Provably Accelerating Ill-Conditioned Low-rank Estimation via Scaled Gradient Descent, Even with Overparameterization

Cong Ma, Xingyu Xu, Tian Tong, and Yuejie Chi

Abstract Many problems encountered in science and engineering can be formulated as estimating a low-rank object (e.g., matrices and tensors) from incomplete, and possibly corrupted, linear measurements. Through the lens of matrix and tensor factorization, one of the most popular approaches is to employ simple iterative algorithms such as gradient descent (GD) to recover the low-rank factors directly, which allow for small memory and computation footprints. However, the convergence rate of GD depends linearly, and sometimes even quadratically, on the condition number of the low-rank object, and therefore, GD slows down painstakingly when the problem is ill-conditioned. This chapter introduces a new algorithmic approach, dubbed scaled gradient descent (ScaledGD), that provably converges linearly at a constant rate independent of the condition number of the low-rank object, while maintaining the low per-iteration cost of gradient descent for a variety of tasks including sensing, robust principal component analysis and completion. In addition, ScaledGD continues to admit fast global convergence to the minimax-optimal solution, again almost independent of the condition number, from a small random initialization when the rank is over-specified in the presence of Gaussian noise. In total, ScaledGD 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 symmetry in low-rank factorization without hurting generalization.

1 Introduction

Low-rank matrix and tensor estimation plays a critical role in fields such as machine learning, signal processing, imaging science, and many others. The central task can be regarded as recovering an d-dimensional object $\boldsymbol{\mathcal{X}}_\star \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ from its highly incomplete observation $\boldsymbol{y} \in \mathbb{R}^m$ given by

$$\boldsymbol{y} \approx \mathcal{A}(\boldsymbol{\mathcal{X}}_\star).$$

Here, $\mathcal{A} : \mathbb{R}^{n_1 \times \cdots \times n_d} \mapsto \mathbb{R}^m$ represents a certain linear map modeling the data collection process. Importantly, the number m of observations is often much smaller than the ambient dimension $\prod_{i=1}^d n_i$ of the data object due to resource or physical constraints, necessitating the need of exploiting low-rank structures to allow for meaningful

Cong Ma
University of Chicago, e-mail: congm@uchicago.edu

Xingyu Xu
Carnegie Mellon University, e-mail: xingyuxu@andrew.cmu.edu

Tian Tong
Amazon, e-mail: tongtn@amazon.com

Yuejie Chi
Carnegie Mellon University, e-mail: yuejiechi@cmu.edu

recovery. It is natural to minimize the least-squares loss function

$$\underset{\boldsymbol{X}\in\mathbb{R}^{n_1\times\cdots\times n_d}}{\text{minimize}} \quad f(\boldsymbol{X}) \coloneqq \frac{1}{2}\|\mathcal{A}(\boldsymbol{X}) - \boldsymbol{y}\|_2^2 \tag{1}$$

subject to some rank constraint. However, naively imposing the rank constraint is computationally intractable, and moreover, as the size of the object increases, the costs involved in optimizing over the full space (i.e., $\mathbb{R}^{n_1\times\cdots\times n_d}$) are prohibitive in terms of both memory and computation.

To cope with these challenges, one popular approach is to represent the object of interest via its low-rank factors, which take a more economical form, and then optimize over the factors instead. Although this leads to a nonconvex optimization problem over the factors, recent breakthroughs have shown that simple iterative algorithms such as vanilla gradient descent (GD), when properly initialized (e.g., via the spectral method), can provably converge to the true low-rank factors under mild statistical assumptions; see [19] for an overview. This enables us to tap into the scalability of gradient descent in solving large-scale problems due to its amenability to computing advances such as parallelism [4].

However, upon closer examination, the computational cost of vanilla gradient descent is still expensive, especially for ill-conditioned objects. Although the per-iteration cost is small, the iteration complexity of gradient descent scales linearly with respect to the condition number of the low-rank matrix [56], which degenerates even worse for higher-order tensors [27]. In fact, the issue of ill-conditioning is quite ubiquitous in real-world data modeling with many contributing factors. One one end, extracting fine-grained and weak information often manifests to estimating ill-conditioned object of interest, when the goals are to separate close-located sources of intelligence, to identify a weak mode nearby a strong one, to predict individualized responses for similar objects, and so on. While the impact of condition numbers on the computational efficacy cannot be ignored in practice, it unfortunately has not been properly addressed in recent algorithmic advances, which often assume the problem is well-conditioned. These together raise an important question:

> Is it possible to design a first-order algorithm with a comparable per-iteration cost as gradient descent, but converges much faster at a rate that is independent of the condition number in a provable manner for a wide variety of low-rank matrix and tensor estimation tasks?

1.1 An overview of ScaledGD

In this chapter, we answer this question affirmatively by setting forth an algorithmic approach dubbed scaled gradient descent (`ScaledGD`), which is instantiated for an array of low-rank matrix and tensor estimation tasks.

ScaledGD for low-rank matrix estimation. By parametrizing the matrix object $\boldsymbol{X} = \boldsymbol{L}\boldsymbol{R}^\top$ via two low-rank factors $\boldsymbol{L} \in \mathbb{R}^{n_1\times r}$ and $\boldsymbol{R} \in \mathbb{R}^{n_2\times r}$ in (1), where r is the rank of the true low-rank object $\boldsymbol{X}_\star$, we arrive at the objective function

$$\underset{\boldsymbol{L}\in\mathbb{R}^{n_1\times r},\boldsymbol{R}\in\mathbb{R}^{n_2\times r}}{\text{minimize}} \quad \mathcal{L}(\boldsymbol{L},\boldsymbol{R}) \coloneqq f(\boldsymbol{L}\boldsymbol{R}^\top). \tag{2}$$

Given an initialization $(\boldsymbol{L}_0, \boldsymbol{R}_0)$, `ScaledGD` proceeds as follows

$$\begin{aligned}\boldsymbol{L}_{t+1} &= \boldsymbol{L}_t - \eta\nabla_{\boldsymbol{L}}\mathcal{L}(\boldsymbol{L}_t,\boldsymbol{R}_t)(\boldsymbol{R}_t^\top\boldsymbol{R}_t)^{-1},\\ \boldsymbol{R}_{t+1} &= \boldsymbol{R}_t - \eta\nabla_{\boldsymbol{R}}\mathcal{L}(\boldsymbol{L}_t,\boldsymbol{R}_t)(\boldsymbol{L}_t^\top\boldsymbol{L}_t)^{-1},\end{aligned} \tag{3}$$

where $\eta > 0$ is the step size and $\nabla_{\boldsymbol{L}}\mathcal{L}(\boldsymbol{L}_t,\boldsymbol{R}_t)$ (resp. $\nabla_{\boldsymbol{R}}\mathcal{L}(\boldsymbol{L}_t,\boldsymbol{R}_t)$) is the gradient of the loss function $\mathcal{L}$ with respect to the factor $\boldsymbol{L}_t$ (resp. $\boldsymbol{R}_t$) at the t-th iteration. Comparing to vanilla gradient descent, the search directions of the low-rank factors $\boldsymbol{L}_t, \boldsymbol{R}_t$ in (3) are *scaled* by $(\boldsymbol{R}_t^\top\boldsymbol{R}_t)^{-1}$ and $(\boldsymbol{L}_t^\top\boldsymbol{L}_t)^{-1}$ respectively. Intuitively, the scaling serves as a preconditioner as in quasi-Newton type algorithms, with the hope of improving the quality of the search

direction to allow larger step sizes. Since computing the Hessian is extremely expensive, it is necessary to design preconditioners that are both theoretically sound and practically cheap to compute. Such requirements are met by `ScaledGD`, where the preconditioners are computed by inverting two $r \times r$ matrices, whose size is much smaller than the dimension of matrix factors. Theoretically, we confirm that `ScaledGD` achieves linear convergence at a rate *independent of* the condition number of the matrix when initialized properly, e.g., using the standard spectral method, for several canonical problems: low-rank matrix sensing, robust PCA, and matrix completion.

Performance of ScaledGD for low-rank matrix completion

Fig. 1 illustrates the relative error of completing a 1000×1000 incoherent matrix (cf. Definition 2) of rank 10 with varying condition numbers from 20% of its entries, using either `ScaledGD` or vanilla GD with spectral initialization. Even for moderately ill-conditioned matrices, the convergence rate of vanilla GD slows down dramatically, while

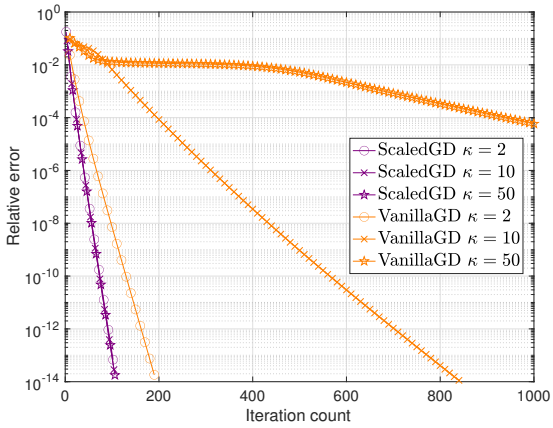


Fig. 1 Performance of `ScaledGD` and vanilla GD for completing a 1000×1000 incoherent matrix of rank 10 with different condition numbers $\kappa = 2, 10, 50$, where each entry is observed independently with probability 0.2. Here, both methods are initialized via the spectral method. It can be seen that `ScaledGD` converges much faster than vanilla GD even for moderately large condition numbers.

it is evident that `ScaledGD` converges at a rate independent of the condition number and therefore is much more efficient.

ScaledGD for low-rank tensor estimation. Turning to the tensor case, we focus on one of the most widely adopted low-rank structures for tensors under the *Tucker* decomposition [57], by assuming the true tensor $\boldsymbol{\mathcal{X}}_\star$ to be low-multilinear-rank, or simply low-rank, when its multilinear rank $\boldsymbol{r} = (r_1, r_2, r_3)$. By parameterizing the order-3 tensor object[1] as $\boldsymbol{\mathcal{X}} = (\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}) \cdot \boldsymbol{\mathcal{S}}$, where $\boldsymbol{F} := (\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}, \boldsymbol{\mathcal{S}})$ consists of the factors $\boldsymbol{U} \in \mathbb{R}^{n_1 \times r_1}$, $\boldsymbol{V} \in \mathbb{R}^{n_2 \times r_2}$, $\boldsymbol{W} \in \mathbb{R}^{n_3 \times r_3}$, and $\boldsymbol{\mathcal{S}} \in \mathbb{R}^{r_1 \times r_2 \times r_3}$, we aim to optimize the objective function:

$$\underset{\boldsymbol{F}}{\text{minimize}} \quad \mathcal{L}(\boldsymbol{F}) := \frac{1}{2} \left\| \mathcal{A}\left((\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}) \cdot \boldsymbol{\mathcal{S}} \right) - \boldsymbol{y} \right\|_2^2 . \tag{4}$$

Given an initialization $\boldsymbol{F}_0 = (\boldsymbol{U}_0, \boldsymbol{V}_0, \boldsymbol{W}_0, \boldsymbol{\mathcal{S}}_0)$, `ScaledGD` proceeds as follows

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

[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.

$$\boldsymbol{\mathcal{S}}_{t+1} = \boldsymbol{\mathcal{S}}_t - \eta \left((\boldsymbol{U}_t^\top \boldsymbol{U}_t)^{-1}, (\boldsymbol{V}_t^\top \boldsymbol{V}_t)^{-1}, (\boldsymbol{W}_t^\top \boldsymbol{W}_t)^{-1} \right) \cdot \nabla_{\boldsymbol{\mathcal{S}}} \mathcal{L}(\boldsymbol{F}_t), \tag{5}$$

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 corresponding variables, and

$$\begin{aligned} \check{\boldsymbol{U}}_t &\coloneqq \mathcal{M}_1 \left((\boldsymbol{I}_{r_1}, \boldsymbol{V}_t, \boldsymbol{W}_t) \cdot \boldsymbol{\mathcal{S}}_t \right)^\top = (\boldsymbol{W}_t \otimes \boldsymbol{V}_t) \mathcal{M}_1(\boldsymbol{\mathcal{S}}_t)^\top, \\ \check{\boldsymbol{V}}_t &\coloneqq \mathcal{M}_2 \left((\boldsymbol{U}_t, \boldsymbol{I}_{r_2}, \boldsymbol{W}_t) \cdot \boldsymbol{\mathcal{S}}_t \right)^\top = (\boldsymbol{W}_t \otimes \boldsymbol{U}_t) \mathcal{M}_2(\boldsymbol{\mathcal{S}}_t)^\top, \\ \check{\boldsymbol{W}}_t &\coloneqq \mathcal{M}_3 \left((\boldsymbol{U}_t, \boldsymbol{V}_t, \boldsymbol{I}_{r_3}) \cdot \boldsymbol{\mathcal{S}}_t \right)^\top = (\boldsymbol{V}_t \otimes \boldsymbol{U}_t) \mathcal{M}_3(\boldsymbol{\mathcal{S}}_t)^\top. \end{aligned} \tag{6}$$

Here, $\mathcal{M}_k(\boldsymbol{\mathcal{S}})$ is the matricization of the tensor $\boldsymbol{\mathcal{S}}$ along the k-th mode ($k = 1, 2, 3$), and $\otimes$ denotes the Kronecker product. We investigate the theoretical properties of `ScaledGD` for tensor regression, tensor robust PCA and tensor completion, which are notably more challenging than the matrix counterpart. It is demonstrated that `ScaledGD`—when initialized properly using appropriate spectral methods—again achieves linear convergence at a rate *independent* of the condition number of the ground truth tensor with near-optimal sample complexities.

ScaledGD for low-rank tensor completion

Fig. 2 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` and regularized GD [27] under different condition numbers for tensor completion under the Bernoulli sampling model. Clearly, the iteration complexity of GD deteriorates at a super linear rate with respect to the condition

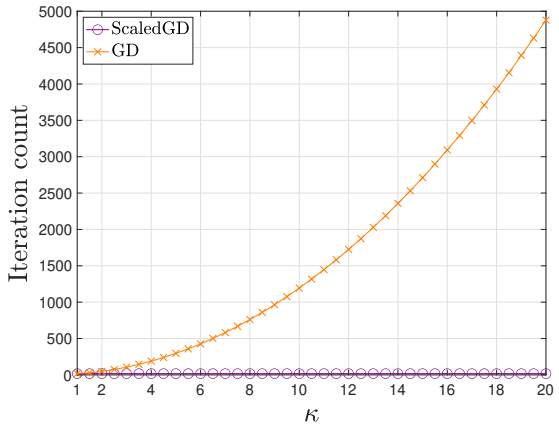


Fig. 2 The iteration complexities of `ScaledGD` and regularized GD to achieve $\|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{X}}_\star\|_{\mathsf{F}} \le 10^{-3} \|\boldsymbol{\mathcal{X}}_\star\|_{\mathsf{F}}$ with respect to different condition numbers for low-rank tensor completion with $n_1 = n_2 = n_3 = 100$, $r_1 = r_2 = r_3 = 5$, and the probability of observation $p = 0.1$.

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!

To highlight, `ScaledGD` possesses many desirable properties appealing to practitioners.

- *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., matrix factorization and tensor factorization). As the sizes of the preconditioners are proportional to the rank rather than the ambient dimension, 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 transformation, the entire trajectory will go through the same reparameterization, leading to an *invariant* sequence of low-rank updates regardless of the parameterization being adopted.
- *Implicit balancing:* `ScaledGD` optimizes the natural loss function in an *unconstrained* manner without requiring additional regularizations or orthogonalizations used in prior literature [27, 23, 34], and the factors stay balanced in an automatic manner as if they are implicitly regularized [40].

In total, the fast convergence rate of `ScaledGD`, together with its low computational and memory costs by operating in the factor space, makes it a highly scalable and desirable method for low-rank estimation tasks.

1.2 Related works

Our work contributes to the growing literature of design and analysis of provable nonconvex optimization procedures for high-dimensional signal estimation; see e.g. [30, 15, 19] for recent overviews. A growing number of problems have been demonstrated to possess benign geometry that is amenable for optimization [42] either globally or locally under appropriate statistical models. On one end, it is shown that there are no spurious local minima in the optimization landscape of matrix sensing and completion [26, 3, 45, 25], phase retrieval [52, 20], dictionary learning [51], kernel PCA [12] and linear neural networks [1, 35]. Such landscape analysis facilitates the adoption of generic saddle-point escaping algorithms [43, 24, 33] to ensure global convergence. However, the resulting iteration complexity is typically high. On the other end, local refinements with carefully-designed initializations often admit fast convergence, for example in phase retrieval [7, 41], matrix sensing [32, 67, 59], matrix completion [53, 16, 41, 13, 68, 17], blind deconvolution [36, 41, 49], quadratic sampling [37], and robust PCA [44, 64, 18], to name a few.

Existing approaches for asymmetric low-rank matrix estimation often require additional regularization terms to balance the two factors, either in the form of $\frac{1}{2}\|\boldsymbol{L}^\top\boldsymbol{L} - \boldsymbol{R}^\top\boldsymbol{R}\|_{\mathsf{F}}^2$ [56, 45] or $\frac{1}{2}\|\boldsymbol{L}\|_{\mathsf{F}}^2 + \frac{1}{2}\|\boldsymbol{R}\|_{\mathsf{F}}^2$ [69, 17, 18], which ease the theoretical analysis but are often unnecessary for the practical success, as long as the initialization is balanced. Some recent work studies the unregularized gradient descent for low-rank matrix factorization and sensing including [11, 22, 40]. However, the iteration complexity of all these approaches scales at least linearly with respect to the condition number κ of the low-rank matrix, e.g. $O(\kappa\log(1/\epsilon))$, to reach ϵ-accuracy, therefore they converge slowly when the underlying matrix becomes ill-conditioned. In contrast, `ScaledGD` enjoys a local convergence rate of $O(\log(1/\epsilon))$, therefore incurring a much smaller computational footprint when κ is large. Last but not least, alternating minimization [32, 28] (which alternatively updates $\boldsymbol{L}_t$ and $\boldsymbol{R}_t$) or singular value projection [44, 31] (which operates in the matrix space) also converge at the rate $O(\log(1/\epsilon))$, but the per-iteration cost is much higher than `ScaledGD`. Another notable algorithm is the Riemannian gradient descent algorithm in [59], which also converges at the rate $O(\log(1/\epsilon))$ under the same sample complexity for low-rank matrix sensing, but requires a higher memory complexity since it operates in the matrix space rather than the factor space.

Turning to the tensor case, unfolding-based approaches typically result in sub-optimal sample complexities since they do not fully exploit the tensor structure. [65] studied directly minimizing the nuclear norm of the tensor, which regrettably is not computationally tractable. [60] proposed a Grassmannian gradient descent algorithm over the factors other than the core tensor for exact tensor completion, whose iteration complexity is not characterized. The statistical rates of tensor completion, together with a spectral method, were investigated in [66, 61], and uncertainty quantifications were recently dealt with in [62]. In addition, for low-rank tensor regression, [46] proposed a general convex optimization approach based on decomposable regularizers, and [47] developed an iterative hard thresholding algorithm. A concurrent work [39] proposed a Riemannian Gauss-Newton algorithm, and obtained an impressive quadratic convergence rate for tensor regression. Compared with `ScaledGD`, this algorithm runs in the tensor space, and the update rule is more sophisticated with higher per-iteration cost by solving a least-squares problem and performing a truncated HOSVD every iteration. Another recent work [6] studies the Riemannian gradient descent algorithm which also achieves an iteration complexity free of condition number, however, the initialization scheme was not studied therein. Riemmannian gradient descent is also applied to low-rank tensor completion with Tucker decomposition in [58].

1.3 Chapter organization and notation

The rest of this chapter is organized as follows. Section 2 and Section 3 describe `ScaledGD` and details its application to sensing, robust PCA and completion with theoretical guarantees in terms of both statistical and computational complexities for the matrix and tensor case respectively. Section 4 discusses a variant of `ScaledGD` when the rank is not specified exactly. Section 5 illustrates the empirical performance of `ScaledGD` on real data, with a particular focus on the issues of rank selection. Finally, we conclude in Section 6.

Before continuing, we introduce several notation used throughout the chapter. First of all, we use boldfaced symbols for vectors and matrices (e.g. $\boldsymbol{x}$ and $\boldsymbol{X}$), and boldface calligraphic letters (e.g. $\boldsymbol{\mathcal{X}}$) to denote tensors. For a vector $\boldsymbol{v}$, we use $\|\boldsymbol{v}\|_0$ to denote its ℓ_0 counting norm, and $\|\boldsymbol{v}\|_2$ to denote the ℓ_2 norm. For any matrix $\boldsymbol{A}$, we use $\sigma_i(\boldsymbol{A})$ to denote its i-th largest singular value, and $\sigma_{\max}(\boldsymbol{A})$ (resp. $\sigma_{\min}(\boldsymbol{A})$) to denote its largest (resp. smallest) nonzero singular value. Let $\mathcal{P}_{\mathsf{diag}}(\boldsymbol{A})$ denote the projection that keeps only the diagonal entries of $\boldsymbol{A}$, and $\mathcal{P}_{\mathsf{off\text{-}diag}}(\boldsymbol{A}) = \boldsymbol{A} - \mathcal{P}_{\mathsf{diag}}(\boldsymbol{A})$, for a square matrix $\boldsymbol{A}$. Let $\boldsymbol{A}_{i,\cdot}$ and $\boldsymbol{A}_{\cdot,j}$ denote its i-th row and j-th column, respectively. In addition, $\|\boldsymbol{A}\|_{\mathsf{F}}$, $\|\boldsymbol{A}\|_{1,\infty}$, $\|\boldsymbol{A}\|_{2,\infty}$, and $\|\boldsymbol{A}\|_{\infty}$ stand for the Frobenius norm, the $\ell_{1,\infty}$ norm (i.e. the largest ℓ_1 norm of the rows), the $\ell_{2,\infty}$ norm (i.e. the largest ℓ_2 norm of the rows), and the entrywise ℓ_∞ norm (the largest magnitude of all entries) of a matrix $\boldsymbol{A}$. The set of invertible matrices in $\mathbb{R}^{r\times r}$ is denoted by $\mathrm{GL}(r)$. The $r\times r$ identity matrix is denoted by $\boldsymbol{I}_r$.

The mode-1 matricization $\mathcal{M}_1(\boldsymbol{\mathcal{X}}) \in \mathbb{R}^{n_1\times(n_2n_3)}$ of a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n_1\times n_2\times n_3}$ is given by $[\mathcal{M}_1(\boldsymbol{\mathcal{X}})]\big(i_1, i_2 + (i_3-1)n_2\big) = \boldsymbol{\mathcal{X}}(i_1,i_2,i_3)$, for $1\le i_k\le n_k$, $k=1,2,3$; $\mathcal{M}_2(\boldsymbol{\mathcal{X}})$ and $\mathcal{M}_3(\boldsymbol{\mathcal{X}})$ can be defined in a similar manner. The inner product between two tensors is defined as

$$\langle \boldsymbol{\mathcal{X}}_1, \boldsymbol{\mathcal{X}}_2\rangle = \sum_{i_1,i_2,i_3} \boldsymbol{\mathcal{X}}_1(i_1,i_2,i_3)\boldsymbol{\mathcal{X}}_2(i_1,i_2,i_3),$$

and the Frobenius norm of a tensor is defined as $\|\boldsymbol{\mathcal{X}}\|_{\mathsf{F}} = \sqrt{\langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{X}}\rangle}$. Define the ℓ_∞ norm of $\boldsymbol{\mathcal{X}}$ as $\|\boldsymbol{\mathcal{X}}\|_\infty = \max_{i_1,i_2,i_3} |\boldsymbol{\mathcal{X}}(i_1,i_2,i_3)|$. With slight abuse of terminology, denote

$$\sigma_{\max}(\boldsymbol{\mathcal{X}}) = \max_{k=1,2,3} \sigma_{\max}(\mathcal{M}_k(\boldsymbol{\mathcal{X}})), \quad \text{and} \quad \sigma_{\min}(\boldsymbol{\mathcal{X}}) = \min_{k=1,2,3} \sigma_{\min}(\mathcal{M}_k(\boldsymbol{\mathcal{X}}))$$

as the maximum and minimum nonzero singular values of $\boldsymbol{\mathcal{X}}$.

For a general tensor $\boldsymbol{\mathcal{X}}$, define $\mathcal{H}_{\boldsymbol{r}}(\boldsymbol{\mathcal{X}})$ as the top-$\boldsymbol{r}$ higher-order SVD (HOSVD) of $\boldsymbol{\mathcal{X}}$ with $\boldsymbol{r} = (r_1, r_2, r_3)$, given by

$$\mathcal{H}_{\boldsymbol{r}}(\boldsymbol{\mathcal{X}}) = (\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W})\cdot \boldsymbol{\mathcal{S}}, \tag{7}$$

where $\boldsymbol{U}$ is the top-r_1 left singular vectors of $\mathcal{M}_1(\boldsymbol{\mathcal{X}})$, $\boldsymbol{V}$ is the top-r_2 left singular vectors of $\mathcal{M}_2(\boldsymbol{\mathcal{X}})$, $\boldsymbol{W}$ is the top-r_3 left singular vectors of $\mathcal{M}_3(\boldsymbol{\mathcal{X}})$, and $\boldsymbol{\mathcal{S}} = (\boldsymbol{U}^\top, \boldsymbol{V}^\top, \boldsymbol{W}^\top)\cdot\boldsymbol{\mathcal{X}}$ is the core tensor.

Let $a\vee b = \max\{a,b\}$ and $a\wedge b = \min\{a,b\}$. Throughout, $f(n)\lesssim g(n)$ or $f(n) = O(g(n))$ means $|f(n)|/|g(n)|\le C$ for some constant $C>0$ when n is sufficiently large; $f(n)\gtrsim g(n)$ means $|f(n)|/|g(n)|\ge C$ for some constant $C>0$ when n is sufficiently large. Last but not least, we use the terminology "with overwhelming probability" to denote the event happens with probability at least $1 - c_1 n^{-c_2}$, where $c_1, c_2 > 0$ are some universal constants, whose values may vary from line to line.

2 ScaledGD for Low-Rank Matrix Estimation

This section is devoted to introducing `ScaledGD` and establishing its statistical and computational guarantees for various low-rank matrix estimation problems; the majority of the results are based on [54]. Table 1 summarizes the performance guarantees of `ScaledGD` in terms of both statistical and computational complexities with comparisons to prior algorithms using GD.

Table 1 Comparisons of `ScaledGD` with `GD` when tailored to various problems (with spectral initialization) [56, 64, 68], where they have comparable per-iteration costs. Here, we say that the output $\boldsymbol{X}$ of an algorithm reaches ϵ-accuracy, if it satisfies $\|\boldsymbol{X}-\boldsymbol{X}_\star\|_{\mathsf{F}} \le \epsilon\,\sigma_r(\boldsymbol{X}_\star)$. Here, $n \coloneqq n_1 \vee n_2 = \max\{n_1, n_2\}$, κ and μ are the condition number and incoherence parameter of $\boldsymbol{X}_\star$.

	Matrix sensing		Matrix robust PCA		Matrix completion	
Algorithms	sample complexity	iteration complexity	corruption fraction	iteration complexity	sample complexity	iteration complexity
`GD`	$nr^2\kappa^2$	$\kappa \log \frac{1}{\epsilon}$	$\frac{1}{\mu r^{3/2}\kappa^{3/2}\vee \mu r\kappa^2}$	$\kappa \log \frac{1}{\epsilon}$	$(\mu \vee \log n)\mu nr^2\kappa^2$	$\kappa \log \frac{1}{\epsilon}$
`ScaledGD`	$nr^2\kappa^2$	$\log \frac{1}{\epsilon}$	$\frac{1}{\mu r^{3/2}\kappa}$	$\log \frac{1}{\epsilon}$	$(\mu\kappa^2 \vee \log n)\mu nr^2\kappa^2$	$\log \frac{1}{\epsilon}$

2.1 Assumptions

Denote by $\boldsymbol{U}_\star\boldsymbol{\Sigma}_\star\boldsymbol{V}_\star^\top$ the compact singular value decomposition (SVD) of the rank-r matrix $\boldsymbol{X}_\star \in \mathbb{R}^{n_1\times n_2}$, i.e.,

$$\boldsymbol{X}_\star = \boldsymbol{U}_\star\boldsymbol{\Sigma}_\star\boldsymbol{V}_\star^\top.$$

Here, $\boldsymbol{U}_\star \in \mathbb{R}^{n_1\times r}$ and $\boldsymbol{V}_\star \in \mathbb{R}^{n_2\times r}$ are composed of r left and right singular vectors, respectively, and $\boldsymbol{\Sigma}_\star \in \mathbb{R}^{r\times r}$ is a diagonal matrix consisting of r singular values of $\boldsymbol{X}_\star$ organized in a non-increasing order, i.e. $\sigma_1(\boldsymbol{X}_\star) \ge \cdots \ge \sigma_r(\boldsymbol{X}_\star) > 0$. Define the ground truth low-rank factors as

$$\boldsymbol{L}_\star \coloneqq \boldsymbol{U}_\star\boldsymbol{\Sigma}_\star^{1/2}, \qquad \text{and} \qquad \boldsymbol{R}_\star \coloneqq \boldsymbol{V}_\star\boldsymbol{\Sigma}_\star^{1/2}, \tag{8}$$

so that $\boldsymbol{X}_\star = \boldsymbol{L}_\star\boldsymbol{R}_\star^\top$. Correspondingly, denote the stacked factor matrix as

$$\boldsymbol{F}_\star \coloneqq \begin{bmatrix}\boldsymbol{L}_\star\\ \boldsymbol{R}_\star\end{bmatrix} \in \mathbb{R}^{(n_1+n_2)\times r}. \tag{9}$$

Key parameters. The condition number of $\boldsymbol{X}_\star$ is defined as follows.

Definition 1 (Matrix condition number) Define

$$\kappa \coloneqq \frac{\sigma_1(\boldsymbol{X}_\star)}{\sigma_r(\boldsymbol{X}_\star)} \tag{10}$$

as the condition number of $\boldsymbol{X}_\star$.

We next introduce the incoherence condition, which is known to be crucial for reliable estimation of the low-rank matrix $\boldsymbol{X}_\star$ in matrix completion and robust PCA [14].

Definition 2 (Matrix incoherence) A rank-r matrix $\boldsymbol{X}_\star \in \mathbb{R}^{n_1\times n_2}$ with compact SVD as $\boldsymbol{X}_\star = \boldsymbol{U}_\star\boldsymbol{\Sigma}_\star\boldsymbol{V}_\star^\top$ is said to be μ-incoherent if

$$\|\boldsymbol{U}_\star\|_{2,\infty} \le \sqrt{\frac{\mu}{n_1}}\|\boldsymbol{U}_\star\|_{\mathsf{F}} = \sqrt{\frac{\mu r}{n_1}}, \quad \text{and} \quad \|\boldsymbol{V}_\star\|_{2,\infty} \le \sqrt{\frac{\mu}{n_2}}\|\boldsymbol{V}_\star\|_{\mathsf{F}} = \sqrt{\frac{\mu r}{n_2}}.$$

2.2 Matrix sensing

Observation model. Assume that we have collected a set of linear measurements about a rank-r matrix $\boldsymbol{X}_\star \in \mathbb{R}^{n_1\times n_2}$, given as

$$\boldsymbol{y} = \mathcal{A}(\boldsymbol{X}_\star) \in \mathbb{R}^m, \tag{11}$$

Algorithm 1 ScaledGD for low-rank matrix sensing with spectral initialization

Spectral initialization: Let $\boldsymbol{U}_0\boldsymbol{\Sigma}_0\boldsymbol{V}_0^\top$ be the top-r SVD of $\mathcal{A}^*(\boldsymbol{y})$, and set

$$\boldsymbol{L}_0 = \boldsymbol{U}_0\boldsymbol{\Sigma}_0^{1/2}, \quad \text{and} \quad \boldsymbol{R}_0 = \boldsymbol{V}_0\boldsymbol{\Sigma}_0^{1/2}. \tag{13}$$

Scaled gradient updates: **for** $t = 0, 1, 2, \ldots, T-1$ **do**

$$\begin{aligned}\boldsymbol{L}_{t+1} &= \boldsymbol{L}_t - \eta\mathcal{A}^*(\mathcal{A}(\boldsymbol{L}_t\boldsymbol{R}_t^\top) - \boldsymbol{y})\boldsymbol{R}_t(\boldsymbol{R}_t^\top\boldsymbol{R}_t)^{-1},\\ \boldsymbol{R}_{t+1} &= \boldsymbol{R}_t - \eta\mathcal{A}^*(\mathcal{A}(\boldsymbol{L}_t\boldsymbol{R}_t^\top) - \boldsymbol{y})^\top\boldsymbol{L}_t(\boldsymbol{L}_t^\top\boldsymbol{L}_t)^{-1}.\end{aligned} \tag{14}$$

where $\mathcal{A}(\boldsymbol{X}) = \{\langle\boldsymbol{A}_i, \boldsymbol{X}\rangle\}_{i=1}^m : \mathbb{R}^{n_1\times n_2} \mapsto \mathbb{R}^m$ is the linear map modeling the measurement process. The goal of low-rank matrix sensing is to recover $\boldsymbol{X}_\star$ from $\boldsymbol{y}$ when the number of measurements $m \ll n_1n_2$, which has wide applications in medical imaging, signal processing, and data compression [9].

Algorithm development. Writing $\boldsymbol{X} \in \mathbb{R}^{n_1\times n_2}$ into a factored form $\boldsymbol{X} = \boldsymbol{L}\boldsymbol{R}^\top$, we consider the following optimization problem:

$$\underset{\boldsymbol{F}\in\mathbb{R}^{(n_1+n_2)\times r}}{\text{minimize}} \quad \mathcal{L}(\boldsymbol{F}) = \frac{1}{2}\left\|\mathcal{A}(\boldsymbol{L}\boldsymbol{R}^\top) - \boldsymbol{y}\right\|_2^2. \tag{12}$$

Here as before, $\boldsymbol{F}$ denotes the stacked factor matrix $[\boldsymbol{L}^\top, \boldsymbol{R}^\top]^\top$. We suggest running ScaledGD (3) with the spectral initialization to solve (12), which performs the top-r SVD on $\mathcal{A}^*(\boldsymbol{y})$, where $\mathcal{A}^*(\cdot)$ is the adjoint operator of $\mathcal{A}(\cdot)$. The full algorithm is stated in Algorithm 1. The low-rank matrix can be estimated as $\boldsymbol{X}_T = \boldsymbol{L}_T\boldsymbol{R}_T^\top$ after running T iterations of ScaledGD.

Theoretical guarantee. To understand the performance of ScaledGD for low-rank matrix sensing, we adopt a standard assumption on the sensing operator $\mathcal{A}(\cdot)$, namely the Restricted Isometry Property (RIP).

Definition 3 (Matrix RIP [48]) The linear map $\mathcal{A}(\cdot)$ is said to obey the rank-r RIP with a constant $\delta_r \in [0, 1)$, if for all matrices $\boldsymbol{M} \in \mathbb{R}^{n_1\times n_2}$ of rank at most r, one has

$$(1-\delta_r)\|\boldsymbol{M}\|_{\mathsf{F}}^2 \le \|\mathcal{A}(\boldsymbol{M})\|_2^2 \le (1+\delta_r)\|\boldsymbol{M}\|_{\mathsf{F}}^2.$$

It is well-known that many measurement ensembles satisfy the RIP property [48, 9]. For example, if the entries of $\boldsymbol{A}_i$'s are composed of i.i.d. Gaussian entries $\mathcal{N}(0, 1/m)$, then the RIP is satisfied for a constant δ_r as long as m is on the order of $(n_1+n_2)r/\delta_r^2$. With the RIP condition in place, the following theorem demonstrates that ScaledGD converges linearly at a constant rate as long as the sensing operator $\mathcal{A}(\cdot)$ has a sufficiently small RIP constant.

Theorem 1 *Suppose that $\mathcal{A}(\cdot)$ obeys the $2r$-RIP with $\delta_{2r} \le 0.02/(\sqrt{r}\kappa)$. If the step size obeys $0 < \eta \le 2/3$, then for all $t \ge 0$, the iterates of the ScaledGD method in Algorithm 1 satisfy*

$$\left\|\boldsymbol{L}_t\boldsymbol{R}_t^\top - \boldsymbol{X}_\star\right\|_{\mathsf{F}} \le (1-0.6\eta)^t 0.15\sigma_r(\boldsymbol{X}_\star).$$

Theorem 1 establishes that the reconstruction error $\|\boldsymbol{L}_t\boldsymbol{R}_t^\top - \boldsymbol{X}_\star\|_{\mathsf{F}}$ contracts linearly at a constant rate, as long as the sample size satisfies $m = O(nr^2\kappa^2)$ with Gaussian random measurements [48], where we recall that $n = n_1 \vee n_2$. To reach ϵ-accuracy, i.e. $\|\boldsymbol{L}_t\boldsymbol{R}_t^\top - \boldsymbol{X}_\star\|_{\mathsf{F}} \le \epsilon\sigma_r(\boldsymbol{X}_\star)$, ScaledGD takes at most $T = O(\log(1/\epsilon))$ iterations, which is *independent* of the condition number κ of $\boldsymbol{X}_\star$. In comparison, GD with spectral initialization in [56] converges in $O(\kappa\log(1/\epsilon))$ iterations as long as $m = O(nr^2\kappa^2)$. Therefore, ScaledGD converges at a much faster rate than GD at the same sample complexity while maintaining a similar per-iteration cost (cf. Table 1).

2.3 Matrix robust principal component analysis

Observation model. Assume that we have observed the data matrix

$$\boldsymbol{Y} = \boldsymbol{X}_\star + \boldsymbol{S}_\star,$$

which is a superposition of a rank-r matrix $\boldsymbol{X}_\star$, modeling the clean data, and a sparse matrix $\boldsymbol{S}_\star$, modeling the corruption or outliers. The goal of robust PCA [8, 10] is to separate the two matrices $\boldsymbol{X}_\star$ and $\boldsymbol{S}_\star$ from their mixture $\boldsymbol{Y}$.

Following [10, 44, 64], we consider a deterministic sparsity model for $\boldsymbol{S}_\star$, in which $\boldsymbol{S}_\star$ contains at most α-fraction of nonzero entries per row and column for some $\alpha \in [0, 1)$, i.e. $\boldsymbol{S}_\star \in \mathcal{S}_\alpha$, where we denote

$$\mathcal{S}_\alpha \coloneqq \{\boldsymbol{S} \in \mathbb{R}^{n_1 \times n_2} : \|\boldsymbol{S}_{i,\cdot}\|_0 \le \alpha n_2 \text{ for all } i, \text{ and } \|\boldsymbol{S}_{\cdot,j}\|_0 \le \alpha n_1 \text{ for all } j\}. \tag{15}$$

Algorithm development. Writing $\boldsymbol{X} \in \mathbb{R}^{n_1 \times n_2}$ into the factored form $\boldsymbol{X} = \boldsymbol{L}\boldsymbol{R}^\top$, we consider the following optimization problem:

$$\underset{\boldsymbol{F} \in \mathbb{R}^{(n_1+n_2)\times r}, \boldsymbol{S} \in \mathcal{S}_\alpha}{\text{minimize}} \quad \mathcal{L}(\boldsymbol{F}, \boldsymbol{S}) = \frac{1}{2}\left\|\boldsymbol{L}\boldsymbol{R}^\top + \boldsymbol{S} - \boldsymbol{Y}\right\|_{\mathsf{F}}^2. \tag{16}$$

It is thus natural to alternatively update $\boldsymbol{F} = [\boldsymbol{L}^\top, \boldsymbol{R}^\top]^\top$ and $\boldsymbol{S}$, where $\boldsymbol{F}$ is updated via the proposed `ScaledGD` algorithm, and $\boldsymbol{S}$ is updated by hard thresholding, which trims the small entries of the residual matrix $\boldsymbol{Y} - \boldsymbol{L}\boldsymbol{R}^\top$. More specifically, for some truncation level $0 \le \bar{\alpha} \le 1$, we define the sparsification operator that only keeps $\bar{\alpha}$ fraction of largest entries in each row and column:

$$(\mathcal{T}_{\bar{\alpha}}[\boldsymbol{A}])_{i,j} = \begin{cases} \boldsymbol{A}_{i,j}, & \text{if } |\boldsymbol{A}|_{i,j} \ge |\boldsymbol{A}|_{i,(\bar{\alpha}n_2)}, \text{ and } |\boldsymbol{A}|_{i,j} \ge |\boldsymbol{A}|_{(\bar{\alpha}n_1),j} \\ 0, & \text{otherwise} \end{cases}, \tag{17}$$

where $|\boldsymbol{A}|_{i,(k)}$ (resp. $|\boldsymbol{A}|_{(k),j}$) denote the k-th largest element in magnitude in the i-th row (resp. j-th column). The `ScaledGD` algorithm with the spectral initialization for solving robust PCA is formally stated in Algorithm 2. Note that, comparing with [64], we do not require a balancing term $\|\boldsymbol{L}^\top\boldsymbol{L} - \boldsymbol{R}^\top\boldsymbol{R}\|_{\mathsf{F}}^2$ in the loss function (16), nor the projection of the low-rank factors onto the $\ell_{2,\infty}$ ball in each iteration.

Algorithm 2 `ScaledGD` for robust PCA with spectral initialization

Spectral initialization: Let $\boldsymbol{U}_0\boldsymbol{\Sigma}_0\boldsymbol{V}_0^\top$ be the top-r SVD of $\boldsymbol{Y} - \mathcal{T}_\alpha[\boldsymbol{Y}]$, and set

$$\boldsymbol{L}_0 = \boldsymbol{U}_0\boldsymbol{\Sigma}_0^{1/2}, \quad \text{and} \quad \boldsymbol{R}_0 = \boldsymbol{V}_0\boldsymbol{\Sigma}_0^{1/2}. \tag{18}$$

Scaled gradient updates: **for** $t = 0, 1, 2, \ldots, T-1$ **do**

$$\begin{aligned} \boldsymbol{S}_t &= \mathcal{T}_{2\alpha}[\boldsymbol{Y} - \boldsymbol{L}_t\boldsymbol{R}_t^\top], \\ \boldsymbol{L}_{t+1} &= \boldsymbol{L}_t - \eta(\boldsymbol{L}_t\boldsymbol{R}_t^\top + \boldsymbol{S}_t - \boldsymbol{Y})\boldsymbol{R}_t(\boldsymbol{R}_t^\top\boldsymbol{R}_t)^{-1}, \\ \boldsymbol{R}_{t+1} &= \boldsymbol{R}_t - \eta(\boldsymbol{L}_t\boldsymbol{R}_t^\top + \boldsymbol{S}_t - \boldsymbol{Y})^\top\boldsymbol{L}_t(\boldsymbol{L}_t^\top\boldsymbol{L}_t)^{-1}. \end{aligned} \tag{19}$$

Theoretical guarantee. The following theorem establishes the performance guarantee of `ScaledGD` as long as the fraction α of corruptions is sufficiently small.

Theorem 2 *Suppose that $\boldsymbol{X}_\star$ is μ-incoherent and that the corruption fraction α obeys $\alpha \le c/(\mu r^{3/2}\kappa)$ for some sufficiently small constant $c > 0$. If the step size obeys $0.1 \le \eta \le 2/3$, then for all $t \ge 0$, the iterates of `ScaledGD` in Algorithm 2 satisfy*

$$\left\|\boldsymbol{L}_t\boldsymbol{R}_t^\top - \boldsymbol{X}_\star\right\|_{\mathsf{F}} \le (1 - 0.6\eta)^t 0.03\sigma_r(\boldsymbol{X}_\star).$$

Theorem 2 establishes that the reconstruction error $\|\boldsymbol{L}_t\boldsymbol{R}_t^\top - \boldsymbol{X}_\star\|_{\mathsf{F}}$ contracts linearly at a constant rate, as long as the fraction of corruptions satisfies $\alpha \lesssim 1/(\mu r^{3/2}\kappa)$. To reach ϵ-accuracy, i.e. $\|\boldsymbol{L}_t\boldsymbol{R}_t^\top - \boldsymbol{X}_\star\|_{\mathsf{F}} \le \epsilon\sigma_r(\boldsymbol{X}_\star)$, `ScaledGD` takes at most $T = O(\log(1/\epsilon))$ iterations, which is *independent* of κ. In comparison, projected gradient

descent with spectral initialization in [64] converges in $O(\kappa \log(1/\epsilon))$ iterations as long as $\alpha \lesssim 1/(\mu r^{3/2}\kappa^{3/2} \vee \mu r \kappa^2)$. Therefore, `ScaledGD` converges at a much faster rate than GD while maintaining a comparable per-iteration cost (cf. Table 1). In addition, our theory unveils that `ScaledGD` automatically maintains the incoherence and balancedness of the low-rank factors without imposing explicit regularizations.

2.4 Matrix completion

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

$$(\mathcal{P}_\Omega(\boldsymbol{X}))_{i,j} = \begin{cases} \boldsymbol{X}_{i,j}, & \text{if } (i,j)\in\Omega, \\ 0, & \text{otherwise} \end{cases}. \tag{20}$$

Here, Ω is generated according to the Bernoulli model in the sense that each $(i,j)\in\Omega$ independent with probability $p\in(0,1]$. The goal of matrix completion is to recover the matrix $\boldsymbol{X}_\star$ from its partial observation $\mathcal{P}_\Omega(\boldsymbol{X}_\star)$.

Algorithm development. Again, writing $\boldsymbol{X}\in\mathbb{R}^{n_1\times n_2}$ into the factored form $\boldsymbol{X}=\boldsymbol{L}\boldsymbol{R}^\top$, we consider the following optimization problem:

$$\underset{\boldsymbol{F}\in\mathbb{R}^{(n_1+n_2)\times r}}{\text{minimize}} \quad \mathcal{L}(\boldsymbol{F}) \coloneqq \frac{1}{2p}\left\|\mathcal{P}_\Omega(\boldsymbol{L}\boldsymbol{R}^\top - \boldsymbol{X}_\star)\right\|_{\mathsf{F}}^2. \tag{21}$$

Similarly to robust PCA, the underlying low-rank matrix $\boldsymbol{X}_\star$ needs to be incoherent (cf. Definition 2) to avoid ill-posedness. One typical strategy to ensure the incoherence condition is to perform projection after the gradient update, by projecting the iterates to maintain small $\ell_{2,\infty}$ norms of the factor matrices. However, the standard projection operator [16] is not covariant with respect to invertible transforms, and consequently, needs to be modified when using scaled gradient updates. To that end, we introduce the following new projection operator: for every $\widetilde{\boldsymbol{F}}\in\mathbb{R}^{(n_1+n_2)\times r} = [\widetilde{\boldsymbol{L}}^\top, \widetilde{\boldsymbol{R}}^\top]^\top$,

$$\begin{aligned} \mathcal{P}_B(\widetilde{\boldsymbol{F}}) = \underset{\boldsymbol{F}\in\mathbb{R}^{(n_1+n_2)\times r}}{\operatorname{argmin}} \quad & \left\|(\boldsymbol{L}-\widetilde{\boldsymbol{L}})(\widetilde{\boldsymbol{R}}^\top\widetilde{\boldsymbol{R}})^{1/2}\right\|_{\mathsf{F}}^2 + \left\|(\boldsymbol{R}-\widetilde{\boldsymbol{R}})(\widetilde{\boldsymbol{L}}^\top\widetilde{\boldsymbol{L}})^{1/2}\right\|_{\mathsf{F}}^2 \\ \text{s.t.} \quad & \sqrt{n_1}\left\|\boldsymbol{L}(\widetilde{\boldsymbol{R}}^\top\widetilde{\boldsymbol{R}})^{1/2}\right\|_{2,\infty} \vee \sqrt{n_2}\left\|\boldsymbol{R}(\widetilde{\boldsymbol{L}}^\top\widetilde{\boldsymbol{L}})^{1/2}\right\|_{2,\infty} \le B \end{aligned}, \tag{22}$$

which finds a factored matrix that is closest to $\widetilde{\boldsymbol{F}}$ and stays incoherent in a weighted sense. Luckily, the solution to the above scaled projection admits a simple closed-form solution, given by

$$\begin{aligned} \mathcal{P}_B(\widetilde{\boldsymbol{F}}) \coloneqq \begin{bmatrix}\boldsymbol{L}\\ \boldsymbol{R}\end{bmatrix}, \quad \text{where } \boldsymbol{L}_{i,\cdot} &\coloneqq \left(1\wedge\frac{B}{\sqrt{n_1}\|\widetilde{\boldsymbol{L}}_{i,\cdot}\widetilde{\boldsymbol{R}}^\top\|_2}\right)\widetilde{\boldsymbol{L}}_{i,\cdot},\ 1\le i\le n_1, \\ \boldsymbol{R}_{j,\cdot} &\coloneqq \left(1\wedge\frac{B}{\sqrt{n_2}\|\widetilde{\boldsymbol{R}}_{j,\cdot}\widetilde{\boldsymbol{L}}^\top\|_2}\right)\widetilde{\boldsymbol{R}}_{j,\cdot},\ 1\le j\le n_2. \end{aligned} \tag{23}$$

With the new projection operator in place, we propose the following `ScaledGD` method with spectral initialization for solving matrix completion, formally stated in Algorithm 3.

Theoretical guarantee. The following theorem establishes the performance guarantee of `ScaledPGD` as long as the number of observations is sufficiently large.

Theorem 3 *Suppose that $\boldsymbol{X}_\star$ is μ-incoherent, and that p satisfies $p \ge C(\mu\kappa^2 \vee \log(n_1\vee n_2))\mu r^2\kappa^2/(n_1\wedge n_2)$ for some sufficiently large constant C. Set the projection radius as $B = C_B\sqrt{\mu r}\sigma_1(\boldsymbol{X}_\star)$ for some constant $C_B \ge 1.02$. If the step size obeys $0<\eta\le 2/3$, then with probability at least $1 - c_1(n_1\vee n_2)^{-c_2}$, for all $t\ge 0$, the iterates of `ScaledGD` in (25) satisfy*

Algorithm 3 ScaledGD for matrix completion with spectral initialization

Spectral initialization: Let $\boldsymbol{U}_0\boldsymbol{\Sigma}_0\boldsymbol{V}_0^\top$ be the top-r SVD of $\frac{1}{p}\mathcal{P}_\Omega(\boldsymbol{X}_\star)$, and set

$$\begin{bmatrix}\boldsymbol{L}_0\\ \boldsymbol{R}_0\end{bmatrix} = \mathcal{P}_B\left(\begin{bmatrix}\boldsymbol{U}_0\boldsymbol{\Sigma}_0^{1/2}\\ \boldsymbol{V}_0\boldsymbol{\Sigma}_0^{1/2}\end{bmatrix}\right). \tag{24}$$

Scaled projected gradient updates: **for** $t = 0, 1, 2, \ldots, T-1$ **do**

$$\begin{bmatrix}\boldsymbol{L}_{t+1}\\ \boldsymbol{R}_{t+1}\end{bmatrix} = \mathcal{P}_B\left(\begin{bmatrix}\boldsymbol{L}_t - \frac{\eta}{p}\mathcal{P}_\Omega(\boldsymbol{L}_t\boldsymbol{R}_t^\top - \boldsymbol{X}_\star)\boldsymbol{R}_t(\boldsymbol{R}_t^\top\boldsymbol{R}_t)^{-1}\\ \boldsymbol{R}_t - \frac{\eta}{p}\mathcal{P}_\Omega(\boldsymbol{L}_t\boldsymbol{R}_t^\top - \boldsymbol{X}_\star)^\top\boldsymbol{L}_t(\boldsymbol{L}_t^\top\boldsymbol{L}_t)^{-1}\end{bmatrix}\right). \tag{25}$$

$$\left\|\boldsymbol{L}_t\boldsymbol{R}_t^\top - \boldsymbol{X}_\star\right\|_{\mathsf{F}} \le (1-0.6\eta)^t 0.03\sigma_r(\boldsymbol{X}_\star).$$

Here $c_1, c_2 > 0$ are two universal constants.

Theorem 3 establishes that the reconstruction error $\|\boldsymbol{L}_t\boldsymbol{R}_t^\top - \boldsymbol{X}_\star\|_{\mathsf{F}}$ contracts linearly at a constant rate, as long as the probability of observation satisfies $p \gtrsim (\mu\kappa^2 \vee \log(n_1 \vee n_2))\mu r^2\kappa^2/(n_1 \wedge n_2)$. To reach ϵ-accuracy, i.e. $\|\boldsymbol{L}_t\boldsymbol{R}_t^\top - \boldsymbol{X}_\star\|_{\mathsf{F}} \le \epsilon\sigma_r(\boldsymbol{X}_\star)$, ScaledPGD takes at most $T = O(\log(1/\epsilon))$ iterations, which is *independent* of κ. In comparison, projected gradient descent [68] with spectral initialization converges in $O(\kappa\log(1/\epsilon))$ iterations as long as $p \gtrsim (\mu \vee \log(n_1 \vee n_2))\mu r^2\kappa^2/(n_1 \wedge n_2)$. Therefore, ScaledGD achieves much faster convergence than its unscaled counterpart, at a slightly higher sample complexity, which we believe can be further improved by finer analysis (cf. Table 1).

2.5 A glimpse of the analysis

At the heart of our analysis is a proper metric to measure the progress of the ScaledGD iterates $\boldsymbol{F}_t := [\boldsymbol{L}_t^\top, \boldsymbol{R}_t^\top]^\top$. Obviously, the factored representation is not unique in that for any invertible matrix $\boldsymbol{Q} \in \mathrm{GL}(r)$, one has $\boldsymbol{L}\boldsymbol{R}^\top = (\boldsymbol{L}\boldsymbol{Q})(\boldsymbol{R}\boldsymbol{Q}^{-\top})^\top$. Therefore, the reconstruction error metric needs to take into account this identifiability issue. More importantly, we need a diagonal scaling in the distance error metric to properly account for the effect of preconditioning. To provide intuition, note that the update rule (3) can be viewed as finding the best local quadratic approximation of $\mathcal{L}(\cdot)$ in the following sense:

$$\begin{aligned}&(\boldsymbol{L}_{t+1}, \boldsymbol{R}_{t+1})\\ &= \operatorname*{argmin}_{\boldsymbol{L},\boldsymbol{R}}\ \mathcal{L}(\boldsymbol{L}_t, \boldsymbol{R}_t) + \langle\nabla_{\boldsymbol{L}}\mathcal{L}(\boldsymbol{L}_t, \boldsymbol{R}_t), \boldsymbol{L} - \boldsymbol{L}_t\rangle + \langle\nabla_{\boldsymbol{R}}\mathcal{L}(\boldsymbol{L}_t, \boldsymbol{R}_t), \boldsymbol{R} - \boldsymbol{R}_t\rangle\\ &\qquad + \frac{1}{2\eta}\left(\left\|(\boldsymbol{L} - \boldsymbol{L}_t)(\boldsymbol{R}_t^\top\boldsymbol{R}_t)^{1/2}\right\|_{\mathsf{F}}^2 + \left\|(\boldsymbol{R} - \boldsymbol{R}_t)(\boldsymbol{L}_t^\top\boldsymbol{L}_t)^{1/2}\right\|_{\mathsf{F}}^2\right),\end{aligned}$$

where it is different from the common interpretation of gradient descent in the way the quadratic approximation is taken by a scaled norm. When $\boldsymbol{L}_t \approx \boldsymbol{L}_\star$ and $\boldsymbol{R}_t \approx \boldsymbol{R}_\star$ are approaching the ground truth, the additional scaling factors can be approximated by $\boldsymbol{L}_t^\top\boldsymbol{L}_t \approx \boldsymbol{\Sigma}_\star$ and $\boldsymbol{R}_t^\top\boldsymbol{R}_t \approx \boldsymbol{\Sigma}_\star$, leading to the following error metric

$$\mathrm{dist}^2(\boldsymbol{F}, \boldsymbol{F}_\star) := \inf_{\boldsymbol{Q}\in\mathrm{GL}(r)}\left\|(\boldsymbol{L}\boldsymbol{Q} - \boldsymbol{L}_\star)\boldsymbol{\Sigma}_\star^{1/2}\right\|_{\mathsf{F}}^2 + \left\|(\boldsymbol{R}\boldsymbol{Q}^{-\top} - \boldsymbol{R}_\star)\boldsymbol{\Sigma}_\star^{1/2}\right\|_{\mathsf{F}}^2. \tag{26}$$

The design and analysis of this new distance metric are of crucial importance in obtaining the improved rate of ScaledGD. In comparison, the previously studied distance metrics (proposed mainly for GD) either do not include the diagonal scaling [40, 56], or only consider the ambiguity class up to orthonormal transforms [56], which fail to unveil the benefit of ScaledGD.

3 ScaledGD for Low-rank Tensor Estimation

This section is devoted to introducing `ScaledGD` and establishing its statistical and computational guarantees for various low-rank tensor estimation problems; the majority of the results are based on [55, 21].

3.1 Assumptions

Suppose 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

$$\boldsymbol{\mathcal{X}}_\star(i_1, i_2, i_3) = \sum_{j_1=1}^{r_1} \sum_{j_2=1}^{r_2} \sum_{j_3=1}^{r_3} \boldsymbol{U}_\star(i_1, j_1) \boldsymbol{V}_\star(i_2, j_2) \boldsymbol{W}_\star(i_3, j_3) \boldsymbol{\mathcal{G}}_\star(j_1, j_2, j_3) \tag{27}$$

for $1 \le i_k \le n_k$, or more compactly,

$$\boldsymbol{\mathcal{X}}_\star = (\boldsymbol{U}_\star, \boldsymbol{V}_\star, \boldsymbol{W}_\star) \cdot \boldsymbol{\mathcal{G}}_\star, \tag{28}$$

where $\boldsymbol{\mathcal{G}}_\star = [\boldsymbol{\mathcal{G}}_\star(j_1, j_2, j_3)] \in \mathbb{R}^{r_1 \times r_2 \times r_3}$ 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

$$\mathcal{M}_1(\boldsymbol{\mathcal{X}}_\star) = \boldsymbol{U}_\star \mathcal{M}_1(\boldsymbol{\mathcal{G}}_\star)(\boldsymbol{W}_\star \otimes \boldsymbol{V}_\star)^\top, \tag{29a}$$
$$\mathcal{M}_2(\boldsymbol{\mathcal{X}}_\star) = \boldsymbol{V}_\star \mathcal{M}_2(\boldsymbol{\mathcal{G}}_\star)(\boldsymbol{W}_\star \otimes \boldsymbol{U}_\star)^\top, \tag{29b}$$
$$\mathcal{M}_3(\boldsymbol{\mathcal{X}}_\star) = \boldsymbol{W}_\star \mathcal{M}_3(\boldsymbol{\mathcal{G}}_\star)(\boldsymbol{V}_\star \otimes \boldsymbol{U}_\star)^\top. \tag{29c}$$

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

$$(\boldsymbol{U}_\star, \boldsymbol{V}_\star, \boldsymbol{W}_\star) \cdot \boldsymbol{\mathcal{G}}_\star = (\boldsymbol{U}_\star \boldsymbol{Q}_1, \boldsymbol{V}_\star \boldsymbol{Q}_2, \boldsymbol{W}_\star \boldsymbol{Q}_3) \cdot ((\boldsymbol{Q}_1^{-1}, \boldsymbol{Q}_2^{-1}, \boldsymbol{Q}_3^{-1}) \cdot \boldsymbol{\mathcal{G}}_\star).$$

We shall fix the ground truth factors such that $\boldsymbol{U}_\star$, $\boldsymbol{V}_\star$ and $\boldsymbol{W}_\star$ are orthonormal matrices consisting of left singular vectors in each mode. Furthermore, the core tensor $\boldsymbol{\mathcal{S}}_\star$ is related to the singular values in each mode as

$$\mathcal{M}_k(\boldsymbol{\mathcal{G}}_\star)\mathcal{M}_k(\boldsymbol{\mathcal{G}}_\star)^\top = \boldsymbol{\Sigma}_{\star,k}^2, \qquad k = 1, 2, 3, \tag{30}$$

where $\boldsymbol{\Sigma}_{\star,k} \coloneqq \mathrm{diag}[\sigma_1(\mathcal{M}_k(\boldsymbol{\mathcal{X}}_\star)), \ldots, \sigma_{r_k}(\mathcal{M}_k(\boldsymbol{\mathcal{X}}_\star))]$ is a diagonal matrix where the diagonal elements are composed of 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 4 (Tensor condition number) The condition number of $\boldsymbol{\mathcal{X}}_\star$ is defined as

$$\kappa \coloneqq \frac{\sigma_{\max}(\boldsymbol{\mathcal{X}}_\star)}{\sigma_{\min}(\boldsymbol{\mathcal{X}}_\star)} = \frac{\max_{k=1,2,3} \sigma_1(\mathcal{M}_k(\boldsymbol{\mathcal{X}}_\star))}{\min_{k=1,2,3} \sigma_{r_k}(\mathcal{M}_k(\boldsymbol{\mathcal{X}}_\star))}. \tag{31}$$

Another parameter is the incoherence parameter, which plays an important role in governing the well-posedness of low-rank tensor RPCA and completion.

Definition 5 (Tensor incoherence) The incoherence parameter of $\boldsymbol{\mathcal{X}}_\star$ is defined as

$$\mu \coloneqq \max\left\{ \frac{n_1}{r_1} \|\boldsymbol{U}_\star\|_{2,\infty}^2,\ \frac{n_2}{r_2} \|\boldsymbol{V}_\star\|_{2,\infty}^2,\ \frac{n_3}{r_3} \|\boldsymbol{W}_\star\|_{2,\infty}^2 \right\}. \tag{32}$$

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.

3.2 Tensor sensing

Observation model. We first consider tensor sensing — also known as tensor regression — with Gaussian design. Assume that we have access to a set of observations given as

$$y_i = \langle \boldsymbol{\mathcal{A}}_i, \boldsymbol{\mathcal{X}}_\star \rangle, \quad i = 1, \dots, m, \qquad \text{or concisely,} \qquad \boldsymbol{y} = \mathcal{A}(\boldsymbol{\mathcal{X}}_\star), \tag{33}$$

where $\boldsymbol{\mathcal{A}}_i \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is the i-th measurement tensor composed of i.i.d. Gaussian entries drawn from $\mathcal{N}(0, 1/m)$, and $\mathcal{A}(\boldsymbol{\mathcal{X}}) = \{\langle \boldsymbol{\mathcal{A}}_i, \boldsymbol{\mathcal{X}} \rangle\}_{i=1}^m$ is a linear map from $\mathbb{R}^{n_1 \times n_2 \times n_3}$ to $\mathbb{R}^m$, whose adjoint operator is given by $\mathcal{A}^*(\boldsymbol{y}) = \sum_{i=1}^m y_i \boldsymbol{\mathcal{A}}_i$. The goal is to recover $\boldsymbol{\mathcal{X}}_\star$ from $\boldsymbol{y}$, by leveraging the low-rank structure of $\boldsymbol{\mathcal{X}}_\star$.

Algorithm development. It is natural to minimize the following loss function

$$\underset{\boldsymbol{F}=(\boldsymbol{U},\boldsymbol{V},\boldsymbol{W},\boldsymbol{\mathcal{G}})}{\text{minimize}} \quad \mathcal{L}(\boldsymbol{F}) \coloneqq \frac{1}{2} \|\mathcal{A}((\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}) \cdot \boldsymbol{\mathcal{G}}) - \boldsymbol{y}\|_2^2 . \tag{34}$$

The proposed `ScaledGD` algorithm to minimize (34) is described in Algorithm 4, where the algorithm is initialized by applying HOSVD to $\mathcal{A}^*(\boldsymbol{y})$, followed by scaled gradient updates given in (35).

Algorithm 4 `ScaledGD` for low-rank tensor sensing

Input parameters: step size η, multilinear rank $\boldsymbol{r} = (r_1, r_2, r_3)$.
Spectral initialization: Let $(\boldsymbol{U}_0, \boldsymbol{V}_0, \boldsymbol{W}_0, \boldsymbol{\mathcal{G}}_0)$ be the factors in the top-$\boldsymbol{r}$ HOSVD of $\mathcal{A}^*(\boldsymbol{y})$ (cf. (7)).
Scaled gradient updates: for $t = 0, 1, 2, \dots, T-1$

$$\begin{aligned}
\boldsymbol{U}_{t+1} &= \boldsymbol{U}_t - \eta \mathcal{M}_1 \left(\mathcal{A}^*(\mathcal{A}((\boldsymbol{U}_t, \boldsymbol{V}_t, \boldsymbol{W}_t) \cdot \boldsymbol{\mathcal{G}}_t) - \boldsymbol{y})\right) \breve{\boldsymbol{U}}_t^\top (\breve{\boldsymbol{U}}_t^\top \breve{\boldsymbol{U}}_t)^{-1}, \\
\boldsymbol{V}_{t+1} &= \boldsymbol{V}_t - \eta \mathcal{M}_2 \left(\mathcal{A}^*(\mathcal{A}((\boldsymbol{U}_t, \boldsymbol{V}_t, \boldsymbol{W}_t) \cdot \boldsymbol{\mathcal{G}}_t) - \boldsymbol{y})\right) \breve{\boldsymbol{V}}_t^\top (\breve{\boldsymbol{V}}_t^\top \breve{\boldsymbol{V}}_t)^{-1}, \\
\boldsymbol{W}_{t+1} &= \boldsymbol{W}_t - \eta \mathcal{M}_3 \left(\mathcal{A}^*(\mathcal{A}((\boldsymbol{U}_t, \boldsymbol{V}_t, \boldsymbol{W}_t) \cdot \boldsymbol{\mathcal{G}}_t) - \boldsymbol{y})\right) \breve{\boldsymbol{W}}_t^\top (\breve{\boldsymbol{W}}_t^\top \breve{\boldsymbol{W}}_t)^{-1}, \\
\boldsymbol{\mathcal{G}}_{t+1} &= \boldsymbol{\mathcal{G}}_t - \eta \left((\boldsymbol{U}_t^\top \boldsymbol{U}_t)^{-1} \boldsymbol{U}_t^\top, (\boldsymbol{V}_t^\top \boldsymbol{V}_t)^{-1} \boldsymbol{V}_t^\top, (\boldsymbol{W}_t^\top \boldsymbol{W}_t)^{-1} \boldsymbol{W}_t^\top\right) \\
&\qquad \cdot \mathcal{A}^*(\mathcal{A}((\boldsymbol{U}_t, \boldsymbol{V}_t, \boldsymbol{W}_t) \cdot \boldsymbol{\mathcal{G}}_t) - \boldsymbol{y}),
\end{aligned} \tag{35}$$

where $\breve{\boldsymbol{U}}_t$, $\breve{\boldsymbol{V}}_t$, and $\breve{\boldsymbol{W}}_t$ are defined in (6).

Theoretical guarantee. Encouragingly, we can guarantee that `ScaledGD` provably recovers the ground truth tensor as long as the sample size is sufficiently large, which is given in the following theorem.

Theorem 4 *Let $n = \max_{k=1,2,3} n_k$ and $r = \max_{k=1,2,3} r_k$. With Gaussian design, suppose that m satisfies*

$$m \gtrsim \epsilon_0^{-1} \sqrt{n_1 n_2 n_3} r^{3/2} \kappa^2 + \epsilon_0^{-2} (n r^2 \kappa^4 \log n + r^4 \kappa^2)$$

for some small constant $\epsilon_0 > 0$. If the step size obeys $0 < \eta \le 2/5$, then with probability at least $1 - c_1 n^{-c_2}$ for universal constants $c_1, c_2 > 0$, for all $t \ge 0$, the iterates of Algorithm 4 satisfy

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

Theorem 4 ensures that the reconstruction error $\|(\boldsymbol{U}_t, \boldsymbol{V}_t, \boldsymbol{W}_t) \cdot \boldsymbol{\mathcal{G}}_t - \boldsymbol{\mathcal{X}}_\star\|_{\mathsf{F}}$ contracts linearly at a constant rate independent of the condition number of $\boldsymbol{\mathcal{X}}_\star$; to find an ε-accurate estimate, i.e. $\|(\boldsymbol{U}_t, \boldsymbol{V}_t, \boldsymbol{W}_t) \cdot \boldsymbol{\mathcal{G}}_t - \boldsymbol{\mathcal{X}}_\star\|_{\mathsf{F}} \le \varepsilon \sigma_{\min}(\boldsymbol{\mathcal{X}}_\star)$, `ScaledGD` needs at most $O(\log(1/\varepsilon))$ iterations, as long as the sample complexity satisfies

$$m \gtrsim n^{3/2} r^{3/2} \kappa^2,$$

where again we keep only terms with dominating orders of n. Compared with regularized GD [27], `ScaledGD` achieves a low computation complexity with robustness to ill-conditioning, improving its iteration complexity by a factor of κ^2, and does not require any explicit regularization.

3.3 Tensor robust principal component analysis

Observation model. Suppose that we collect a set of corrupted observations of $\boldsymbol{\mathcal{X}}_\star$ as

$$\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{X}}_\star + \boldsymbol{\mathcal{S}}_\star, \tag{36}$$

where $\boldsymbol{\mathcal{S}}_\star$ is the corruption tensor. The problem of tensor RPCA seeks to separate $\boldsymbol{\mathcal{X}}_\star$ and $\boldsymbol{\mathcal{S}}_\star$ from their sum $\boldsymbol{\mathcal{Y}}$ as efficiently and accurately as possible. Similar to the matrix case, we consider a deterministic sparsity model following the matrix case [10, 44, 64], where $\boldsymbol{\mathcal{S}}_\star$ contains at most a small fraction of nonzero entries per fiber. Formally, the corruption tensor $\boldsymbol{\mathcal{S}}_\star$ is said to be α-fraction sparse, i.e., $\boldsymbol{\mathcal{S}}_\star \in \boldsymbol{\mathcal{S}}_\alpha$, where

$$\boldsymbol{\mathcal{S}}_\alpha := \Big\{ \boldsymbol{\mathcal{S}} \in \mathbb{R}^{n_1 \times n_2 \times n_3} : \ \|\boldsymbol{\mathcal{S}}_{i_1,i_2,:}\|_0 \le \alpha n_3, \ \|\boldsymbol{\mathcal{S}}_{i_1,:,i_3}\|_0 \le \alpha n_2, \\ \|\boldsymbol{\mathcal{S}}_{:,i_2,i_3}\|_0 \le \alpha n_1, \ \text{for all } 1 \le i_k \le n_k, \quad k = 1,2,3 \Big\}. \tag{37}$$

ScaledGD algorithm. It is natural to optimize the following objective function:

$$\underset{\boldsymbol{F},\boldsymbol{\mathcal{S}}}{\text{minimize}} \ \mathcal{L}(\boldsymbol{F}, \boldsymbol{\mathcal{S}}) := \frac{1}{2} \left\| (\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}) \cdot \boldsymbol{\mathcal{G}} + \boldsymbol{\mathcal{S}} - \boldsymbol{\mathcal{Y}} \right\|_{\mathsf{F}}^2, \tag{38}$$

where $\boldsymbol{F} = (\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}, \boldsymbol{\mathcal{G}})$ and $\boldsymbol{\mathcal{S}}$ are the optimization variables for the tensor factors and the corruption tensor, respectively. Our algorithm alternates between corruption removal and factor refinements, as detailed in Algorithm 5. To remove the corruption, we use the following soft-shrinkage operator that trims the magnitudes of the entries by the amount of some carefully pre-set threshold.

Definition 6 (Soft-shrinkage operator) For an order-3 tensor $\boldsymbol{\mathcal{X}}$, the soft-shrinkage operator $\mathcal{T}_\zeta^{\text{soft}}[\cdot] : \mathbb{R}^{n_1 \times n_2 \times n_3} \mapsto \mathbb{R}^{n_1 \times n_2 \times n_3}$ with threshold $\zeta > 0$ is defined as

$$\left[\mathcal{T}_\zeta^{\text{soft}}[\boldsymbol{\mathcal{X}}]\right]_{i_1,i_2,i_3} := \text{sgn}\left([\boldsymbol{\mathcal{X}}]_{i_1,i_2,i_3}\right) \cdot \max\left(0, \left|[\boldsymbol{\mathcal{X}}]_{i_1,i_2,i_3}\right| - \zeta\right).$$

The soft-shrinkage operator sets entries with magnitudes smaller than ζ to 0, while uniformly shrinking the magnitudes of the other entries by ζ. At the beginning of each iteration, the corruption tensor is updated via applying the soft-thresholding operator to the current residual $\boldsymbol{\mathcal{Y}} - (\boldsymbol{U}_t, \boldsymbol{V}_t, \boldsymbol{W}_t) \cdot \boldsymbol{\mathcal{G}}_t$ using some properly selected threshold ζ_t, followed by updating the tensor factors $\boldsymbol{F}_t$ via scaled gradient descent with respect to $\mathcal{L}(\boldsymbol{F}_t, \boldsymbol{\mathcal{S}}_{t+1})$ in (38). To complete the algorithm description, we still need to specify how to initialize the algorithm. This is again achieved by the spectral method, which computes the rank-$\boldsymbol{r}$ HOSVD of the observation after applying the soft-shrinkage operator

Theoretical guarantee. Fortunately, the `ScaledGD` algorithm provably recovers the ground truth tensor—as long as the fraction of corruptions is not too large—with proper choices of the tuning parameters, as captured in following theorem.

Theorem 5 *Let $\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{X}}_\star + \boldsymbol{\mathcal{S}}_\star \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, where $\boldsymbol{\mathcal{X}}_\star$ is μ-incoherent with multilinear rank $\boldsymbol{r} = (r_1, r_2, r_3)$, and $\boldsymbol{\mathcal{S}}_\star$ is α-sparse. Suppose that the threshold values $\{\zeta_k\}_{k=0}^\infty$ obey that $\|\boldsymbol{\mathcal{X}}_\star\|_\infty \le \zeta_0 \le 2\|\boldsymbol{\mathcal{X}}_\star\|_\infty$ and $\zeta_{t+1} = \rho\zeta_t$, $t \ge 1$, for some properly tuned $\zeta_1 := 8\sqrt{\frac{\mu^3 r_1 r_2 r_3}{n_1 n_2 n_3}}\sigma_{\min}(\boldsymbol{\mathcal{X}}_\star)$ and $\frac{1}{7} \le \eta \le \frac{1}{4}$, where $\rho = 1 - 0.45\eta$. Then, the iterates $\boldsymbol{\mathcal{X}}_t = (\boldsymbol{U}_t, \boldsymbol{V}_t, \boldsymbol{W}_t) \cdot \boldsymbol{\mathcal{G}}_t$ satisfy*

Algorithm 5 `ScaledGD` for tensor robust principal component analysis

Input: observed tensor $\boldsymbol{\mathcal{Y}}$, multilinear rank $\boldsymbol{r}$, learning rate η, and threshold schedule $\{\zeta_t\}_{t=0}^{T}$.
Spectral initialization: Set $\boldsymbol{\mathcal{S}}_0 = \mathcal{T}_{\zeta_0}^{\text{soft}}[\boldsymbol{\mathcal{Y}}]$ and $(\boldsymbol{U}_0, \boldsymbol{V}_0, \boldsymbol{W}_0, \boldsymbol{\mathcal{G}}_0)$ as the factors in the top-$\boldsymbol{r}$ HOSVD of $\boldsymbol{\mathcal{Y}} - \boldsymbol{\mathcal{S}}_0$ (cf. (7)).
Scaled gradient updates: **for** $t = 0, 1, 2, \ldots, T-1$ **do**

$$\boldsymbol{\mathcal{S}}_{t+1} = \mathcal{T}_{\zeta_{t+1}}^{\text{soft}}\left[\boldsymbol{\mathcal{Y}} - (\boldsymbol{U}_t, \boldsymbol{V}_t, \boldsymbol{W}_t)\cdot\boldsymbol{\mathcal{G}}_t\right], \tag{39a}$$

$$\boldsymbol{U}_{t+1} = \boldsymbol{U}_t - \eta\left(\boldsymbol{U}_t\check{\boldsymbol{U}}_t^\top + \mathcal{M}_1(\boldsymbol{\mathcal{S}}_{t+1}) - \mathcal{M}_1(\boldsymbol{\mathcal{Y}})\right)\check{\boldsymbol{U}}_t(\check{\boldsymbol{U}}_t^\top\check{\boldsymbol{U}}_t)^{-1}, \tag{39b}$$

$$\boldsymbol{V}_{t+1} = \boldsymbol{V}_t - \eta\left(\boldsymbol{V}_t\check{\boldsymbol{V}}_t^\top + \mathcal{M}_2(\boldsymbol{\mathcal{S}}_{t+1}) - \mathcal{M}_2(\boldsymbol{\mathcal{Y}})\right)\check{\boldsymbol{V}}_t(\check{\boldsymbol{V}}_t^\top\check{\boldsymbol{V}}_t)^{-1}, \tag{39c}$$

$$\boldsymbol{W}_{t+1} = \boldsymbol{W}_t - \eta\left(\boldsymbol{W}_t\check{\boldsymbol{W}}_t^\top + \mathcal{M}_1(\boldsymbol{\mathcal{S}}_{t+1}) - \mathcal{M}_1(\boldsymbol{\mathcal{Y}})\right)\check{\boldsymbol{W}}_t(\check{\boldsymbol{W}}_t^\top\check{\boldsymbol{W}}_t)^{-1}, \tag{39d}$$

$$\begin{aligned}\boldsymbol{\mathcal{G}}_{t+1} = \boldsymbol{\mathcal{G}}_t - \eta\Big((\boldsymbol{U}_t^\top\boldsymbol{U}_t)^{-1}\boldsymbol{U}_t^\top, (\boldsymbol{V}_t^\top\boldsymbol{V}_t)^{-1}\boldsymbol{V}_t^\top, (\boldsymbol{W}_t^\top\boldsymbol{W}_t)^{-1}\boldsymbol{W}_t^\top\Big)\\ \cdot\left((\boldsymbol{U}_t, \boldsymbol{V}_t, \boldsymbol{W}_t)\cdot\boldsymbol{\mathcal{G}}_t + \boldsymbol{\mathcal{S}}_{t+1} - \boldsymbol{\mathcal{Y}}\right),\end{aligned} \tag{39e}$$

where $\check{\boldsymbol{U}}_t$, $\check{\boldsymbol{V}}_t$, and $\check{\boldsymbol{W}}_t$ are defined in (6).

$$\|\boldsymbol{\mathcal{X}}_t - \boldsymbol{\mathcal{X}}_\star\|_{\mathsf{F}} \le 0.03\rho^t\sigma_{\min}(\boldsymbol{\mathcal{X}}_\star), \tag{40a}$$

$$\|\boldsymbol{\mathcal{X}}_t - \boldsymbol{\mathcal{X}}_\star\|_{\infty} \le 8\rho^t\sqrt{\frac{\mu^3 r_1 r_2 r_3}{n_1 n_2 n_3}}\sigma_{\min}(\boldsymbol{\mathcal{X}}_\star) \tag{40b}$$

for all $t \ge 0$, as long as the level of corruptions obeys $\alpha \le \frac{c_0}{\mu^2 r_1 r_2 r_3 \kappa}$ for some sufficiently small $c_0 > 0$.

Theorem 5 implies that upon appropriate choices of the parameters, if the level of corruptions α is small enough, i.e. not exceeding the order of $\frac{1}{\mu^2 r_1 r_2 r_3 \kappa}$, we can ensure that `ScaledGD` converges at a linear rate independent of the condition number and exactly recovers the ground truth tensor $\boldsymbol{\mathcal{X}}_\star$ even when the gross corruptions are arbitrary and adversarial. Furthermore, when $\mu = O(1)$ and $r = O(1)$, the entrywise error bound (40b)—which is smaller than the Frobenius error (40a) by a factor of $\sqrt{\frac{1}{n_1 n_2 n_3}}$—suggests the errors are distributed in an even manner across the entries for incoherent and low-rank tensors.

3.4 Tensor completion

Observation model. 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 such that

$$[\mathcal{P}_\Omega(\boldsymbol{\mathcal{X}}_\star)](i_1, i_2, i_3) = \begin{cases}\boldsymbol{\mathcal{X}}_\star(i_1, i_2, i_3), & \text{if } (i_1, i_2, i_3) \in \Omega,\\ 0, & \text{otherwise.}\end{cases} \tag{41}$$

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

$$(i_1, i_2, i_3) \in \Omega \quad \text{independently with probability } p \in (0, 1]. \tag{42}$$

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

$$\min_{\boldsymbol{F}=(\boldsymbol{U},\boldsymbol{V},\boldsymbol{W},\boldsymbol{\mathcal{S}})} \mathcal{L}(\boldsymbol{F}) := \frac{1}{2p}\left\|\mathcal{P}_\Omega\big((\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W})\cdot\boldsymbol{\mathcal{S}}\big) - \boldsymbol{\mathcal{Y}}\right\|_{\mathsf{F}}^2. \tag{43}$$

Algorithm development. To guarantee faithful recovery from partial observations, the underlying low-rank tensor $\boldsymbol{\mathcal{X}}_\star$ needs to be incoherent (cf. Definition 5) to avoid ill-posedness. One typical strategy, as employed in the matrix setting, to ensure the incoherence condition is to trim the rows of the factors after the scaled gradient update. We

introduce the scaled projection as follows,

$$(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}, \boldsymbol{\mathcal{S}}) = \mathcal{P}_B(\boldsymbol{U}_+, \boldsymbol{V}_+, \boldsymbol{W}_+, \boldsymbol{\mathcal{S}}_+), \tag{44}$$

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

$$\begin{aligned}
\boldsymbol{U}(i_1,:) &= \left(1 \wedge \frac{B}{\sqrt{n_1}\|\boldsymbol{U}_+(i_1,:)\check{\boldsymbol{U}}_+^\top\|_2}\right)\boldsymbol{U}_+(i_1,:), \qquad 1 \le i_1 \le n_1;\\
\boldsymbol{V}(i_2,:) &= \left(1 \wedge \frac{B}{\sqrt{n_2}\|\boldsymbol{V}_+(i_2,:)\check{\boldsymbol{V}}_+^\top\|_2}\right)\boldsymbol{V}_+(i_2,:), \qquad 1 \le i_2 \le n_2;\\
\boldsymbol{W}(i_3,:) &= \left(1 \wedge \frac{B}{\sqrt{n_3}\|\boldsymbol{W}_+(i_3,:)\check{\boldsymbol{W}}_+^\top\|_2}\right)\boldsymbol{W}_+(i_3,:), \qquad 1 \le i_3 \le n_3;\\
\boldsymbol{\mathcal{S}} &= \boldsymbol{\mathcal{S}}_+.
\end{aligned}$$

Here, we recall $\check{\boldsymbol{U}}_+$, $\check{\boldsymbol{V}}_+$, $\check{\boldsymbol{W}}_+$ are analogously defined in (6) 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 $\boldsymbol{U}_+\check{\boldsymbol{U}}_+^\top$ (resp. $\boldsymbol{V}_+\check{\boldsymbol{V}}_+^\top$ and $\boldsymbol{W}_+\check{\boldsymbol{W}}_+^\top$), which is the mode-1 (resp. mode-2 and mode-3) matricization of the tensor $(\boldsymbol{U}_+, \boldsymbol{V}_+, \boldsymbol{W}_+) \cdot \boldsymbol{\mathcal{S}}_+$. It is a straightforward observation that the projection can be computed efficiently.

With the scaled projection $\mathcal{P}_B(\cdot)$ defined in hand, we are in a position to describe the details of the proposed `ScaledGD` algorithm, summarized in Algorithm 6. It consists of two stages: spectral initialization followed by iterative refinements using the scaled projected gradient updates in (45). For the spectral initialization, we take advantage of the subspace estimators proposed in [5, 61] for highly unbalanced 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 $p^{-1}\mathcal{M}_1(\boldsymbol{\mathcal{Y}})$, denoted as

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

and the other two factors $\boldsymbol{V}_+$ and $\boldsymbol{W}_+$ are estimated similarly. The core tensor is then estimated as

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

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

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

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

Theorem 6 *Let $n = \max_{k=1,2,3} n_k$ and $r = \max_{k=1,2,3} r_k$. Suppose that $\boldsymbol{\mathcal{X}}_\star$ is μ-incoherent, $n_k \gtrsim \epsilon_0^{-1}\mu r_k^{3/2}\kappa^2$ for $k = 1, 2, 3$, and that p satisfies*

$$pn_1n_2n_3 \gtrsim \epsilon_0^{-1}\sqrt{n_1n_2n_3}\mu^{3/2}r^{5/2}\kappa^3\log^3 n + \epsilon_0^{-2}n\mu^3r^4\kappa^6\log^5 n$$

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 6 satisfy

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

Theorem 6 ensures that to find an ε-accurate estimate, i.e. $\|(\boldsymbol{U}_t, \boldsymbol{V}_t, \boldsymbol{W}_t) \cdot \boldsymbol{\mathcal{S}}_t - \boldsymbol{\mathcal{X}}_\star\|_{\mathsf{F}} \le \varepsilon\sigma_{\min}(\boldsymbol{\mathcal{X}}_\star)$, `ScaledGD` takes at most $O(\log(1/\varepsilon))$ iterations, which is independent of the condition number of $\boldsymbol{\mathcal{X}}_\star$, as long as the sample

Algorithm 6 `ScaledGD` for low-rank tensor completion

Input parameters: step size η, multilinear 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 $\mathcal{P}_{\text{off-diag}}(p^{-2}\mathcal{M}_1(\boldsymbol{\mathcal{Y}})\mathcal{M}_1(\boldsymbol{\mathcal{Y}})^\top)$, and similarly for $\boldsymbol{V}_+, \boldsymbol{W}_+$, and $\boldsymbol{\mathcal{S}}_+ = p^{-1}(\boldsymbol{U}_+^\top, \boldsymbol{V}_+^\top, \boldsymbol{W}_+^\top) \cdot \boldsymbol{\mathcal{Y}}$. Set $(\boldsymbol{U}_0, \boldsymbol{V}_0, \boldsymbol{W}_0, \boldsymbol{\mathcal{S}}_0) = \mathcal{P}_B(\boldsymbol{U}_+, \boldsymbol{V}_+, \boldsymbol{W}_+, \boldsymbol{\mathcal{S}}_+)$.
Scaled gradient updates: for $t = 0, 1, 2, \ldots, T-1$, **do**

$$
\begin{aligned}
\boldsymbol{U}_{t+} &= \boldsymbol{U}_t - \frac{\eta}{p}\mathcal{M}_1\left(\mathcal{P}_\Omega((\boldsymbol{U}_t, \boldsymbol{V}_t, \boldsymbol{W}_t)\cdot\boldsymbol{\mathcal{S}}_t) - \boldsymbol{\mathcal{Y}}\right)\check{\boldsymbol{U}}_t(\check{\boldsymbol{U}}_t^\top\check{\boldsymbol{U}}_t)^{-1},\\
\boldsymbol{V}_{t+} &= \boldsymbol{V}_t - \frac{\eta}{p}\mathcal{M}_2\left(\mathcal{P}_\Omega((\boldsymbol{U}_t, \boldsymbol{V}_t, \boldsymbol{W}_t)\cdot\boldsymbol{\mathcal{S}}_t) - \boldsymbol{\mathcal{Y}}\right)\check{\boldsymbol{V}}_t(\check{\boldsymbol{V}}_t^\top\check{\boldsymbol{V}}_t)^{-1},\\
\boldsymbol{W}_{t+} &= \boldsymbol{W}_t - \frac{\eta}{p}\mathcal{M}_3\left(\mathcal{P}_\Omega((\boldsymbol{U}_t, \boldsymbol{V}_t, \boldsymbol{W}_t)\cdot\boldsymbol{\mathcal{S}}_t) - \boldsymbol{\mathcal{Y}}\right)\check{\boldsymbol{W}}_t(\check{\boldsymbol{W}}_t^\top\check{\boldsymbol{W}}_t)^{-1},\\
\boldsymbol{\mathcal{S}}_{t+} &= \boldsymbol{\mathcal{S}}_t - \frac{\eta}{p}\left((\boldsymbol{U}_t^\top\boldsymbol{U}_t)^{-1}\boldsymbol{U}_t^\top, (\boldsymbol{V}_t^\top\boldsymbol{V}_t)^{-1}\boldsymbol{V}_t^\top, (\boldsymbol{W}_t^\top\boldsymbol{W}_t)^{-1}\boldsymbol{W}_t^\top\right)\\
&\qquad\cdot\left(\mathcal{P}_\Omega((\boldsymbol{U}_t, \boldsymbol{V}_t, \boldsymbol{W}_t)\cdot\boldsymbol{\mathcal{S}}_t) - \boldsymbol{\mathcal{Y}}\right),
\end{aligned}
\tag{45}
$$

where $\check{\boldsymbol{U}}_t$, $\check{\boldsymbol{V}}_t$, and $\check{\boldsymbol{W}}_t$ are defined in (6). Set

$$(\boldsymbol{U}_{t+1}, \boldsymbol{V}_{t+1}, \boldsymbol{W}_{t+1}, \boldsymbol{\mathcal{S}}_{t+1}) = \mathcal{P}_B(\boldsymbol{U}_{t+}, \boldsymbol{V}_{t+}, \boldsymbol{W}_{t+}, \boldsymbol{\mathcal{S}}_{t+}).$$

Algorithms	Sample complexity	Iteration complexity	Parameter space
Unfolding + nuclear norm min. [29]	$n^2 r \log^2 n$	polynomial	tensor
Tensor nuclear norm min. [65]	$n^{3/2} r^{1/2} \log^{3/2} n$	NP-hard	tensor
Grassmannian GD [60]	$n^{3/2} r^{7/2} \kappa^4 \log^{7/2} n$	N/A	factor
`ScaledGD`	$n^{3/2} r^{5/2} \kappa^3 \log^3 n$	$\log \frac{1}{\varepsilon}$	factor

Table 2 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 3.1). 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.

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 [2]. Compared with existing algorithms collected in Table 2, `ScaledGD` is the *first* algorithm that simultaneously achieves a near-optimal sample complexity and a near-linear run time complexity in a provable manner.

4 Preconditioning Meets Overparameterization

In this section we treat a more complicated scenario, where the correct rank r of the ground truth $\boldsymbol{X}_\star$ is not known *a priori*. In this case, a practical solution is to *overparameterize*, i.e., to choose some $r' > r$, and proceed as if r' is the correct rank. It turns out that `ScaledGD` needs some simple modification to work robustly in this setting. The modified algorithm, which we call `ScaledGD`(λ), will be introduced in the rest of this section, along with theoretical analysis on its global convergence.

Motivation. We begin with inspecting the behavior of ScaledGD in the overparameterized setting. Assume we already find some $\boldsymbol{L}_t \in \mathbb{R}^{n_1\times r'}, \boldsymbol{R}_t \in \mathbb{R}^{n_2\times r'}$ that are close to the ground truth: the first r columns of $\boldsymbol{L}_t$ are close to $\boldsymbol{L}_\star$, while the rest $r'-r$ columns are close to zero; *mutatis mutandis* for $\boldsymbol{R}_t$.

Recall that in the update equation (3) of ScaledGD:

$$\begin{aligned}\boldsymbol{L}_{t+1} &= \boldsymbol{L}_t - \eta\nabla_{\boldsymbol{L}}\mathcal{L}(\boldsymbol{L}_t,\boldsymbol{R}_t)(\boldsymbol{R}_t^\top\boldsymbol{R}_t)^{-1},\\ \boldsymbol{R}_{t+1} &= \boldsymbol{R}_t - \eta\nabla_{\boldsymbol{R}}\mathcal{L}(\boldsymbol{L}_t,\boldsymbol{R}_t)(\boldsymbol{L}_t^\top\boldsymbol{L}_t)^{-1},\end{aligned}$$

the preconditioners are chosen to be the inverse of the $r'\times r'$ matrices $\boldsymbol{L}_t^\top\boldsymbol{L}_t, \boldsymbol{R}_t^\top\boldsymbol{R}_t$. However, since the last $r'-r$ columns for $\boldsymbol{L}_t$ (respectively, $\boldsymbol{R}_t$) are close to zero, it is clear that $\boldsymbol{L}_t^\top\boldsymbol{L}_t$ (respectively, $\boldsymbol{R}_t^\top\boldsymbol{R}_t$) are approximately of rank at most r. When $r'>r$, this means $\boldsymbol{L}_t^\top\boldsymbol{L}_t$ and $\boldsymbol{R}_t^\top\boldsymbol{R}_t$ are close to being degenerate since their approximate ranks (no larger than r) are smaller than their dimensions r', thus taking inverse of them are numerically unstable.

ScaledGD(λ): ScaledGD with overparameterization. One of the simplest remedies to such instability is to *regularize* the preconditioner. Before taking inverse of $\boldsymbol{L}_t^\top\boldsymbol{L}_t$ and $\boldsymbol{R}_t^\top\boldsymbol{R}_t$, we add a regularizer $\lambda\boldsymbol{I}$ to avoid degeneracy, where $\lambda>0$ is a regularization parameter; the preconditioner thus becomes $(\boldsymbol{L}_t^\top\boldsymbol{L}_t+\lambda\boldsymbol{I})^{-1}$ and $(\boldsymbol{R}_t^\top\boldsymbol{R}_t+\lambda\boldsymbol{I})^{-1}$. The new update rule is specified by

$$\begin{aligned}\boldsymbol{L}_{t+1} &= \boldsymbol{L}_t - \eta\nabla_{\boldsymbol{L}}\mathcal{L}(\boldsymbol{L}_t,\boldsymbol{R}_t)(\boldsymbol{R}_t^\top\boldsymbol{R}_t+\lambda\boldsymbol{I})^{-1},\\ \boldsymbol{R}_{t+1} &= \boldsymbol{R}_t - \eta\nabla_{\boldsymbol{R}}\mathcal{L}(\boldsymbol{L}_t,\boldsymbol{R}_t)(\boldsymbol{L}_t^\top\boldsymbol{L}_t+\lambda\boldsymbol{I})^{-1}.\end{aligned} \tag{46}$$

This regularized version of ScaledGD is called ScaledGD(λ). It turns out that the simple regularization trick works out well: ScaledGD(λ) not only (almost) inherits the κ-free convergence rate, but also has the advantage of being robust to overparameterization and enjoying provable global convergence to any prescribed accuracy level, when initialized by a small random initialization. This is established in [63] formally for the matrix sensing setting studied in Section 2.4, where the two factors $\boldsymbol{L}_\star$, $\boldsymbol{R}_\star$ in $\boldsymbol{X}_\star=\boldsymbol{L}_\star\boldsymbol{R}_\star^\top$ are equal, i.e. $\boldsymbol{X}_\star=\boldsymbol{L}_\star\boldsymbol{L}_\star^\top$. Under such situation, ScaledGD(λ) instantiates to

$$\boldsymbol{L}_{t+1} = \boldsymbol{L}_t - \eta\mathcal{A}^*(\mathcal{A}(\boldsymbol{L}_t\boldsymbol{L}_t^\top)-\boldsymbol{y})\boldsymbol{L}_t(\boldsymbol{L}_t^\top\boldsymbol{L}_t+\lambda\boldsymbol{I})^{-1}. \tag{47}$$

Similar to [38, 50], we set the initialization $\boldsymbol{L}_0$ to be a small random matrix, i.e.,

$$\boldsymbol{L}_0 = \alpha\boldsymbol{G}, \tag{48}$$

where $\boldsymbol{G}\in\mathbb{R}^{n\times r'}$ is some matrix considered to be normalized and $\alpha>0$ controls the magnitude of the initialization. To simplify exposition, we take $\boldsymbol{G}$ to be a standard random Gaussian matrix, that is, $\boldsymbol{G}$ is a random matrix with i.i.d. entries distributed as $\mathcal{N}(0,1/n)$.

Theoretical guarantee. We have the performance guarantee of ScaledGD(λ) under the standard RIP assumption as follows.

Theorem 7 *Suppose that $\mathcal{A}(\cdot)$ satisfies the rank-$(r+1)$ RIP with $\delta_{r+1}=:\delta$. Furthermore, there exist a sufficiently small constant $c_\delta>0$ and a sufficiently large constant $C_\delta>0$ such that*

$$\delta \le c_\delta r^{-1/2}\kappa^{-C_\delta}. \tag{49}$$

Assume there exist some universal constants $c_\eta, c_\lambda, C_\alpha>0$ such that (η,λ,α) in ScaledGD(λ) satisfy the following conditions:

$$\text{(learning rate)}\qquad \eta \le c_\eta, \tag{50a}$$

$$\text{(damping parameter)}\qquad \frac{1}{100}c_\lambda\sigma_{\min}^2(\boldsymbol{L}_\star) \le \lambda \le c_\lambda\sigma_{\min}^2(\boldsymbol{L}_\star), \tag{50b}$$

$$\text{(initialization size)}\qquad \log\frac{\|\boldsymbol{L}_\star\|}{\alpha} \ge \frac{C_\alpha}{\eta}\log(2\kappa)\cdot\log(2\kappa n). \tag{50c}$$

With high probability (with respect to the realization of the random initialization $\boldsymbol{G}$), there exists a universal constant $C_{\min} > 0$ such that for some $T \le T_{\min} := \frac{C_{\min}}{\eta} \log \frac{\|\boldsymbol{L}_\star\|}{\alpha}$, the iterates $\boldsymbol{L}_t$ of (47), obey

$$\|\boldsymbol{L}_T \boldsymbol{L}_T^\top - \boldsymbol{X}_\star\|_{\mathsf{F}} \le \alpha^{1/3} \|\boldsymbol{L}_\star\|^{5/3}.$$

In particular, for any prescribed accuracy target $\epsilon \in (0,1)$, by choosing a sufficiently small α fulfilling both (50c) and $\alpha \le \epsilon^3 \|\boldsymbol{L}_\star\|$, we have $\|\boldsymbol{L}_T \boldsymbol{L}_T^\top - \boldsymbol{X}_\star\|_{\mathsf{F}} \le \epsilon \|\boldsymbol{X}_\star\|$.

Theorem 7 shows that by choosing an appropriate α, `ScaledGD(`λ`)` finds an ϵ-accurate solution, i.e., $\|\boldsymbol{L}_T \boldsymbol{L}_T^\top - \boldsymbol{X}_\star\|_{\mathsf{F}} \le \epsilon \|\boldsymbol{X}_\star\|$, in no more than an order of

$$\log \kappa \cdot \log(\kappa n) + \log(1/\epsilon)$$

iterations. Roughly speaking, this asserts that `ScaledGD(`λ`)` converges at a constant linear rate independent of the condition number κ after an initial phase of approximately $O(\log \kappa \cdot \log(\kappa n))$ iterations. In contrast, overparameterized `GD` requires $O(\kappa^4 + \kappa^3 \log(\kappa n/\epsilon))$ iterations to converge from a small random initialization to ϵ-accuracy; see [50, 38]. Thus, the convergence of overparameterized `GD` is much slower than `ScaledGD(`λ`)` even for mildly ill-conditioned matrices. Furthermore, our sample complexity depends only on the true rank r, but not on the overparameterized rank r' — a crucial feature in order to provide meaningful guarantees when the overparameterized rank r' is uninformative of the true rank, i.e., close to the full dimension n. The dependency on κ in the sample complexity, on the other end, has been generally unavoidable in nonconvex low-rank estimation [19].

5 Numerical Experiments

We illustrate the performance of `ScaledGD` and `ScaledGD(`λ`)` via a real data experiment to highlight the consideration of rank selection, when the data matrix is approximately low-rank, a scenario which occurs frequently in practice. We consider a dataset[2] that measures chlorine concentrations in a drinking water distribution system, over different junctions and recorded once every 5 minutes during 15 days. The data matrix of interest, $\boldsymbol{X}_\star \in \mathbb{R}^{120\times 180}$, corresponds to the data extracted at 120 junctions over 15 hours. Fig. 3 plots the spectrum of $\boldsymbol{X}_\star$, where its singular values decay rapidly, suggesting it can be well approximated by a low-rank matrix.

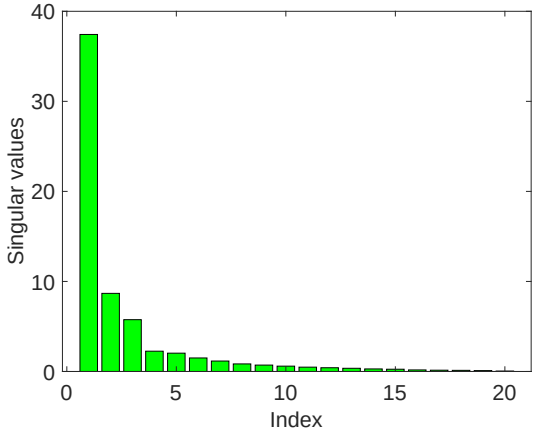
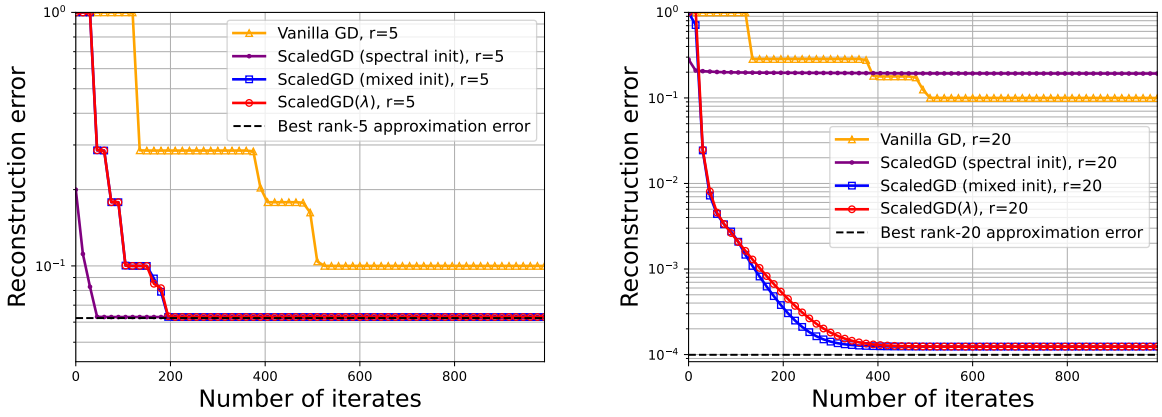


Fig. 3 The spectrum of the chlorine concentration data matrix, where its singular values decay rapidly.

[2] The dataset can be accessed from http://www.cs.cmu.edu/afs/cs/project/spirit-1/www/

Since the data matrix $\boldsymbol{X}_\star$ is not exactly low-rank, the choice of the rank r determines how good the low-rank approximation is as well as the condition number κ: choosing a larger r leads to a lower approximation error but also a higher κ. We explore the behavior of `ScaledGD` and `ScaledGD(`λ`)` in comparison with vanilla gradient descent under different choices of r in Fig. 4. Moreover, apart from the spectral initialized `ScaledGD`, we consider yet another variant of `ScaledGD` which we found useful in practice: we start with `ScaledGD(`λ`)` for a few iterations, and switch to `ScaledGD` after it is detected $\sigma_{\min}^2(\boldsymbol{L}_t) \gtrsim \lambda$. We call this variant `ScaledGD` with *mixed initialization*. The philosophy of this variant will be introduced after we discuss the results in Fig. 4.

We consider the matrix completion setting, where we randomly observe 80% of the entries in the data matrix. Fig. 4 (a) illustrates the performance of different algorithms when $r = 5$. `ScaledGD` with spectral initialization achieves the fastest convergence, while `ScaledGD(`λ`)` and `ScaledGD` with mixed initialization take a few more iterations at the beginning to warm up. All variants of `ScaledGD` converge considerably faster than vanilla GD and approach the optimal rank-r approximation error. Fig. 4 (b) illustrates the performance of different algorithms when $r = 20$, where the situation becomes different: `ScaledGD` with spectral initialization no longer converges, while `ScaledGD(`λ`)` and `ScaledGD` with mixed initialization still demonstrate fast convergence to the optimal rank-r approximation error. Vanilla GD, on the other hand, is still significantly slower.



(a) $r = 5$ (b) $r = 20$

Fig. 4 Performance comparison of different algorithms for matrix completion on the chlorine concentration dataset, under $r = 5$ (a) and $r = 20$ (b).

The experimental results help to explain the motivation of using `ScaledGD` with mixed initialization. The reason for the instability of spectrally initialized `ScaledGD` for large r stems from the fact that spectral initialization does not cope well with overparameterization. On the other hand, small random initialization is known to help stabilize with overparameterization [38], but does not integrate with `ScaledGD` since the preconditioner $(\boldsymbol{L}_0^\top \boldsymbol{L}_0)^{-1}$ at the first iteration would be extremely large if the initialization $\boldsymbol{L}_0$ were small. Therefore, initializing with `ScaledGD(`λ`)`, which is provably robust against overparameterization and can be integrated perfectly into `ScaledGD`, becomes a reasonable choice.

6 Conclusions

This chapter highlights a novel approach to provably accelerate ill-conditioned low-rank estimation via `ScaledGD`. Its fast convergence, together with low computational and memory costs by operating in the factor space, makes it a highly scalable and desirable method in practice. The performance of `ScaledGD` is also robust when the data matrix is only approximately low-rank and the observations are noisy; we refer interested readers to [63] for further details. In terms of future directions, it is of great interest to explore the design of effective preconditioners for

other statistical estimation and learning tasks, as well as further understand the implications of preconditioning in the presence of overparameterization for the asymmetric setting.

Acknowledgements This work is supported in part by Office of Naval Research under N00014-19-1-2404, and by National Science Foundation under CCF-1901199, DMS-2134080 and ECCS-2126634 to Y. Chi.

References

1. Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks*, 2(1):53–58, 1989.
2. Boaz Barak and Ankur Moitra. Noisy tensor completion via the sum-of-squares hierarchy. In *Conference on Learning Theory*, pages 417–445. PMLR, 2016.
3. Srinadh Bhojanapalli, Behnam Neyshabur, and Nati Srebro. Global optimality of local search for low rank matrix recovery. In *Advances in Neural Information Processing Systems*, pages 3873–3881, 2016.
4. Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, 2018.
5. Changxiao Cai, Gen Li, Yuejie Chi, H Vincent Poor, and Yuxin Chen. Subspace estimation from unbalanced and incomplete data matrices: $\ell_{2,\infty}$ statistical guarantees. *The Annals of Statistics*, 49(2):944–967, 2021.
6. Jian-Feng Cai, Jingyang Li, and Dong Xia. Generalized low-rank plus sparse tensor estimation by fast Riemannian optimization. *arXiv preprint arXiv:2103.08895*, 2021.
7. Emmanuel Candès, Xiaodong Li, and Mahdi Soltanolkotabi. Phase retrieval via Wirtinger flow: Theory and algorithms. *IEEE Transactions on Information Theory*, 61(4):1985–2007, 2015.
8. Emmanuel J. Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM*, 58(3):11:1–11:37, 2011.
9. 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.
10. Venkat Chandrasekaran, Sujay Sanghavi, Pablo Parrilo, and Alan Willsky. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2):572–596, 2011.
11. 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.
12. Ji Chen and Xiaodong Li. Model-free nonconvex matrix completion: Local minima analysis and applications in memory-efficient kernel PCA. *Journal of Machine Learning Research*, 20(142):1–39, 2019.
13. Ji Chen, Dekai Liu, and Xiaodong Li. Nonconvex rectangular matrix completion via gradient descent without $\ell_{2,\infty}$ regularization. *IEEE Transactions on Information Theory*, 66(9):5806–5841, 2020.
14. Yudong Chen. Incoherence-optimal matrix completion. *IEEE Transactions on Information Theory*, 61(5):2909–2923, 2015.
15. Yudong Chen and Yuejie Chi. Harnessing structures in big data via guaranteed low-rank matrix estimation: Recent theory and fast algorithms via convex and nonconvex optimization. *IEEE Signal Processing Magazine*, 35(4):14 – 31, 2018.
16. Yudong Chen and Martin J Wainwright. Fast low-rank estimation by projected gradient descent: General statistical and algorithmic guarantees. *arXiv preprint arXiv:1509.03025*, 2015.
17. Yuxin Chen, Yuejie Chi, Jianqing Fan, Cong Ma, and Yuling Yan. Noisy matrix completion: Understanding statistical guarantees for convex relaxation via nonconvex optimization. *SIAM Journal on Optimization*, 30(4):3098–3121, 2020.
18. Yuxin Chen, Jianqing Fan, Cong Ma, and Yuling Yan. Bridging convex and nonconvex optimization in robust PCA: Noise, outliers, and missing data. *The Annals of Statistics*, 49(5):2948–2971, 2021.
19. 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.
20. Damek Davis, Dmitriy Drusvyatskiy, and Courtney Paquette. The nonsmooth landscape of phase retrieval. *IMA Journal of Numerical Analysis*, 40(4):2652–2695, 2020.
21. Harry Dong, Tian Tong, Cong Ma, and Yuejie Chi. Fast and provable tensor robust principal component analysis via scaled gradient descent. *Information and Inference: A Journal of the IMA*, 12(3):iaad019, 2023.
22. Simon S Du, Wei Hu, and Jason D Lee. Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced. In *Advances in Neural Information Processing Systems*, pages 384–395, 2018.
23. Abraham Frandsen and Rong Ge. Optimization landscape of Tucker decomposition. *Mathematical Programming*, pages 1–26, 2020.
24. Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points-online stochastic gradient for tensor decomposition. In *Conference on Learning Theory (COLT)*, pages 797–842, 2015.
25. Rong Ge, Chi Jin, and Yi Zheng. No spurious local minima in nonconvex low rank problems: A unified geometric analysis. In *International Conference on Machine Learning*, pages 1233–1242, 2017.
26. Rong Ge, Jason D Lee, and Tengyu Ma. Matrix completion has no spurious local minimum. In *Advances in Neural Information Processing Systems*, pages 2973–2981, 2016.

27. 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, 2022.
28. Moritz Hardt and Mary Wootters. Fast matrix completion without the condition number. In *Proceedings of The 27th Conference on Learning Theory*, pages 638–678, 2014.
29. Bo Huang, Cun Mu, Donald Goldfarb, and John Wright. Provable models for robust low-rank tensor completion. *Pacific Journal of Optimization*, 11(2):339–364, 2015.
30. Prateek Jain and Purushottam Kar. Non-convex optimization for machine learning. *Foundations and Trends® in Machine Learning*, 10(3-4):142–336, 2017.
31. 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.
32. Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing*, pages 665–674, 2013.
33. Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M Kakade, and Michael I Jordan. How to escape saddle points efficiently. In *International Conference on Machine Learning*, pages 1724–1732, 2017.
34. Hiroyuki Kasai and Bamdev Mishra. Low-rank tensor completion: a Riemannian manifold preconditioning approach. In *International Conference on Machine Learning*, pages 1012–1021, 2016.
35. Kenji Kawaguchi. Deep learning without poor local minima. In *Advances in Neural Information Processing Systems*, pages 586–594, 2016.
36. 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.
37. Yuanxin Li, Cong Ma, Yuxin Chen, and Yuejie Chi. Nonconvex matrix factorization from rank-one measurements. *IEEE Transactions on Information Theory*, 67(3):1928–1950, 2021.
38. Yuanzhi Li, Tengyu Ma, and Hongyang Zhang. Algorithmic regularization in over-parameterized matrix sensing and neural networks with quadratic activations. In *Conference On Learning Theory*, pages 2–47. PMLR, 2018.
39. Yuetian Luo and Anru R Zhang. Low-rank tensor estimation via Riemannian Gauss-Newton: Statistical optimality and second-order convergence. *arXiv preprint arXiv:2104.12031*, 2021.
40. Cong Ma, Yuanxin Li, and Yuejie Chi. Beyond Procrustes: Balancing-free gradient descent for asymmetric low-rank matrix sensing. *IEEE Transactions on Signal Processing*, 69:867–877, 2021.
41. 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.
42. Song Mei, Yu Bai, and Andrea Montanari. The landscape of empirical risk for nonconvex losses. *The Annals of Statistics*, 46(6A):2747–2774, 2018.
43. Yurii Nesterov and Boris T Polyak. Cubic regularization of Newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.
44. Praneeth Netrapalli, UN Niranjan, Sujay Sanghavi, Animashree Anandkumar, and Prateek Jain. Non-convex robust PCA. In *Advances in Neural Information Processing Systems*, pages 1107–1115, 2014.
45. Dohyung Park, Anastasios Kyrillidis, Constantine Carmanis, and Sujay Sanghavi. Non-square matrix sensing without spurious local minima via the Burer-Monteiro approach. In *Artificial Intelligence and Statistics*, pages 65–74, 2017.
46. Garvesh Raskutti, Ming Yuan, and Han Chen. Convex regularization for high-dimensional multiresponse tensor regression. *The Annals of Statistics*, 47(3):1554–1584, 2019.
47. Holger Rauhut, Reinhold Schneider, and Željka Stojanac. Low rank tensor recovery via iterative hard thresholding. *Linear Algebra and its Applications*, 523:220–262, 2017.
48. 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.
49. Laixi Shi and Yuejie Chi. Manifold gradient descent solves multi-channel sparse blind deconvolution provably and efficiently. *IEEE Transactions on Information Theory*, 67(7):4784–4811, 2021.
50. Dominik Stöger and Mahdi Soltanolkotabi. Small random initialization is akin to spectral learning: Optimization and generalization guarantees for overparameterized low-rank matrix reconstruction. *Advances in Neural Information Processing Systems*, 34:23831–23843, 2021.
51. Ju Sun, Qing Qu, and John Wright. Complete dictionary recovery using nonconvex optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 2351–2360, 2015.
52. Ju Sun, Qing Qu, and John Wright. A geometric analysis of phase retrieval. *Foundations of Computational Mathematics*, 18(5):1131–1198, 2018.
53. Ruoyu Sun and Zhi-Quan Luo. Guaranteed matrix completion via non-convex factorization. *IEEE Transactions on Information Theory*, 62(11):6535–6579, 2016.
54. Tian Tong, Cong Ma, and Yuejie Chi. Accelerating ill-conditioned low-rank matrix estimation via scaled gradient descent. *Journal of Machine Learning Research*, 22(150):1–63, 2021.
55. 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.
56. Stephen Tu, Ross Boczar, Max Simchowitz, Mahdi Soltanolkotabi, and Benjamin Recht. Low-rank solutions of linear matrix equations via Procrustes flow. In *International Conference Machine Learning*, pages 964–973, 2016.
57. Ledyard R Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

58. Haifeng Wang, Jinchi Chen, and Ke Wei. Implicit regularization and entrywise convergence of Riemannian optimization for low tucker-rank tensor completion. *arXiv preprint arXiv:2108.07899*, 2021.
59. 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.
60. Dong Xia and Ming Yuan. On polynomial time methods for exact low-rank tensor completion. *Foundations of Computational Mathematics*, 19(6):1265–1313, 2019.
61. 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):76–99, 2021.
62. Dong Xia, Anru R Zhang, and Yuchen Zhou. Inference for low-rank tensors–no need to debias. *arXiv preprint arXiv:2012.14844*, 2020.
63. Xingyu Xu, Yandi Shen, Yuejie Chi, and Cong Ma. The power of preconditioning in overparameterized low-rank matrix sensing. *arXiv preprint arXiv:2302.01186*, 2023.
64. Xinyang Yi, Dohyung Park, Yudong Chen, and Constantine Caramanis. Fast algorithms for robust PCA via gradient descent. In *Advances in Neural Information Processing Systems*, pages 4152–4160, 2016.
65. Ming Yuan and Cun-Hui Zhang. On tensor completion via nuclear norm minimization. *Foundations of Computational Mathematics*, 16(4):1031–1068, 2016.
66. Anru Zhang and Dong Xia. Tensor SVD: Statistical and computational limits. *IEEE Transactions on Information Theory*, 64(11):7311–7338, 2018.
67. 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.
68. Qinqing Zheng and John Lafferty. Convergence analysis for rectangular matrix completion using Burer-Monteiro factorization and gradient descent. *arXiv preprint arXiv:1605.07051*, 2016.
69. Zhihui Zhu, Qiuwei Li, Gongguo Tang, and Michael B Wakin. Global optimality in low-rank matrix optimization. *IEEE Transactions on Signal Processing*, 66(13):3614–3628, 2018.