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.

1 INTRODUCTION

We consider the low-rank matrix recovery problem, which seeks to recover an \( n \times n \) ground truth matrix \( M^* \) of low rank \( r^* \), given a small number \( m \) of measurement matrices \( A_i \) and noisy observations \( y_i = \langle A_i, M^* \rangle + \epsilon_i \), for indices \( i \in \{1, \ldots, m\} \). The main challenge lies in the fact that the presence of measurement noise \( \epsilon_i \) reduces the “amount of useful information” contained within the observations; it usually becomes impossible to recover \( M^* \) exactly. Instead, one aims to compute a minimax optimal estimate \( M \approx M^* \), which is roughly defined as the closest estimate of the ground truth \( M^* \) 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)].

Figure 1: Preconditioned gradient descent for a 60 megapixel medical image denoising task. We denoise a 2400-frame ultrafast ultrasound image of a rat tail (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.
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 Wainwright (2015)), for many real-world datasets it is unable to do so with a sufficiently large search rank \( r \), as in

\[
\min_{U, V} f(U, V) = \frac{1}{m} \sum_{i=1}^{m} (y_i - \langle A_i, UV^{T} \rangle)^2
\]

(1)

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.

1.1 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); Tu 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 \( M^* \) often have excessively large condition numbers \( \kappa = \lambda_1(M^*)/\lambda_r(M^*) \). In particular, if the search rank is over-parameterized as \( r > r^* \) with respect to the unknown true rank \( r^* \), 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 slowdown with an excessively large or even diverging \( \kappa \). 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):

\[ U_{new} = U - \alpha \nabla_U f(U, V) (V^T V + \eta I)^{-1}, \]
\[ V_{new} = V - \alpha \nabla_V f(U, V) (U^T U + \eta I)^{-1}, \]
\[ \eta_{new} = \beta \eta, \]

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 \( \eta \). 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 \( (V^T V + \eta I)^{-1} \) and \( (U^T U + \eta I)^{-1} \) 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 reminiscent 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 \( \eta \) 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 \( \eta \) to a constant for all iterations (Xu et al. 2023). Our main message in this paper is that choosing \( \eta \) 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 \( \eta \) 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 \( A_1, \ldots, A_m \) satisfy the restricted isometry property (RIP) and that the measurement noise come from a zero-mean Gaussian \( \varepsilon_i \sim N(0, \sigma^2) \), then non-convex gradient descent with rank \( r = O(r^*) \) converges to an estimate \( M \) with Frobenius norm error \( \|M - M^*\|_F = O(\sigma^2 nr^* \log n) \) (Chen and Wainwright 2015; Zhao 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 (Zhao 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 \( \sigma^2 \) (which is clearly unreasonable in practice), or a complicated and expensive cross-validation procedure to estimate the noise variance \( \sigma^2 \).

Our main result is that, under the same setting as the above, the preconditioned iterations in (3) with geometrically decaying \( \eta \) 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 \( O(\sigma^2 nr^*) \); (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 \( \|UV^T - M^*\|_F \) reaches the same order of magnitude as the regularization parameter \( \eta \), but an excessively small \( \eta \) can actually cause the iterations to diverge. Therefore, the most natural and principled way to set \( \eta \) is to allow it to geometrically decay alongside the error norm \( \|UV^T - M^*\|_F \), 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 \( M^* \) 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 \( \eta \); 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 \cite{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 \cite{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 local-approximation 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 \cite{Tu et al. (2016); Zheng and Laflerty (2015); Charisopoulos et al. (2021)}, matrix completion \cite{Sun et al. (2015); Jain et al. (2013); Chen et al. (2020)}, phase retrieval \cite{Candes et al. (2013); Netrapalli et al. (2013); Ma et al. (2018)} and other related problems \cite{Li et al. (2019); Yi et al. (2016); Chen et al. (2020)}. For matrix sensing in particular, both Chen and Wainwright \cite{Chen and Wainwright (2015) and Zhao et al. (2021)} showed that gradient descent achieves a statistical error of $O(\sigma^2nr \log n)$, where $r$ is the search rank. Under the assumption that $r = O(r^*)$, this error matches the minimax error noted in \cite{Candes et al. (2011)} up to log factors. In this work we prove that our method converges to the same statistical error as both Chen and Wainwright \cite{Chen and Wainwright (2015); Zhao 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 $r = O(r^*)$ 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 $(V^TV + \eta I)^{-1}$ and $(U^TU + \eta I)^{-1}$ was first suggested in \cite{Mishra et al. (2012)}, and its convergence properties for the noiseless case were later studied in detail in \cite{Tong et al. (2020)} resulting in a method known as ScaledGD. Its extension to SGD was first proposed in \cite{Mishra and Sepulchre (2016), and studied in \cite{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 \cite{Zhang et al. (2021) as a means to counteract the effects of over-parameterization $r > r^*$, which resulted in a method known as PrecGD. A similar regularization was subsequently studied in \cite{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 \cite{Tong et al. (2020) can diverge in the case $r > r^*$. While this divergence is avoided by adding a regularization parameter as in \cite{Zhang et al. (2013)} and \cite{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 \cite{Li et al. (2018), Stöger and Soltanolkotabi (2021); Ma and Fattahi (2021); Ding et al. (2021); Jin et al. (2023)} for the general over-parameterized case. Similar results have also been obtained in the asymmetric case \cite{Jiang et al. (2023)} \cite{Chou et al. (2023)}. For preconditioned methods, a similar analysis has been done in \cite{Xu et al. (2023)} for a variant of the ScaledGD \cite{Tong et al. (2020)}. In the noisy setting, \cite{Ding et al. (2022)} first showed that GD with small random initialization converges to the minimax error by extending the analysis in \cite{Stöger and Soltanolkotabi (2021)} for the general over-parameterized case. 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 \( (A, B) = \text{tr}(A^TB) \) 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-\( r^* \) ground truth matrix \( M^* \succeq 0 \), from a small number \( m \) of possibly noisy measurements \( y = A(M^*) + \varepsilon \), where the linear measurement operator \( A \) is defined as

\[
A(M^*) = [(A_1, M^*), (A_2, M^*), \ldots, (A_m, M^*)]^T.
\]

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 \( \varepsilon \) is normally distributed

\[
\varepsilon_i \sim N(0, \sigma^2) \quad \text{for all } i \in \{1, \ldots, m\}
\]

and that \( A \) satisfies the restricted isometry property (RIP) \([\text{Candes (2008)}]\).

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

\[
(1 - \delta)\|M\|_F^2 \leq \|A(M)\|^2 \leq (1 + \delta)\|M\|_F^2. \quad (4)
\]

Specifically, we will always assume throughout the paper that \( A \) 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 \( M^* \) 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

\[
X_{\text{new}} = X - \alpha \nabla f(X)(X^TX + \eta I)^{-1}.
\]

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 \( \gamma_0 \) satisfy \( \gamma_0 \geq 2 \sqrt{f_0(X_0)} \), and let the decay rate \( \beta \) satisfy \( 1 > \beta \geq \sqrt{1 - \frac{\rho}{c}} \). Here \( f_0(X_0) \) is the noiseless function value defined in (1), and \( \mu, L \) are both constants depending only on \( r, r^* \) and \( \delta \), 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 \( \|A(X_0)^T - M^*)\|_2^2 < \rho^2(1 - \delta)\lambda_{r^*}(M^*)^2 \) with a radius \( \rho > 0 \) that satisfies \( \rho^2 / (1 - \rho^2) \leq (1 - \delta^2)/2 \). Let the step-size \( \alpha \) satisfy \( \alpha \leq 1/L \), where \( L > 0 \) is a constant that only depends on \( \delta \). At the \( t \)-th iteration, with high probability, we have

\[
\|X_tX_t^T - M^*\|_2^2 \lesssim \max \left\{ \beta^2, \|X_0^T X_0^T - M^*\|_F^2, \varepsilon_{\text{opt}} \right\},
\]

where \( \varepsilon_{\text{opt}} = \sigma^2 n r \log n / m \). Here the inequality \( \lesssim \) hides a constant that only depends on \( \delta \).

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 \( \|A(X_0^T) - M^*)\|_2^2 < \rho^2(1 - \delta)\lambda_{r^*}(M^*)^2 \). 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 $1 > \beta \geq \sqrt{1 - \frac{\mu}{\eta}}$. Although $\mu$ and $L$ are in general hard to estimate, we find that in practice $\beta$ is extremely robust. In our experiments, any value of $\beta$ 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 $\kappa$, since $\mu$ and $L$ has no dependence on $\kappa$. In addition, the statistical error that our algorithm converges to exactly matches that of Chen and Wainwright (2015) and Zhao et al. (2021), which proved that GD converges to an error of $O(\sigma^2nr\log n)$. In other words, our preconditioner exponentially accelerates GD without amplifying the noise at all. Under the mild assumption $r = O(\sigma^2)$, 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:

$$X_{t+1} = X_t - \alpha \nabla f(X_t)(X_t^T X_t + \eta_t I)^{-1},$$

$$C_{lb}\|X_t X_t^T - M^*\| \leq \eta_t \leq C_{ub}\|X_t X_t^T - M^*\|,$$

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 $\eta_t$ must be within a constant factor of the error $\|X_t X_t^T - M^*\|$. In the noiseless case, simply setting $\eta_t = \sqrt{f(X_t)}$ will imply $\eta_t = \Theta(\|X_t X_t^T - M^*\|)$. However, in the noisy setting finding the right $\eta_t$ 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 $\eta_t$ 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 $\eta_t$ is too large, gradient dominance is lost. When $\eta_t$ is too small, Lipschitz smoothness is lost. The analysis in Zhang et al. (2021) suggests that the choice of $\eta_t$ is extremely fragile and delicate.

Surprisingly, we find that this is not the case. In fact, we will show that the choice of $\eta_t$ is not delicate, but rather robust. Our method avoids the need to choose the optimal regularization parameter altogether by simply letting $\eta_t$ 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 $\eta_t$ and $\mathcal{E}_t = \|X_t X_t^T - M^*\|$, are connected using a rubber band. When $\eta_t$ 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 $\eta_t$ itself controls how fast $\mathcal{E}_t$ decays. If the regularization parameter $\eta_t$ is large compared to $\mathcal{E}_t$, then our algorithm behaves more like gradient descent. As a result, our algorithm briefly stagnates, allowing $\eta_t$ to catch up and become close to $\mathcal{E}_t$ again. Similarly, if $\eta_t$ is small, our algorithm begins to converge faster. Thus, the error $\mathcal{E}_t$ decays quickly, and will eventually catch up to $\eta_t$.

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 $r > r^*$, can be restored under a change of norm, as long as the preconditioner $P = (X_t^T X_t + \eta_t I)^{-1}$ has the “correct” amount of regularization $\eta_t$. The second component is the observation that $\eta_t$ and $\mathcal{E}_t = \|X_t X_t^T - M^*\|$ 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

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

where $f_c(X) = \frac{1}{m}\|A(XX^T - M^*)\|^2$ 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...
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: $\|\nabla f_c(X)\|_P^2 \geq \mu f_c(X)$.

First, we consider an ideal case: suppose that there exists some constant $C > 1$ such that $\eta_t \leq \sqrt{f_c(X_t)} \leq C\eta_t$. This is exactly the regime for $\eta_t$ 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 $\eta_t \leq \sqrt{f_c(X_t)} \leq C\eta_t$, 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 $\eta_t$ and $\sqrt{f_c(X_t)}$ are changing. Instead, we have to consider scenarios where $\eta_t$ deviates from this ideal range. To complete the proof of Theorem 2.1, we need to show that $\eta_t$ 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 $\eta_t$, our algorithm will start to behave more like gradient descent and slow down. Hence with a fixed decay rate, $\eta_t$ 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($\lambda$) [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($\lambda$) have trouble converging to

![Convergence of our algorithm and ScaledGD(\lambda) using spectral initialization. Left: Noiseless measurements. Right: Noisy measurements with noise variance $\sigma = 10^{-6}$.](image-url)
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(\(\lambda\)) 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 an 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 \times 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 \(\eta_t\) as \(\beta = 0.85\) in noiseless case, and \(\beta = 0.5\) in noisy case.

Our algorithm v.s. ScaledGD(\(\lambda\)) In Figure 2 we plot the convergence of our algorithm and ScaledGD(\(\lambda\)) under four values of \(\lambda = \{10^{-2}, 10^{-3}, 10^{-6}, 10^{-9}\}\). 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(\(\lambda\)) converges to an error of \(O(\lambda)\). It is important to note
that we cannot set $\lambda$ to be too small because it would cause ScaledGD($\lambda$) 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($\lambda$) 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($\lambda$) becomes numerically unstable when $\lambda$ 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($\lambda$) and GD. In this experiment, our algorithm is initialized using spectral initialization, and both ScaledGD($\lambda$) 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($\lambda$) 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**


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


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


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 $P > 0$ 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

$$\|X\|_P \overset{\text{def}}{=} \|XP^{1/2}\|_2, \quad \|X\|_{P^*} \overset{\text{def}}{=} \|XP^{-1/2}\|_2.$$  

(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 $M^* = ZZ^T \in \mathbb{R}^{n \times n}$. We denote the true rank by $r^* = \text{rank}(M^*)$. Our goal is to recovery $M^*$ from a small number of measurements of the form $y = A(ZZ^T) + \epsilon \in \mathbb{R}^m$. Here $\epsilon$ is a vector with independent Gaussian entries with zero mean and variance $\sigma^2$.

To do so, we minimize the non-convex objective function

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

where $f_c(X) \overset{\text{def}}{=} \frac{1}{m} \|A(XX^T - M^*)\|^2$ 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 $\alpha$ 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 $X_{t+1} = X_t - \alpha \nabla f(X_t)(X_t^TX_t + \eta_t I)^{-1}$, we get

$$\text{vec} X_{t+1} = \text{vec} X_t - \alpha \text{vec} \nabla f(X_t)(X_t^TX_t + \eta_t I)^{-1} = \text{vec} X_t - (X_t^TX_t + \eta_t I) \otimes I)^{-1} \text{vec}(\nabla f(X_t)),$$

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 $x_{t+1} = x_t - \alpha P^{-1} \nabla f(x_t)$, with $P = (X_t^TX_t + \eta_t I) \otimes I$.

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 $\eta$ 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 $M^* \neq 0$. Suppose that $X$ satisfies $f(X) \leq \rho^2 \cdot (1 - \delta) \lambda^2_{\text{min}}(M^*)$ with radius $\rho > 0$ that satisfies $\rho^2/(1 - \rho^2) \leq (1 - \delta^2)/2$. Then, we have

$$\eta \leq C_{\text{rob}} \|XX^T - M^*\|_F \implies \|\nabla f(X)\|^2_{P^*} \geq 2 \mu f(X)$$

where

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

(10)

Here $C_1$ is a constant that only depend on $\delta$. 

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 $\eta$ is small compared to the true error $\|XX^T - M^*\|$, 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 $\eta$ 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 $\eta$ 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

\[
 f_c(X - \alpha D) \leq f_c(X) - \alpha \langle \nabla f_c(X), D \rangle + \frac{\alpha^2}{2} \| D \nabla^2 f_c(X)[D] \|_F^2 + (1 + \delta)\frac{\alpha^4}{m} \| D \|_F^2 (2 \| DX^T \|_F + \alpha \| D \|_F^2).
\]

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 $A$.

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 $\eta_t \leq \sqrt{f_c(X_t)} \leq C\eta_t$, 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 $\eta_t \leq \sqrt{f_c(X_t)} \leq C\eta_t$ is no longer satisfied, we want to show that $\eta_t$ will automatically return to the ideal interval $\eta_t \leq \sqrt{f_c(X_t)} \leq C\eta_t$ 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 $\eta_t \leq \sqrt{f_c(X_t)} \leq C\eta_t$. 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 $\eta_t$ 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 $\eta_t$ decay with some constant rate $\beta$ suffices to guarantee the linear convergence of our algorithm, even in the noisy setting. One of our key observations is that the condition $\eta_t \leq \sqrt{f_c(X_t)} \leq C\eta_t$ does not have to be satisfied at all times. In fact, we can allow $\eta_t$ to dip below $\sqrt{f_c(X_t)}$ in our algorithm, because of the “coupling” effect that we discussed previously: $\eta_t$ 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. $\eta_t \leq \sqrt{f_c(X_t)} \leq C\eta_t$
2. $\sqrt{f_c(X_t)} \leq \eta_t$.

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., $\| \nabla f_c(X_t) \|_p \geq \sqrt{\frac{\sigma^2 \eta_t \log n}{m}}$, 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 $\eta_t$ satisfies $\eta_t \leq \sqrt{f_c(X_t)} \leq C\eta_t$ for some $C > 1$. Furthermore, suppose that $\| \nabla f_c(X_t) \|_p \geq \sqrt{\frac{\sigma^2 \eta_t \log n}{m}}$. Then for $\alpha \leq 1/L$, with high probability
we have
\[ f_c(X_{t+1}) \leq \left( 1 - \frac{\mu}{2L} \right) f_c(X_t). \]
Here \( L = O(C^2) \) is the constant defined in (16), and \( \mu \) 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 \( x = \text{vec}(X) \). Consequently, we write \( f(x) \in \mathbb{R}^{nr} \) and \( \nabla f(x) \in \mathbb{R}^{nr} \) 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
\[ x_{k+1} = x_k - \alpha \text{P}^{-1} \nabla f(x), \quad \text{where } \text{P} = (X^T X + \eta_1 I_r) \otimes I_n. \]
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 \( \eta_t \leq \sqrt{f_c(X_t)} \leq C \eta_t \) to bound the progress of our algorithm at each iteration. In particular, when \( \eta_t \) is small, i.e., \( \eta_t \leq \sqrt{f_c(X_t)} \), then Theorem A.1 guarantees the gradient dominance. On the other hand, the lower bound \( \sqrt{f_c(X_t)} \leq C \eta_t \) allows us to apply Lemma A.2 to guarantee that the step-size \( \alpha \) 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
\[ d = \text{vec}(\nabla f(X)) P^{-1} = \text{P}^{-1} \nabla f_c(x) - \frac{2}{m} \text{P}^{-1} \left( I_r \otimes \sum_{i=1}^m \epsilon_i A_i \right) x. \]
Here we have dissected the gradient descent direction into two parts: \( \text{P}^{-1} \nabla f_c(x) \), which corresponds to “correct” gradient and a remaining error term \( \text{P}^{-1} \mathcal{E}(x) \), where
\[ \mathcal{E}(x) \overset{\text{def}}{=} \frac{2}{m} \left( I_r \otimes \sum_{i=1}^m \epsilon_i A_i \right) x. \]
In other words we have \( d = \text{P}^{-1} (\nabla f_c(x) - \mathcal{E}(x)) \). 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., \( \| \nabla f_c(X) \|_p \geq \sqrt{\frac{\sigma^2 n \log n}{m}} \).

In vectorized notation, Lemma A.2 can be written as
\[ f_c(x - \alpha d) \leq f_c(x) - \alpha \nabla f_c(x)^T d + \alpha^2 \frac{d^T \nabla^2 f_c(x) d}{2} + \frac{(1 + \delta)\alpha^3}{m} \| d \|^2 \left( \| d \| + \alpha \| d \|^2 \right), \]
where we define \( \mathbf{J} : \mathbb{R}^{nr} \to \mathbb{R}^{n^2} \) as the linear operator satisfying \( \mathbf{J} d = \text{vec}(XD^T + DX^T) \) (recall that \( d = \text{vec}(D) \)). Now setting \( d = \text{P}^{-1} (\nabla f_c(x) - \mathcal{E}(x)) \) in the formula above yields
\[ f_c(x - \alpha d) \leq f_c(x) - \alpha \| \nabla f_c(x) \|^2_\mathcal{F} + T_1 + T_2 + T_3, \]
where
\[ T_1 = \alpha \nabla f_c(x)^T \text{P}^{-1} \mathcal{E}(x) \]
\[ T_2 = \frac{\alpha^2}{2} \left( \nabla f_c(x)^T \text{P}^{-1} \nabla^2 f_c(x) \text{P}^{-1} \nabla f_c(x) \right)^T \mathcal{E}(x) \]
\[ T_3 = (1 + \delta) \alpha^3 \left( \| \text{P}^{-1} \nabla f_c(x) - \text{P}^{-1} \mathcal{E}(x) \|^2 \right) \left( \| d \| + \alpha \| d \|^2 \right), \]
\[ \text{P} = (X^T X + \eta_1 I_r) \otimes I_n. \]
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 $E(x)$ and $P$. With the right choice of $\eta$, i.e., with $\sqrt{f_c(X_t)} \leq C \eta$, the preconditioner $P$ is well-conditioned, so that all the errors in $T_1, T_2, T_3$ will remain small as long as $E(x)$ is small. Specifically, we can bound the error term as

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

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

$$T_1 \leq \alpha \Delta \sqrt{C_e \frac{\sigma^2 r n \log n}{m}},$$

$$T_2 \leq 2 \alpha^2 L_\delta \Delta^2 + 2 \alpha^2 L_\delta \frac{\sigma^2 r n \log n}{m},$$

$$T_3 \leq 4 \frac{(1 + \delta) \alpha^3}{\eta} \left( \Delta^2 + \frac{C_e \sigma^2 r n \log n}{m} \right) \left( \frac{\alpha \Delta^2}{\eta} + \frac{\alpha C_e \sigma^2 r n \log n}{\eta m} + 2 \sqrt{2} \Delta + 2 \sqrt{2} \left( \frac{C_e \sigma^2 r n \log n}{m} \right) \right).$$

Here $L_\delta$ is a constant that depends only on the RIP constant $\delta$. Now plugging these error bounds back into (11) yields

$$f_c(x - \alpha d) \leq f_c(x) - \alpha \Delta^2 + \alpha \sqrt{C_e \frac{\sigma^2 r n \log n}{m}} + 2 \alpha^2 L_\delta \Delta^2 + 2 C_e \frac{\alpha \sigma^2 r n \log n}{m}$$

$$+ 4 \frac{(1 + \delta) \alpha^3}{\eta} \left( \Delta^2 + \frac{C_e \sigma^2 r n \log n}{m} \right) \left( \frac{\alpha \Delta^2}{\eta} + \frac{\alpha C_e \sigma^2 r n \log n}{\eta m} + 2 \sqrt{2} \Delta + 2 \sqrt{2} \left( \frac{C_e \sigma^2 r n \log n}{m} \right) \right). \tag{15}$$

In the case $\Delta \geq 2 \sqrt{C_e \frac{\sigma^2 r n \log n}{m}}$ 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

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

Now, assuming that the step-size satisfies

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

we obtain $f_c(x - \alpha d) \leq f_c(x) - \frac{\alpha \Delta^2}{2} \leq \left( 1 - \frac{\alpha n}{4} \right) f_c(x)$, where in the last step we used the fact that $\eta \leq \sqrt{f_c(X_t)}$, so the conditions of Theorem 1.1 are satisfied, so gradient dominance holds. This completes the proof.

$\square$

**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.
We will show that in either case, the next iterate satisfies

\[ \sqrt{f_c(X_t)} \leq C \eta_t \]

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

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

We will show that in either case, the next iterate satisfies \( \sqrt{f_c(X_{t+1})} \leq C \eta_{t+1} \). The core idea behind this proof is that the values of \( \eta_t \) 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 \( \eta_t \leq \sqrt{f_c(X_t)} \leq C \eta_t \), the behavior of our algorithm plus is exactly the same as PrecGD, since \( \eta_t \) is bounded both above and below by a constant factor of \( \sqrt{f_c(X_{t+1})} \). Thus, according to Lemma \ref{lem:linear_convergence}, we converge linearly (at least for the current iteration). In fact, we have chosen the decay rate \( \beta \) so that \( f_c(X) \) will decay faster than \( \beta \) when \( \eta_t \leq \sqrt{f_c(X_t)} \leq C \eta_t \). Specifically, we have

\[ \sqrt{f_c(X_{t+1})} \leq \left( \frac{1}{\sqrt{2L}} \right) \sqrt{f_c(X_t)} \leq \beta \cdot C \eta_t = C \eta_{t+1} \]

where the second inequality follows from \( \beta = \sqrt{1 - \frac{\mu}{2L}} \) and the assumption \( \sqrt{f_c(X_t)} \leq C \eta_t \). 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 \( \eta_t \) 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 \ref{lem:linear_convergence}, i.e., \( \eta_t \leq \sqrt{f_c(X_t)} \leq C \eta_t \), will no longer hold. However, in this case, the function values are still decaying monotonically. Since the stepsize satisfies \( C > 1/L \), where \( L \) is the constant defined in \( \ref{eq:stepsize} \), we can use Lemma \ref{lem:linear_convergence} again to get \( f_c(X_{t+1}) \leq f_c(X_t) \). Thus

\[ \sqrt{f_c(X_{t+1})} \leq \sqrt{f_c(X_t)} \leq \eta_t = \beta^{-1} \eta_{t+1} \leq C \eta_{t+1} \]

Here we note that for the last step to hold we need \( \beta^{-1} < C \), which is equivalent to \( \sqrt{1 - \frac{\mu}{2L}} \cdot C > 1 \). In fact, this is the key step that keeps us from choosing the decay rate \( \beta \) to be too small so that we get any linear convergence rate we like. By definition \( L = C_3 \cdot C^2 \), where \( C_3 \) is a constant that only depends on \( \delta \). Thus this condition is always satisfied for some \( C \geq C_\beta \), where \( C_\beta \) is a constant lower bound that only depends on \( \delta \).

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

1. \( \| \nabla f_c(X_t) \|_p \leq \sqrt{\frac{\sigma^2 r \log n}{m}} \)
2. \( \| \nabla f_c(X_t) \|_p \geq \sqrt{\frac{\sigma^2 r \log n}{m}} \).

In the first case, we can use Theorem \ref{thm:mu} to conclude that \( \mu f(X_t) \leq (\| \nabla f_c(X_t) \|_p)^2 \). Since \( \mu \) is a constant, we have \( f(X_t) \lesssim \frac{\sigma^2 r \log n}{m} \). Now consider second case. Here we can apply Lemma \ref{lem:linear_convergence} again which guarantees that \( f(X_{t+1}) \leq f(X_t) \), so the function value is decreasing. Consequently, we have \( f(X_t) \lesssim \frac{\sigma^2 r \log n}{m} \) for all \( t > T \). This completes the proof. \( \square \)
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 $M^* = Q \Sigma Q^T$, we initialized $X_0 = Q \Sigma^{1/2} + 0.1 \cdot Q$, where $Q \in \mathbb{R}^{n \times r}$ 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 $10 \times 10$ ground truth matrix $M^*$. The rank of $M^*$ is set to 2. To generate $M^*$, we first randomly generate an orthonormal matrix $Q \in \mathbb{R}^{10 \times 2}$ and then set $M^* = Q \Sigma Q^T$. For Figure 2, 3 and 4, we set $\Sigma = \text{diag}(1, 10^{-2})$ so that $M^*$ is ill-conditioned with condition number $\kappa = 10^2$. For Figure 5, we set $\Sigma = \text{diag}(1, 1)$ so that $M^*$ is well-conditioned with condition number $\kappa = 1$.
- **1-bit matrix sensing:** For the experiment results shown in Figure 7, the ground truth matrix $M^*$ 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 $M^* = zz^T$. 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 x 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 $P_M = 20 \log_2 (\mu_M \sum_i M_i^2)$ where $\mu_M$ 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 x 2400 ground truth matrix $M^*$, 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 $M = [\text{vec}(M_1) \ldots \text{vec}(M_m)] = Q \Sigma S^T$ where $Q$ and $S$ are orthonormal matrices and $\Sigma$ is diagonal. The noiseless space-time matrix $M^*$ can then be computed by keeping the top $r$ singular values in $\Sigma$, i.e. $M^* = Q_r \Sigma_r S_r^T$ where $Q_r$ and $S_r$ denotes the first $r$ columns of $Q$ and $S$, respectively, and $\Sigma_r$ denotes the top $r \times r$ block of $\Sigma$. 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 prohibist 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 $M^*$ through low-rank matrix completion. Specifically, we treat the noiseless space-time matrix $M^*$ as the ground truth, and perform low-rank matrix completion $M^* = UV^T$ using noisy measurements $y_{ij} = M^*_{ij} + \epsilon_{ij} = M_{ij}$, 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. $U = Q_r \Sigma_r^{1/2}$ and $V = S_r \Sigma_r^{1/2}$.

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 $26000 \times 2400$ ground truth matrix $M^*$ given 50% of its noisy entries. In our experiment, we randomly sample (without replacement) 50% of the entries in $M = [\text{vec}(M_1) \ldots \text{vec}(M_{2400})]$ as our noisy measurements: each measurement takes the form $y_{ij} = \langle e_i e_j^T, M \rangle = \langle e_i e_j^T, M^* \rangle + \epsilon_{ij}$, which is a noisy measurement on $M^*$. We approximate the noiseless ground truth by first setting $M^* = UV^T$ and minimize the following loss function over rank-$r$ matrices $U$ and $V$:

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

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 $y_{ij} = \langle e_i e_j^T, M \rangle$, the two gradient terms $\nabla_U f(U, V)$ and $\nabla_V f(U, V)$ in (3) can be efficiently calculated in $O(n_1 r |\Omega|)$ time and $O(n_2 r |\Omega|)$ 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 $\nabla_V f(U, V) = E^T U$ where:

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

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 $26000 \times 100$ matrix and $V$ is a size $2400 \times 100$ 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 $\hat{\sigma} = 5 \times 10^{-3}$ 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 \( y_i = \langle A_i, M^* \rangle \) on the ground truth matrix \( M^* \in \mathbb{R}^{10 \times 10} \) using 80 linearly independent measurement matrices \( A_i \in \mathbb{R}^{10 \times 10} \) drawn from standard Gaussian. Substituting \( M^* = XX^T \), the loss function for Gaussian matrix sensing is defined as

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

We perform Gaussian matrix sensing under four different settings.

The exactly-parameterized, noiseless case Recall that the truth rank of \( M^* \) 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 \( y_i = \langle A_i, M^* \rangle + \varepsilon_i \). 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 \( \eta_i = \sqrt{f(X_i) - \hat{\sigma}^2} \), and ScaledGD(λ) is implemented with \( \lambda = 0.01 \). The learning rate for all four methods are set to \( \alpha = 0.1 \).

Figure 6 plots the convergence of our algorithm, PrecGD, ScaledGD(λ) and GD. The first setting corresponds to the case where \( r^* \) 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 slowdown 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 \( \varepsilon_i \sim \mathcal{N}(0, 10^{-1}) \). 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 \( \eta_i = \sqrt{f(X_i) - \hat{\sigma}^2} \). 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 minimax 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.

Gavin Zhang, Hong-Ming Chiu, Richard Y. Zhang
E.2 1-bit matrix sensing

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

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

where $x_i^T$ is the $i$-th row in $X$, i.e. $X = [x_1 \ldots x_{10}]^T$. 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 $M^*$ is 2. In the exactly-parameterized case, we set $X$ to be a size $10 \times 2$ matrix and minimize $f(X)$ using our algorithm, PrecGD,
ScaledGD(\(\lambda\)) and GD for 200 iterations. We set \(\beta = 0.4\) in our algorithm, and \(\lambda = 0\) in ScaledGD(\(\lambda\)). 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 \(\sigma(M^*_ij + \varepsilon_{ij})\) and \(y_{ij} = 0\) with probability \(1 - \sigma(M^*_ij + \varepsilon_{ij})\). We set \(X\) to be a size 10 \(\times\) 4 matrix and minimize \(f(X)\) using our algorithm, PrecGD, ScaledGD(\(\lambda\)) 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 \(\eta_t = p|f(X_t) - \hat{\sigma}^2|\), 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 \(r > r^*\), but PrecGD, ScaledGD(\(\lambda\)) and GD showdown dramatically.

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

### E.3 Phase retrieval

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

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

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 \(M^*\) is 1. In the exactly-parameterized case, we set \(X\) to be a size 10 \(\times\) 1 complex matrix and minimize \(f(X)\) using our algorithm, PrecGD, ScaledGD(\(\lambda\)) and GD for 1000 iterations. We set \(\beta = 0.1\) in our algorithm, and \(\lambda = 0\) in ScaledGD(\(\lambda\)). 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 \(y_i = \langle a_i a_i^T, M^* \rangle + \varepsilon_i\). We set \(X\) to be a size 10 \(\times\) 2 matrix and minimize \(f(X)\) using our algorithm,
PrecGD, ScaledGD($\lambda$) 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 algorithm, PrecGD, ScaledGD($\lambda$) and GD. Again, our algorithm again converge linearly to the minimax error.

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

**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($\lambda$) 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($\lambda$) 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. Column 3: image denoised from ScaledGD(λ). Column 4: image denoised from PrecGD. Column 5: image denoised from GD.