Fast and Accurate Estimation of Low-Rank Matrices from Noisy Measurements via Preconditioned Non-Convex Gradient Descent

Gavin Zhang **Hong-Ming Chiu** **Richard Y. Zhang**
Department of Electrical and Computer Engineering, University of Illinois-Urbana Champaign

Abstract

Non-convex gradient descent is a common approach for estimating a low-rank $n \times n$ ground truth matrix from noisy measurements, because it has per-iteration costs as low as $O(n)$ time, and is in theory capable of converging to a minimax optimal estimate. However, the practitioner is often constrained to just tens to hundreds of iterations, and the slow and/or inconsistent convergence of non-convex gradient descent can prevent a high-quality estimate from being obtained. Recently, the technique of *preconditioning* was shown to be highly effective at accelerating the local convergence of non-convex gradient descent when the measurements are noiseless. In this paper, we describe how preconditioning should be done for noisy measurements to accelerate local convergence to minimax optimality. For the symmetric matrix sensing problem, our proposed preconditioned method is guaranteed to locally converge to minimax error at a linear rate that is immune to ill-conditioning and/or over-parameterization. Using our proposed preconditioned method, we perform a 60 megapixel medical image denoising task, and observe significantly reduced noise levels compared to previous approaches.

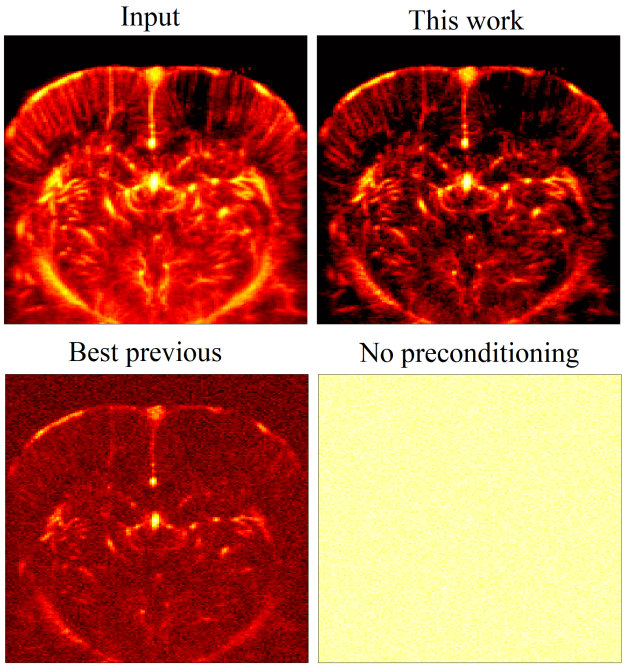

Figure 1: **Preconditioned gradient descent for a 60 megapixel medical image denoising task.** We denoise a 2400-frame ultrafast ultrasound image of a rat rain (200×130 pixels per frame) by running 30 iterations of the low-rank denoising procedure in Demené et al. (2015). Top-left: original noisy input. Top-right: image denoised and reconstructed by our preconditioning scheme in (3). Bottom-left: image obtained via the preconditioning scheme in Zhang et al. (2021), which is the previous state-of-the-art. Bottom-right: image obtained by naive non-convex gradient descent without preconditioning.

1 INTRODUCTION

We consider the *low-rank matrix recovery* problem, which seeks to recover an $n \times n$ ground truth matrix $M^\star$ of low rank $r^\star$, given a small number m of measurement matrices A_i and noisy observations $y_i = \langle A_i, M^\star \rangle + \varepsilon_i$, for indices $i \in \{1, \dots, m\}$. The main challenge lies in the fact that the presence of measurement noise ε_i reduces the "amount of useful information" contained within the observations; it usually becomes impossible to recover $M^\star$ exactly. Instead, one aims to compute a *minimax optimal* estimate $M \approx M^\star$, which is roughly defined as the closest estimate of the ground truth $M^\star$ for the worst-possible scenario Candes and Plan (2010, 2011). Informally, a minimax optimal estimate is the best achievable given the finite amount of useful information contained with the noisy observations Lehmann and Casella (2006).

Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238.

Minimax optimal estimations are highly desirable for real-world applications of low-rank matrix recovery. In medical imaging, for example, minimax optimality would assure the highest possible level of reconstruction accuracy, in order to minimize the chances of diagnostic errors, detect subtle changes or anomalies, and reduce the need for repeated scans or reanalysis. In response, an extensive body of literature has been developed over the past two decades on techniques for solving low-rank matrix recovery to guaranteed minimax optimality; see our detailed literature review in Section 1.4 below. Unfortunately, despite significant progress, existing state-of-the-art algorithms still have trouble achieving minimax optimality on many real-world applications, due to two perennial difficulties.

The first perennial difficulty is the enormous scale of real-world datasets. Today, the most common approach is *non-convex gradient descent* (Zheng and Lafferty (2015); Zhao et al. (2015); Tu et al. (2016); Sun and Luo (2016)), or factored gradient descent (Chen and Wainwright (2015); Park et al. (2017, 2018)), which is to factor a candidate estimate $M = UV^T$, and to directly optimize over its $n \times r$ low-rank factor matrices U, V, as in

$$\min_{U,V \in \mathbb{R}^{n\times r}} f(U,V) = \frac{1}{m}\sum_{i=1}^{m}(y_i - \langle A_i, UV^T\rangle)^2 \quad (1)$$

using an iterative local optimization algorithm like gradient descent

$$\begin{aligned} U_{\text{new}} &= U - \alpha\nabla_U f(U,V), \\ V_{\text{new}} &= V - \alpha\nabla_V f(U,V), \end{aligned} \quad (2)$$

in which $\alpha \in (0,1]$ is the step-size / learning rate. With a small rank $r \ll n$, each iteration costs as low as $O(m+n)$ time and memory, and so in principle, the approach can scale to arbitrarily large values of m and n. But real-world datasets routinely have m and n on the order of tens to hundreds of millions, and in practice, even a single iteration can take many minutes to several hours.

The considerable expense of performing even a single iteration often constrains the practitioner to just a few tens to low hundreds of iterations. But this further exacerbates the second perennial difficulty, which is the inconsistent and sometimes extremely slow convergence of non-convex gradient descent. While the method is known to converge to minimax optimality given a *sufficiently large* number of iterations (Chen and Wainwright (2015)), for many real-world datasets it is unable to do so with a *reasonable* number of iterations.

1.1 Accelerating convergence via preconditioning

Recently, there has been exciting progress on the use of *preconditioning* to accelerate the local convergence of non-convex gradient descent for low-rank matrix recovery (Mishra et al. (2012); Tong et al. (2020); Zhang et al. (2021, 2022); Xu et al. (2023); Zhang et al. (2023)). This line of work is motivated by the observation that real-world ground truth matrices $M^\star$ often have excessively large condition numbers $\kappa = \lambda_1(M^\star)/\lambda_r(M^\star)$. In particular, if the search rank is over-parameterized as $r > r^\star$ with respect to the unknown true rank $r^\star$, then the condition number even diverges as $\kappa \to \infty$. Non-convex gradient descent is known to locally converge with a linear convergence rate like $\rho = 1 - c/\kappa$ with absolute constant $c > 0$ (Zheng and Lafferty (2015); Tu et al. (2016)), and therefore experiences a significant slow-down with an excessively large or even diverging κ. In both cases, suitable preconditioning was shown to restore the linear convergence rate back to $\rho = 1 - c$ (Tong et al. (2020); Zhang et al. (2021)), as if the condition number were perfect $\kappa = 1$.

However, the existing literature on preconditioning primarily focuses on the *noiseless* instance of low-rank matrix recovery, which assumes $\varepsilon_i = 0$ for all i. Indeed, it remains unclear how preconditioning should be done in the presence of measurement noise $\varepsilon_i \neq 0$. Experimentally, existing preconditioning methods for the noiseless case do not consistently accelerate convergence in the presence of noise; the acceleration is either lost at a much coarser error level than minimax optimal, or the iterates sporadically diverge. (See our experiments in Section 4)

The existence of a significant gap between the noiseless and noisy cases is less surprising if we consider the underlying mechanism that allow preconditioning to work in the first place. Intuitively, preconditioning works by inverting one ill-conditioned matrix against another ill-conditioned matrix, in order to "cancel out" their ill-conditioning and obtain a well-conditioned matrix. This mechanism necessarily requires a precise alignment between the two matrices; in the presence of noise, slight "misalignments" can nullify the cancellation and render the preconditioning ineffective, or at worst even amplify the noise and cause divergence to occur.

1.2 Our contribution: How to precondition in the presence of measurement noise?

Our primary goal in this paper is to provide a *principled* way to perform preconditioning on non-convex gradient descent, that is both effective and reliable for real-world applications of *noisy* low-rank matrix recovery. To this

end, we propose the following iterations for solving (1):

$$\begin{aligned} U_{\text{new}} &= U - \alpha \nabla_U f(U,V)(V^TV + \eta I)^{-1}, \\ V_{\text{new}} &= V - \alpha \nabla_V f(U,V)(U^TU + \eta I)^{-1}, \\ \eta_{\text{new}} &= \beta \eta, \end{aligned} \tag{3}$$

where $\alpha \in (0,1]$ is the step-size / learning rate as before in (2), and $\beta \in [0,1)$ is a geometric decay rate for the regularization parameter η. Here, each gradient $\nabla_U f(U,V)$ and $\nabla_V f(U,V)$ is an $n \times r$ matrix, so the equation (2) says that each $r \times r$ matrix preconditioner $(V^TV + \eta I)^{-1}$ and $(U^TU + \eta I)^{-1}$ should be applied as a right matrix-matrix product onto the respective gradient. It is easily verified that the additional overhead of computing and applying the preconditioner is $O(r^3 + nr^2) = O(n)$ time and $O(r^2) = O(1)$ memory, again assuming a small rank $r \ll n$.

The basic form of the preconditioned iterations (3) is reminescent of previous work on noiseless low-rank matrix recovery (Mishra et al. (2012); Mishra and Sepulchre (2016); Tong et al. (2020); Zhang et al. (2021, 2022, 2023); Xu et al. (2023)). In the noisy setting, we provide strong theoretical and empirical evidence to argue that the most principled way to adjust the regularization parameter η is to make it decay geometrically. This is in contrast to prior work, that either set $\eta = \sqrt{f(U,V)}$ at each iteration (Zhang et al. (2021)), or simply fix η to a constant for all iterations (Xu et al. (2023)). Our main message in this paper is that choosing η correctly has a significant and outsized impact on the quality of acceleration in the noisy setting; the previous choices deliver an acceleration only at coarser error levels, and could even cause divergence. As shown in Figure 1, our method significantly improves upon the previous state-of-the-art on a real-world instance of low-rank matrix recovery arising in medical image denoising.

Rigorously, we prove, for the *symmetric matrix sensing* instance of low-rank matrix recovery, that the preconditioned iterations in (3) with a geometrically decaying η locally converges to minimax optimality, at an accelerated linear rate that is immune to ill-conditioning and over-parameterization. It was previously shown that, if the measurements $A_1, \dots, A_m$ satisfy the restricted isometry property (RIP) and that the measurement noise come from a zero-mean Gaussian $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$, then non-convex gradient descent with rank $r = O(r^\star)$ converges to an estimate M with Frobenius norm error $\|M - M^\star\|_F = O(\sigma^2 n r^\star \log n)$ (Chen and Wainwright (2015); Zhuo et al. (2021)), which is indeed minimax optimal up to log factors (Candes and Plan (2011)). However, the actual convergence rate can be dramatically slowed by ill-conditioning and over-parameterization, to be as slow as sublinear (Zhuo et al. (2021); Zhang et al. (2021)). A variant of (3) known as PrecGD was proposed in Zhang et al. (2021) to accelerate convergence to minimax error, but this prior work required either perfect knowledge of the noise variance σ^2 (which is clearly unreasonable in practice), or a complicated and expensive cross-validation procedure to estimate the noise variance σ^2.

Our main result is that, under the same setting as the above, the preconditioned iterations in (3) with geometrically decaying η is guaranteed to converge to minimax optimal error, at the same local convergence rate as non-convex gradient descent with a perfect condition number $\kappa = 1$. In particular: (i) the accelerated convergence rate is maintained all the way down to the minimax optimal error level of $O(\sigma^2 n r^\star)$; (ii) the acceleration is applicable to all initial points within a neighborhood of the ground truth. To the best of our knowledge, our result is the first to rigorously guarantee both two properties. Our analysis reveals a simple mechanism that explains the inconsistent performance of previous preconditioners in the noisy setting: the accelerated convergence is only maintained until the current error norm $\|UV^T - M^\star\|_F$ reaches the same order of magnitude as the regularization parameter η, but an excessively small η can actually cause the iterations to diverge. Therefore, the most natural and principled way to set η is to allow it to geometrically decay alongside the error norm $\|UV^T - M^\star\|_F$, which is exactly what we proposed in (3).

1.3 Limitations

There are two main limitations with our theoretical analysis. First, we make two idealized assumptions (via the symmetric matrix sensing problem) that may not be satisfied in real-world datasets: (i) the underlying ground truth $M^\star$ is symmetric positive definite; (ii) that the measurements A_i satisfy RIP. Here, we emphasize that the purpose of our theoretical analysis is to provide a *rigorous justification* for the geometric decay of the regularization parameter η; in particular, it is not to guarantee performance on real-world datasets, as these will rarely satisfy the idealized assumptions (like RIP and incoherence) needed for a theoretical analysis to be possible. The symmetry assumption is primarily made to simplify our presentation; it can be mechanically overcome by repeating the analyses in Park et al. (2017); Tong et al. (2020), but we expect our conclusions to transfer largely verbatim to the non-symmetric case. We leave the extension of our theoretical results to the non-symmetric RIP setting as future work, and emphasize that our large-scale experiments on real-world datasets are indeed performed for a non-symmetric dataset that does not satisfy RIP.

Second, our theoretical analysis focuses on local convergence given a sufficient good initialization. For the

sake of an end-to-end guarantee, we initialize our theoretical analysis using the standard technique of spectral initialization (Tu et al. (2016); Chen et al. (2021a)). In practice, the enormous scale of real-world datasets often constrains the practitioner to just a few tens to low hundreds of iterations, so *heuristic warm starts* are widely used to maximize the effectiveness of these few iterations (Bercoff et al. (2011); Zhang et al. (2019)). It is important to point out that slow convergence remains a critical issue even when a high-quality heuristic warm start is provided, as further progress towards minimax optimality is slowed by the slow convergence of the iterative algorithm. In this regard, our work in this paper answers the practical question: "given a heuristic warm start, how do we refine this warm start to the best accuracy possible, while using as few iterations as possible?"

1.4 Related Work

Non-convex gradient descent converges to minimax optimality For a wide range of problems relating to low-rank matrix recovery, if we are given a warm-start solution, local refinements via GD is often capable of converging towards the ground truth. This has been shown rigorously for matrix sensing (Tu et al. (2016); Zheng and Lafferty (2015); Charisopoulos et al. (2021)), matrix completion (Sun et al. (2015); Jain et al. (2013); Chen et al. (2020)), phase retrieval (Candes et al. (2015); Netrapalli et al. (2013); Ma et al. (2018)) and other related problems (Li et al. (2019); Yi et al. (2016); Chen et al. (2021b)).

For matrix sensing in particular, both Chen and Wainwright (2015) and Zhuo et al. (2021) showed that gradient descent achieves a statistical error of $O(\sigma^2 nr \log n)$, where r is the search rank. Under the assumption that $r = O(r^\star)$, this error matches the minimax error noted in Candes and Plan (2011) up to log factors. In this work we prove that our method converges to the same statistical error as both Chen and Wainwright (2015); Zhuo et al. (2021), under exactly the same assumptions. In other words, our method does preconditioning without amplifying the statistical error at all. Under the warm-start setting, the question of whether the assumption $r = O(r^\star)$ is necessary for achieving minimax error is an open question even without preconditioning, and we do not attempt to resolve it in this work.

Accelerating local convergence via preconditioning The basic idea to precondition the gradient against $(V^TV)^{-1}$ and $(U^TU)^{-1}$ was first suggested in Mishra et al. (2012), and its convergence properties for the noiseless case were later studied in detail in Tong et al. (2020) resulting in a method known as ScaledGD. Its extension to SGD was first proposed in Mishra and Sepulchre (2016), and studied in Zhang et al. (2022). The idea to regularize with an identity perturbation and precondition against $(V^TV + \eta I)^{-1}$ and $(U^TU + \eta I)^{-1}$ was first suggested in Zhang et al. (2021) as a means to counteract the effects of over-parameterization $r > r^\star$, which resulted in a method known as PrecGD. A similar regularization was subsequently studied in Xu et al. (2023).

All of these methods, as well as our proposed method, are able to overcome the slow convergence of non-convex gradient descent in the the noiseless setting. However, previous methods do not provide theoretical guarantees in the noisy setting. Empirically, their behaviors are inconsistent under noise. Specifically, the ScaledGD of Tong et al. (2020) can diverge in the case $r > r^*$. While this divergence is avoided by adding a regularization parameter as in Zhang et al. (2013) and Xu et al. (2023), the regularization parameter itself can cause these methods to stagnate at a higher noise level, as seen in our experimental section (Figures 2 and 3). In contrast, our method is the only one that maintains its acceleration all the way down to minimax optimality.

Small random initialization While most early work in non-convex gradient descent focused on local refinement of a warm-start initialization, a separate line of recent work focused on using a *small* random initialization (Li et al. (2018); Stöger and Soltanolkotabi (2021); Ma and Fattahi (2021); Ding et al. (2021); Jin et al. (2023)). For matrix sensing in particular, global convergence of GD was first proven in Li et al. (2018) in the case $r = n$, and later refined in Stöger and Soltanolkotabi (2021) for the general over-parameterized case. Similar results have also been obtained in the asymmetric case (Jiang et al. (2023); Chou et al. (2023)). For preconditioned methods, a similar analysis has been done in Xu et al. (2023) for a variant of the ScaledGD Tong et al. (2020). In the noisy setting, Ding et al. (2022) first showed that GD with small random initialization converges to the minimax error by extending the analysis in Stöger and Soltanolkotabi (2021). A major strength of their theoretical analysis is that it no longer requires the assumption $r = O(r^*)$.

However, we emphasize that these theoretical results for small random initialization are not directly comparable to the results in this work. First, we provide theoretical guarantees for all initializations close to the ground truth. In contrast, small initialization relies on tracing a very specific and rapidly converging trajectory. In fact, we believe that this is the main reason that small random initialization achieves minimax optimality without requiring $r = O(r^*)$. However, in order to trace this specific trajectory, small random initialization forces an already good initial solution to

be thrown away. In our experience, this means that GD has to use many more iterations to get back the warm-start that was thrown away (see Figure 4).

Notations We use $\|\cdot\|_F$ to denote the Frobenius norm and $\|\cdot\|$ to denote the spectral norm of a matrix. We use $\langle A, B\rangle = \mathrm{tr}(A^T B)$ to denote the standard matrix inner product. We use $\lesssim$ to denote an inequality that hides a constant factor. For a scalar function $f : \mathbb{R}^{n\times r} \to \mathbb{R}$, the gradient $\nabla f(X)$ is a matrix of size $n \times r$. For any matrix M, the eigenvalues and singular values are denoted by $\lambda_i(M)$ and $\sigma_i(M)$, arranged in decreasing order.

2 MAIN RESULTS

In our theoretical analysis, we consider the variant of low-rank matrix recovery known as *symmetric matrix sensing*, which aims to recover a positive semidefinite, rank-$r^\star$ ground truth matrix $M^\star \succeq 0$, from a small number m of possibly noisy measurements $y = \mathcal{A}(M^\star)+\varepsilon$, where the linear measurement operator $\mathcal{A}$ is defined

$$\mathcal{A}(M^\star) = [\langle A_1, M^\star\rangle, \langle A_2, M^\star\rangle, \ldots, \langle A_m, M^\star\rangle]^T.$$

Without loss of generality, we assume that the measurement matrices A_i are symmetric. In addition, we will adopt the standard assumption that the unknown measurement noise modeled via the length-m vector ϵ is normally distributed

$$\epsilon_i \sim \mathcal{N}(0, \sigma^2) \text{ for all } i \in \{1, \ldots, m\}$$

and that $\mathcal{A}$ satisfies the *restricted isometry property* (RIP) Candes (2008).

Definition 2.1 (Restricted Isometry)**.** The linear operator $\mathcal{A}$ satisfies RIP with parameters $(2r, \delta)$ if there exists constant $0 \le \delta < 1$ such that, for every rank-$2r$ matrix M, we have

$$(1-\delta)\|M\|_F^2 \le \|\mathcal{A}(M)\|^2 \le (1+\delta)\|M\|_F^2. \quad (4)$$

Specifically, we will always assume throughout the paper that $\mathcal{A}$ satisfies RIP with parameters $(2r, \delta)$. We note that the RIP assumption is in line with existing work on the statistical optimality of gradient descent (Chen and Wainwright (2015); Zhuo et al. (2021)) and preconditioned gradient descent (Tong et al. (2020); Zhang et al. (2021)). Under the mild assumption $r = O(r^*)$, it is also in line with prior work on convex methods (Candes (2008); Candes and Plan (2010); Candès et al. (2011)) and small random initialization (Li et al. (2018); Stöger and Soltanolkotabi (2021); Ma and Fattahi (2021); Ding et al. (2021); Jin et al. (2023); Xu et al. (2023)).

Given a warm-start close to the ground truth, our goal is to refine this warm-start by using gradient descent. In particular, we want to minimize the non-convex loss function in (1) up to minimax optimal error.

Since the ground truth $M^\star$ and the measurement matrices A_i's are symmetric, if both the left and right factors U, V in (3) start at the same initial point, they will always stay the same. Therefore, if we denote $X = U = V$, with $X \in \mathbb{R}^{n\times r}$, then the iterations simplify to

$$\begin{aligned} X_{\text{new}} &= X - \alpha\nabla f(X)(X^T X + \eta I)^{-1} \\ \eta_{\text{new}} &= \beta\eta \end{aligned}. \quad (5)$$

Now we are ready to state our main result, which says that our algorithm always converges linearly to the minimax optimal error, at a linear rate that is affected by neither ill-conditioning nor over-parameterization.

In particular, in equation (5), let the initial regularization η_0 satisfy $\eta_0 \ge 2\sqrt{f_c(X_0)}$, and let the decay rate β satisfy $1 > \beta \ge \sqrt{1 - \frac{\mu}{4L}}$. Here $f_c(X_0)$ is the noiseless function value defined in (6), and μ, L are both constants depending only on r, r^* and δ, which we define rigorously in Appendix A.2. Then we have the following result.

Theorem 2.1. *Suppose that the initial point X_0 satisfies $\|\mathcal{A}(X_0X_0^T - M^*)\|^2 < \rho^2(1-\delta)\lambda_{r^*}(M^\star)^2$ with a radius $\rho > 0$ that satisfies $\rho^2/(1-\rho^2) \le (1-\delta^2)/2$. Let the step-size α satisfy $\alpha \le 1/L$, where $L > 0$ is a constant that only depends on δ. At the t-th iteration, with high probability, we have*

$$\|X_t X_t^T - M^\star\|_F^2 \lesssim \max\left\{\beta^{2t}\cdot\|X_0X_0^T - M^\star\|_F^2, \mathcal{E}_{opt}\right\},$$

where $\mathcal{E}_{opt} = \frac{\sigma^2 nr\log n}{m}$. Here the inequality $\lesssim$ hides a constant that only depends on δ.

A complete proof of Theorem 2.1 is presented in the appendix. In the next section, we sketch out the key ideas behind its proof. First, we make a few important observations.

In Theorem 2.1, we require an initial point that satisfies $\|\mathcal{A}(X_0X_0^T - M^*)\|^2 < \rho^2(1-\delta)\lambda_{r^*}(M^\star)^2$. This requirement is standard and appeared in all previous works on preconditioned methods (Tong et al. (2020); Zhang et al. (2021, 2023)). The only exception is Xu et al. (2023), which uses a small random initialization. In practice, a common way to obtain such an initial point is through domain specific heuristics. In ultrafast ultrasound (Bercoff et al. (2011)) for instance, the noisy version of the ultrasound image itself can serve as a warm-start, since it is already close to the ground truth. However, even without heuristics, such an initial point can be achieved using spectral initialization

(see Proposition 6 of Zhang et al. (2021)), in which we simply need to compute one SVD factorization.

In addition, we also need a decay rate that satisfies $1 > \beta \geq \sqrt{1 - \frac{\mu}{4L}}$. Although μ and L are in general hard to estimate, we find that in practice β is extremely robust. In our experiments, any value of β satisfying $0.5 \leq \beta < 1$ was sufficient for linear convergence.

We also note that in Theorem 2.1, the convergence rate is crucially independent of the condition number κ, since μ and L has no dependence on κ. In addition, the statistical error that our algorithm converges to exactly matches that of Chen and Wainwright (2015) and Zhuo et al. (2021), which proved that GD converges to an error of $O(\sigma^2 nr \log n)$. In other words, our preconditioner exponentially accelerates GD without amplifying the noise at all. Under the mild assumption $r = O(r^*)$, this rate matches the minimax rate in Candes and Plan (2011).

3 KEY IDEA and PROOF SKETCH

The recent work of Zhang et al. (2021) proposed a preconditioned variant of gradient descent called PrecGD to restore its linear convergence rate:

$$X_{t+1} = X_t - \alpha \nabla f(X_t)(X_t^T X_t + \eta_t I)^{-1},$$
$$C_{lb}\|X_t X_t^T - M^\star\| \leq \eta_t \leq C_{ub}\|X_t X_t^T - M^\star\|,$$

where C_{lb}, C_{ub} are fixed constants. To understand why our algorithm achieves minimax optimality and immunity to ill-conditioning and over-parameterization all at the same time, it is instructive to first see how PrecGD can fail in the noisy case.

To maintain linear convergence, the key contribution of Zhang et al. (2021) is the crucial observation that the regularization parameter η_t must be within a constant factor of the error $\|X_t X_t^T - M^\star\|$. In the noiseless case, simply setting $\eta_t = \sqrt{f(X_t)}$ will imply $\eta_t = \Theta(\|X_t X_t^T - M^\star\|)$. However, in the noisy setting finding the right η_t requires an accurate estimate of the noise variance, which is in general very difficult.

3.1 Key Innovations

Maintaining the right amount of regularization of is the most important ingredient for our method to succeed. The regularization η_t used in PrecGD has to be perfect because linear convergence requires two contradictory properties to intersect, namely gradient dominance (also known as the PL-inequality) and Lipschitz smoothness. When η_t is too large, gradient dominance is lost. When η_t is too small, Lipschitz smoothness is lost. The analysis in Zhang et al. (2021) suggests that the choice of η_t is extremely fragile and delicate.

Surprisingly, we find that this is not the case. In fact, we will show that the choice of η_t is not delicate, but rather robust. Our method avoids the need to choose the optimal regularization parameter altogether by simply letting η decay with some rate $\beta < 1$. It turns out that this extremely simple choice of the regularization parameter will *automatically* maintain the right amount of regularization needed for linear convergence due to a phenomenon we call "coupling".

This phenomenon can be intuitively understood as a race in which the two runners η_t and $\mathcal{E}_t = \|X_t X_t^T - M^\star\|_F$, are connected using a rubber band. When η_t and $\mathcal{E}_t$ begin to grow apart, the rubber band will exert a counteracting force and pull them back together. As a result, the amount of regularization is always right. This happens because that η_t itself controls how fast $\mathcal{E}_t$ decays. If the regularization parameter η_t is large compared to $\mathcal{E}_t$, then our algorithm behaves more like gradient descent. As a result, our algorithm briefly stagnates, allowing η_t to catch up and become close to $\mathcal{E}_t$ again. Similarly, if η_t is small, our algorithm begins to converge faster. Thus, the error $\mathcal{E}_t$ decays quickly, and will eventually catch up to η_t.

This coupling of the regularization parameter and the error is precisely why we can avoid the expensive procedure used in PrecGD to estimate the noise variance and approximate $\mathcal{E}_t$. We *implicitly* maintain the right amount of regularization, so that our algorithm always converges linearly, even in ill-conditioned, over-parameterized, and noisy settings.

3.2 Proof Sketch

In this section we sketch the main steps of the proof of our main result, Theorem 2.1, and defer the full proof to the appendix. Our proof consists of two components: the first is the observation that the PL-inequality, which is lost in the case $r > r^\star$, can be restored under a change of norm, as long as the preconditioner $P = (X_t^T X_t + \eta_t \cdot I)^{-1}$ has the "correct" amount of regularization η_t. The second component is the observation that η_t and $\mathcal{E}_t = \|X_t X_t^T - M^\star\|$ are coupled together, meaning that they can never be too far apart.

To begin, note that the objective function in (1) can be written as

$$f(X) = f_c(X) + \frac{\|\varepsilon\|^2}{m} - \frac{2}{m}\langle \mathcal{A}(XX^T - M^\star), \varepsilon \rangle, \quad (6)$$

where $f_c(X) = \frac{1}{m}\|\mathcal{A}(XX^T - M^\star)\|^2$ is defined to be the objective function with clean measurements that are not corrupted by noise.

The first component in our proof consists of showing that the iterates of (5) can be viewed as gradient descent under a change of norm. In particular, let

$P = X^T X + \eta_t I$ be a real symmetric, positive definite $r \times r$ matrix. We define a corresponding P-norm and its dual P-norm on $\mathbb{R}^{n\times r}$ as follows

$$\|X\|_P \overset{\text{def}}{=} \|XP^{1/2}\|, \quad \|X\|_{P*} \overset{\text{def}}{=} \|XP^{-1/2}\|. \quad (7)$$

Consider a descent direction D. Suppose that the following inequality holds with some constant L:

$$f_c(X - \alpha D) \le f_c(X) - \alpha\langle \nabla f_c(X), D\rangle + \frac{\alpha^2 L}{2}\|D\|_P^2 \quad (8)$$

and that the PL-inequality holds under the P-norm: $\|\nabla f_c(X)\|_{P*}^2 \ge \mu(f_c(X))$ with $\mu > 0$. Then plugging in the descent direction $D = \nabla f_c(X)P^{-1}$ yields linear convergence since $f_c(X - \alpha D) \le \left(1 - \frac{\mu}{2L}\right) f_c(X)$.

Therefore, to complete our proof, we need to demonstrate the following conditions:

1. The inequality (8) holds with some constant L
2. The PL-inequality holds under the P-norm: $\|\nabla f_c(X)\|_{P*}^2 \ge \mu_(f_c(X))$.

First, we consider an ideal case: suppose that there exists some constant $C > 1$ such that $\eta_t \le \sqrt{f_c(X_t)} \le C\eta_t$. This is exactly the regime for η_t where our algorithm is well-behaved: both Lipschitz gradients and the PL-inequality is satisfied in the P-norm. The proof of these two facts, especially the second one, is quite involved, but it is similar to the proof of Corollary 5 in Zhang et al. (2021), so they are deferred to the appendix.

As a result, in the ideal case where $\eta_t \le \sqrt{f_c(X_t)} \le C\eta_t$, if we go in the direction $D = \nabla f_c(X)P^{-1}$, then linear convergence is already achieved. However, due to noise, the descent direction is $\nabla f(X)$, instead of $\nabla f_c(X)$, since we cannot access the true gradient. Fortunately, if the norm of the gradient is large compared to a statistical error, we can prove that the difference between $\nabla f_c(X)$ and $\nabla f(X)$ is negligible, and our algorithm will still make enough progress at each iteration to ensure linear convergence.

Essentially, if $\eta_t \approx \sqrt{f_c(X_t)}$, then our algorithm will converge linearly up to some statistical error. Unfortunately, $\eta_t \approx \sqrt{f_c(X_t)}$ does not always hold, because *both* η_t and $\sqrt{f_c(X_t)}$ are changing. Instead, we have to consider scenarios where η_t deviates from this ideal range. To complete the proof of Theorem 2.1, we need to show that η_t never deviates too far from the ideal range. This is the key difficulty in our proof. Intuitively, if $\sqrt{f_c(X_{t+1})}$ becomes too small compared to η_t, our algorithm will start to behave more like gradient descent and slow down. Hence with a fixed decay rate, η_t will quickly be on the same order as $\sqrt{f_c(X_{t+1})}$ again. As a result, linear convergence is always maintained.

4 NUMERICAL SIMULATIONS

In this section, we compare our method against two state of the art preconditioned methods: PrecGD Zhang et al. (2021) and ScaledGD(λ) Xu et al. (2023). To validate our theoretical results, we first perform experiments using Gaussian measurements on a synthetic low-rank matrix. In addition, we also perform experiments on a real-world medical imaging application, specifically in denoising an ultrafast ultrasound scan as shown in Figure 1. We leave the details of this experiment to the appendix.

With spectral initialization, we show that our method is the only algorithm that is able to consistently achieve minimax error at a linear rate. In Figures 2 and 3, both PrecGD and ScaledGD(λ) have trouble converging to

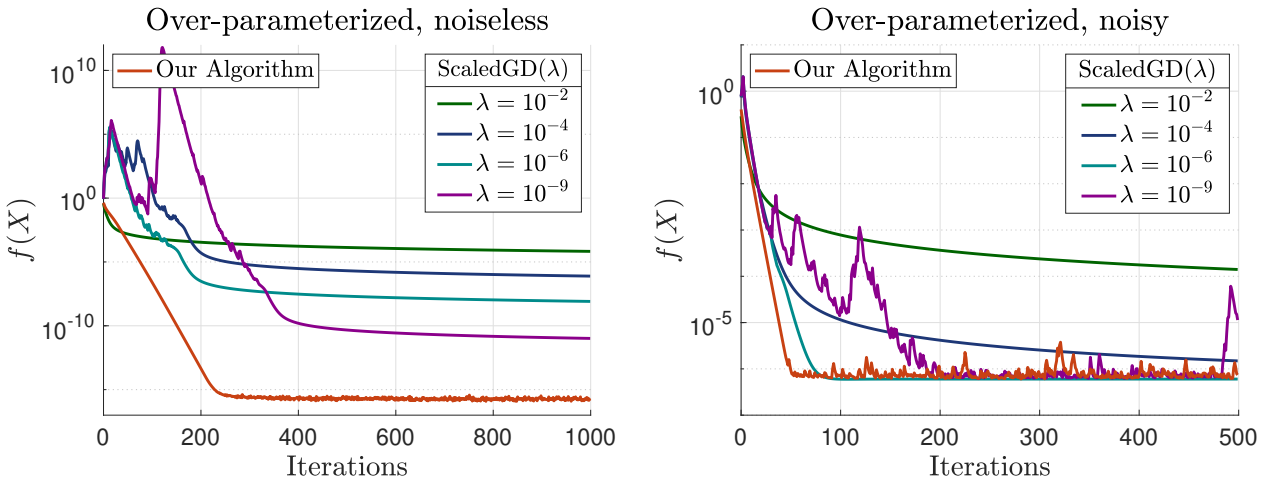

Figure 2: **Convergence of our algorithm and ScaledGD(λ) using spectral initialization**. Left: Noiseless measurements. Right: Noisy measurements with noise variance $\sigma = 10^{-6}$.

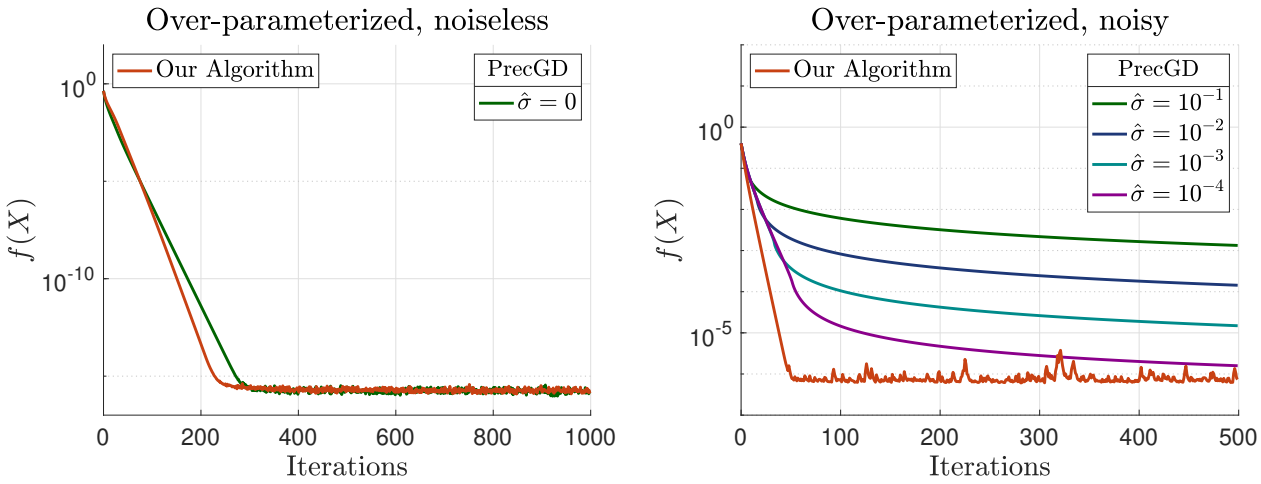

Figure 3: **Convergence of our algorithm and PrecGD using spectral initialization**. Left: Noiseless measurements. Right: Noisy measurements with noise variance $\sigma = 10^{-6}$.

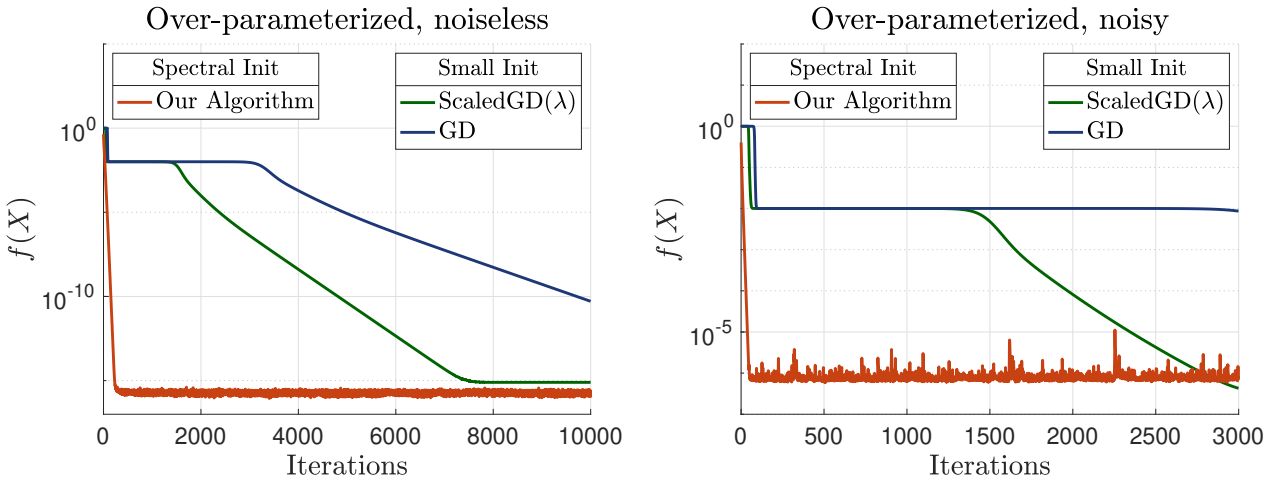

Figure 4: **Convergence of our algorithm (spectral init.), ScaledGD(λ) (small init.) and GD (small init.) for Gaussian matrix sensing**. Left: Noiseless measurements. Right: Noisy measurements with noise variance $\sigma = 10^{-6}$.

the minimax optimal unless a perfect regularization parameter is chosen. Without the perfect choice, they either stagnate at a high noise level, or even diverge.

In Figure 4, we see that with small initialization, both ScaledGD(λ) and GD seem to stagnate for a significantly long time before converging to the next eigenvalue. Therefore, in cases where a good heuristic initialization is available, throwing such a initialization away and using small initialization instead can come at a great cost, since it can take many iterations to get it back.

4.1 Gaussian matrix sensing

In this experiment, we consider a matrix recovery problem on a 10×10 ground truth matrix $M^\star$ with truth rank $r^\star = 2$. The condition number of $M^\star$ is set to $\kappa = 10^2$. We take measurements on $M^\star$ using linearly independent measurement matrices $A_1, \ldots, A_m$ drawn from the standard Gaussian distribution. In Figure 2, 3 and 4, we set the search rank to be $r = 8$ and draw $m = 2nr$ measurements from $M^\star$. In noisy setting, we corrupt the measurements with noise $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ where $\sigma = 10^{-6}$. For our algorithm, we set $\eta_0 = \sqrt{f(X_0)}$ and the decay rate for η_t as $\beta = 0.85$ in noiseless case, and $\beta = 0.5$ in noisy case.

Our algorithm v.s. ScaledGD(λ) In Figure 2, we plot the convergence of our algorithm and ScaledGD(λ) under four values of $\lambda = \{10^{-2}, 10^{-4}, 10^{-6}, 10^{-9}\}$. We spectrally initialized both methods at the same initial point and set the learning rate to be $\alpha = 0.1$. In noiseless case, we see that our algorithm converges to minimax error at a linear rate, while ScaledGD(λ) converges to an error of $O(\lambda)$. It is important to note

that we cannot set λ to be too small because it would cause ScaledGD(λ) to diverge or become numerically unstable, as depicted in Figure 2. In noisy case, we obtain similar results as in the noiseless case. The only difference is that ScaledGD(λ) can converge to the same error as our algorithm when $\lambda = \{10^{-6}, 10^{-9}\}$ as the minimax error in the noisy case is around 10^{-6}. However, we again observe that ScaledGD(λ) becomes numerically unstable when λ is too small.

Our algorithm v.s. PrecGD In Figure 3, we plot the convergence of our algorithm and PrecGD. We spectrally initialize both methods at the same initial point and set the learning rate to be $\alpha = 0.1$. Here, PrecGD is implemented with a proxy variance $\hat{\sigma}$ so that $\eta_t = \sqrt{|f(X_t) - \hat{\sigma}^2|}$. We see that our algorithm converges linearly to the minimax error in both the noiseless and noisy case. While PrecGD also converges linearly to the minimax error in the noiseless case, in the noisy case, however, its error depends crucially on the value of proxy variance $\hat{\sigma}$; it requires $\hat{\sigma} \approx \sigma$ to achieve minimax error.

Small init. v.s. spectral init. In Figure 4, we plot the convergence of our algorithm, ScaledGD(λ) and GD. In this experiment, our algorithm is initialized using spectral initialization, and both ScaledGD(λ) and GD are initialized using small initialization with initialization scale 10^{-12}. We set the learning rate for all three methods to be $\alpha = 0.1$. Again, our algorithm converges linearly to the minimax error in both noiseless and noisy setting. Both ScaledGD(λ) and GD learn the solution incrementally (see Jin et al. (2023) for a precise characterization of incremental learning) and hence reach the minimax error a lot slower than our algorithm.

ACKNOWLEDGMENTS

The authors are grateful to Pengfei Song and YiRang Shin for their help and advice on the ultrafast ultrasound application, and for sharing the rat brain datasets used in our medical imaging experiments.

Financial support for this work was provided in part by NSF CAREER Award ECCS-2047462.

References

Jeremy Bercoff, Gabriel Montaldo, Thanasis Loupas, David Savery, Fabien Mézière, Mathias Fink, and Mickael Tanter. Ultrafast compound doppler imaging: Providing full blood flow characterization. *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, 58(1):134–147, 2011.

Emmanuel J Candes. The restricted isometry property and its implications for compressed sensing. *Comptes rendus mathematique*, 346(9-10):589–592, 2008.

Emmanuel J Candes and Yaniv Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6): 925–936, 2010.

Emmanuel J Candes 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.

Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):1–37, 2011.

Emmanuel J Candes, Xiaodong Li, and Mahdi Soltanolkotabi. Phase retrieval via wirtinger flow: Theory and algorithms. *IEEE Transactions on Information Theory*, 61(4):1985–2007, 2015.

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*, 21(6):1505–1593, 2021.

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.

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.

Yuxin Chen, Yuejie Chi, Jianqing Fan, Cong Ma, et al. Spectral methods for data science: A statistical perspective. *Foundations and Trends® in Machine Learning*, 14(5):566–806, 2021a.

Yuxin Chen, Jianqing Fan, Cong Ma, and Yuling Yan. Bridging convex and nonconvex optimization in robust pca: Noise, outliers, and missing data. *Annals of statistics*, 49(5):2948, 2021b.

Hung-Hsu Chou, Johannes Maly, and Dominik Stöger. How to induce regularization in generalized linear models: A guide to reparametrizing gradient flow. *arXiv preprint arXiv:2308.04921*, 2023.

Charlie Demené, Thomas Deffieux, Mathieu Pernot, Bruno-Félix Osmanski, Valérie Biran, Jean-Luc Gennisson, Lim-Anna Sieu, Antoine Bergel, Stephanie Franqui, Jean-Michel Correas, et al. Spatiotemporal clutter filtering of ultrafast ultrasound data highly increases doppler and fultrasound sensitivity. *IEEE transactions on medical imaging*, 34(11):2271–2285, 2015.

Lijun Ding, Liwei Jiang, Yudong Chen, Qing Qu, and Zhihui Zhu. Rank overspecified robust matrix recovery: Subgradient method and exact recovery. *arXiv preprint arXiv:2109.11154*, 2021.

Lijun Ding, Zhen Qin, Liwei Jiang, Jinxin Zhou, and Zhihui Zhu. A validation approach to over-parameterized matrix and image recovery. *arXiv preprint arXiv:2209.10675*, 2022.

Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 665–674, 2013.

Liwei Jiang, Yudong Chen, and Lijun Ding. Algorithmic regularization in model-free overparametrized asymmetric matrix factorization. *SIAM Journal on Mathematics of Data Science*, 5(3):723–744, 2023.

Jikai Jin, Zhiyuan Li, Kaifeng Lyu, Simon Shaolei Du, and Jason D Lee. Understanding incremental learning of gradient descent: A fine-grained analysis of matrix sensing. In *International Conference on Machine Learning*, pages 15200–15238. PMLR, 2023.

Erich L Lehmann and George Casella. *Theory of point estimation*. Springer Science & Business Media, 2006.

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.

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.

Cong Ma, Kaizheng Wang, Yuejie Chi, and Yuxin Chen. Implicit regularization in nonconvex statistical estimation: Gradient descent converges linearly for phase retrieval and matrix completion. In *International Conference on Machine Learning*, pages 3345–3354. PMLR, 2018.

Jianhao Ma and Salar Fattahi. Implicit regularization of sub-gradient method in robust matrix recovery: Don't be afraid of outliers. *arXiv preprint arXiv:2102.02969*, 2021.

Bamdev Mishra and Rodolphe Sepulchre. Scaled stochastic gradient descent for low-rank matrix completion. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 2820–2825. IEEE, 2016.

Bamdev Mishra, K Adithya Apuroop, and Rodolphe Sepulchre. A riemannian geometry for low-rank matrix completion. *arXiv preprint arXiv:1211.1550*, 2012.

Yurii Nesterov. *Lectures on convex optimization*, volume 137. Springer, 2018.

Praneeth Netrapalli, Prateek Jain, and Sujay Sanghavi. Phase retrieval using alternating minimization. *Advances in Neural Information Processing Systems*, 26, 2013.

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. PMLR, 2017.

Dohyung Park, Anastasios Kyrillidis, Constantine Caramanis, and Sujay Sanghavi. Finding low-rank solutions via nonconvex matrix factorization, efficiently and provably. *SIAM Journal on Imaging Sciences*, 11(4):2165–2204, 2018.

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.

Ju Sun, Qing Qu, and John Wright. When are nonconvex problems not scary? *arXiv preprint arXiv:1510.06096*, 2015.

Ruoyu Sun and Zhi-Quan Luo. Guaranteed matrix completion via non-convex factorization. *IEEE Transactions on Information Theory*, 62(11):6535–6579, 2016.

Tian Tong, Cong Ma, and Yuejie Chi. Accelerating ill-conditioned low-rank matrix estimation via scaled gradient descent. *arXiv preprint arXiv:2005.08898*, 2020.

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

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.

Xinyang Yi, Dohyung Park, Yudong Chen, and Constantine Caramanis. Fast algorithms for robust pca via gradient descent. *Advances in neural information processing systems*, 29, 2016.

Gavin Zhang, Hong-Ming Chiu, and Richard Y Zhang. Accelerating sgd for highly ill-conditioned huge-scale online matrix completion. *arXiv preprint arXiv:2208.11246*, 2022.

Gavin Zhang, Salar Fattahi, and Richard Y Zhang. Preconditioned gradient descent for overparameterized

nonconvex burer-monteiro factorization with global optimality certification. *J. Mach. Learn. Res.*, 24: 163–1, 2023.

Jialun Zhang, Salar Fattahi, and Richard Y Zhang. Preconditioned gradient descent for over-parameterized nonconvex matrix factorization. *Advances in Neural Information Processing Systems*, 34:5985–5996, 2021.

Min Zhang, Zheng-Hai Huang, and Ying Zhang. Restricted p-isometry properties of nonconvex matrix recovery. *IEEE Transactions on Information Theory*, 59(7):4316–4323, 2013.

Richard Y Zhang, Javad Lavaei, and Ross Baldick. Spurious local minima in power system state estimation. *IEEE transactions on control of network systems*, 6 (3):1086–1096, 2019.

Tuo Zhao, Zhaoran Wang, and Han Liu. A nonconvex optimization framework for low rank matrix estimation. *Advances in Neural Information Processing Systems*, 28, 2015.

Qinqing Zheng and John Lafferty. A convergent gradient descent algorithm for rank minimization and semidefinite programming from random linear measurements. *arXiv preprint arXiv:1506.06081*, 2015.

Jiacheng Zhuo, Jeongyeol Kwon, Nhat Ho, and Constantine Caramanis. On the computational and statistical complexity of over-parameterized matrix sensing. *arXiv preprint arXiv:2102.02756*, 2021.

A Proof of Main Results

A.1 Prelminaries

In addition to the notation used in the main paper, we define some additional notation that will be used throughout the appendix. Let X by an $n \times r$ matrix, and let $P \succ 0$ be a fixed $r \times r$ positive definite matrix. We define a corresponding P-norm and its dual P-norm on $\mathbb{R}^{n\times r}$ as follows

$$\|X\|_P \stackrel{\text{def}}{=} \|XP^{1/2}\|, \quad \|X\|_{P*} \stackrel{\text{def}}{=} \|XP^{-1/2}\|. \tag{9}$$

We use $\text{vec}(X)$ to denote the vectorization operator that stacks the column of X into a single column vector. As before, we use $\otimes$ to denote the Kronecker product between two matrices. For a scalar-valued function of a matrix, $f(X)$, we use $\nabla^2 f(X)[V]$ to denote the Hessian vector product, defined by

$$\nabla^2 f(X)[V] = \lim_{t\to 0} \frac{\nabla f(X+tV) - \nabla f(X)}{t}.$$

Note that here $\nabla^2 f(X)[V]$ is a matrix of the same size as $\nabla f(X)$. With a slight abuse of notation, we use lower case letters x_t to denote the vectorized version of X_t, so $x_t = \text{vec}(X_t)$. We denote the corresponding gradient by $\nabla f(x_t)$.

For symmetric matrix sensing, we denote our ground truth by $M^\star = ZZ^T \in \mathbb{R}^{n\times n}$. We denote the true rank by $r^* = \text{rank}(M^*)$. Our goal is to recovery $M^\star$ from a small number of measurements of the form $y = \mathcal{A}(ZZ^T) + \epsilon \in \mathbb{R}^m$. Here ϵ is a vector with independent Gaussian entries with zero mean and variance σ^2. To do so, we minimize the non-convex objective function

$$f(X) = \frac{1}{m}\|\mathcal{A}(XX^T) - y\|^2 = f_c(X) + \frac{1}{m}\|\epsilon\|^2 - \frac{2}{m}\langle \mathcal{A}(XX^T - M^\star), \epsilon\rangle,$$

where $f_c(X) \stackrel{def}{=} \frac{1}{m}\|\mathcal{A}(XX^T - M^\star)\|^2$ is the objective function with clean measurements that are not corrupted with noise. Here X is a matrix of size $n \times r$, where r is known as the search rank.

We make a few additional simplifications on notations. As before, we will use α to denote the step-size and D to denote the local search direction. In our proof below, it will often be easier to use the *vectorized* form of our gradient updates: vectorizing both sides of the update $X_{t+1} = X_t - \alpha\nabla f(X_t)(X_t^T X_t + \eta_t I)^{-1}$, we get

$$\begin{aligned}\text{vec}\, X_{t+1} &= \text{vec}\, X_t - \alpha\, \text{vec}\, \nabla f(X_t)(X_t^T X_t + \eta_t I)^{-1} \\ &= \text{vec}\, X_t - ((X_t^T X_t + \eta_t I)\otimes I)^{-1}\, \text{vec}(\nabla f(X_t)),\end{aligned}$$

where in the second line we used the standard identity $\text{vec}(AXB) = (B^T \otimes A)\,\text{vec}(X)$ for the Kronecker product. Using lower case letters x and d to refer to $\text{vec}\,(X)$ and $\text{vec}\,(D)$ respectively, the update above can be written as $x_{t+1} = x_t - \alpha\mathbf{P}^{-1}\nabla f(x_t)$, with $\mathbf{P} = (X_t^T X_t + \eta_t I)\otimes I$.

A.2 Auxiliary Results

In this section we collect two results from Zhang et al. (2021) that will be used in the proof of our main result. The first theorem shows that when the regularization η is small, the PL-inequality holds within a small neighborhood around the ground truth.

Theorem A.1 (Noiseless gradient dominance)**.** *Let $\min_X f(X) = 0$ for $M^\star \neq 0$. Suppose that X satisfies $f(X) \le \rho^2 \cdot (1-\delta)\lambda^2_{r^\star}(M^\star)$ with radius $\rho > 0$ that satisfies $\rho^2/(1-\rho^2) \le (1-\delta^2)/2$. Then, we have*

$$\eta \le C_{ub}\|XX^T - M^\star\|_F \quad \Longrightarrow \quad \|\nabla f(X)\|^2_{P*} \ge 2\mu f(X)$$

where

$$\mu = \left(\sqrt{\frac{1+\delta^2}{2}} - \delta\right)^2 \cdot \min\left\{\left(\frac{C_1}{\sqrt{2}-1}\right)^{-1}, \left(1 + 3C_1\sqrt{\frac{(r-r^\star)}{1-\delta^2}}\right)^{-1}\right\}. \tag{10}$$

Here C_1 is a constant that only depend on δ.

We recall that in the theorem above $\|\cdot\|_{P^*}$ denotes the dual norm of $\|\cdot\|_P$ defined in (9), with $P = X^TX + \eta I$. Essentially, this theorem says that when η is small compared to the true error $\|XX^T - M^\star\|$, the PL-inequality is restored under the local norm defined by P. The difficulty of applying this theorem directly in our case arises from two issues: first, if η is too small, then the gradients of $f(X)$ are no longer Lipschitz under the P-norm. As a result, the iterates can diverge. Moreover, it is very difficult to gauge the 'right' size of η in the noisy setting, since we have no access to the true error. The proof of Theorem A.1 can be found in Zhang et al. (2021) so we do not repeat it here.

We also state an lemma from Zhang et al. (2021) that directly characterizes the progress of gradient descent at each iteration in a fashion similar to the descent lemma (see e.g. Nesterov (2018)). For general smooth functions with Lipschitz gradients, the decrement in the function value at each iteration can be characterized by a quadratic upper bound (the so-called descent lemma). However, for matrix sensing, we can in fact obtain a tighter upper bound because $f_c(X - \alpha D)$ itself is just a quartic polynomial. This allows us to characterize the progress made at each iteration directly, using the following result.

Lemma A.2. *For any descent direction $D \in \mathbb{R}^{n\times r}$ and step-size $\alpha > 0$ we have*

$$f_c(X - \alpha D) \le f_c(X) - \alpha\langle \nabla f_c(X), D\rangle + \frac{\alpha^2}{2}\langle D, \nabla^2 f_c(X)[D]\rangle \tag{11}$$

$$+ \frac{(1+\delta)\alpha^3}{m}\|D\|_F^2\left(2\|DX^T + XD^T\|_F + \alpha\|D\|_F^2\right). \tag{12}$$

The proof of this lemma is quite straightforward so we do not repeat it here. The Lemma follows simply from expanding the function $f_c(X - \alpha D)$ and bounding the third and fourth order terms using the restricted isometry property of $\mathcal{A}$.

In section 3 of the main paper, we sketched out the main idea behind our proof of Theorem 2.1. In particular, we stated that our proof mainly consists of two parts. First, in the ideal case where $\eta_t \le \sqrt{f_c(X_t)} \le C\eta_t$, if we go in the direction $D = \nabla f_c(X)P^{-1}$, then linear convergence is already achieved in the sense that the function value decreases by a constant factor. Second, in the case where $\eta_t \le \sqrt{f_c(X_t)} \le C\eta_t$ is no longer satisfied, we want to show that η_t will automatically return to the ideal interval $\eta_t \le \sqrt{f_c(X_t)} \le C\eta_t$ after a few iterations. Thus overall, we have linear convergence. In the following sections, we make these two parts of our proof precise.

A.3 Linear Convergence in Ideal Case

For the PrecGD algorithm of Zhang et al. (2021) to succeed, it is crucial that the regularization parameter satisfies $\eta_t \le \sqrt{f_c(X_t)} \le C\eta_t$. If so, then within a local neighborhood of the ground truth, Theorem A.1 and Theorem A.2 can be used to establish linear convergence in the noiseless setting. However, as we have argued in the main paper, this requirement for η_t is difficult if not impossible to maintain explicitly in the noisy setting. This makes it extremely difficult for PrecGD to achieve a minimax optimal error.

Our main result, Theorem 2.1, states that letting η_t decay with some constant rate β suffices to guarantee the linear convergence of our algorithm, even in the noisy setting. One of our key observations is that the condition $\eta_t \le \sqrt{f_c(X_t)} \le C\eta_t$ *does not* have to be satisfied at all times. In fact, we can allow η_t to dip below $\sqrt{f_c(X_t)}$ in our algorithm, because of the "coupling" effect that we discussed previously: η_t can never deviate too far from $\sqrt{f_c(X_t)}$.

Therefore, in our proof of Theorem 2.1, we will consider two cases:

1. $\eta_t \le \sqrt{f_c(X_t)} \le C\eta_t$
2. $\sqrt{f_c(X_t)} \le \eta_t$.

The first case is the "good" situation, because the conditions for linear convergence is satisfied by assumption. In this case, we show that as long as the gradient is large compared to the noise, i.e., $\|\nabla f_c(X_t)\|_P^* \gtrsim \sqrt{\frac{\sigma^2 rn\log n}{m}}$, our algorithm will converge linearly. This behavior is stated rigorously in the following lemma.

Lemma A.3. *Suppose that at the t-th iteration, the regularization parameter η_t satisfies $\eta_t \le \sqrt{f_c(X_t)} \le C\eta_t$ for some $C > 1$. Furthermore, suppose that $\|\nabla f_c(X_t)\|_P^* \gtrsim \sqrt{\frac{\sigma^2 rn\log n}{m}}$. Then for $\alpha \le 1/L$, with high probability*

we have

$$f_c(X_{t+1}) \le \left(1 - \frac{\mu}{2L}\right) f_c(X_t).$$

Here $L = O(C^2)$ is the constant defined in (16)*, and μ is the constant defined in Theorem A.1.*

The proof of Lemma A.3 is long but it is mainly computational. The overall idea is similar to the proof of Theorem 20 in Zhang et al. (2021): our goal is to show that when the local norm of the gradient is large compared to the noise level, the decrement we make at each iteration 'overcomes' the error caused by the noisy measurements. Our main tool here is Lemma A.2, which allows us to directly compute the decrement and bound the error terms.

It turns out that in this proof, it will be slightly easier to deal with the vectorized version of this problem: we use $f(x)$ to denote original objective function $f(X)$ as a function of the vector $x = \mathrm{vec}(X)$. Consequently, we write $f(x) \in \mathbb{R}^{nr}$ and $\nabla f(x) \in R^{nr}$ as the vectorized versions of $f(X)$ and its gradient. We use the same vectorized notation for the "true" function value $f_c(X)$. Thus, in vectorized form, the iterates of our algorithm can be written as

$$x_{k+1} = x_k - \alpha \mathbf{P}^{-1} \nabla f(x), \quad \text{where } \mathbf{P} = (X^T X + \eta I_r) \otimes I_n.$$

We note that all the norms we consider remain unchanged after vectorization, meaning that $\|\nabla f(x)\|_P = \|\nabla f(X)\|_P$ and $\|\nabla f(x)\|_{P^*} = \|\nabla f(X)\|_{P^*}$. Now we are ready to prove this lemma.

Proof. The main idea of the proof is to use the inequality $\eta_t \le \sqrt{f_c(X_t)} \le C\eta_t$ to bound the progress of our algorithm at each iteration. In particular, when η_t is small, i.e., $\eta_t \le \sqrt{f_c(X_t)}$, then Theorem A.1 guarantees the gradient dominance. On the other hand, the lower bound $\sqrt{f_c(X_t)} \le C\eta_t$ allows us to apply Lemma A.2 to guarantee that the step-size α can be large enough so that we get linear convergence.

First, note that *vectorized* version of the gradient update $X_+ = X - \alpha D$ (where $D = \nabla f(X) P^{-1}$) can be written as $x_+ = x - \alpha d$, where

$$d = \mathrm{vec}\left(\nabla f(X) P^{-1}\right) \quad = \mathbf{P}^{-1} \nabla f_c(x) - \frac{2}{m} \mathbf{P}^{-1} \left(I_r \otimes \sum_{i=1}^m \epsilon_i A_i \right) x. \tag{13}$$

Here we have dissected the gradient descent direction into two parts: $\mathbf{P}^{-1}\nabla f_c(x)$, which corresponds to "correct" gradient and a remaining error term $\mathbf{P}^{-1}\mathcal{E}(x)$, where

$$\mathcal{E}(x) \stackrel{def}{=} \frac{2}{m} \left(I_r \otimes \sum_{i=1}^m \epsilon_i A_i \right) x.$$

In other words we have $d = \mathbf{P}^{-1}(\nabla f_c(x) - \mathcal{E}(x))$. If $\mathcal{E}(x) = 0$, then our proof reduces to the noiseless case. Here we want to show that the error is small compared the decrement we make in the function value. As we will see, this happens precisely in the regime where the gradient is large, i.e., $\|\nabla f_c(X)\|_P^* \gtrsim \sqrt{\frac{\sigma^2 rn \log n}{m}}$.

In vectorized notation, Lemma A.2 can be written as

$$f_c(x - \alpha d) \le f_c(x) - \alpha \nabla f_c(x)^T d + \frac{\alpha^2}{2} d^T \nabla^2 f_c(x) d + \frac{(1+\delta)\alpha^3}{m} \|d\|^2 \left(2\|\mathbf{J}d\| + \alpha \|d\|^2\right), \tag{14}$$

where we define $\mathbf{J} : \mathbb{R}^{nr} \to \mathbb{R}^{n^2}$ as the linear operator satisfying $\mathbf{J}d = \mathrm{vec}(XD^T + DX^T)$ (recall that $d = \mathrm{vec}(D)$). Now setting $d = \mathbf{P}^{-1}(\nabla f_c(x) - \mathcal{E}(x))$ in the formula above yields

$$f_c(x - \alpha d) \le f_c(x) - \alpha \|\nabla f_c(x)\|_{P^*}^2 + T_1 + T_2 + T_3$$

where

$$\begin{aligned}
T_1 =& \alpha \nabla f_c(x)^T \mathbf{P}^{-1} \mathcal{E}(x) \\
T_2 =& \frac{\alpha^2}{2} \Big(\nabla f_c(x)^T \mathbf{P}^{-1} \nabla^2 f_c(x) \mathbf{P}^{-1} \nabla f_c(x) + \mathcal{E}(x)^T \mathbf{P}^{-1} \nabla^2 f_c(x) \mathbf{P}^{-1} \mathcal{E}(x) \\
& - 2 \nabla f_c(x)^T \mathbf{P}^{-1} \nabla^2 f_c(x) \mathbf{P}^{-1} \mathcal{E}(x) \Big) \\
T_3 =& (1+\delta)\alpha^3 \left(\|\mathbf{P}^{-1}\nabla f_c(x) - \mathbf{P}^{-1}\mathcal{E}(x)\|^2 \right) \left(2\|\mathbf{J}\mathbf{P}^{-1}\nabla f_c(x)\| + 2\|\mathbf{J}\mathbf{P}^{-1}\mathcal{E}(x)\| \right. \\
& \left. + \alpha \|\mathbf{P}^{-1}\nabla f_c(x) - \mathbf{P}^{-1}\mathcal{E}(x)\|^2 \right).
\end{aligned}$$

Our goal is to show that all three terms T_1, T_2, T_3 are small compared to the decrement that we make at each iteration. The key observation here is that all of these terms depend on $\mathcal{E}(x)$ and $\mathbf{P}$. With the right choice of η, i.e., with $\sqrt{f_c(X_t)} \leq C\eta_t$, the preconditioner $\mathbf{P}$ is well-conditioned, so that all the errors in T_1, T_2, T_3 will remain small as long as $\mathcal{E}(x)$ is small. Specifically, we can bound the error term as

$$\begin{aligned}
\|\mathcal{E}(x)\|_{P^*}^2 = \mathcal{E}(x)^T \mathbf{P}^{-1} \mathcal{E}(x) &= \left\| \left(\frac{2}{m} \sum_{i=1}^m \epsilon_i A_i \right) X(X^T X + \eta I)^{-1/2} \right\|_F^2 \\
&\leq \left\| \left(\frac{2}{m} \sum_{i=1}^m \epsilon_i A_i \right) \right\|_2^2 \left\| X(X^T X + \eta I)^{-1/2} \right\|_F^2 \\
&\overset{(i)}{\leq} C_e \frac{\sigma^2 n \log n}{m} \left(\sum_{i=1}^r \frac{\sigma_i^2(X)}{\sigma_i(X)^2 + \eta} \right) \\
&\leq C_e \frac{\sigma^2 r n \log n}{m},
\end{aligned}$$

where C_e is an absolute constant and (i) follows from a standard concentration bound (see Candes and Plan (2011) or Lemma 16 of Zhang et al. (2021)).

Now, denoting $\Delta = \|\nabla f_c(x)\|_{P^*}$ and using the bound for the error above, we get after some computations that

$$\begin{aligned}
T_1 &\leq \alpha \Delta \sqrt{\frac{C_e \sigma^2 r n \log n}{m}}, \\
T_2 &\leq 2\alpha^2 L_\delta \Delta^2 + 2\alpha^2 L_\delta \frac{\sigma^2 r n \log n}{m} \\
T_3 &\leq \frac{4(1+\delta)\alpha^3}{\eta} \left(\Delta^2 + \frac{C_e \sigma^2 r n \log n}{m} \right) \left(\frac{\alpha \Delta^2}{\eta} + \frac{\alpha C_e \sigma^2 r n \log n}{\eta m} + 2\sqrt{2}\Delta + 2\sqrt{2} \sqrt{\frac{C_e \sigma^2 r n \log n}{m}} \right).
\end{aligned}$$

Here L_δ is a constant that depends only on the RIP constant δ. Now plugging these error bounds back into (11) yields

$$\begin{aligned}
f_c(x - \alpha d) \leq f_c(x) &- \alpha \Delta^2 + \alpha \Delta \sqrt{\frac{C \sigma^2 r n \log n}{m}} + 2\alpha^2 L_\delta \Delta^2 + 2C\alpha^2 L_\delta \frac{\sigma^2 r n \log n}{m} \\
&+ \frac{4(1+\delta)\alpha^3}{\eta} \left(\Delta^2 + \frac{C \sigma^2 r n \log n}{m} \right) \left(\frac{\alpha \Delta^2}{\eta} + \frac{\alpha C \sigma^2 r n \log n}{\eta m} + 2\sqrt{2}\Delta + 2\sqrt{2} \sqrt{\frac{C \sigma^2 r n \log n}{m}} \right). \quad (15)
\end{aligned}$$

In the case $\Delta \geq 2\sqrt{\frac{C_e \sigma^2 r n \log n}{m}}$ all the terms above can be bounded so that the decrement in the function value dominates all the error. In particular, plugging this lower bound into the inequality above yields

$$f_c(x - \alpha d) \leq f_c(x) - \frac{\alpha}{2} \Delta^2 \left(1 - \frac{5}{2} L_\delta \alpha - 60\sqrt{2}\alpha^2(1+\delta) - 25\alpha^3(1+\delta)^2 \right).$$

Now, assuming that the step-size satisfies

$$\alpha \leq \min \left\{ \frac{L_\delta}{60\sqrt{2}(1+\delta) + 25(1+\delta)^2}, \frac{1}{7L_\delta} \right\} \overset{def}{=} \frac{1}{L} \quad (16)$$

we obtain $f_c(x - \alpha d) \leq f_c(x) - \frac{t\Delta^2}{4} \leq \left(1 - \frac{\alpha\mu}{4}\right) f_c(x)$, where in the last step we used the fact that $\eta_t \leq \sqrt{f_c(X_t)}$, so the conditions of Theorem A.1 are satisfied, so gradient dominance holds. This completes the proof.

□

B Proof of Theorem 2.1

In this section we provide a complete proof of Theorem 2.1, filling out some of the missing details left out in the main paper.

Proof. Let $T > 0$ be the smallest index such that $\eta_T < 2\sqrt{\frac{C_0\sigma^2 rn\log n}{\mu m}}$. Suppose that $t < T$. Similar to the noiseless case, we will show that there exists some constant $C > 1$, which depends only on δ, such that the following holds: if at the t-th iterate we have $\sqrt{f_c(X_t)} \le C\eta_t$, then $\sqrt{f_c(X_{t+1})} \le C\eta_{t+1}$. As before, this implies that $f_c(X_t) \le C^2\beta^{2t}\eta_0$ for all $t \le T$.

At the t-iterate, suppose that $\sqrt{f_c(X_t)} \le C\eta_t$. We consider two cases:

1. $\eta_t \le \sqrt{f_c(X_t)} \le C\eta_t$
2. $\sqrt{f_c(X_t)} \le \eta_t$.

We will show that in either case, the next iterate satisfies $\sqrt{f_c(X_{t+1})} \le C\eta_{t+1}$. The core idea behind this proof is that the values of η_t and $\sqrt{f_c(X_{t+1})}$ are "coupled", meaning that they can not deviate too far from each other. In the first case, where $\eta_t \le \sqrt{f_c(X_t)} \le C\eta_t$, the behavior of our algorithm plus is exactly the same as PrecGD, since η_t is bounded both above and below by a constant factor of $\sqrt{f_c(X_{t+1})}$. Thus, according to Lemma A.3, we converge linearly (at least for the current iteration). In fact, we have chosen the decay rate β so that $f_c(X)$ will decay faster than β when $\eta_t \le \sqrt{f_c(X_t)} \le C\eta_t$. Specifically, we have

$$\sqrt{f_c(X_{t+1})} \le \sqrt{\left(1 - \frac{\mu}{4L}\right)}\sqrt{f_c(X_t)} \le \beta \cdot C\eta_t = C\eta_{t+1}$$

where the second inequality follows from $\beta = \sqrt{\left(1 - \frac{\mu}{8L}\right)}$ and the assumption $\sqrt{f_c(X_t)} \le C\eta_t$. Thus, in this case, $\sqrt{f_c(X_{t+1})}$ will continue to be upper bounded by $C\eta_{t+1}$. If this remains true for all t, then we are already done since η_t decays exponentially, which means that the function value will also decay exponentially fast. However, if $\sqrt{f_c(X_{t+1})}$ decays too fast, the condition for applying Lemma A.3, i.e., $\eta_t \le \sqrt{f_c(X_t)} \le C\eta_t$, will no longer hold. However, in this case, the function values are still decaying monotonically. Since the stepsize satisifies $\alpha \le 1/L$, where L is the constant defined in (16), we can use Lemma A.3 again to get $f_c(X_{t+1}) \le f_c(X_t)$. Thus

$$\sqrt{f_c(X_{t+1})} \le \sqrt{f_c(X_t)} \le \eta_t = \beta^{-1}\eta_{t+1} \le C\eta_{t+1}.$$

Here we note that for the last step to hold we need $\beta^{-1} < C$, which is equivalent to $\sqrt{1 - \frac{\mu}{4L}} \cdot C > 1$. In fact, this is the key step that keeps us from choosing the decay rate β to be too small so that we get any linear convergence rate we like. By definition $L = C_\delta \cdot C^2$, where C_δ is a constant that only depends on δ. Thus this condition is always satisfied for some $C \ge C_{lb}$, where C_{lb} is a constant lower bound that only depends on δ.

Finally, at the T-th iteration, we have $\sqrt{f_c(X_T)} \le C \cdot \eta_T \lesssim \sqrt{\frac{\sigma^2 rn\log n}{m}}$. Now for all $t > T$, we again consider two cases:

1. $\|\nabla f_c(X_t)\|_P^* \le \sqrt{\frac{\sigma^2 rn\log n}{m}}$
2. $\|\nabla f_c(X_t)\|_P^* \ge \sqrt{\frac{\sigma^2 rn\log n}{m}}$.

In the first case, we can use Theorem A.1 to conclude that $\mu f(X_t) \le (\|\nabla f_c(X_t)\|_P^*)^2$. Since μ is a constant, we have $f(X_t) \lesssim \frac{\sigma^2 rn\log n}{m}$. Now consider second case. Here we can apply Lemma A.3 again which guarantees that $f(X_{t+1}) \le f(X_t)$, so the function value is decreasing. Consequently, we have $f(X_t) \lesssim \frac{\sigma^2 rn\log n}{m}$ for all $t > T$. This completes the proof.

□

C Experimental details

Experimental setups We perform all the experiments in this paper on an Apple MacBook Pro, running a silicon M1 pro chip with 10-core CPU, 16-core GPU, and 32GB of RAM. We implement our algorithm in MATLAB R2021a.

Initialization

- **Spectral initialization:** For spectral initialization with respect to a ground truth matrix $M^\star = Q\Sigma Q^T$, we initialized $X_0 = Q\Sigma^{1/2} + 0.1 \cdot \hat{Q}$, where $\hat{Q} \in \mathbb{R}^{n\times r}$ is drawn from standard Gaussian.
- **Small initialization:** For small initialization, we set $X_0 = \hat{\alpha} \cdot \hat{Q}$, where $\hat{\alpha}$ is the initialization scale and $\hat{Q}$ is a $n \times r$ matrix drawn from standard Gaussian.

Datasets The datasets we use for the experiments in the main paper and Appendix E are described below.

- **Gaussian matrix sensing:** For the experiment results shown in Figure 2, 3, 4 and 5 we synthetically generate a 10×10 ground truth matrix $M^\star$. The rank of $M^\star$ is set to 2. To generate $M^\star$, we first randomly generate an orthonormal matrix $Q \in \mathbb{R}^{10\times 2}$ and then set $M^\star = Q\Sigma Q^T$. For Figure 2, 3 and 4, we set $\Sigma = \mathrm{diag}(1, 10^{-2})$ so that $M^\star$ is ill-conditioned with condition number $\kappa = 10^2$. For Figure 5, we set $\Sigma = \mathrm{diag}(1, 1)$ so that $M^\star$ is well-conditioned with condition number $\kappa = 1$.
- **1-bit matrix sensing:** For the experiment results shown in Figure 7, the ground truth matrix $M^\star$ is exactly the same as the one in Figure 5.
- **Phase retrieval:** For the experiment results shown in Figure 8, we synthetically generate a length 10 complex ground truth vector z and set $M^\star = zz^T$. The real and imaginary parts of z are drawn from standard Gaussian.
- **Ultrafast ultrasound image denoising task:** For the experiment results shown in Figure 1 and 9. We take an ultrafast ultrasound scan on a rat brain provided from our collaborator. The ultrasound scan consists of 2400 frames of size 200×130 images. We note that in order to show the entire ultrasound scan in 2D, in Figure 1 and 9, the ultrasound scans are shown in power Doppler Bercoff et al. (2011). In particular, let $M_i \in \mathbb{R}^{200\times 130}$ be the i-th frame of the ultrasound scan, the power Doppler of the scan is defined as $P_M = 20\log\left(\mu_M \sum_i M_i^2\right)$ where μ_M is a normalization constant that normalizes the entries in $\mu_M \sum_i M_i^2$ to between 0 and 1. Here, M_i^2 denotes the elementwise squaring. In this case, the ultrasound image denoising problem can be treated as a low-rank matrix completion problem on a size 26000×2400 ground truth matrix $M^\star$, which we will elaborate in Appendix D.

D Ultrafast ultrasound image denoising task

Ultrafast ultrasound is an advanced imaging technique that leverages high frame rate of plane wave imaging, reaching up to thousands of frames per second. The significant increase in frame rate has revolutionized ultrasound imaging, particularly in ultrafast Doppler Bercoff et al. (2011), providing enhanced temporal resolution for precise evaluation of high-speed blood flows and improved sensitivity in detecting subtle flow within small vessels. However, clutter signals originated from stationary and slow moving tissue introduce significant artifacts during the acquisition of ultrasound image, preventing it from capturing a clear visualization of vascular paths, and measuring blood flows in small vessels. Therefore, effective denoising techniques for removing these artifacts are often required to obtain a high-quality ultrasound image, in order to minimize the chances of diagnostic errors, detect subtle changes or anomalies, and reduce the need for repeated scans or reanalysis.

One effective denoising technique to suppress clutter signals is based on computing the truncated SVD on a space-time matrix Demené et al. (2015). In particular, let M_i be the i-th frame of the m frames ultrasound scan, Demené et al. (2015) proposed to compute the SVD on the space-time matrix $M = [\mathrm{vec}(M_1) \dots \mathrm{vec}(M_m)] = Q\Sigma S^T$ where Q and S are orthonormal matrices and Σ is diagonal. The noiseless space-time matrix $M^\star$ can then be computed by keeping the top r singular values in Σ, i.e. $M^\star = Q_r\Sigma_r S_r^T$ where Q_r and S_r denotes the first r columns of Q and S, respectively, and Σ_r denotes the top $r \times r$ block of Σ. This technique is effective as signals

from stationary and slow moving tissue generally correspond to low frequency components in the spectral domain, as they change slowly over time. However, due to the high frame rates of the ultrafast ultrasound, with prolonged acquisition, it would require sufficiently large memories to store M. In many cases, computing the SVD on a large M also become computationally prohibit as its complexity is cubic in the number of frames m.

One possible way to address these limitations is to downsampled the space-time matrix M and then use it to approximate the noiseless space-time matrix $M^\star$ through low-rank matrix completion. Specifically, we treat the noiseless space-time matrix $M^\star$ as the ground truth, and perform low-rank matrix completion $M^\star = UV^T$ using noisy measurements $y_{ij} = M^\star_{i,j} + \epsilon_{i,j} = M_{i,j}$. Here, U and V are matrices with exactly $r \ll m$ columns. Notice that if the matrix completion problem achieves minimax error, it is exactly coincides with the truncated SVD, i.e. $U = Q_r\Sigma^{1/2}$ and $V = S_r\Sigma^{1/2}$.

In the experimental results shown in Figure 1, we are interested in denoising a 60 megapixel (2400-frames, 200×130 pixels per frame) ultrafast ultrasound image by running 30 iterations of the low-rank denoising procedure in Demené et al. (2015) with 50% sampling rate. As described above, this ultrasound image denoising task can be viewed as a matrix completion problem on a size 26000×2400 ground truth matrix $M^\star$ given 50% of its noisy entries. In our experiment, we randomly sample (without replacement) 50% of the entries in $M = [\mathrm{vec}(M_1)\dots\mathrm{vec}(M_{2400})]$ as our noisy measurements: each measurement takes the form $y_{ij} = \langle e_ie_j^T, M\rangle = \langle e_ie_j^T, M^\star\rangle + \epsilon_{i,j}$, which is a noisy measurement on $M^\star$. We approximate the noiseless ground truth by first setting $M^\star = UV^T$ and minimize the following loss function over rank-r matrices U and V

$$f(U,V) = \frac{1}{|\Omega|}\sum_{(i,j)\in\Omega}\left(\langle e_ie_j^T, UV^T\rangle - y_{ij}\right)^2$$

where the set $\Omega = \{(i,j)\}$ contains indices for which we know the value of M_{ij}.

In Figure 1, we show the ultrasound image recovered from our algorithm, PrecGD (best previous) and GD (no preconditioning) in power Doppler Bercoff et al. (2011). As shown in Figure 1, our algorithm is the only algorithm that achieves the best denoising effect, making the image even sharper. We also emphasize that the per-iteration cost of our algorithm is almost identical to gradient descent. All three experiments take approximately 3 minutes.

Time complexity for gradient evaluation In this experiment, because the measurements is of the form $y_{ij} = \langle e_ie_j^T, M\rangle$, the two gradient terms $\nabla_U f(U,V)$ and $\nabla_V f(U,V)$ in (3) can be efficiently calculated in $O(n_1r|\Omega|)$ time and $O(n_2r|\Omega|)$ time, respectively. Here, we let n_1 denote the number of rows in U and n_2 denote the number of rows in V. To see why this is the case, observe that the two gradient terms can be expressed as $\nabla_U f(U,V) = EV$ and $\nabla_V f(U,V) = E^TU$ where

$$E = \frac{2}{|\Omega|}\sum_{(i,j)\in\Omega}\langle e_ie_j^T, UV^T\rangle\cdot e_ie_j^T$$

is a size $n_1\times n_2$ sparse matrix with exactly $|\Omega|$ nonzero entries, which can be efficiently formed in $O(r|\Omega|)$ time and $O(|\Omega|)$ memory. Hence, despite the large number of measurements in this experiment ($|\Omega| = 31.2$ million), in our practical implementation, evaluating both gradient terms at each iteration only takes approximately 6 seconds.

Ultrasound image denoising task In the experiment results shown in Figure 1, we set the search rank to be $r = 100$, so that U is a size 26000×100 matrix and V is a size 2400×100 matrix. We apply our algorithm, PrecGD and GD to minimize $f(U,V)$ for 30 iterations. Here, our algorithm is implemented with $\beta = 0.05$, and PrecGD is implemented with proxy variance $\hat\sigma = 5\times10^{-3}$ so that $\eta_t = \sqrt{|f(U_t,V_t) - \hat\sigma^2|}$. The learning rate for our algorithm, PrecGD and GD are chosen to be as large as possible. For our algorithm and PrecGD, the learning rate is set to be $\alpha = 10^7$. For GD, the learning rate is set to be $\alpha = 10^3$.

E Additional experiments

In this section, we perform an additional experiment on Gaussian matrix sensing. We also perform additional experiments to gauge the performance of our algorithm for applications outside of the assumptions of our theoretical results. In particular, we consider two common problems considered in the existing literature that do not satisfy conditions under which Theorem 2.1 applies: phase retrieval and 1-bit matrix sensing. For these problems, we see almost identical results to Gaussian matrix sensing: our algorithm succeeds in converging to a minimax optimal error, while GD, PrecGD and ScaledGD(λ) struggle.

E.1 Gaussian matrix sensing

The problem formulation is described in the main paper. In this experiment, we take 80 measurements $y_i = \langle A_i, M^\star \rangle$ on the ground truth matrix $M^\star \in \mathbb{R}^{10\times 10}$ using 80 linearly independent measurement matrices $A_i \in \mathbb{R}^{10\times 10}$ drawn from standard Gaussian. Substituting $M^\star = XX^T$, the loss function for Gaussian matrix sensing is defined as

$$f(X) = \frac{1}{80}\sum_{i=1}^{80} \left(\langle A_i, XX^T\rangle - y_i\right)^2.$$

We perform Gaussian matrix sensing under four different settings.

The exactly-parameterized, noiseless case Recall that the truth rank of $M^\star$ is 2. In the exactly-parameterized case, we set X to be a size 10×2 matrix and minimize $f(X)$ using our algorithm, PrecGD, ScaledGD(λ) and GD for 500 iterations. We set $\beta = 0.1$ in our algorithm, and $\lambda = 0$ in ScaledGD(λ). The learning rate for all four methods are set to $\alpha = 0.1$.

The over-parameterized, noisy case In this setting, we corrupt the measurements with noise $\varepsilon_i \sim \mathcal{N}(0, 10^{-6})$ such that $y_i = \langle A_i, M^\star\rangle + \varepsilon_i$. We set X to be a size 10×4 matrix and minimize $f(X)$ using our algorithm, PrecGD, ScaledGD(λ) and GD for 500 iterations. Here, our algorithm is implemented with $\beta = 0.1$, PrecGD is implemented with proxy variance $\hat{\sigma} = 10^{-5}$ so that $\eta_t = \sqrt{|f(X_t) - \hat{\sigma}^2|}$, and ScaledGD($\lambda$) is implemented with $\lambda = 0.01$. The learning rate for all four methods are set to $\alpha = 0.1$.

Figure 5 plots the convergence of our algorithm, PrecGD, ScaledGD(λ) and GD. The first setting corresponds to the case where $r^\star$ is known, and our measurements are perfect. In this highly unrealistic scenario, we see that the all four methods behave identically, converging linearly to machine error. In the second setting, we see that our algorithm converges to a minimax error of around 10^{-6}, while PrecGD, ScaledGD(λ) and GD struggles to attain the same error. Here the slow down of GD is due to over-parameterization, while the showdown of PrecGD and ScaledGD(λ) are due to an inaccurate estimate of scaling parameters.

High noise setting In the first plot of Figure 6, we plot the convergence of our algorithm under higher noise setting. In particular, we corrupt the measurements with noise $\epsilon_i \sim \mathcal{N}(0, 10^{-1})$. To accommodate higher noise, we set X to be a size 10×8 matrix and minimize $f(X)$ using our algorithm and PrecGD for 500 iterations. In this experiment, our algorithm is implemented with $\beta = 0.97$ and PrecGD is implemented with four different proxy variance $\hat{\sigma} = 1, 0.7, 0.5$ and 0.1 so that $\eta_t = \sqrt{|f(X_t) - \hat{\sigma}^2|}$. The learning rate for both methods are set to $\alpha = 0.01$. From Figure 6, we again see that our algorithm converges linearly to the minmax error while PrecGD slows down when the proxy variance is incorrectly estimated.

Comparison with small initialization In the second plot of Figure 6, we compare the runtime of our algorithm (which is initialized using spectral initialization) against ScaledGD(λ) and GD that are initialized using small initialization. We note that we include the time for calculating the spectral initial point into the runtime of our algorithm. In this experiment, we set X to be a size 10×8 matrix and minimize $f(X)$ using our algorithm, ScaledGD(λ) and GD for around 0.5 seconds. Here, our algorithm is implemented with $\beta = 0.5$, and ScaledGD(λ) is implemented with $\lambda = 0.01$. The learning rate for all three methods are set to $\alpha = 0.1$. In Figure 6, we see that, despite having to spend extra time to computing the spectral initial points, our algorithm is still significantly faster than ScaledGD(λ) and GD.

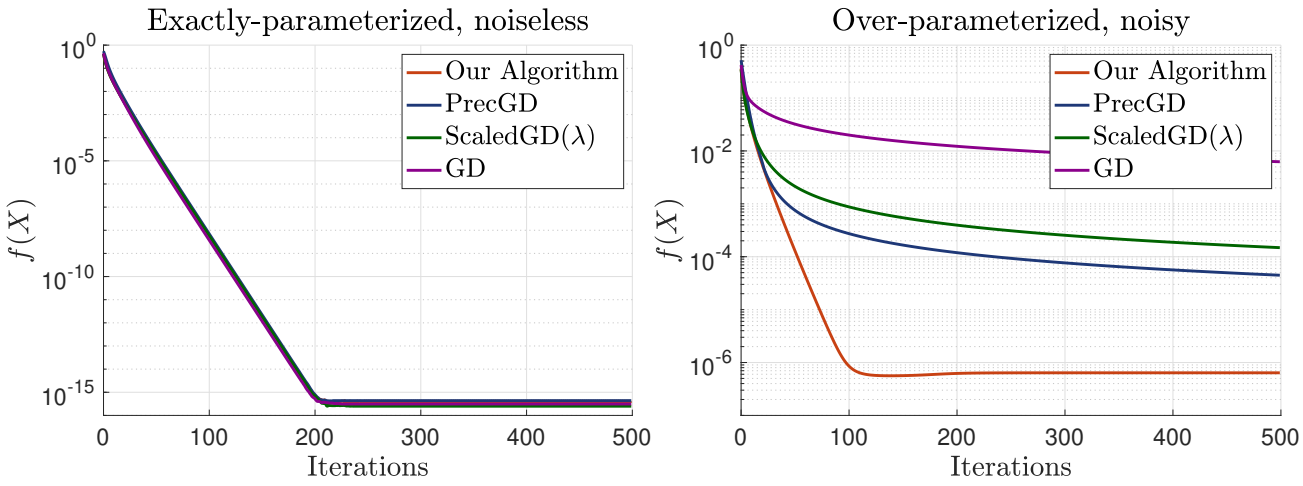

Figure 5: **Convergence of our algorithm, PrecGD, ScaledGD(λ) and GD for Gaussian matrix sensing**. Left: Noiseless measurements with $r = r^\star$. Right: Noisy measurements with $r > r^\star$.

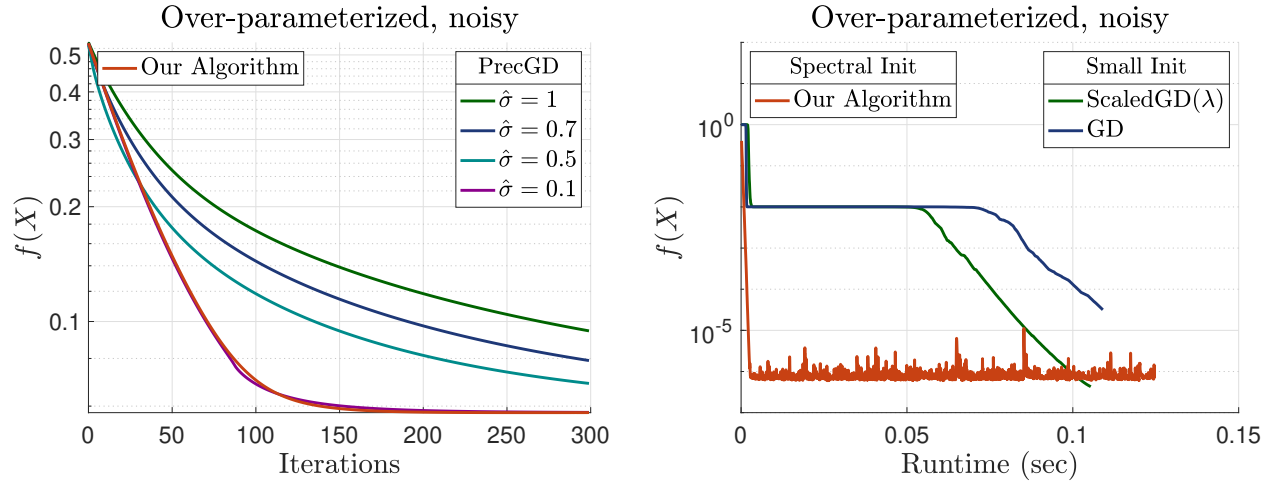

Figure 6: **Comparison under higher noise setting and small initialization.**. Left: Comparison between our algorithm and PrecGD under higher noise setting. Right: Runtime comparison between our algorithm (with spectral initialization), and ScaledGD(λ) and GD (with small initialization).

E.2 1-bit matrix sensing

The goal of 1-bit matrix sensing is to recover a ground truth matrix $M^\star$ from 1-bit measurements of each entry in $M^\star$. In particular, the measurements y_{ij} of each entry $M^\star_{ij}$ are quantized, so that they are 1 with some probability $\sigma(M^\star_{ij})$ and 0 with probability $1 - \sigma(M^\star_{ij})$ where $\sigma(\cdot)$ is the sigmoid function. In our experiment on a size 10×10 ground truth matrix $M^\star$, we measure each y_{ij} for a number of times and let α_{ij} denote the percentage of y_{ij} that is equal to 1. To recover the ground truth matrix $M^\star$, we substitute $M^\star = XX^T$ and minimize the following loss function

$$f(X) = \frac{1}{100} \sum_{i=1}^{10} \sum_{j=1}^{10} -\alpha_{ij} \log\left(\sigma(x_i^T x_j)\right) - (1 - \alpha_{ij}) \log\left(1 - \sigma(x_i^T x_j)\right)$$

where x_i^T is the i-th row in X, i.e. $X = [x_1 \dots x_{10}]^T$. We perform 1-bit matrix sensing under two settings: the exactly-parameterized, noiseless case; and the over-parameterized, noisy case.

The exactly-parameterized, noiseless case Recall that the truth rank of $M^\star$ is 2. In the exactly-parameterized case, we set X to be a size 10×2 matrix and minimize $f(X)$ using our algorithm, PrecGD,

ScaledGD(λ) and GD for 200 iterations. We set $\beta = 0.4$ in our algorithm, and $\lambda = 0$ in ScaledGD(λ). The learning rate for all four methods are set to $\alpha = 1$.

The over-parameterized, noisy case In this setting, we corrupt the measurements with noise $\varepsilon_{ij} \sim \mathcal{N}(0, 10^{-6})$ such that $y_{ij} = 1$ with probability $\sigma(M^\star_{ij} + \varepsilon_{ij})$ and $y_{ij} = 0$ with probability $1 - \sigma(M^\star_{ij} + \varepsilon_{ij})$. We set X to be a size 10×4 matrix and minimize $f(X)$ using our algorithm, PrecGD, ScaledGD(λ) and GD for 200 iterations. Here, our algorithm is implemented with $\beta = 0.4$, PrecGD is implemented with proxy variance $\hat{\sigma} = 10^{-5}$ so that $\eta_t = \sqrt{|f(X_t) - \hat{\sigma}^2|}$, and ScaledGD($\lambda$) is implemented with $\lambda = 0.01$. The learning rate for all four methods are set to $\alpha = 1$.

From Figure 7, we again see that the results are almost identical to that of Figure 5: our algorithm is able to converge linearly to a minimax error rate as soon as $r > r^\star$, but PrecGD, ScaledGD(λ) and GD showdown dramatically.

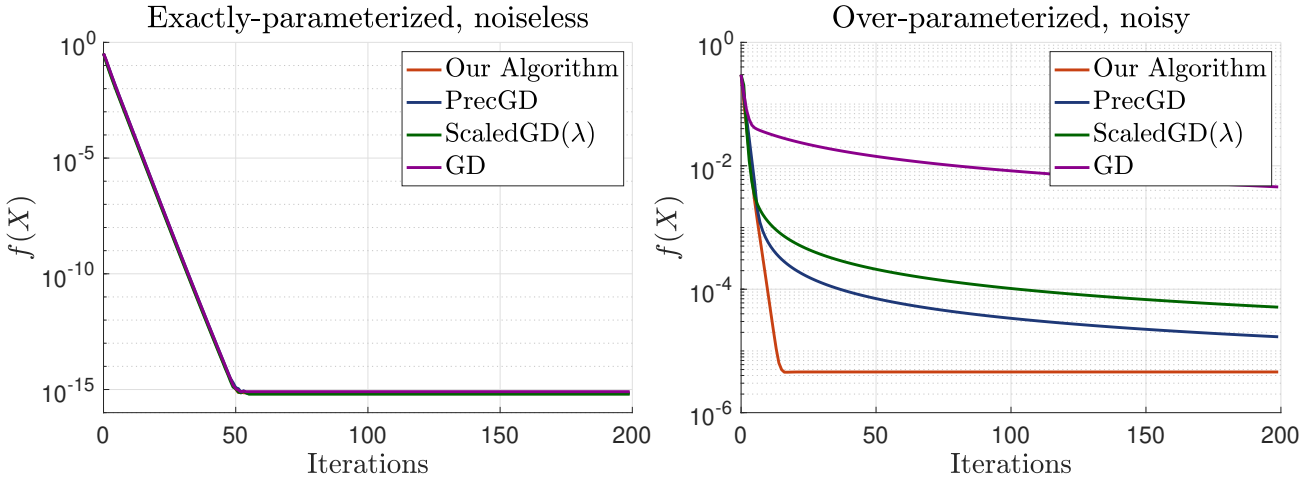

Figure 7: **Convergence of our algorithm, PrecGD, ScaledGD(λ) and GD for 1-bit matrix sensing**. Left: Noiseless measurements with $r = r^\star$. Right: Noisy measurements with $r > r^\star$.

E.3 Phase retrieval

The goal of phase retrieval is to recover a vector $z \in \mathcal{C}^n$ from the phaseless measurements of the form $y_i = |\langle a_i, z\rangle|^2$ where $a_i \in \mathcal{C}^n$ are the measurement vectors. Equivalently, we can view this problem as recovering a complex matrix $M^\star$ from measurements $y_i = \langle a_i a_i^T, M^\star\rangle$, subjecting to a constraint that $M^\star$ is rank-1. In our experiment on a length 10 ground truth vector z, we set $M^\star = zz^T$ and take 80 measurements on $M^\star \in \mathcal{C}^{10\times 10}$ using 80 linearly independent measurement vectors $a_i \in \mathcal{C}^{10}$ drawn from standard Gaussian. Substituting $M^\star = XX^T$, the loss function of phase retrieval is defined as

$$f(X) = \frac{1}{80}\sum_{i=1}^{80} \left(\langle a_i a_i^T, XX^T\rangle - y_i\right)^2.$$

We again perform phase retrieval under two settings: the exactly-parameterized, noiseless case; and the over-parameterized, noisy case.

The exactly-parameterized, noiseless case Recall that the truth rank of $M^\star$ is 1. In the exactly-parameterized case, we set X to be a size 10×1 complex matrix and minimize $f(X)$ using our algorithm, PrecGD, ScaledGD(λ) and GD for 1000 iterations. We set $\beta = 0.1$ in our algorithm, and $\lambda = 0$ in ScaledGD(λ). The learning rate for all four methods are set to $\alpha = 0.02$.

The over-parameterized, noisy case In this setting, we corrupt the measurements with noise $\varepsilon_i \sim \mathcal{N}(0, 10^{-6})$ such that $y_i = \langle a_i a_i^T, M^\star\rangle + \varepsilon_i$. We set X to be a size 10×2 matrix and minimize $f(X)$ using our algorithm,

PrecGD, ScaledGD(λ) and GD for 1000 iterations. Here, our algorithm is implemented with $\beta = 0.1$, PrecGD is implemented with proxy variance $\hat{\sigma} = 10^{-5}$ so that $\eta_t = \sqrt{|f(X_t) - \hat{\sigma}^2|}$, and ScaledGD($\lambda$) is implemented with $\lambda = 0.01$. The learning rate for all four methods are set to $\alpha = 0.02$.

Figure 8 shows the convergence of our alogirthm, PrecGD, ScaledGD(λ) and GD. Again, our algorithm again converge linearly to the minimax error.

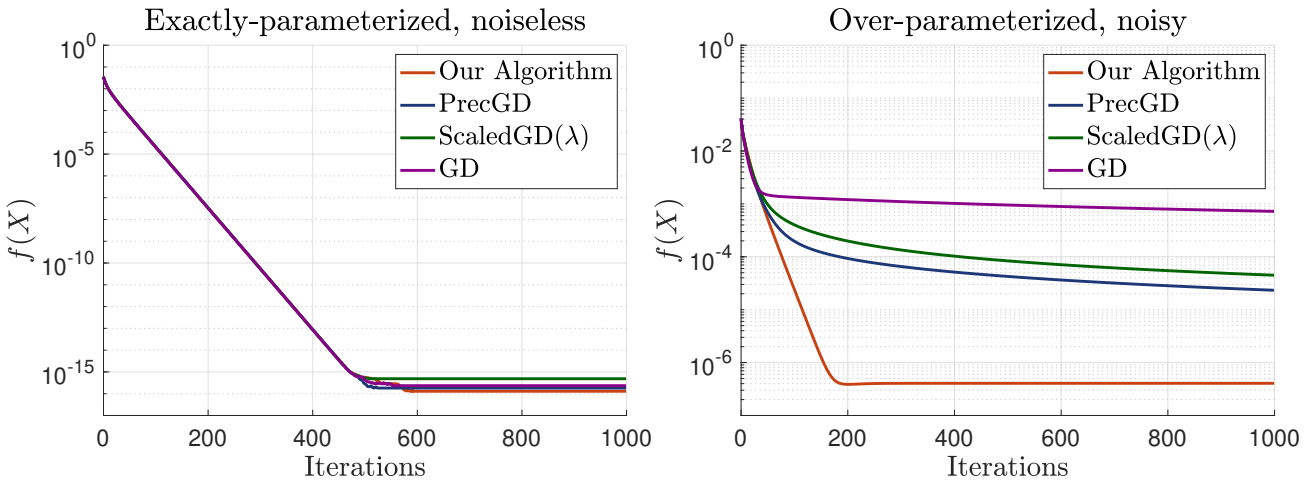

Figure 8: **Convergence of our algorithm, PrecGD, ScaledGD(λ) and GD for phase retrieval**. Left: Noiseless measurements with $r = r^\star$. Right: Noisy measurements with $r > r^\star$.

F Additional Experiments on Ultrasound Image Recovery

We repeat the experiment on ultrasound image denoising task under 7 downsampling rates: 50%, 45%, 40%, 35%, 30%, 25% and 20%. In all 7 cases, we set the search rank to be $r = 100$ and apply our algorithm, PrecGD, ScaledGD(λ) and GD to minimize the corresponding loss function $f(U, V)$ for 30 iterations. Our algorithm, PrecGD and GD are implemented using the same hyperparameters and learning rates in Figure 1. ScaledGD(λ) is implemented with $\lambda = 5 \times 10^{-2}$ and learning rate $\alpha = 10^7$. As shown in Figure 9, our algorithm is able to almost perfectly denoise the ultrasound image when the downsampling rate is above 25%.

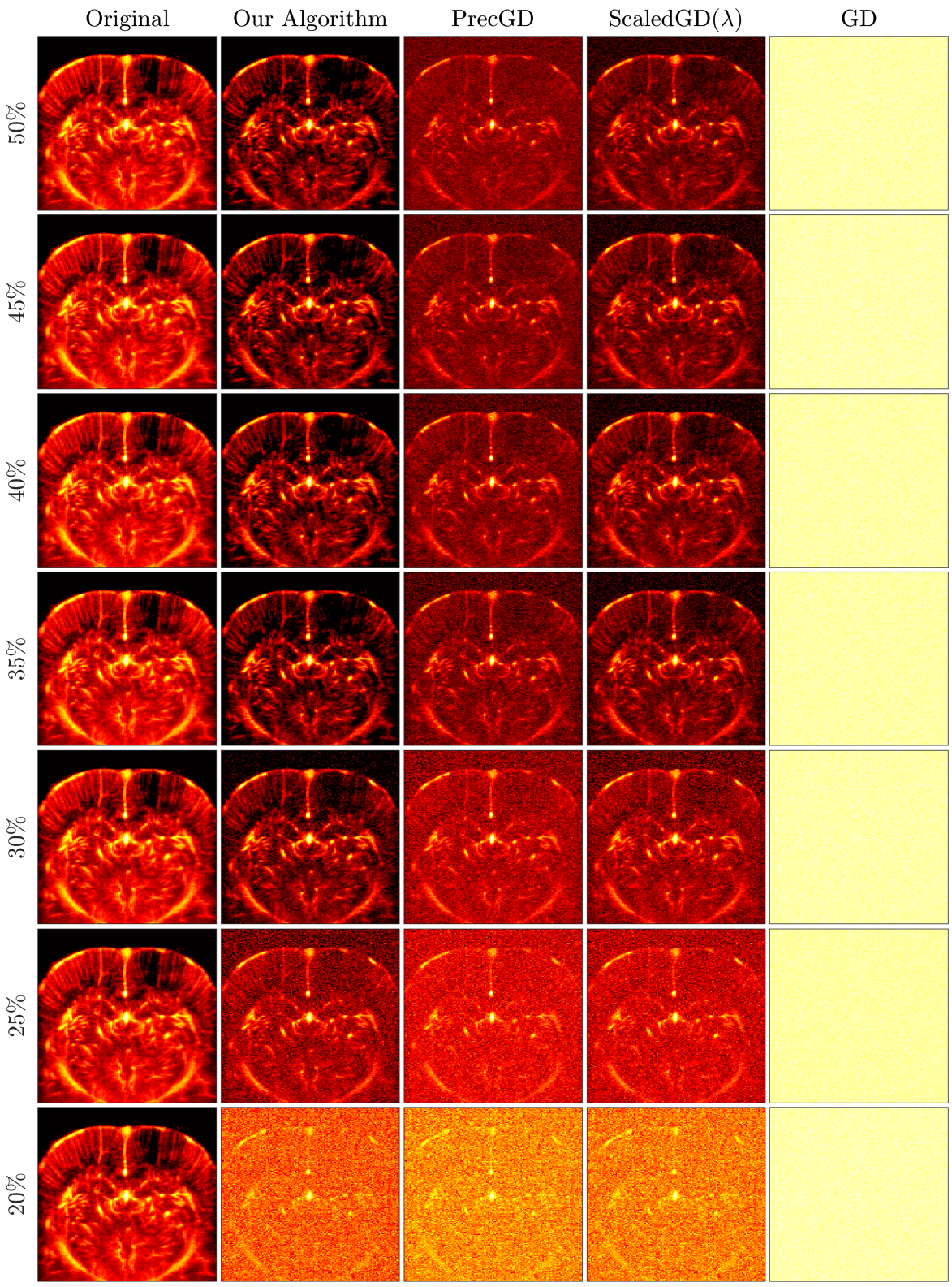

Figure 9: **Denoising an ultrafast ultrasound image under different downsampling rate**. We denoise the ultrasound image in Figure 1 under 7 downsampling rates: 50%, 45%, 40%, 35%, 30%, 25% and 20%. The ultrasound images are shown using power Doppler Bercoff et al. (2011). Column 1: original image. Column 2: image denoised from our algorithm (3). Column 3: image denoised from ScaledGD(λ). Column 4: image denoised from PrecGD. Column 5: image denoised from GD.