}_\star$ is defined as

$$\kappa := \frac{\sigma_{\max}(\boldsymbol{\mathcal{X}}_\star)}{\sigma_{\min}(\boldsymbol{\mathcal{X}}_\star)} = \frac{\max_{k=1,2,3}\sigma_1(\mathcal{M}_k(\boldsymbol{\mathcal{X}}_\star))}{\min_{k=1,2,3}\sigma_{r_k}(\mathcal{M}_k(\boldsymbol{\mathcal{X}}_\star))}. \tag{8}$$

Another parameter is the incoherence parameter, which is crucial in determining the well-posedness of low-rank tensor completion.

Definition 2 (Incoherence)**.** The incoherence parameter of $\boldsymbol{\mathcal{X}}_\star$ is defined as

$$\mu := \max\left\{\frac{n_1}{r_1}\|\boldsymbol{U}_\star\|_{2,\infty}^2, \frac{n_2}{r_2}\|\boldsymbol{V}_\star\|_{2,\infty}^2, \frac{n_3}{r_3}\|\boldsymbol{W}_\star\|_{2,\infty}^2\right\}. \tag{9}$$

Roughly speaking, a small incoherence parameter ensures that the energy of the tensor is evenly distributed across its entries, so that a small random subset of its elements still reveals substantial information about the latent structure of the entire tensor.

2.2 A warm-up case: ScaledGD for tensor factorization

To shed light on the design insights, we now introduce the ScaledGD algorithm for the tensor factorization problem, which aims to minimize the following loss function:

$$\mathcal{L}(\boldsymbol{F}) := \frac{1}{2}\|(\boldsymbol{U},\boldsymbol{V},\boldsymbol{W})\boldsymbol{\cdot}\boldsymbol{\mathcal{S}} - \boldsymbol{\mathcal{X}}_\star\|_{\mathsf{F}}^2. \tag{11}$$

Recalling the update rule (2), ScaledGD proceeds as

$$\begin{aligned}
\boldsymbol{U}_{t+1} &= \boldsymbol{U}_t - \eta\mathcal{M}_1\left(\boldsymbol{\mathcal{X}}_t - \boldsymbol{\mathcal{X}}_\star\right)\breve{\boldsymbol{U}}_t^\top\left(\breve{\boldsymbol{U}}_t^\top\breve{\boldsymbol{U}}_t\right)^{-1},\\
\boldsymbol{V}_{t+1} &= \boldsymbol{V}_t - \eta\mathcal{M}_2\left(\boldsymbol{\mathcal{X}}_t - \boldsymbol{\mathcal{X}}_\star\right)\breve{\boldsymbol{V}}_t^\top\left(\breve{\boldsymbol{V}}_t^\top\breve{\boldsymbol{V}}_t\right)^{-1},\\
\boldsymbol{W}_{t+1} &= \boldsymbol{W}_t - \eta\mathcal{M}_3\left(\boldsymbol{\mathcal{X}}_t - \boldsymbol{\mathcal{X}}_\star\right)\breve{\boldsymbol{W}}_t^\top\left(\breve{\boldsymbol{W}}_t^\top\breve{\boldsymbol{W}}_t\right)^{-1},\\
\boldsymbol{\mathcal{S}}_{t+1} &= \boldsymbol{\mathcal{S}}_t - \eta\Big((\boldsymbol{U}_t^\top\boldsymbol{U}_t)^{-1}\boldsymbol{U}_t^\top,\\
&\qquad (\boldsymbol{V}_t^\top\boldsymbol{V}_t)^{-1}\boldsymbol{V}_t^\top, (\boldsymbol{W}_t^\top\boldsymbol{W}_t)^{-1}\boldsymbol{W}_t^\top\Big)\boldsymbol{\cdot}\left(\boldsymbol{\mathcal{X}}_t - \boldsymbol{\mathcal{X}}_\star\right),
\end{aligned} \tag{12}$$

where $\boldsymbol{\mathcal{X}}_t = (\boldsymbol{U}_t,\boldsymbol{V}_t,\boldsymbol{W}_t)\boldsymbol{\cdot}\boldsymbol{\mathcal{S}}_t$, with $\breve{\boldsymbol{U}}_t$, $\breve{\boldsymbol{V}}_t$, and $\breve{\boldsymbol{W}}_t$ defined in (3).

ScaledGD as a quasi-Newton algorithm. One way to think of ScaledGD is through the lens of quasi-Newton methods. We can equivalently rewrite the ScaledGD update (12) as

$$\mathrm{vec}(\boldsymbol{F}_{t+1}) = \mathrm{vec}(\boldsymbol{F}_t) - \eta\boldsymbol{H}_t^{-1}\nabla_{\mathrm{vec}(\boldsymbol{F})}\mathcal{L}(\boldsymbol{F}_t), \tag{13}$$

where the diagonal blocks of the Hessian of the loss function (11) are given precisely as

$$\begin{aligned}
\boldsymbol{H}_t &:= \mathrm{diag}\left[\nabla^2_{\mathrm{vec}(\boldsymbol{U}),\mathrm{vec}(\boldsymbol{U})}\mathcal{L}(\boldsymbol{F}_t), \nabla^2_{\mathrm{vec}(\boldsymbol{V}),\mathrm{vec}(\boldsymbol{V})}\mathcal{L}(\boldsymbol{F}_t),\right.\\
&\qquad \left.\nabla^2_{\mathrm{vec}(\boldsymbol{W}),\mathrm{vec}(\boldsymbol{W})}\mathcal{L}(\boldsymbol{F}_t), \nabla^2_{\mathrm{vec}(\boldsymbol{\mathcal{S}}),\mathrm{vec}(\boldsymbol{\mathcal{S}})}\mathcal{L}(\boldsymbol{F}_t)\right]\\
&= \mathrm{diag}\left[(\breve{\boldsymbol{U}}_t^\top\breve{\boldsymbol{U}}_t)\otimes\boldsymbol{I}_{n_1}, (\breve{\boldsymbol{V}}_t^\top\breve{\boldsymbol{V}}_t)\otimes\boldsymbol{I}_{n_2},\right.\\
&\qquad \left.(\breve{\boldsymbol{W}}_t^\top\breve{\boldsymbol{W}}_t)\otimes\boldsymbol{I}_{n_3}, (\boldsymbol{W}_t^\top\boldsymbol{W}_t)\otimes(\boldsymbol{V}_t^\top\boldsymbol{V}_t)\otimes(\boldsymbol{U}_t^\top\boldsymbol{U}_t)\right].
\end{aligned}$$

Therefore, by vectorization of (12), ScaledGD can be regarded as a quasi-Newton method where the preconditioner is designed as the inverse of the diagonal approximation of the Hessian.

2.3 ScaledGD for tensor completion

Assume that we have observed a subset of entries in $\boldsymbol{\mathcal{X}}_\star$, given as $\boldsymbol{\mathcal{Y}} = \mathcal{P}_\Omega(\boldsymbol{\mathcal{X}}_\star)$, where $\mathcal{P}_\Omega : \mathbb{R}^{n_1\times n_2\times n_3} \mapsto \mathbb{R}^{n_1\times n_2\times n_3}$ is a projection operator such that

$$[\mathcal{P}_\Omega(\boldsymbol{\mathcal{X}}_\star)](i_1,i_2,i_3) = \begin{cases}\boldsymbol{\mathcal{X}}_\star(i_1,i_2,i_3), & \text{if } (i_1,i_2,i_3)\in\Omega,\\ 0, & \text{otherwise.}\end{cases} \tag{14}$$

Here, Ω is generated according to the Bernoulli observation model in the sense that

$$(i_1,i_2,i_3)\in\Omega \quad \text{i.i.d. with prob. } p\in(0,1]. \tag{15}$$

The goal of tensor completion is to recover the tensor $\boldsymbol{\mathcal{X}}_\star$ from its partial observation $\mathcal{P}_\Omega(\boldsymbol{\mathcal{X}}_\star)$. This can be achieved by minimizing the loss function

$$\min_{\boldsymbol{F}=(\boldsymbol{U},\boldsymbol{V},\boldsymbol{W},\boldsymbol{\mathcal{S}})} \mathcal{L}(\boldsymbol{F}) := \frac{1}{2p}\left\|\mathcal{P}_\Omega\big((\boldsymbol{U},\boldsymbol{V},\boldsymbol{W})\boldsymbol{\cdot}\boldsymbol{\mathcal{S}}\big) - \boldsymbol{\mathcal{Y}}\right\|_{\mathsf{F}}^2. \tag{16}$$

Preparation: a scaled projection operator. To guarantee faithful recovery from partial observations, the underlying low-rank tensor $\boldsymbol{\mathcal{X}}_\star$ needs to be incoherent (cf. Definition 2) to avoid ill-posedness. One typical strategy, frequently employed in the matrix setting, to ensure the incoherence condition is to trim the rows of the factors (Chen and Wainwright, 2015) after the gradient update. For ScaledGD, this needs to be done in a careful manner to preserve the equivariance with respect to invertible transforms. Motivated by Tong et al. (2021a), we introduce the scaled projection as follows,

$$(\boldsymbol{U},\boldsymbol{V},\boldsymbol{W},\boldsymbol{\mathcal{S}}) = \mathcal{P}_B(\boldsymbol{U}_+,\boldsymbol{V}_+,\boldsymbol{W}_+,\boldsymbol{\mathcal{S}}_+), \tag{17a}$$

where $B>0$ is the projection radius, and

$$\begin{aligned}
\boldsymbol{U}(i_1,:) &= \left(1\wedge\frac{B}{\sqrt{n_1}\|\boldsymbol{U}_+(i_1,:)\breve{\boldsymbol{U}}_+^\top\|_2}\right)\boldsymbol{U}_+(i_1,:);\\
\boldsymbol{V}(i_2,:) &= \left(1\wedge\frac{B}{\sqrt{n_2}\|\boldsymbol{V}_+(i_2,:)\breve{\boldsymbol{V}}_+^\top\|_2}\right)\boldsymbol{V}_+(i_2,:);
\end{aligned} \tag{17b}$$

```
Algorithm 1 ScaledGD for low-rank tensor completion
  Input parameters: step size $\eta$, rank $\boldsymbol{r} = (r_1, r_2, r_3)$, probability of observation $p$, projection radius $B$.
  Spectral initialization: Let $\boldsymbol{U}_+$ be the top-$r_1$ eigenvectors of $\mathcal{P}_{\textsf{off-diag}}(p^{-2}\mathcal{M}_1(\boldsymbol{\mathcal{Y}})\mathcal{M}_1(\boldsymbol{\mathcal{Y}})^\top)$, and similarly
  for $\boldsymbol{V}_+, \boldsymbol{W}_+$, and $\boldsymbol{\mathcal{S}}_+ = p^{-1}(\boldsymbol{U}_+^\top, \boldsymbol{V}_+^\top, \boldsymbol{W}_+^\top)\boldsymbol{\cdot}\boldsymbol{\mathcal{Y}}$. Set $(\boldsymbol{U}_0, \boldsymbol{V}_0, \boldsymbol{W}_0, \boldsymbol{\mathcal{S}}_0) = \mathcal{P}_B\big(\boldsymbol{U}_+, \boldsymbol{V}_+, \boldsymbol{W}_+, \boldsymbol{\mathcal{S}}_+\big)$.
  Scaled projected gradient updates: for $t = 0, 1, 2, \dots, T-1$ do
      $\boldsymbol{U}_{t+} = \boldsymbol{U}_t - \frac{\eta}{p}\mathcal{M}_1\left(\mathcal{P}_\Omega\big((\boldsymbol{U}_t, \boldsymbol{V}_t, \boldsymbol{W}_t)\boldsymbol{\cdot}\boldsymbol{\mathcal{S}}_t\big) - \boldsymbol{\mathcal{Y}}\right)\breve{\boldsymbol{U}}_t\big(\breve{\boldsymbol{U}}_t^\top\breve{\boldsymbol{U}}_t\big)^{-1},$
      $\boldsymbol{V}_{t+} = \boldsymbol{V}_t - \frac{\eta}{p}\mathcal{M}_2\left(\mathcal{P}_\Omega\big((\boldsymbol{U}_t, \boldsymbol{V}_t, \boldsymbol{W}_t)\boldsymbol{\cdot}\boldsymbol{\mathcal{S}}_t\big) - \boldsymbol{\mathcal{Y}}\right)\breve{\boldsymbol{V}}_t\big(\breve{\boldsymbol{V}}_t^\top\breve{\boldsymbol{V}}_t\big)^{-1},$
      $\boldsymbol{W}_{t+} = \boldsymbol{W}_t - \frac{\eta}{p}\mathcal{M}_3\left(\mathcal{P}_\Omega\big((\boldsymbol{U}_t, \boldsymbol{V}_t, \boldsymbol{W}_t)\boldsymbol{\cdot}\boldsymbol{\mathcal{S}}_t\big) - \boldsymbol{\mathcal{Y}}\right)\breve{\boldsymbol{W}}_t\big(\breve{\boldsymbol{W}}_t^\top\breve{\boldsymbol{W}}_t\big)^{-1},$      (10)
      $\boldsymbol{\mathcal{S}}_{t+} = \boldsymbol{\mathcal{S}}_t - \frac{\eta}{p}\left((\boldsymbol{U}_t^\top\boldsymbol{U}_t)^{-1}\boldsymbol{U}_t^\top, (\boldsymbol{V}_t^\top\boldsymbol{V}_t)^{-1}\boldsymbol{V}_t^\top, (\boldsymbol{W}_t^\top\boldsymbol{W}_t)^{-1}\boldsymbol{W}_t^\top\right)\boldsymbol{\cdot}\left(\mathcal{P}_\Omega\big((\boldsymbol{U}_t, \boldsymbol{V}_t, \boldsymbol{W}_t)\boldsymbol{\cdot}\boldsymbol{\mathcal{S}}_t\big) - \boldsymbol{\mathcal{Y}}\right),$
  where $\breve{\boldsymbol{U}}_t$, $\breve{\boldsymbol{V}}_t$, and $\breve{\boldsymbol{W}}_t$ are defined in (3). Set $(\boldsymbol{U}_{t+1}, \boldsymbol{V}_{t+1}, \boldsymbol{W}_{t+1}, \boldsymbol{\mathcal{S}}_{t+1}) = \mathcal{P}_B(\boldsymbol{U}_{t+}, \boldsymbol{V}_{t+}, \boldsymbol{W}_{t+}, \boldsymbol{\mathcal{S}}_{t+})$.
```

$$\boldsymbol{W}(i_3,:) = \left(1 \wedge \frac{B}{\sqrt{n_3}\|\boldsymbol{W}_+(i_3,:)\breve{\boldsymbol{W}}_+^\top\|_2}\right)\boldsymbol{W}_+(i_3,:);$$
$$\boldsymbol{\mathcal{S}} = \boldsymbol{\mathcal{S}}_+,$$

for $1 \le i_1 \le n_1$, $1 \le i_2 \le n_2$, and $1 \le i_3 \le n_3$. Here, $\breve{\boldsymbol{U}}_+$, $\breve{\boldsymbol{V}}_+$, $\breve{\boldsymbol{W}}_+$ are analogously defined in (3) using $(\boldsymbol{U}_+, \boldsymbol{V}_+, \boldsymbol{W}_+, \boldsymbol{\mathcal{S}}_+)$. As can be seen, each row of $\boldsymbol{U}_+$ (resp. $\boldsymbol{V}_+$ and $\boldsymbol{W}_+$) is scaled by a scalar based on the row ℓ_2 norms of $\boldsymbol{U}_+\breve{\boldsymbol{U}}_+^\top$ (resp. $\boldsymbol{V}_+\breve{\boldsymbol{V}}_+^\top$ and $\boldsymbol{W}_+\breve{\boldsymbol{W}}_+^\top$), which is the mode-1 (resp. mode-2 and mode-3) matricization of the tensor $(\boldsymbol{U}_+, \boldsymbol{V}_+, \boldsymbol{W}_+)\boldsymbol{\cdot}\boldsymbol{\mathcal{S}}_+$. It is a straightforward observation that $\mathcal{P}_B$ can be computed efficiently.

Algorithm description. With the scaled projection $\mathcal{P}_B(\cdot)$ defined in hand, we are in position to describe the details of the proposed ScaledGD algorithm, summarized in Algorithm 1. It consists of two stages: spectral initialization followed by iterative refinements using the scaled projected gradient updates in (10). It is worth emphasizing that all the factors are updated simultaneously, which can be achieved in a parallel manner to accelerate run time.

For spectral initialization, we take advantage of the subspace estimators proposed in Cai et al. (2021) for highly unbalanced data matrices. Specifically, we estimate the subspace spanned by $\boldsymbol{U}_\star$ by that spanned by top-r_1 eigenvectors $\boldsymbol{U}_+$ of the diagonally-deleted Gram matrix of $p^{-1}\mathcal{M}_1(\boldsymbol{\mathcal{Y}})$:

$$\mathcal{P}_{\textsf{off-diag}}(p^{-2}\mathcal{M}_1(\boldsymbol{\mathcal{Y}})\mathcal{M}_1(\boldsymbol{\mathcal{Y}})^\top),$$

where $\mathcal{P}_{\textsf{off-diag}}(\boldsymbol{M})$ sets the diagonal entries of the matrix $\boldsymbol{M}$ as zeros; the other two factors $\boldsymbol{V}_+$ and $\boldsymbol{W}_+$ are estimated similarly. The core tensor is then estimated via

$$\boldsymbol{\mathcal{S}}_+ = p^{-1}(\boldsymbol{U}_+^\top, \boldsymbol{V}_+^\top, \boldsymbol{W}_+^\top)\boldsymbol{\cdot}\boldsymbol{\mathcal{Y}}.$$

To ensure the initialization is incoherent, we pass it through the scaled projection operator to obtain the final initial estimate:

$$(\boldsymbol{U}_0, \boldsymbol{V}_0, \boldsymbol{W}_0, \boldsymbol{\mathcal{S}}_0) = \mathcal{P}_B\big(\boldsymbol{U}_+, \boldsymbol{V}_+, \boldsymbol{W}_+, \boldsymbol{\mathcal{S}}_+\big).$$

2.4 Theoretical guarantees

The following theorem establishes the performance guarantee of ScaledGD for tensor completion, as soon as the sample size is sufficiently large.

Theorem 1 (ScaledGD for tensor completion)**.** *Suppose that $\boldsymbol{\mathcal{X}}_\star$ is μ-incoherent, and that p satisfies*

$$pn_1n_2n_3 \gtrsim \epsilon_0^{-2}\mu^{3/2}r^{5/2}\kappa^3\sqrt{n_1n_2n_3}\log^3 n \\ +\epsilon_0^{-4}\mu^3r^4\kappa^6 n\log^5 n$$

for some small constant $\epsilon_0 > 0$. Set the projection radius as $B = C_B\sqrt{\mu r}\sigma_{\max}(\boldsymbol{\mathcal{X}}_\star)$ for some constant $C_B \ge (1+\epsilon_0)^3$. If the step size obeys $0 < \eta \le 2/5$, then with probability at least $1 - c_1n^{-c_2}$ for universal constants $c_1, c_2 > 0$, for all $t \ge 0$, the iterates of Algorithm 1 satisfy

$$\|(\boldsymbol{U}_t, \boldsymbol{V}_t, \boldsymbol{W}_t)\boldsymbol{\cdot}\boldsymbol{\mathcal{S}}_t - \boldsymbol{\mathcal{X}}_\star\|_{\mathsf{F}} \le 3\epsilon_0(1-0.6\eta)^t\sigma_{\min}(\boldsymbol{\mathcal{X}}_\star).$$

Theorem 1 ensures that ScaledGD finds an ε-accurate estimate, i.e. $\|(\boldsymbol{U}_t, \boldsymbol{V}_t, \boldsymbol{W}_t)\boldsymbol{\cdot}\boldsymbol{\mathcal{S}}_t - \boldsymbol{\mathcal{X}}_\star\|_{\mathsf{F}} \le \varepsilon\sigma_{\min}(\boldsymbol{\mathcal{X}}_\star)$, in at most $O(\log(1/\varepsilon))$ iterations, which is *independent* of the condition number of $\boldsymbol{\mathcal{X}}_\star$, as long as the sample complexity is large enough. Assuming that $\mu = O(1)$ and $r \vee \kappa \ll n^\delta$ for some small constant δ to keep only terms with dominating orders of n, the sample complexity simplifies to

$$pn_1n_2n_3 \gtrsim n^{3/2}r^{5/2}\kappa^3\log^3 n,$$

which is near-optimal in view of the conjecture that no polynomial-time algorithm will be successful if the sample complexity is less than the order of $n^{3/2}$ for tensor completion (Barak and Moitra, 2016). Compared with existing algorithms collected in Table 1, ScaledGD is the *first* algorithm that simultaneously achieves a near-optimal sample complexity and a near-linear run time complexity for tensor completion in a provable manner. In particular, while Yuan and Zhang (2016); Xia and Yuan (2019) achieve a sample complexity comparable to ours, the tensor nuclear norm minimization algorithm in Yuan and Zhang (2016) is NP-hard to compute, and the Grassmannian GD algorithm in Xia and Yuan (2019) does not offer an explicit iteration complexity, except that each iteration can be computed in polynomial time.

3 ANALYSIS

In this section, we provide some intuitions and sketch the proof of our main theorem. Before continuing, we highlight an important property of ScaledGD: if starting from an equivalent estimate

$$\widetilde{\boldsymbol{U}}_t = \boldsymbol{U}_t\boldsymbol{Q}_1, \quad \widetilde{\boldsymbol{V}}_t = \boldsymbol{V}_t\boldsymbol{Q}_2, \quad \widetilde{\boldsymbol{W}}_t = \boldsymbol{W}_t\boldsymbol{Q}_3,$$
$$\widetilde{\boldsymbol{\mathcal{S}}}_t = (\boldsymbol{Q}_1^{-1}, \boldsymbol{Q}_2^{-1}, \boldsymbol{Q}_3^{-1}) \boldsymbol{\cdot} \boldsymbol{\mathcal{S}}_t$$

for some invertible matrices[5] $\boldsymbol{Q}_k \in \mathrm{GL}(r_k)$ (i.e. replacing $\boldsymbol{U}_t$ by $\boldsymbol{U}_t\boldsymbol{Q}_1$, and so on), by plugging the above estimate in (2) it is easy to check that the next iterate of ScaledGD is covariant with respect to invertible transforms, meaning

$$\widetilde{\boldsymbol{U}}_{t+1} = \boldsymbol{U}_{t+1}\boldsymbol{Q}_1, \; \widetilde{\boldsymbol{V}}_{t+1} = \boldsymbol{V}_{t+1}\boldsymbol{Q}_2, \; \widetilde{\boldsymbol{W}}_{t+1} = \boldsymbol{W}_{t+1}\boldsymbol{Q}_3,$$
$$\widetilde{\boldsymbol{\mathcal{S}}}_{t+1} = (\boldsymbol{Q}_1^{-1}, \boldsymbol{Q}_2^{-1}, \boldsymbol{Q}_3^{-1}) \boldsymbol{\cdot} \boldsymbol{\mathcal{S}}_{t+1}.$$

In other words, ScaledGD produces an invariant sequence of low-rank tensor estimates

$$\boldsymbol{\mathcal{X}}_t = (\boldsymbol{U}_t, \boldsymbol{V}_t, \boldsymbol{W}_t) \boldsymbol{\cdot} \boldsymbol{\mathcal{S}}_t = (\widetilde{\boldsymbol{U}}_t, \widetilde{\boldsymbol{V}}_t, \widetilde{\boldsymbol{W}}_t) \boldsymbol{\cdot} \widetilde{\boldsymbol{\mathcal{S}}}_t$$

regardless of the representation of the tensor factors with respect to the underlying symmetry group. This is one of the key reasons behind the insensitivity of ScaledGD to ill-conditioning and factor imbalance.

A key scaled distance metric. To track the progress of ScaledGD throughout the entire trajectory, one needs a distance metric that properly takes account of the factor ambiguity due to invertible transforms, as well as the effect of scaling. To that end, we define the scaled distance between factor quadruples $\boldsymbol{F} = (\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}, \boldsymbol{\mathcal{S}})$ and $\boldsymbol{F}_\star = (\boldsymbol{U}_\star, \boldsymbol{V}_\star, \boldsymbol{W}_\star, \boldsymbol{\mathcal{S}}_\star)$ as

$$\mathrm{dist}^2(\boldsymbol{F}, \boldsymbol{F}_\star) \coloneqq \inf_{\boldsymbol{Q}_k \in \mathrm{GL}(r_k)} \left\|(\boldsymbol{U}\boldsymbol{Q}_1 - \boldsymbol{U}_\star)\boldsymbol{\Sigma}_{\star,1}\right\|_{\mathsf{F}}^2$$
$$+ \left\|(\boldsymbol{V}\boldsymbol{Q}_2 - \boldsymbol{V}_\star)\boldsymbol{\Sigma}_{\star,2}\right\|_{\mathsf{F}}^2 + \left\|(\boldsymbol{W}\boldsymbol{Q}_3 - \boldsymbol{W}_\star)\boldsymbol{\Sigma}_{\star,3}\right\|_{\mathsf{F}}^2$$
$$+ \left\|(\boldsymbol{Q}_1^{-1}, \boldsymbol{Q}_2^{-1}, \boldsymbol{Q}_3^{-1}) \boldsymbol{\cdot} \boldsymbol{\mathcal{S}} - \boldsymbol{\mathcal{S}}_\star\right\|_{\mathsf{F}}^2. \tag{18}$$

The distance is closely related to the ℓ_2 distances between the corresponding tensors. In fact, it can be shown that as long as $\boldsymbol{F}$ and $\boldsymbol{F}_\star$ are not too far apart, e.g. $\mathrm{dist}(\boldsymbol{F}, \boldsymbol{F}_\star) \le 0.2\sigma_{\min}(\boldsymbol{\mathcal{X}}_\star)$, it holds that $\mathrm{dist}(\boldsymbol{F}, \boldsymbol{F}_\star) \asymp \|(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}) \boldsymbol{\cdot} \boldsymbol{\mathcal{S}} - \boldsymbol{\mathcal{X}}_\star\|_{\mathsf{F}}$ in the sense that (see Tong et al. (2021c) for the proof):

$$\tfrac{1}{3}\left\|(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}) \boldsymbol{\cdot} \boldsymbol{\mathcal{S}} - \boldsymbol{\mathcal{X}}_\star\right\|_{\mathsf{F}} \le \mathrm{dist}(\boldsymbol{F}, \boldsymbol{F}_\star) \le$$
$$(\sqrt{2}+1)^{3/2}\left\|(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}) \boldsymbol{\cdot} \boldsymbol{\mathcal{S}} - \boldsymbol{\mathcal{X}}_\star\right\|_{\mathsf{F}}.$$

3.1 Proof outline of Theorem 1

Armed with the insights from the tensor factorization case, we now provide a proof outline of our main theorem on tensor completion, which can be viewed as perturbations of tensor factorization with incomplete measurements, combined with properly designed initialization schemes. We start with the guarantee for the spectral initialization for tensor completion.

Lemma 1 (Initialization for tensor completion)**.** *Suppose that $\boldsymbol{\mathcal{X}}_\star$ is μ-incoherent, and that p satisfies*

$$pn_1n_2n_3 \gtrsim \epsilon_0^{-2}\mu^{3/2}r^{5/2}\kappa^2\sqrt{n_1n_2n_3}\log^3 n$$
$$+\epsilon_0^{-4}\mu^2r^4\kappa^4 n\log^5 n$$

for some small constant $\epsilon_0 > 0$. Then with overwhelming probability (i.e. at least $1 - c_1n^{-c_2}$), the spectral initialization before projection $\boldsymbol{F}_+ = (\boldsymbol{U}_+, \boldsymbol{V}_+, \boldsymbol{W}_+, \boldsymbol{S}_+)$ in Algorithm 1 satisfies $\mathrm{dist}(\boldsymbol{F}_+, \boldsymbol{F}_\star) \le \epsilon_0\sigma_{\min}(\boldsymbol{\mathcal{X}}_\star)$.

Under a suitable sample size condition, Lemma 1 guarantees that $\mathrm{dist}(\boldsymbol{F}_+, \boldsymbol{F}_\star) \le \epsilon_0\sigma_{\min}(\boldsymbol{\mathcal{X}}_\star)$ for some small constant ϵ_0. To proceed, we need to know what would happen for the spectral estimate $\boldsymbol{F}_0 = \mathcal{P}_B(\boldsymbol{F}_+)$ after projection. In fact, the scaled projection is non-expansive w.r.t. the scaled distance. More importantly, the output is guaranteed to be incoherent. Both properties are stated in the following lemma.

Lemma 2 (Properties of scaled projection)**.** *Suppose that $\boldsymbol{\mathcal{X}}_\star$ is μ-incoherent, and $\mathrm{dist}(\boldsymbol{F}_+, \boldsymbol{F}_\star) \le \epsilon\sigma_{\min}(\boldsymbol{\mathcal{X}}_\star)$ for some $\epsilon < 1$. Set $B = C_B\sqrt{\mu r}\sigma_{\max}(\boldsymbol{\mathcal{X}}_\star)$ for some constant $C_B \ge (1+\epsilon)^3$, then $\boldsymbol{F} = (\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}, \boldsymbol{\mathcal{S}}) \coloneqq \mathcal{P}_B(\boldsymbol{F}_+)$ satisfies the non-expansiveness property*

$$\mathrm{dist}(\boldsymbol{F}, \boldsymbol{F}_\star) \le \mathrm{dist}(\boldsymbol{F}_+, \boldsymbol{F}_\star),$$

and the incoherence condition

$$\sqrt{n_1}\|\boldsymbol{U}\breve{\boldsymbol{U}}^\top\|_{2,\infty} \vee \sqrt{n_2}\|\boldsymbol{V}\breve{\boldsymbol{V}}^\top\|_{2,\infty}$$
$$\vee\sqrt{n_3}\|\boldsymbol{W}\breve{\boldsymbol{W}}^\top\|_{2,\infty} \le B. \tag{19}$$

Now we are ready to state the following lemma that ensures the linear contraction of the iterative refinements given by the ScaledGD updates.

[5] $\mathrm{GL}(r)$ denotes the set of invertible matrices in $\mathbb{R}^{r\times r}$.

Lemma 3 (Local refinements for tensor completion). *Suppose that $\boldsymbol{\mathcal{X}}_\star$ is μ-incoherent, and that p satisfies*

$$pn_1n_2n_3 \gtrsim \mu^{3/2}r^2\kappa^3\sqrt{n_1n_2n_3}\log^3 n + \mu^3r^4\kappa^6 n\log^5 n.$$

Under an event $\mathcal{E}$ which happens with overwhelming probability, if the t-th iterate satisfies $\operatorname{dist}(\boldsymbol{F}_t, \boldsymbol{F}_\star) \le \epsilon\sigma_{\min}(\boldsymbol{\mathcal{X}}_\star)$ *for some small constant ϵ, then* $\|(\boldsymbol{U}_t, \boldsymbol{V}_t, \boldsymbol{W}_t)\boldsymbol{\cdot}\boldsymbol{\mathcal{S}}_t - \boldsymbol{\mathcal{X}}_\star\|_{\mathsf{F}} \le 3\operatorname{dist}(\boldsymbol{F}_t, \boldsymbol{F}_\star)$. *In addition, if the t-th iterate satisfies the incoherence condition*

$$\sqrt{n_1}\|\boldsymbol{U}_t\breve{\boldsymbol{U}}_t^\top\|_{2,\infty} \vee \sqrt{n_2}\|\boldsymbol{V}_t\breve{\boldsymbol{V}}_t^\top\|_{2,\infty} \\ \vee\sqrt{n_3}\|\boldsymbol{W}_t\breve{\boldsymbol{W}}_t^\top\|_{2,\infty} \le B,$$

with $B = C_B\sqrt{\mu r}\sigma_{\max}(\boldsymbol{\mathcal{X}}_\star)$ for some constant $C_B \ge (1+\epsilon)^3$, then the $(t+1)$-th iterate of Algorithm 1 satisfies

$$\operatorname{dist}(\boldsymbol{F}_{t+1}, \boldsymbol{F}_\star) \le (1-0.6\eta)\operatorname{dist}(\boldsymbol{F}_t, \boldsymbol{F}_\star),$$

and the incoherence condition

$$\sqrt{n_1}\|\boldsymbol{U}_{t+1}\breve{\boldsymbol{U}}_{t+1}^\top\|_{2,\infty} \vee \sqrt{n_2}\|\boldsymbol{V}_{t+1}\breve{\boldsymbol{V}}_{t+1}^\top\|_{2,\infty} \\ \vee\sqrt{n_3}\|\boldsymbol{W}_{t+1}\breve{\boldsymbol{W}}_{t+1}^\top\|_{2,\infty} \le B.$$

By combining Lemma 1 and Lemma 2, we can ensure that the spectral initialization $\boldsymbol{F}_0 = \mathcal{P}_B(\boldsymbol{F}_+)$ satisfies the conditions required in Lemma 3, which further enables us to repetitively apply Lemma 3 to finish the proof of Theorem 1. The proofs to these lemmas can be found in Tong et al. (2021c).

4 NUMERICAL EXPERIMENTS

We present several demonstrative numerical experiments of ScaledGD, with codes available at

https://github.com/Titan-Tong/ScaledGD.

The simulations are performed in Matlab with a 3.6 GHz Intel Xeon Gold 6244 CPU. Due to space limits, more experiments are provided in Tong et al. (2021c).

We illustrate the numerical performance of ScaledGD for tensor completion to corroborate our findings, especially its computational advantage over the regularized GD algorithm (Han et al., 2020) that is closest to our design. Their algorithm was originally proposed for tensor regression, nevertheless, it naturally applies to tensor completion and exhibits similar results. Since the scaled projection does not visibly impact the performance, we implement ScaledGD without performing the projection. Also, we empirically find that the regularization used in Han et al. (2020) has no visible benefits, hence we implement GD without the regularization. For simplicity, we set $n_1 = n_2 = n_3 = n$, and $r_1 = r_2 = r_3 = r$. Each entry of the tensor is observed i.i.d. with probability $p \in (0, 1]$.

Phase transition of ScaledGD. We construct the ground truth tensor $\boldsymbol{\mathcal{X}}_\star = (\boldsymbol{U}_\star, \boldsymbol{V}_\star, \boldsymbol{W}_\star)\boldsymbol{\cdot}\boldsymbol{\mathcal{S}}_\star$ by generating $\boldsymbol{U}_\star$, $\boldsymbol{V}_\star$ and $\boldsymbol{W}_\star$ as random orthonormal matrices, and the core tensor $\boldsymbol{\mathcal{S}}_\star$ composed of i.i.d. standard Gaussian entries, i.e. $\boldsymbol{\mathcal{S}}_\star(j_1, j_2, j_3) \sim \mathcal{N}(0,1)$ for $1 \le j_k \le r$, $k = 1, 2, 3$. For each set of parameters, we run 100 random tests and count the success rate, where the recovery is regarded as successful if the recovered tensor has a relative error $\|\boldsymbol{\mathcal{X}}_T - \boldsymbol{\mathcal{X}}_\star\|_{\mathsf{F}}/\|\boldsymbol{\mathcal{X}}_\star\|_{\mathsf{F}} \le 10^{-3}$.

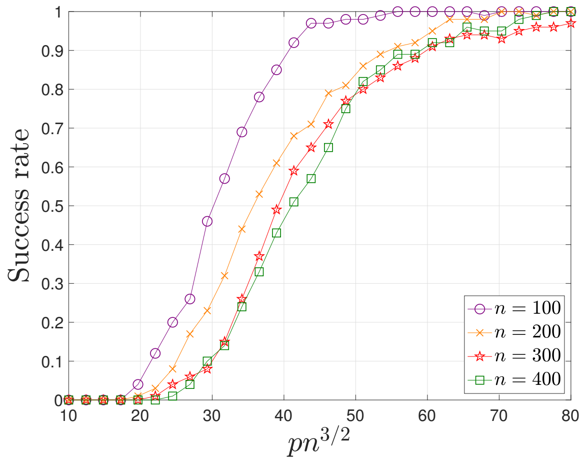

Figure 2: The success rate of ScaledGD with respect to the scaled sample size for tensor completion with $r = 5$, when the core tensor is composed of i.i.d. standard Gaussian entries, for various tensor size n.

Figure 2 illustrates the success rate with respect to the (scaled) sample size for different tensor sizes n, which implies that the recovery is successful when the sample size is moderately large.

Comparison with GD. We next compare the performance of ScaledGD with GD. For a fair comparison, both ScaledGD and GD start from the same spectral initialization, and we use the following update rule of GD as

$$\begin{aligned} \boldsymbol{U}_{t+1} &= \boldsymbol{U}_t - \eta\sigma_{\max}^{-2}(\boldsymbol{\mathcal{X}}_\star)\nabla_{\boldsymbol{U}}\mathcal{L}(\boldsymbol{F}_t), \\ \boldsymbol{V}_{t+1} &= \boldsymbol{V}_t - \eta\sigma_{\max}^{-2}(\boldsymbol{\mathcal{X}}_\star)\nabla_{\boldsymbol{V}}\mathcal{L}(\boldsymbol{F}_t), \\ \boldsymbol{W}_{t+1} &= \boldsymbol{W}_t - \eta\sigma_{\max}^{-2}(\boldsymbol{\mathcal{X}}_\star)\nabla_{\boldsymbol{W}}\mathcal{L}(\boldsymbol{F}_t), \\ \boldsymbol{\mathcal{S}}_{t+1} &= \boldsymbol{\mathcal{S}}_t - \eta\nabla_{\boldsymbol{\mathcal{S}}}\mathcal{L}(\boldsymbol{F}_t). \end{aligned} \tag{20}$$

Throughout the experiments, we used the ground truth value $\sigma_{\max}(\boldsymbol{\mathcal{X}}_\star)$ in running (20), while in practice, this parameter needs to estimated; to put it differently, the step size of GD is not *scale-invariant*, whereas the step size of ScaledGD is.

To ensure the ground truth tensor $\boldsymbol{\mathcal{X}}_\star = (\boldsymbol{U}_\star, \boldsymbol{V}_\star, \boldsymbol{W}_\star)\boldsymbol{\cdot}\boldsymbol{\mathcal{S}}_\star$ has a prescribed condition number

κ, we generate the core tensor $\boldsymbol{\mathcal{S}}_\star \in \mathbb{R}^{r\times r\times r}$ according to $\boldsymbol{\mathcal{S}}_\star(j_1, j_2, j_3) = \sigma_{j_1}/\sqrt{r}$ if $j_1 + j_2 + j_3 \equiv 0 \pmod r$ and 0 otherwise, where $\{\sigma_{j_1}\}_{1\le j_1\le r}$ take values spaced equally from 1 to $1/\kappa$. It then follows that $\sigma_{\max}(\boldsymbol{\mathcal{X}}_\star) = 1$, $\sigma_{\min}(\boldsymbol{\mathcal{X}}_\star) = 1/\kappa$, and the condition number of $\boldsymbol{\mathcal{X}}_\star$ is exactly κ.

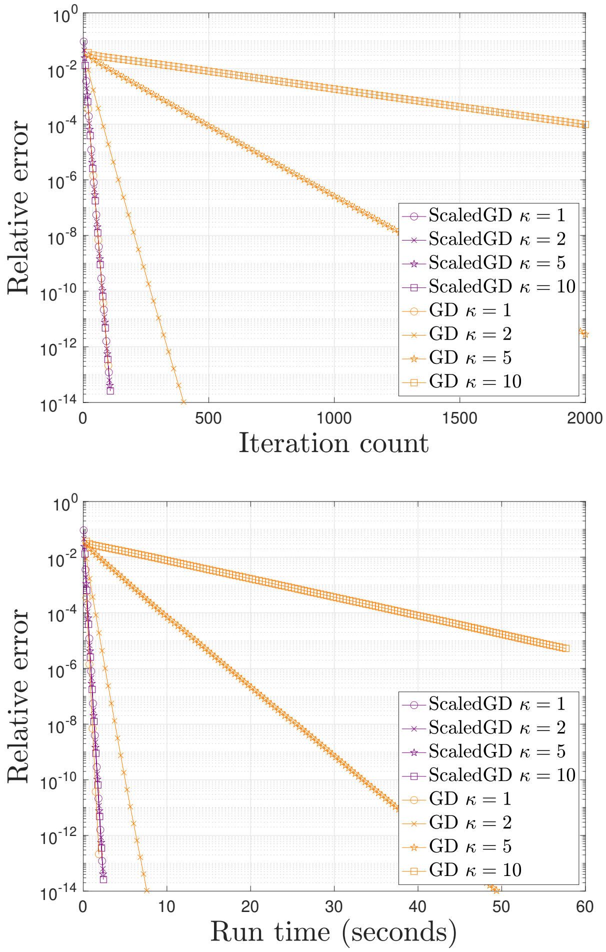

Figure 3: The relative errors of ScaledGD and GD with respect to the iteration count and run time (in seconds) under different condition numbers $\kappa = 1, 2, 5, 10$ for tensor completion with $n = 100$, $r = 5$, and $p = 0.1$.

Figure 3 compares the relative errors of ScaledGD and GD for tensor completion with respect to the iteration count and run time (in seconds) under different condition numbers $\kappa = 1, 2, 5, 10$. This experiment verifies that ScaledGD converges rapidly at a rate independent of the condition number, and matches the fastest rate of GD with perfect conditioning $\kappa = 1$. In contrast, the convergence rate of GD deteriorates quickly with the increase of κ even at a moderate level. The advantage of ScaledGD carries over to the run time as well, since the scaled gradient only adds a negligible overhead to the gradient computation.

5 DISCUSSIONS

This paper develops a scaled gradient descent algorithm over the factor space for low-rank tensor completion with provable sample and computational guarantees, leading to a highly scalable approach especially when the ground truth tensor is ill-conditioned and high-dimensional. Several future directions are worth exploring, which we briefly discuss below.

- *Preconditioning for other tensor decompositions.* Preconditioning will likely also accelerate vanilla gradient descent for low-rank tensor estimation using other decomposition models, such as CP decomposition (Cai et al., 2019), which is worth investigating.

- *Entrywise error control for tensor completion.* In this paper, we focused on controlling the ℓ_2 error of the reconstructed tensor in tensor completion, whereas another strong form of statistical guarantees deals with the ℓ_∞ error, as done in Ma et al. (2019) for matrix completion and in Cai et al. (2019) for tensor completion with CP decomposition. It is hence of interest to develop similar strong entrywise error guarantees of ScaledGD for tensor completion with Tucker decomposition.

- *Stable and robust low-rank tensor estimation.* In practice, the observations are corrupted by noise and even outliers (Li et al., 2020), therefore, it is necessary to examine the stability and robustness of ScaledGD in more depths, such as by pinning down the statistical error rates and extending the scaled subgradient method in Tong et al. (2021b) to the tensor case.

Acknowledgments

The work of T. Tong and Y. Chi is supported in part by Office of Naval Research under N00014-18-1-2142 and N00014-19-1-2404, by Army Research Office under W911NF-18-1-0303, by Air Force Research Laboratory under FA8750-20-2-0504, and by National Science Foundation under CAREER ECCS-1818571, CCF-1806154, CCF-1901199 and ECCS-2126634. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

References

Anandkumar, A., Ge, R., Hsu, D., Kakade, S., and Telgarsky, M. (2014). Tensor decompositions for learning latent variable models. *Journal of Machine Learning Research*, 15:2773–2832.

Barak, B. and Moitra, A. (2016). Noisy tensor completion via the sum-of-squares hierarchy. In *Conference on Learning Theory*, pages 417–445. PMLR.

Cai, C., Li, G., Chi, Y., Poor, H. V., and Chen, Y. (2021). Subspace estimation from unbalanced and incomplete data matrices: $\ell_{2,\infty}$ statistical guarantees. *The Annals of Statistics*, 49(2):944–967.

Cai, C., Li, G., Poor, H. V., and Chen, Y. (2019). Nonconvex low-rank tensor completion from noisy data. In *Advances in Neural Information Processing Systems*, pages 1863–1874.

Candès, E. J. and Tao, T. (2010). The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080.

Chen, Y. and Wainwright, M. J. (2015). Fast low-rank estimation by projected gradient descent: General statistical and algorithmic guarantees. *arXiv preprint arXiv:1509.03025*.

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

Dian, R., Fang, L., and Li, S. (2017). Hyperspectral image super-resolution via non-local sparse tensor factorization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5344–5353.

Frandsen, A. and Ge, R. (2020). Optimization landscape of Tucker decomposition. *Mathematical Programming*, pages 1–26.

Friedland, S. and Lim, L.-H. (2018). Nuclear norm of higher-order tensors. *Mathematics of Computation*, 87(311):1255–1281.

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

Hackbusch, W. (2012). *Tensor spaces and numerical tensor calculus*, volume 42. Springer.

Han, R., Willett, R., and Zhang, A. (2020). An optimal statistical and computational framework for generalized tensor estimation. *arXiv preprint arXiv:2002.11255*.

Huang, B., Mu, C., Goldfarb, D., and Wright, J. (2015). Provable models for robust low-rank tensor completion. *Pacific Journal of Optimization*, 11(2):339–364.

Karatzoglou, A., Amatriain, X., Baltrunas, L., and Oliver, N. (2010). Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the 4th ACM Conference on Recommender Systems*, pages 79–86.

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

Kolda, T. G. and Bader, B. W. (2009). Tensor decompositions and applications. *SIAM Review*, 51(3):455–500.

Li, Y., Chi, Y., Zhang, H., and Liang, Y. (2020). Non-convex low-rank matrix recovery with arbitrary outliers via median-truncated gradient descent. *Information and Inference: A Journal of the IMA*, 9(2):289–325.

Liu, J., Musialski, P., Wonka, P., and Ye, J. (2012). Tensor completion for estimating missing values in visual data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):208–220.

Ma, C., Li, Y., and Chi, Y. (2021). Beyond Procrustes: Balancing-free gradient descent for asymmetric low-rank matrix sensing. *IEEE Transactions on Signal Processing*, 69:867–877.

Ma, C., Wang, K., Chi, Y., and Chen, Y. (2019). 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.

Mu, C., Huang, B., Wright, J., and Goldfarb, D. (2014). Square deal: Lower bounds and improved relaxations for tensor recovery. In *International Conference on Machine Learning*, pages 73–81. PMLR.

Paatero, P. (2000). Construction and analysis of degenerate PARAFAC models. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 14(3):285–299.

Papalexakis, E. E., Faloutsos, C., and Sidiropoulos, N. D. (2016). Tensors for data mining and data fusion: Models, applications, and scalable algorithms. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 8(2):1–44.

Potechin, A. and Steurer, D. (2017). Exact tensor completion with sum-of-squares. In *Conference on Learning Theory*, pages 1619–1673. PMLR.

Sidiropoulos, N. D., De Lathauwer, L., Fu, X., Huang, K., Papalexakis, E. E., and Faloutsos, C. (2017). Tensor decomposition for signal processing and machine learning. *IEEE Transactions on Signal Processing*, 65(13):3551–3582.

Tong, T., Ma, C., and Chi, Y. (2021a). Accelerating ill-conditioned low-rank matrix estimation via scaled gradient descent. *Journal of Machine Learning Research*, 22(150):1–63.

Tong, T., Ma, C., and Chi, Y. (2021b). Low-rank matrix recovery with scaled subgradient methods: Fast and robust convergence without the condition number. *IEEE Transactions on Signal Processing*, 69:2396–2409.

Tong, T., Ma, C., Prater-Bennette, A., Tripp, E., and Chi, Y. (2021c). Scaling and scalability: Provable nonconvex low-rank tensor estimation from incomplete measurements. *arXiv preprint arXiv:2104.14526*.

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

Xia, D. and Yuan, M. (2019). On polynomial time methods for exact low-rank tensor completion. *Foundations of Computational Mathematics*, 19(6):1265–1313.

Xiong, L., Chen, X., Huang, T.-K., Schneider, J., and Carbonell, J. G. (2010). Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *Proceedings of the 2010 SIAM International Conference on Data Mining*, pages 211–222. SIAM.

Yuan, M. and Zhang, C.-H. (2016). On tensor completion via nuclear norm minimization. *Foundations of Computational Mathematics*, 16(4):1031–1068.

Zhang, Z., Ely, G., Aeron, S., Hao, N., and Kilmer, M. (2014). Novel methods for multilinear data completion and de-noising based on tensor-SVD. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3842–3849.