Fast and Provable Algorithms for Learning Two-Layer Polynomial Neural Networks

Mohammadreza Soltani and Chinmay Hegde

***Abstract*—We study the problem of (provably) learning the weights of a two-layer neural network with quadratic activations. We focus on the under-parametrized regime where the number of neurons in the hidden layer is smaller than the dimension of the input. Our main approach is to "lift" the learning problem into a higher dimension, which enables us to borrow algorithmic techniques from low-rank matrix estimation. Using this intuition, we propose three novel, non-convex training algorithms. We support our algorithms with rigorous theoretical analysis, and show that all three enjoy a linear convergence, fast running time per iteration, and near-optimal sample complexity. Finally, we complement our theoretical results with numerical experiments.**

I. Introduction

A. Setup

The re-emergence of neural networks has had a remarkable impact on various machine learning problems including object recognition in images, natural language processing, and automated drug discovery. However, despite their successful empirical performance, *provable* algorithms for training neural networks remain relatively less well understood. In this paper, we propose a series of training algorithms for a simple class of shallow neural networks by making connections to the area of *low-rank matrix estimation*. Our work can be viewed as a bridge between matrix recovery and neural network learning, and we hope that that our building blocks of our theoretical analysis leads to insights for more complex networks.

Mathematically, we consider the following neural network architecture. The network comprises p input nodes, a single hidden layer with r neurons with activation function $\sigma(z)$, first layer weights $\{w_j\}_{j=1}^r \subset \mathbb{R}^p$, and an output layer comprising of a single node and weights $\{\alpha_j\}_{j=1}^r \subset \mathbb{R}$. If $\sigma(z) = z^2$, then the above network is called a *polynomial neural network* [1]. The input-output relationship between an input, $x \in \mathbb{R}^p$, and the corresponding output, $y \in \mathbb{R}$, is given by:

$$\hat{y} = \sum_{j=1}^{r} \alpha_j \sigma(w_j^T x) = \sum_{j=1}^{r} \alpha_j \langle w_j, x \rangle^2.$$

In this paper, our focus is in the so-called "under-parameterized" regime where $r \ll p$. While quadratic activation functions are less common than (say) sigmoid activations or rectified linear units (ReLUs), they have been shown to be have competitive expressive power [1], while enabling important safety protocols in verifiable computing [2].

The authors are with the ECE Department at Iowa State University. Email: {msoltani,chinmay@iastate.edu}. A conference version featuring the first two algorithms of this paper appears in the proceedings of AISTATS 2018.

Our goal is to learn this network, given a set of training input-output pairs $\{(x_i, y_i)\}_{i=1}^m$. We do so by finding a set of weights $\{\alpha_j, w_j\}_{j=1}^r$ that minimize the *empirical risk*:

$$\min_{W \in \mathbb{R}^{r \times p}, \alpha \in \mathbb{R}^r} \quad F(W, \alpha) = \frac{1}{2m} \sum_{i=1}^{m} \left(y_i - \hat{y_i}\right)^2, \qquad (1)$$

where the rows of W and the entries of α indicate the weights of the two layers. Numerous recent papers have explored (provable) algorithms to learn the weights of such a network under distributional assumptions on the input data [1], [3], [4], [5], [6], [7], [8].

Clearly, the empirical risk defined in (1) is highly nonconvex function which is difficult to optimize. However, we can circumvent this difficulty using a *lifting* trick: if we define the matrix variable $L_* = \sum_{j=1}^r \alpha_j w_j w_j^T$, then the input-output relationship becomes:

$$\hat{y_i} = x_i^T L_* x_i = \langle x_i x_i^T, L_* \rangle, \qquad (2)$$

where $x_i \in \mathbb{R}^p$ denotes the i^{th} training sample. Moreover, the variable L_* is a rank-r matrix of size $p \times p$ Therefore, (1) can be viewed as an instance of learning a fixed (but unknown) rank-r symmetric matrix $L_* \in \mathbb{R}^{p \times p}$ with $r \ll p$, from a small number of *rank-one* linear observations given by $A_i = x_i x_i^T$. While still non-convex, low-rank matrix estimation problems such as (2) are much better understood. Some specific instances of low-rank estimation in statistical signal processing and machine learning include matrix sensing and matrix completion [9], [10], covariance sketching [11], [12], and generalized phase retrieval [13], [14].

B. Our Contributions

In this paper, we make concrete algorithmic progress on solving low-rank matrix estimation problems of the form (2). In the context of learning polynomial neural networks, once we have estimated a rank-r symmetric matrix L_*, we can always produce weights $\{\alpha_j, w_j\}$ by an eigendecomposition of L_*.

In general, a range of algorithms for solving (2) (or variants thereof) exist in the literature, and can be broadly classified into two categories: (i) *convex* approaches, all of which involve enforcing the rank-r assumption in terms of a convex penalty term, such as the nuclear norm [10], [11], [12]; (ii) *nonconvex* approaches based on either alternating minimization [15], [3] or greedy approximation [1], [16].

Both types of approaches suffer from severe computational difficulties (which we make more precise below). Typically, they require multiple invocations of singular value decomposition (SVD), which can incur *cubic* ($\Omega(p^3)$) running time.

TABLE I: *Our contributions and comparison with existing algorithms. Here, $\beta = \frac{\sigma_1}{\sigma_r}$ denotes the condition number of L_*.*

Algorithm	Sample complexity (m)	Running Time
[12], [11], [20]	$\mathcal{O}(pr)$	$\mathcal{O}\left(\frac{p^3}{\sqrt{\epsilon}}\right)$
[1], [16]	N/A	$\mathcal{O}\left(\frac{p^2 \log(p)\text{poly}(r)}{\epsilon}\right)$
[15]	$\mathcal{O}\left(pr^4 \log^2(p)\beta^2 \log(\frac{1}{\epsilon})\right)$	$\mathcal{O}\left(mpr \log(\frac{1}{\epsilon}) + p^3\right)$
[3]	$\mathcal{O}(pr^3\beta^2 \log(\frac{1}{\epsilon}))$	$\mathcal{O}\left(mpr \log(\frac{1}{\epsilon}) + p^3\right)$
Algorithm 1	$\mathcal{O}\left(pr^2 \log^4(p) \log(\frac{1}{\epsilon})\right)$	$\mathcal{O}\left(mp^2 \log(\frac{1}{\epsilon})\right)$
Algorithm 2	$\mathcal{O}\left(pr^3 \log^4(p) \log(\frac{1}{\epsilon})\right)$	$\mathcal{O}\left(mpr \log(p) \log(\frac{1}{\epsilon})\right)$
Algorithm 3	$\mathcal{O}\left(pr\right)$	$\mathcal{O}\left(mpr \log(p) \log(\frac{1}{\epsilon})\right)$

Moreover, most non-convex approaches require an accurate initialization, and also require that the underlying matrix L_* is well-conditioned; if this is not the case, the running time of all available methods again inflates to $\Omega(p^3)$, or worse.

In this paper, we take a different approach, and show how to leverage recent results in low-rank approximation to our advantage [17], [18], [19]. In contrast with all earlier works, our methods do not require any full SVD calculations. Specifically, we propose three iterative algorithms. In our first two algorithms, we use a careful concatenation of *randomized, approximate* SVD methods, coupled with appropriately defined gradient steps, to arrive at an ε-accurate matrix estimate. To our knowledge, these approaches constitute the first *linearly convergent* methods for low-rank matrix estimation from rank-one observations.

On the other hand, both these algorithms (at least in theory) require freshly chosen independent samples in each iteration, which is somewhat impractical. To resolve this, we propose a third algorithm that optimizes for a somewhat different loss than the empirical risk defined in (1). For this algorithm, we prove that the sample complexity matches the optimal bound already established by convex methods [11], [12]. This algorithm also exhibits linear convergence. See Table I for a comparison.

C. Our Techniques

At a high level, our method can be viewed as a variant of the seminal, nonconvex algorithms proposed in [21] and [22], which perform iterative projected gradient descent over the manifold of rank-r matrices. However, since computing SVD in high dimensions can be a bottleneck, we cannot use this approach directly. To this end, we use the approximation-based matrix estimation framework proposed in [18]. This work demonstrates how to carefully integrate approximate SVD methods into singular value projection (SVP)-based matrix estimation algorithms; In particular, they use algorithms that satisfy certain "head" and "tail" projection properties (see Section III). Crucially, this framework eliminates the need to compute *even a single SVD*.

We also use this "head-tail" framework. However, a direct application does not succeed for matrix estimation problems obeying the model (2). Two major obstacles arise:

Obstacle 1. It is well-known that the linear operator that maps L_* to y does not satisfy the requisite concentration property (specifically, the Restricted Isometry Property, or RIP, over rank-r matrices) [11], [12], [15]. Therefore, the theoretical analysis of [18] no longer applies. To be more precise, define the operator $\mathcal{A}$ such that $(\mathcal{A}(L_*))_i = x_i^T L_* x_i$ for $i = 1, \ldots, m$, where x_i is a standard normal random vector. It is easy to see that if L_t is the current estimate of the underlying matrix variable, then we have: $\mathbb{E}\mathcal{A}^*\mathcal{A}(L_t - L_*) = 2(L_t - L_*) + Tr(L_t - L_*)I$, where $Tr(\cdot)$ denotes the trace of a matrix.

Obstacle 2. The algebraic structure of the rank-one observations in (2) inflates the running time of computing even a simple gradient update to $\mathcal{O}(p^3)$ (irrespective of the cost of exact or approximate rank-r projection).

We resolve Obstacle 1 by studying the concentration properties of certain linear operators of the form of rank-one projections, leveraging an approach first proposed in [15]. We show that a non-trivial "bias correction" step, coupled with projected descent-type methods, within each iteration is sufficient to achieve fast (linear) convergence. We resolve Obstacle 2 by carefully exploiting the rank-one structure of the observations through developing a modification of the randomized block-Krylov SVD (or BK-SVD) algorithm of [17]. This enables us to achieve fast per-iteration running time.

While the above approach produces fast running time (up to poly-logarithmic factors), its theoretical success depends on the idea of fresh samples in each iteration. Our next algorithm removes this restriction by using the ℓ_1-loss function instead of the squared loss used in (1). However, the ℓ_1-loss is non-differentiable, nor does it satisfy our previous concentration property of the gradient. This motivates us to use the so-called RIP(ℓ_1, ℓ_2) [23], [12], [11]. For this, we propose a projected sub-gradient algorithm which does not require use of fresh samples within each iteration, and enjoys linear convergence with an optimal sample complexity.

II. Related Work

In the context of provable methods for learning neural networks, two-layer networks have received special attention. For instance, [1] has considered a two-layer network with quadratic activation function (identical to the model proposed above), and proposed a greedy, *improper* learning algorithm. Recently, in [6], the authors have proposed a linearly convergent algorithm for learning two-layer networks for several classes of activation functions, together with rigorous upper bounds on the sample complexity of their algorithm. However, their theory does not consider the case of quadratic activations. This paper closes this gap.

Other works have also studied similar two-layer setups, including [4], [5], [7], [8]. In contrast with these results, our framework does not assume the over-parameterized setting where the number of hidden neurons r is greater than p. We now briefly contrast our method with other algorithmic techniques for low-rank matrix estimation. Broadly, two classes of such techniques exist. The first class is based on convex relaxation of the rank constraint [11], [12], [20]. For instance, the authors in [11], [12] demonstrate that the observation operator $\mathcal{A}$ satisfies a specialized mixed-norm isometry condition. Further, they show that the sample complexity of matrix estimation using rank-one projections matches the optimal rate $\mathcal{O}(pr)$. However, these methods advocate using either semidefinite programming (SDP) or proximal sub-gradient algorithms [24], both of which are too slow for very high-dimensional problems.

The second class are non-convex approaches, which are all based on a factorization-based approach initially advocated by [25]. Here, the underlying low-rank matrix variable is factorized as $L_* = UV^T$, where $U, V \in \mathbb{R}^{p\times r}$ [26]. In the Altmin-LRROM method proposed by [15], U and V are updated in alternative fashion. However, the setup in [15] is different from this paper, as it uses an asymmetric observation model. In a subsequent work (called the generalized factorization machine) by [27], U and V are updated based on the construction of certain sequences of moment estimators. Both the approaches of [15] and [27] require a spectral initialization which involves running a rank-r SVD on a given $p \times p$ matrix, and hence the running time heavily depends on the condition number of L_*.

Finally, Frank-Wolfe type greedy algorithms for solving (1) also exist [16], [1]. However, their rate of convergence is sub-linear, and they provide no sample-complexity guarantees. Indeed, the main motivating factor of our paper was to accelerate the running time of such greedy approximation techniques. We complete this line of work by providing (a) rigorous analysis that precisely establishes upper bounds on the number of samples required for learning such networks, and (b) algorithms that provably exhibits *linear* convergence, as well as *(near) linear* per iteration running time.

III. Main results

A. Preliminaries

Let us first introduce some notation. Throughout this paper, $\|\cdot\|_F$ and $\|\cdot\|_2$ denote the matrix Frobenius and spectral norm, respectively, and $\mathrm{Tr}(\cdot)$ denotes matrix trace. Also, $\|\cdot\|_1$ denotes the ℓ_1-norm of a vector. The phrase "with high probability" indicates an event whose failure rate is exponentially small. We assume that the training data samples $(x, y) \in \mathbb{R}^p \times \mathbb{R}$ obey a noisy generative model (2) written as:

$$y = \sum_{j=1}^{r} \alpha_j^* \sigma(\langle w_j^*, x\rangle) = x^T L_* x + e', \tag{3}$$

where $L_* \in \mathbb{R}^{p\times p}$ is the "ground-truth" matrix (with rank equal to r), and $e' \in \mathbb{R}$ is an additive noise. Define $\mathcal{A} : \mathbb{R}^{p\times p} \to \mathbb{R}^m$ such that: $\mathcal{A}(L_*) = [x_1^T L_* x_1, x_2^T L_* x_2, \ldots, x_m^T L_* x_m]^T$, and each $x_i \overset{i.i.d}{\sim} \mathcal{N}(0, I)$ is a normal random vector in $\mathbb{R}^p$ for $i = 1, \ldots, m$. The adjoint operator of $\mathcal{A}$ is defined as $\mathcal{A}^*(y) = \sum_{i=1}^m y_i x_i x_i^T$. Throughout the paper (for the purpose of analysis) we assume that e is zero-mean, subgaussian random vector with i.i.d entries, and independent of x_i.

The analysis of our first two algorithms require that the operators $\mathcal{A}$ and $\mathcal{A}^*$ satisfy the following regularity condition with respect to the set of low-rank matrices. We call this the *Conditional Unbiased Restricted Isometry Property*, or *CU-RIP*(ρ):

Definition 1. *Consider fixed rank-r matrices L_1 and L_2. Then, $\mathcal{A}$ is said to satisfy CU-RIP(ρ) if there exists $0 < \rho < 1$ such that $\left\| L_1 - L_2 - \frac{1}{2m}\mathcal{A}^*\mathcal{A}(L_1 - L_2) - \frac{1}{2m}\mathbf{1}^T\left(\mathcal{A}(L_1) - \mathcal{A}(L_2)\right) I \right\|_2 \le \rho \left\| L_1 - L_2 \right\|_2$.*

Let $\mathbb{U}_r$ denote the set of all rank-r matrix subspaces, i.e., subspaces of $\mathbb{R}^{p\times p}$ which are spanned by any r atoms of the form uv^T where $u, v \in \mathbb{R}^p$ are unit ℓ_2-norm vectors. We use the idea of *head* and *tail* approximate projections with respect to $\mathbb{U}_r$ first proposed in [28], and instantiated in the context of low-rank approximation in [18].

Algorithm 1

Initialization: $L_0 \leftarrow 0$, $t \leftarrow 0$
Calculate: $\bar{y} = \frac{1}{m}\sum_{i=1}^m y_i$
while $t \le K$ **do**
  $\mathrm{Bias}(L_t) := (\frac{1}{2m}\mathbf{1}^T\mathcal{A}(L_t) - \frac{1}{2}\bar{y})I$
  $g(L_t) = \frac{1}{2m}\sum_{i=1}^m \left((x_i^t)^T L_t x_i^t - y_i\right) x_i^t (x_i^t)^T - \mathrm{Bias}(L_t)$
  $L_{t+1} = \mathcal{P}_r\left(L_t - g(L_t)\right)$
  $t \leftarrow t + 1$
end while
Return: $\widehat{L} = L_K$

Definition 2 (Approximate tail projection)**.** *$\mathcal{T} : \mathbb{R}^{p\times p} \to \mathbb{U}_r$ is a ε-approximate tail projection algorithm if for all $L \in \mathbb{R}^{p\times p}$, $\mathcal{T}$ returns a subspace $W = \mathcal{T}(L)$ that satisfies: $\|L - \mathcal{P}_W L\|_F \le (1+\varepsilon)\|L - L_r\|_F$, where L_r is the optimal rank-r approximation of L.*

Definition 3 (Approximate head projection)**.** *$\mathcal{H} : \mathbb{R}^{p\times p} \to \mathbb{U}_r$ is a ε-approximate head projection if for all $L \in \mathbb{R}^{p\times p}$, the returned subspace $V = \mathcal{H}(L)$ satisfies: $\|\mathcal{P}_V L\|_F \ge (1-\varepsilon)\|L_r\|_F$, where L_r is the optimal rank-r approximation of L.*

We also need the following mixed RIP definition due to [23], [12], [11] for our third algorithm, proposed in Section III-C.

Definition 4 (RIP(ℓ_1, ℓ_2) for low-rank matrices)**.** *A linear operator $\mathcal{B}$ satisfies RIP(ℓ_1, ℓ_2) if for any two rank-r matrices L_1 and L_2, there exists constants $0 < \alpha < \beta$ such the following holds: $\alpha\|L_1 - L_2\|_F \le \frac{1}{m}\|\mathcal{B}(L_1 - L_2)\|_1 \le \beta\|L_1 - L_2\|_F$.*

B. Proposed Algorithms

We now propose methods to estimate L_* given knowledge of $\{x_i, y_i\}_{i=1}^m$. Our first method is somewhat computationally inefficient, but achieves nearly good sample complexity and serves to illustrate the overall algorithmic approach. Consider the non-convex, constrained risk minimization problem:

$$\min_{L\in\mathbb{R}^{p\times p}} F(L) = \frac{1}{2m}\sum_{i=1}^m \left(y_i - x_i^T L x_i\right)^2 \text{ s.t. } \mathrm{rank}(L) \le r. \tag{4}$$

To solve this problem, we first propose an algorithm, described in pseudocode form as Algorithm 1[1]. The following theoretical result establishes statistical and optimization convergence rates of our algorithm. More precisely, we derive an upper bound on the estimation error in terms of the spectral norm. (All proofs are deferred to the appendix.)

Theorem 5. *Assume that in each iteration the linear operator $\mathcal{A}$ satisfies CU-RIP(ρ) for some $0 < \rho < \frac{1}{2}$. Then Alg. 1 outputs a sequence of estimates L_t such that (where $0 < q < 1$):*

$$\|L_{t+1} - L_*\|_2 \le q\left\|L_t - L_*\right\|_2 + \frac{1}{2m}(|\mathbf{1}^T e| + \left\|\mathcal{A}^* e\right\|_2), \tag{5}$$

[1] In Alg 1, $\mathcal{P}_r$ denotes the projection operator onto the set of rank-r matrices.

Algorithm 2

Initialization: $L_0 \leftarrow 0$, $t \leftarrow 0$
Calculate: $\bar{y} = \frac{1}{m}\sum_{i=1}^{m} y_i$
while $t \leq K$ **do**
$\text{Bias}(L_t) = (\frac{1}{2m}\mathbf{1}^T\mathcal{A}(L_t) - \frac{1}{2}\bar{y})I$
$g(L_t) = \frac{1}{2m}\sum_{i=1}^{m}\left((x_i^t)^T L_t x_i^t - y_i\right) x_i^t (x_i^t)^T - \text{Bias}(L_t)$
$L_{t+1} = \mathcal{T}\left(L_t - \mathcal{H}\left(g(L_t)\right)\right)$
$t \leftarrow t+1$
end while
Return: $\widehat{L} = L_K$

The contraction factor, q, in Equation (5) can be made small enough if we choose m sufficiently large, and we elaborate on this point in Theorem (7). The second and third term in (5) represent the statistical error rate. Next, we show that these error terms can be suitably bounded.

Theorem 6. *Consider the generative model* (3) *with zero-mean subgaussian noise vector $e \in \mathbb{R}^m$ with i.i.d. entries (and independent of the x_i's) such that $\tau = \max_{1\leq j\leq m}\|e_j\|_{\psi_2}$ (Here, $\|\cdot\|_{\psi_2}$ denotes the ψ_2-norm; subgaussian norm). Then, with probability at least $1-\gamma$, we have:*

$$\frac{1}{m}|\mathbf{1}^T e| + \left\|\frac{1}{m}\mathcal{A}^* e\right\|_2 \leq C_\tau''\sqrt{\frac{p\log^2 p}{m}\log(\frac{p}{\gamma})}. \qquad (6)$$

where $C_\tau'' > 0$ is constant which depends on τ.

To establish linear convergence of our algorithm, we assume that the CU-RIP holds at *each* iteration. The following theorem certifies this assumption.

Theorem 7. *At any iteration t of Alg. 1, with probability at least $1-\xi$, CU-RIP(ρ) is satisfied with parameter $\rho < \frac{1}{2}$ provided that $m = \mathcal{O}\left(\frac{1}{\delta^2}pr^2\log^3 p\log(\frac{p}{\xi})\right)$ for some $\delta > 0$.*

Integrating the above results, together with the assumption of availability of a batch of m independent samples (fresh samples) in each iteration, and all the assumptions in theorem 5, Alg. 1 needs $K = \mathcal{O}(\log(\frac{\|L_*\|_2}{\epsilon}))$ iterations to achieve ϵ-accuracy in terms of the spectral norm. Furthermore, the sample complexity scales as $m = \mathcal{O}\left(pr^2\log^4 p\log(\frac{1}{\epsilon})\right)$ based on Theorems 6 and 7. Although the assumption of "fresh samples" is an artifact of our proof techniques; nonetheless, it is a standard mechanism for theoretically analyzing non-convex problems [29], [6]. In Section III-C, we revisit this issue.

While the above algorithm exhibits linear convergence, the per-iteration complexity is still high since it requires projection onto the space of rank-r matrices. This necessitates the application of SVD. In the absence of any spectral assumptions on the input to the SVD, the per-iteration running time can be as high as *cubic* ($\Omega(p^3)$). Overall, we obtain a running time of $\widetilde{\mathcal{O}}(p^3r^2)$ in order to achieve ϵ-accuracy (please see Section 5.3 in the appendix of [30] for more details).

To reduce the running time, one can instead replace a standard SVD with approximate heuristics such as Lanczos iterations [31]; however, these do not result in rigorous algorithms with provable convergence guarantees. Instead, following [18], we can use a pair of *inaccurate* rank-r projections (in particular, tail-and head-approximate projection operators). Based on this idea, we propose our second algorithm, displayed in pseudocode form as Algorithm 2.

The specific choice of approximate SVD algorithms that simulate the operators $\mathcal{T}(.)$ and $\mathcal{H}(.)$ is flexible. We note that tail-approximate projections have been widely studied in the numerical linear algebra literature [32], [33], [34]; however, head-approximate projection methods are less well-known. In our method, we use the randomized Block Krylov SVD (BK-SVD) method proposed by [17], which has been shown to satisfy both types of approximation guarantees [18]. One can alternatively use LazySVD, recently proposed by [35], which also satisfies both guarantees. The nice feature of these two approaches is that their running time is ***independent of the spectral gap*** of the matrix, which provides an asymptotic improvements over partial SVD algorithms such as the power method; see [35].

We now establish that Alg. 2 also exhibits linear convergence:

Theorem 8. *Consider the sequence of iterates (L_t) obtained in Alg. 2. Assume that in each iteration t, $\mathcal{A}$ satisfies CU-RIP(ρ') for some $0 < \rho' < 1$, then Alg. 2 outputs a sequence of estimates L_t such that:*

$$\|L_{t+1} - L_*\|_F \leq q_1'\|L_t - L_*\|_F + q_2'\left(|\mathbf{1}^T e| + \left\|\mathcal{A}^* e\right\|_2\right), \qquad (7)$$

where $q_1' = (2+\varepsilon)(\rho' + \sqrt{1-\phi^2})$, $q_2' = \frac{\sqrt{r}}{2m}\left(2-\varepsilon+\frac{\phi(2-\varepsilon)(2+\varepsilon)}{\sqrt{1-\phi^2}}\right)$, and $\phi = (1-\varepsilon)(1-\rho') - \rho'$.

Similar to Theorem 7, we can show that CU-RIP is satisfied in each iteration with probability at least $1-\xi$, provided that $m = \mathcal{O}\left(\frac{1}{\delta^2}pr^3\log^3 p\log(\frac{p}{\xi})\right)$. Hence, we require a factor-r increase compared to our first algorithm.

The above analysis shows that instead of using exact rank-r projections using SVDs (as in Alg. 1), one can use instead tail and head approximate projection, as implemented by the BK-SVD method of [17]. The running time for this method is given by $\widetilde{\mathcal{O}}(p^2r)$ if $r \ll p$. While the running time of the projection step is gap-independent, the calculation of the *gradient* (i.e., the input to the head projection method $\mathcal{H}$) is itself the major bottleneck. In essence, this is related to the calculation of the adjoint operator, $\mathcal{A}^*(d) = \sum_{i=1}^m d^{(i)}x_i x_i^T$, which requires $\mathcal{O}(p^2)$ operations for each sample. Coupled with the sample-complexity of $m = \Omega(pr^3)$, this means that the running time per-iteration scaled as $\Omega(p^3r^3)$, which overshadows any gains achieved during the projection step.

To address this challenge, we propose a modified version of BK-SVD for head approximate projection which uses the special rank-one structures involved in the calculation of the gradients. We call this method *Modified BK-SVD*, or MBK-SVD. The basic idea is to *implicitly* evaluate each Krylov-subspace iteration within BK-SVD, and avoid any explicit calculation of the adjoint operator $\mathcal{A}^*$ applied to the current estimate. Due to space constraints, the pseudocode as well as the running time analysis of MBK-SVD (the proof of the following theorem) is given in [30].

Algorithm 3

Initialization: $L_0 \leftarrow 0$, $t \leftarrow 0$
while $t \leq K$ **do**
$L_{t+1} = \mathcal{T}_{\kappa r}\left(L_t - \frac{\eta_t}{m}\mathcal{B}^*\text{sgn}(\mathcal{B}(L_t) - y)\right)$
$t \leftarrow t + 1$
end while
Return: $\widehat{L} = L_K$

Theorem 9. *Algorithm 2 (with modified BK-SVD) runs in time* $K = \mathcal{O}\left(p^2r^4\log^2(\frac{1}{\epsilon})\text{polylog}(p)\right)$.

C. Achieving Optimal Sample Complexity

In the previous section, we saw that both our proposed algorithms result in suboptimal sample complexity by logarithmic factors, primarily because their analysis requires a set of fresh samples in each iteration. In this section, we propose a third algorithm that removes this assumption and achieves asymptotically optimal sample complexity, i.e., $m = \mathcal{O}(pr)$. Referring back to Table I, we observe that convex methods [12], [11] exhibits the same sample complexity, but are very slow. However, we show that our new algorithm enjoys a fast running time.

To overcome the issue of fresh samples, our key intuition is to replace squared loss with the absolute deviation loss function (i.e., ℓ_1-loss), and the CU-RIP with the RIP(ℓ_1, ℓ_2). For simplicity, in this section we ignore noise, while noting that our analysis seamlessly carries over to the noisy case with a somewhat more tedious (but straightforward) extension.

We introduce a (slightly) different observation model: $y' = \mathcal{B}(L_*)$, where $\mathcal{B} : \mathbb{R}^{p\times p} \to \mathbb{R}^m$ denotes a linear operator such that $\mathcal{B}(L)_i = \mathcal{A}(L)_{2i} - \mathcal{A}(L)_{2i-1}$, with $\mathcal{A}$ as defined in observation model (3). It is easy to see that $\mathcal{B}$ can be implemented by doubling the number of training samples. The reason why $\mathcal{B}$ is constructed in this way is inspired by [12], where the authors have shown that $\mathcal{B}$ satisfies RIP(ℓ_1, ℓ_2) if the number of samples $m = O(pr)$. Based on the above model, we consider the following risk minimization problem:

$$\min_{L\in\mathbb{R}^{p\times p}} \quad F(L) = \frac{1}{m}\|y - \mathcal{B}(L)\|_1 \quad \text{s.t.} \quad \text{rank}(L) \leq r. \quad (8)$$

To solve this problem, we propose an approximate projected *sub*-gradient algorithm, displayed in pseudocode as Algorithm 3.

Compared to the previous algorithms, Alg. 3 has three major differences. First, it has only one approximation operator – an approximate tail projection $\mathcal{T}$ – which projects its argument onto a *larger* set of matrices with rank κr for $\kappa > 1$. To implement this operator, we use the modified BK-SVD algorithm similar to the discussion above. The idea of projecting onto a larger space was first proposed by [22], and subsequently has been extended for approximate tail projections by [19]. In that work, we showed that this idea essentially removes the need for an inner "head" projection. Second, the objective function in (8) is not differentiable; hence, we have to use a sub-gradients which, for our case, is given by $\partial F(L) = \frac{1}{m}\mathcal{B}^*\text{sgn}(\mathcal{B}(L_t) - y)$. Third, Algorithm 3 requires a time-varying step-size η_t specified below.

We now prove that with sufficiently many samples, Algorithm 3 converges linearly, and at termination, provides an accurate estimate of the true low-rank matrix. This proof complements the theoretical analysis of [12], [11] with a simple and easily implementable algorithm with provably fast convergence guarantees. For establishing the proof, we need the following Lemma, proved in [19] and adapted to our notation.

Lemma 10. *Let* $\kappa > (1 + \frac{1}{1-\varepsilon})$. *For any matrices* $L, L_* \in \mathbb{R}^{p\times p}$ *with rank*$(L_*) = r$, *we have*

$$\|\mathcal{T}_{\kappa r}(L) - L_*\|_F^2 \leq \left(1 + \frac{2}{\sqrt{1-\varepsilon}\sqrt{\kappa-1}}\right)\|L - L_*\|_F^2,$$

where $\mathcal{T} : \mathbb{R}^{p\times p} \to \mathbb{U}_{\kappa r}$ *denotes the approximate tail projection defined in Definition 2 and* $\varepsilon > 0$ *is the corresponding approximation ratio.*

This lemma says that the near-contraction factor $\nu = 1 + \frac{2}{\sqrt{1-\varepsilon}\sqrt{\kappa-1}}$ can be made arbitrary close to 1, provided that we increase the parameter κ accordingly. We now establish the linear convergence of Algorithm 3:

Theorem 11. *Suppose that the linear map* $\mathcal{B}$ *is constructed such that it satisfies RIP*(ℓ_1, ℓ_2) *property with constants* α *and* β *in each iteration. Set* $\kappa > 1 + \max\left\{\frac{4\left((\frac{\alpha^2}{\beta^2})-1\right)^2}{1-\varepsilon}, \frac{1}{1-\varepsilon}\right\}$. *Choose step size as* $\eta_t = \frac{\|\mathcal{B}(L_t)-y\|_1}{\beta^2}$. *Then, Algorithm 3 produces a sequence of estimates* L_t *for* $t = 1, 2\ldots$ *such that*

$$\|L_{t+1} - L_*\|_2 \leq \lambda\|L_t - L_*\|_2. \quad (9)$$

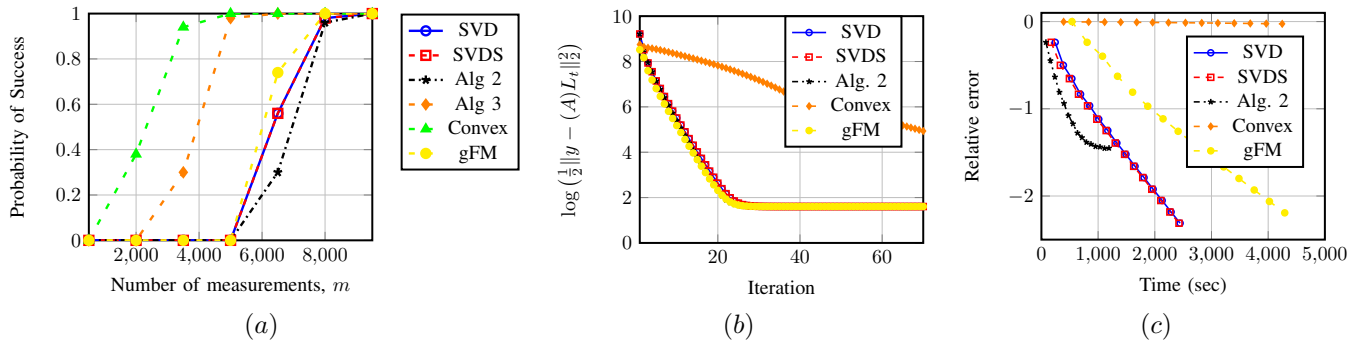
where $\lambda = \sqrt{\nu(1 - \frac{\alpha^2}{\beta^2})}$.

The RIP assumption for $\mathcal{B}$ is justified by Proposition 1 in [12], and the fact that in each iteration, the input matrix of $\mathcal{B}$ is a matrix with rank at most equals to $2\kappa r + r^*$ (see the proof). In addition, the running time of Algorithm 3 scales as $\mathcal{O}(p^2r^2\text{log(p)}\log(\frac{1}{\varepsilon}))$ as a result of implementing $\mathcal{T}$ with MBK-SVD.

IV. Experimental results

We conclude by provided the results of some numerical experiments to support our proposed algorithms.

We compare Alg. 1 and Alg. 2 with convex (nuclear norm) minimization as well as the gFM algorithm of [3]. To solve the nuclear norm minimization, we use FASTA [24] (accelerated proximal sub-gradient). In addition, SVD and SVDS denote the projection step in Alg. 1 using Matlab's SVD and SVDs functions respectively. In all the experiments, we generate a low-rank matrix, $L_* = UU^T$, such that $U \in \mathbb{R}^{p\times r}$ with $r = 5$ where the entries of U is randomly chosen according to the standard normal distribution. Figures 1(a) and 1(b) show the phase transition of successful estimation as well as the evolution of the objective function, $\frac{1}{2}\|y - \mathcal{A}(L_t)\|_2^2$ versus the iteration count t for all the algorithms. In figure 1(a), we have used 50 Monte Carlo trials and the phase transition plot is generated based on the empirical probability of success; here, success is when the relative error between $\hat{L}$ (the estimate of L_*) and the ground truth L_* (measured in terms of spectral norm) is less than 0.05. For solving convex problem, we set the



(a) (b) (c)

Fig. 1: *Comparison of algorithms. (a) Phase transition plot with* $p = 100$. *(b) Evolution of the objective function versus number of iterations with* $p = 100$, $m = 8500$, *and noise level* $\sigma = 0.1$. *(c) Running time of the algorithm with* $p = 1000$ *and* $m = 75000$.

Lagrangian parameter, μ via a grid search. In Figure 1(a), there is no additive noise. As we can see in this Figure, the phase transition for the convex method and Alg 3 has comparable phase transition as predicted by theory, and they are slightly better than those for non-convex algorithms, which is consistent with known theoretical results. However, the convex method is *improper*, i.e., the rank of $\hat{L}$ is much higher than the target rank. In Figure 1(b) we consider an additive standard normal noise with standard deviation equal to 0.1, and average over 10 Monte Carlo trials. As illustrated in this plot, all non-convex algorithm have much better performance in decreasing the objective function compared to convex method.

In Figure 1(c), we compare the algorithms in the high-dimensional regime where $p = 1000$, $m = 75000$, and $r = 5$ in terms of running time. We let all the algorithms run 15 iterations, and then compute the CPU time in seconds for each of them. The y-axis denotes the logarithm of relative error in spectral norm and we report averages over 10 Monte Carlo trials. As we can see, convex methods are the slowest (as expected); the non-convex methods are comparable to each other, while our method is the fastest. This plot verifies that Alg. 2 is faster than other non-convex methods, which makes it a promising approach for high-dimensional matrix estimation applications.

Discussion. It seems plausible that the matrix-based techniques of this paper can be extended to learn networks with similar polynomial-like activation functions (such as the squared ReLU). Finally, similar algorithms can be plausibly used to train multi-layer networks using a greedy (layer-by-layer) learning strategy.

REFERENCES

[1] R. Livni, S. Shalev-Shwartz, and O. Shamir, "On the computational efficiency of training neural networks," in *Adv. Neural Inf. Proc. Sys. (NIPS)*, 2014, pp. 855–863.

[2] Z. Ghodsi, T. Gu, and S. Garg, "Safetynets: Verifiable execution of deep neural networks on an untrusted cloud," in *Adv. Neural Inf. Proc. Sys. (NIPS)*, 2017, pp. 4675–4684.

[3] M. Lin and J. Ye, "A non-convex one-pass framework for generalized factorization machine and rank-one matrix sensing," in *In Adv. Neural Inf. Proc. Sys. (NIPS)*, 2016, pp. 1633–1641.

[4] M. Janzamin, H. Sedghi, and A. Anandkumar, "Beating the perils of non-convexity: Guaranteed training of neural networks using tensor methods," *arXiv preprint arXiv:1506.08473*, 2015.

[5] Y. Tian, "Symmetry-breaking convergence analysis of certain two-layered neural networks with relu nonlinearity," *In Submitted to ICLR 2017*, 2016.

[6] K. Zhong, Z. Song, P. Jain, P. L Bartlett, and I. Dhillon, "Recovery guarantees for one-hidden-layer neural networks," 2016.

[7] M. Soltanolkotabi, Adel J., and J. Lee, "Theoretical insights into the optimization landscape of over-parameterized shallow neural networks," *arXiv preprint arXiv:1707.04926*, 2017.

[8] Y. Li and Y. Yuan, "Convergence analysis of two-layer neural networks with relu activation," *In Adv. Neural Inf. Proc. Sys. (NIPS)*, 2017.

[9] E. Candès and B. Recht, "Exact matrix completion via convex optimization.," *Found. Comput. Math.*, vol. 9, no. 6, 2009.

[10] B. Recht, M. Fazel, and P. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM review*, vol. 52, no. 3, pp. 471–501, 2010.

[11] T. Cai and A. Zhang, "Rop: Matrix recovery via rank-one projections," *Ann. Stat.*, vol. 43, no. 1, pp. 102–138, 2015.

[12] Y. Chen, Y. Chi, and A. Goldsmith, "Exact and stable covariance estimation from quadratic sampling via convex programming," *IEEE Trans. Inform. Theory*, vol. 61, no. 7, pp. 4034–4059, 2015.

[13] E. Candes, T. Strohmer, and V. Voroninski, "Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming," *Comm. Pure Appl. Math.*, vol. 66, no. 8, pp. 1241–1274, 2013.

[14] P. Netrapalli, P. Jain, and S. Sanghavi, "Phase retrieval using alternating minimization," in *Adv. Neural Inf. Proc. Sys. (NIPS)*, 2013, pp. 2796–2804.

[15] K. Zhong, P. Jain, and I. Dhillon, "Efficient matrix sensing using rank-1 gaussian measurements," in *Int. Conf on Algorithmic Learning Theory*. Springer, 2015, pp. 3–18.

[16] S. Shalev-Shwartz, A. Gonen, and O. Shamir, "Large-scale convex minimization with a low-rank constraint," in *Proc. Int. Conf. Machine Learning*, 2011, pp. 329–336.

[17] C. Musco and C. Musco, "Randomized block krylov methods for stronger and faster approximate singular value decomposition," in *Adv. Neural Inf. Proc. Sys. (NIPS)*, 2015, pp. 1396–1404.

[18] C. Hegde, P. Indyk, and L. Schmidt, "Fast recovery from a union of subspaces," in *Adv. Neural Inf. Proc. Sys. (NIPS)*, 2016.

[19] M. Soltani and C. Hegde, "Fast low-rank matrix estimation without the condition number," *arXiv preprint arXiv:1712.03281*, 2017.

[20] R. Kueng, H. Rauhut, and U. Terstiege, "Low rank matrix recovery from rank one measurements," *Appl. Comput. Harmon. Anal.*, vol. 42, no. 1, pp. 88–116, 2017.

[21] P. Jain, R. Meka, and I. Dhillon, "Guaranteed rank minimization via singular value projection," in *Adv. Neural Inf. Proc. Sys. (NIPS)*, 2010, pp. 937–945.

[22] P. Jain, A. Tewari, and P. Kar, "On iterative hard thresholding methods for high-dimensional m-estimation," in *Adv. Neural Inf. Proc. Sys. (NIPS)*, 2014, pp. 685–693.

[23] S. Foucart, "Flavors of compressive sensing," in *International Conference Approximation Theory*. Springer, 2016, pp. 61–104.

[24] T. Goldstein, C. Studer, and R. Baraniuk, "FASTA: A generalized implementation of forward-backward splitting," January 2015, http://arxiv.org/abs/1501.04979.

[25] S. Burer and R. Monteiro, "A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization," *Mathematical Programming*, vol. 95, no. 2, pp. 329–357, 2003.

[26] S. Tu, R. Boczar, M. Simchowitz, M. Soltanolkotabi, and B. Recht, "Low-rank solutions of linear matrix equations via procrustes flow," in *icml*, 2016, pp. 964–973.

[27] M. Lin, S. Qiu, B. Hong, and J. Ye, "The second order linear model," *arXiv preprint arXiv:1703.00598*, 2017.

[28] C. Hegde, P. Indyk, and L. Schmidt, "Approximation algorithms for model-based compressive sensing," *IEEE Trans. Inform. Theory*, vol. 61, no. 9, pp. 5129–5147, 2015.

[29] M. Hardt, "Understanding alternating minimization for matrix completion," in *Proc. IEEE Symp. Found. Comp. Science (FOCS)*. IEEE, 2014, pp. 651–660.

[30] M. Soltani and C. Hegde, "Towards provable learning of polynomial neural networks using low-rank matrix estimation," in *Proc. Int. Conf. Art. Intell. Stat. (AISTATS)*, 2017.

[31] C. Lanczos, "An iteration method for the solution of the eigenvalue problem of linear differential and integral operators1," *Journal of Research of the National Bureau of Standards*, vol. 45, no. 4, 1950.

[32] K. L Clarkson and D. Woodruff, "Low rank approximation and regression in input sparsity time," in *Proc. ACM Symp. Theory of Comput.* ACM, 2013, pp. 81–90.

[33] M. Mahoney and P. Drineas, "Cur matrix decompositions for improved data analysis," *Proc. Natl. Acad. Sci.*, vol. 106, no. 3, pp. 697–702, 2009.

[34] V. Rokhlin, A. Szlam, and M. Tygert, "A randomized algorithm for principal component analysis," *SIAM J. Matrix Anal. Appl.*, vol. 31, no. 3, pp. 1100–1124, 2009.

[35] Z. Allen-Zhu and Y. Li, "Lazysvd: Even faster svd decomposition yet without agonizing pain," in *Adv. Neural Inf. Proc. Sys. (NIPS)*, 2016, pp. 974–982.

[36] R. Vershynin, "Introduction to the non-asymptotic analysis of random matrices," *arXiv preprint arXiv:1011.3027*, 2010.

V. Appendix

Proof of Theorem 5. The proof of the theorem is special case and simpler version of the Theorem 8. Hence, we give a proof sketch (see [30] for more details). The idea is analogous to the proof of IHT algorithm in compressive sensing literature where the projection step along with the triangle inequality results $\left\|L_{t+1}-L_*\right\|_2 \leq 2\left\|b-L_*\right\|_2$ where b is the update rule vector. Then, we invoke CU-RIP by the assumption of the theorem. This establishes our desired linear convergence with contraction factor $q=2\rho<1$. □

Proof of Theorem 8. Assume that $Y \in \mathcal{M}(\mathbb{U}_{2r})$ such that $L_t - L_* \in Y$ and

$$V := V_t = \mathcal{H}\left(\mathcal{A}^*(\mathcal{A}(L_t)-y) - Tr(L_t-\bar{y})I\right).$$

Also, define $\text{Bias}(L_t) := (\frac{1}{2m}\mathbf{1}^T\mathcal{A}(L_t) - \frac{1}{2}\bar{y})I$

$$\begin{aligned} b' &= L_t - \mathcal{H}\left(\frac{1}{m}\sum_{i=1}^{2m}\left(x_i L_t x_i^T - y_i\right)x_i x_i^T - \text{Bias}(L_t)\right) \\ &= L_t - \frac{1}{2m}\mathcal{H}\left(\mathcal{A}^*(\mathcal{A}(L_t)-y) - \text{Bias}(L_t)\right). \end{aligned}$$

Furthermore, by definition of approximate tail projection, $L_t \in \mathcal{M}(\mathbb{U}_r)$. Now, we have:

$$\begin{aligned} &\|L_{t+1}-L_*\|_F = \|L_* - \mathcal{T}(b')\|_F \\ &\leq \|L^* - b'\|_F + \|b' - \mathcal{T}(b')\|_F \overset{a_1}{\leq} (2+\varepsilon)\|b'-L_*\|_F \\ &= (2+\varepsilon)\Big\|L_t - L_* - \mathcal{H}(\frac{1}{2m}\mathcal{A}^*(\mathcal{A}(L_t)-y) \\ &\qquad - (\frac{1}{2m}\mathbf{1}^T\mathcal{A}(L_t) - \frac{1}{2}\bar{y})I)\Big\|_F \\ &\overset{a_2}{=} (2+\varepsilon)\Big\|L_t - L_* - \mathcal{P}_V(\frac{1}{2m}\mathcal{A}^*(\mathcal{A}(L_t)-y) \\ &\qquad - (\frac{1}{2m}\mathbf{1}^T\mathcal{A}(L_t) - \frac{1}{2}\bar{y})I)\Big\|_F, \end{aligned}$$

where a_1 is implied by the triangle inequality and the definition of approximate tail projection, and inequality a_2 holds by the definition of approximate head projection. Next, we have:

$$\begin{aligned} &\|L_{t+1}-L_*\|_F \overset{a_3}{\leq} (2+\varepsilon)\Big\|\mathcal{P}_V(L_t-L_*) + \mathcal{P}_{V^\perp}(L_t-L_*) \\ &\quad - \mathcal{P}_V(\frac{1}{2m}\mathcal{A}^*(\mathcal{A}(L_t)-y) - (\frac{1}{2m}\mathbf{1}^T\mathcal{A}(L_t) - \frac{1}{2}\bar{y})I)\Big\|_F \\ &\overset{a_4}{\leq} (2+\varepsilon)\Big\|\mathcal{P}_V(L_t-L_*) - \mathcal{P}_V(\frac{1}{2m}\mathcal{A}^*\mathcal{A}(L_t-L_*) \\ &\quad - (\frac{1}{2m}\mathbf{1}^T\mathcal{A}(L_t) - \frac{1}{2}\bar{y})I)\Big\|_F + (2+\varepsilon)\Big\|\mathcal{P}_{V^\perp}(L_t-L_*)\Big\|_F \\ &\qquad + \frac{2+\varepsilon}{2m}\Big\|\mathcal{P}_V\mathcal{A}^*e\Big\|_F \\ &\overset{a_5}{\leq} (2+\varepsilon)\Big\|\mathcal{P}_{V+Y}(L_t - L_* - (\frac{1}{2m}\mathcal{A}^*\mathcal{A}(L_t-L_*) \\ &\quad - \frac{1}{2m}\mathbf{1}^T\mathcal{A}(L_t-L_*)I))\Big\|_F + (2+\varepsilon)\Big\|\mathcal{P}_{V^\perp}(L_t-L_*)\Big\|_F \\ &\qquad + \frac{2+\varepsilon}{2m}\left(|\mathbf{1}^Te| + \Big\|\mathcal{P}_V\mathcal{A}^*e\Big\|_F\right), \end{aligned} \tag{10}$$

where a_3 follows by decomposing the residual $L_t - L_*$ on the two subspaces V and $V^\perp$, and a_4 is due to the triangle inequality, the fact that $L_t - L_* \in Y$, and $V \subseteq V+Y$.

Now, we need to bound the three terms in (10). The third and fourth terms can be bounded by using Theorem 6. For the first term, we have:

$$\begin{aligned} &(2+\varepsilon)\Big\|\mathcal{P}_{V+Y}(L_t - L_* - (\frac{1}{2m}\mathcal{A}^*\mathcal{A}(L_t-L_*) \\ &\qquad - \frac{1}{2m}\mathbf{1}^T\mathcal{A}(L_t-L_*)I))\Big\|_F \\ &\overset{a_1}{\leq} (2+\varepsilon)\Big\|L_t - L_* - (\frac{1}{2m}\mathcal{A}^*\mathcal{A}(L_t-L_*) \\ &\qquad - \frac{1}{2m}\mathbf{1}^T\mathcal{A}(L_t-L_*)I)\Big\|_F \\ &\overset{a_2}{\leq} (2+\varepsilon)\rho'\Big\|L_t - L_*\Big\|_F, \end{aligned} \tag{11}$$

above, a_1 holds by the properties of the Frobenius and spectral norm, and a_2 is due to the CU-RIP assumption in the theorem. To bound the second term in (10), $(2+\varepsilon)\Big\|\mathcal{P}_{V^\perp}(L_t-L_*)\Big\|_F$, we have:

$$\begin{aligned} &\Big\|\mathcal{P}_V\left(\frac{1}{2m}\mathcal{A}^*(\mathcal{A}(L_t)-y) - (\frac{1}{2m}\mathbf{1}^T\mathcal{A}(L_t)-\bar{y})I\right)\Big\|_F \\ &\overset{a_1}{\geq} (1-\varepsilon)\Big\|\mathcal{P}_Y\left(\frac{1}{2m}\mathcal{A}^*(\mathcal{A}(L_t)-y) - (\frac{1}{2m}\mathbf{1}^T\mathcal{A}(L_t)-\bar{y})I\right)\Big\|_F \\ &\overset{a_2}{\geq} (1-\varepsilon)\Big\|\mathcal{P}_Y\left(\frac{1}{2m}\mathcal{A}^*\mathcal{A}(L_t-L_*) - \frac{1}{2m}\mathbf{1}^T\mathcal{A}(L_t-L_*)I\right)\Big\|_F \\ &\qquad - \frac{1-\varepsilon}{2m}|\mathbf{1}^Te| - \frac{1-\varepsilon}{2m}\Big\|\mathcal{P}_V\mathcal{A}^*e\Big\|_F \\ &\overset{a_3}{\geq} (1-\varepsilon)(1-\rho')\Big\|L_t-L_*\Big\|_F - \frac{1-\varepsilon}{2m}\left(|\mathbf{1}^Te| + \Big\|\mathcal{P}_V\mathcal{A}^*e\Big\|_F\right), \end{aligned} \tag{12}$$

Here, a_1 holds by the definition of approximate head projection, a_2 is followed by triangle inequality, a_3 is due to Corollary 15,

and finally a_4 holds due to the fact that $\text{rank}(L_t - L_*) \leq 2r$. For the upper bound, we have:

$$\begin{aligned}
&\left\|\mathcal{P}_V\left(\frac{1}{2m}\mathcal{A}^*(\mathcal{A}(L_t)-y) - (\frac{1}{2m}\mathbf{1}^T\mathcal{A}(L_t)-\bar{y})I\right)\right\|_F \\
&\overset{a_1}{\leq} \left\|\mathcal{P}_{V+Y}\left(\frac{1}{2m}\mathcal{A}^*\mathcal{A}(L_t-L_*) - \frac{1}{2m}\mathbf{1}^T\mathcal{A}(L_t-L_*)I\right)\right. \\
&\qquad \left. - \mathcal{P}_{V+Y}(L_t-L_*)\right\|_F + \left\|\mathcal{P}_V(L_t-L_*)\right\|_F \\
&\qquad + \frac{1}{2m}\left(|\mathbf{1}^Te| + \left\|\mathcal{P}_V\mathcal{A}^*e\right\|_F\right) \\
&\overset{a_2}{\leq} \left\|L_t - L_* - \frac{1}{2m}\mathcal{A}^*(\mathcal{A}(L_t)-y) + \frac{1}{2m}\mathbf{1}^T\mathcal{A}(L_t-L_*)I\right\|_F \\
&\qquad + \left\|\mathcal{P}_V(L_t-L_*)\right\|_F + \frac{1}{2m}\left(|\mathbf{1}^Te| + \left\|\mathcal{P}_V\mathcal{A}^*e + \right\|_F\right) \\
&\overset{a_3}{\leq} \rho'\left\|L_t - L_*\right\|_F + \left\|\mathcal{P}_V(L_t-L_*)\right\|_F \\
&\qquad + \frac{1}{2m}\left(|\mathbf{1}^Te| + \left\|\mathcal{P}_V\mathcal{A}^*e\right\|_F\right), \qquad (13)
\end{aligned}$$

above, a_1 holds by triangle inequality and the fact that projection onto the extended subspace $V+Y$ ($V \subseteq V+Y$) does not decrease the Frobenius norm, a_2 is due to the inequality $\|AB\|_F \leq \|A\|_2\|B\|_F$, and finally a_3 is followed by CU-RIP assumption and the fact that $\text{rank}(L_t - L_*) \leq 2r$. Putting together (12) and (13), we obtain:

$$\begin{aligned}
\left\|\mathcal{P}_V(L_t-L_*)\right\|_F &\geq ((1-\varepsilon)(1-\rho')-\rho')\left\|L_t-L_*\right\|_F \\
&\quad - \frac{2-\varepsilon}{2m}\left(|\mathbf{1}^Te| + \left\|\mathcal{P}_V\mathcal{A}^*e\right\|_F\right). \qquad (14)
\end{aligned}$$

By the Pythagoras theorem, we know $\left\|\mathcal{P}_V(L_t-L_*)\right\|_F^2 + \left\|\mathcal{P}_{V^\perp}(L_t-L_*)\right\|_F^2 = \left\|L_t-L_*\right\|_F^2$, and hence we can bound the second term in (10). To use this fact, we apply (14) in [18] which results:

$$\begin{aligned}
(2+\varepsilon)\left\|\mathcal{P}_{V^\perp}(L_t-L_*)\right\|_F &\leq (2+\varepsilon)\sqrt{1-\phi^2}\left\|L_t-L_*\right\|_F \\
&\quad + \frac{\phi(2-\varepsilon)(2+\varepsilon)}{2m\sqrt{1-\phi^2}}\left(|\mathbf{1}^Te| + \left\|\mathcal{P}_V\mathcal{A}^*e\right\|_F\right), \qquad (15)
\end{aligned}$$

where $\phi = (1-\varepsilon)(1-\rho') - \rho'$. Putting all the bounds in (11), and (15) altogether, we obtain:

$$\begin{aligned}
\|L_{t+1}-L_*\|_F &\leq \left((2+\varepsilon)\rho' + (2+\varepsilon)\sqrt{1-\phi^2}\right)\left\|L_t-L_*\right\|_F \\
&\quad + \frac{\sqrt{r}}{2m}\left(2-\varepsilon+\frac{\phi(2-\varepsilon)(2+\varepsilon)}{\sqrt{1-\phi^2}}\right)\left(|\mathbf{1}^Te| + \left\|\mathcal{A}^*e\right\|_2\right) \\
&= q_1'\left\|L_t-L_*\right\|_F + q_2'\left(|\mathbf{1}^Te| + \left\|\mathcal{A}^*e\right\|_2\right). \qquad (16)
\end{aligned}$$

We choose $q_1' = (2+\varepsilon)(\rho' + \sqrt{1-\phi^2})$, and $q_2' = \frac{\sqrt{r}}{2m}\left(2-\varepsilon+\frac{\phi(2-\varepsilon)(2+\varepsilon)}{\sqrt{1-\phi^2}}\right)$. Now in order to have convergence, we have to make sure that $0 < \phi < 1$ and $q_1' < 1$. These conditions are achieved if we let choose m sufficiently large such that $\rho' < \frac{1}{2+\varepsilon} - \sqrt{1-\phi^2}$. The completes the proof. □

Lemma 12. *(Bernstein-type inequality for symmetric random matrices). Consider a sequence of symmetric and random independent identical distributed matrices $\{S_i\}_{i=1}^m$ with dimension $p \times p$. Also, assume that $\|S_i - \mathbb{E}S_i\|_2 \leq R$ for $i = 1, \ldots, m$. Then for all $t \geq 0$,*

$$\mathbb{P}\left(\left\|\frac{1}{m}\sum_{i=1}^m S_i - \mathbb{E}S_i\right\|_2 \geq t\right) \leq 2p\exp\left(\frac{-mt^2}{\sigma + Rt/3}\right),$$

where $\sigma = \|\mathbb{E}\left(S - \mathbb{E}S\right)^2\|_2$ and S is a independent copy of S_i's.

Before verifying CU-RIP, we need the following lemmas. In the first lemma, we show that $\bar{y} = \frac{1}{m}\sum_{i=1}^m y_i$ is concentrated around its mean with high probability.

Lemma 13 (Concentration of $\bar{y}$)**.** *Let $\mathcal{A} : \mathbb{R}^{p\times p} \to \mathbb{R}^m$ be a linear operator defined as* (3) *and $L \in \mathbb{R}^{p\times p}$ be some symmetric matrix. Then with probability at least $1-\xi_1$, we have for some constant $C > 0$:*

$$|\frac{1}{m}\mathbf{1}^T\mathcal{A}(L) - Tr(L)| \leq C\sqrt{\frac{1}{m}\log(\frac{p}{\xi_1})}\|L\|_2. \qquad (17)$$

Proof. In all the following expressions, $c_l > 0$ for $l = 1, \ldots, 4$ are absolute constants. We start by noting that: $\mathbb{E}\mathcal{A}(L) = \mathbb{E}Tr(x_ix_i^TL) = Tr(L)$ where we have used the fact that $x_i \overset{i.i.d}{\sim} \mathcal{N}(0, I)$. We have for all $t > 0$:

$$\begin{aligned}
&\mathbb{P}\left(\left|\frac{1}{m}\mathbf{1}^T\mathcal{A}(L) - Tr(L)\right| \geq t\right) \\
&= \mathbb{P}\left(\left|\frac{1}{m}\sum_{i=1}^m\langle x_ix_i^T, L\rangle - Tr(L)\right| \geq t\right) \\
&= \mathbb{P}\left(\left|\frac{1}{m}\sum_{i=1}^m\sum_{u,v}(x_i^ux_i^vL^{uv}) - Tr(L)\right| \geq t\right) \\
&= \mathbb{P}(\Big|\sum_u\frac{1}{m}\sum_{i=1}^m((x_i^u)^2L^{uu} - L^{uu}) \\
&\qquad + \sum_{u\neq v}\frac{1}{m}\sum_{i=1}^m(x_i^ux_i^vL^{uv})\Big| \geq t). \qquad (18)
\end{aligned}$$

Now we bound two probabilities. First, $\forall\ t_1 \geq 0$: $\mathbb{P}\left(\left|\sum_u\frac{1}{m}\sum_{i=1}^m((x_i^u)^2L^{uu} - L^{uu})\right| \geq t_1\right) \leq p\exp\left(-c_1\frac{mt_1^2}{\|L\|_2^2}\right)$, where the inequality is due to the union bound over p diagonal variables and by the fact that $(x_i^u)^2$ is a χ^2 random variable with mean 1 and $\|\chi^2\|_{\psi_1} = 2$; as a result, we can use the scalar version of Bernstein inequality. Now by choosing $t_1 \geq c_2\|L\|_2\sqrt{\frac{\log(\frac{p}{\xi_1'})}{m}}$, with probability at least $1-\xi_1'$, we have:

$$\Big|\sum_u\frac{1}{m}\sum_{i=1}^m((x_i^u)^2L^{uu} - L^{uu})\Big| \leq \sqrt{\frac{c_2}{m}\log(\frac{p}{\xi_1'})}\|L\|_2. \qquad (19)$$

Second, let $k = \max_{u\neq v}(L^{uv})^2$. Thus, $\forall t_2 \geq 0$, we have, $\mathbb{P}\left(\left|\sum_{u\neq v}\frac{1}{m}\sum_{i=1}^m(x_i^ux_i^vL^{uv})\right| \geq t_2\right) \overset{a_2}{\leq} p^2\exp\left(-c_2\frac{mt_2^2}{k^2}\right)$, where a_2 holds by a union bound over $p^2 - p$ off-diagonal variables, and the fact that $x_i^ux_i^v$ is a zero mean subexponential random variable. Hence, we can again use the scalar version

of Bernstein inequality. By choosing $t_2 \geq \sqrt{\frac{c_3}{m}\log(\frac{p}{\xi_1^{''}})}$, with probability at least $1-\xi_1^{''}$, we have:

$$\Big|\sum_{u\neq v}\frac{1}{m}\sum_{i=1}^{m}(x_i^u x_i^v L^{uv})\Big| \leq \sqrt{\frac{c_3}{m}\log(\frac{p}{\xi_1^{''}})}. \tag{20}$$

Now from (18), (19), and (20) and by choosing $t = t_1 + t_2$ with probability at least $1-\xi_1$ where $\xi_1 = \xi_1^{'} + \xi_1^{''}$, we obtain:

$$\mathbb{P}\left(|\frac{1}{m}\mathbf{1}^T\mathcal{A}(L) - Tr(L)| \geq t\right) \leq \sqrt{\frac{c_4}{m}\log(\frac{p}{\xi_1^{'}})}\|L\|_2.$$

which proves the stated claim. □

Lemma 14 (Concentration of $\frac{1}{m}\mathcal{A}^*\mathcal{A}(M)$)**.** *Let $M \in \mathbb{R}^{p\times p}$ be a fixed matrix with rank r and let $S_i = x_i x_i^T(M)x_i x_i^T$ for $i = 1,\ldots,m$. Consider the linear operator $\mathcal{A}$ in model* (3) *independent of M. Then with probability at least $1-\xi_2$, we have:*

$$\Big\|\frac{1}{m}\sum_{i=1}^{m}S_i - \mathbb{E}S_i\Big\|_2 \leq C'\sqrt{\frac{pr^2\log^3 p}{m}\log(\frac{p}{\xi_2})}\|M\|_2. \tag{21}$$

where $C' > 0$ is a constant.

Proof. In all the following expressions, $C_l > 0$ for $l = 1,\ldots,11$ are absolute constants. First we note that by some calculations, one can show that $\mathbb{E}\left(\frac{1}{m}\mathcal{A}^*\mathcal{A}(M)\right) = \mathbb{E}S_i = 2(M) + Tr(M)I$. Our technique to establish the concentration of $\mathcal{A}^*\mathcal{A}(L_t - L_*)$ is based on the matrix Bernstein inequality. As stated in lemma (12), there should be a spectral bound on the summands, $S_i = x_i x_i^T(M)x_i x_i^T$ for $i = 1,\ldots,m$. Since the entries of a_i are Gaussian, the spectral norm is not absolutely bounded; hence, we cannot directly use the matrix Bernstein inequality. Inspired by [15], we will use a *truncation* trick to make sure that the spectral norm of summands are bounded. Define the random variable $\widetilde{x_i}^{(j)}$ as follows:

$$\widetilde{x_i}^{(j)} = \begin{cases} x_i^{(j)}, & |x_i^{(j)}| \leq C_1\sqrt{\log mp} \\ 0, & otherwise, \end{cases} \tag{22}$$

where $x_i^{(j)}$ is the j^{th} entry of the random vector x_i. By this definition, we immediately have the following properties:

- $\mathbb{P}\left(x_i^{(j)} = \widetilde{x_i}^{(j)}\right) \geq 1 - \frac{1}{(mp)^{C_2}}$,
- $\mathbb{E}\left(\widetilde{x_i}^{(j)}\widetilde{x_i}^{(k)}\right) = 0$, for $j \neq k$,
- $\mathbb{E}\widetilde{x_i}^{(j)} = 0$ for $j = 1,\ldots,p$,
- $\mathbb{E}\left(\widetilde{x_i}^{(j)}\right)^2 \leq \mathbb{E}\left(x_i^{(j)}\right)^2 = 1$, for $j = 1,\ldots,p$,

Let $\widetilde{S_i} = \widetilde{x_i}\widetilde{x_i}^T M\widetilde{x_i}\widetilde{x_i}^T$ for $i = 1,\ldots,m$. We need to bound parameters R and σ in the matrix Bernstein inequality. Denote the SVD of M by $M = U_M\Sigma V_M^T$. Since x_i is a normal random vector, it is rotationally invariant. As a result, w.l.o.g., we can assume that $U_M = [e_1, e_2, \ldots, e_r]$ and $V_M = [e_1, e_2, \ldots, e_r]$ as long as the random vector x_i is independent of M. Here, e_j denotes the j^{th} canonical basis vector in $\mathbb{R}^p$. To make sure this happens, we use m fresh samples of x_i's in each iteration of the algorithm. Now, we have for each i:

$$\begin{aligned}
&\|\widetilde{x_i}\widetilde{x_i}^T M\widetilde{x_i}\widetilde{x_i}^T\|_2 = \|\widetilde{x_i}\widetilde{x_i}^T U_M\Sigma V_M^T\widetilde{x_i}\widetilde{x_i}^T\|_2 \\
&\leq |\widetilde{x_i}^T U_M\Sigma V_M^T\widetilde{x_i}|\|\widetilde{x_i}\widetilde{x_i}^T\|_2 \\
&\leq \|U_M^T\widetilde{x_i}\|_2\|V_M^T\widetilde{x_i}\|_2\|\widetilde{x_i}\|_2^2\|M\|_2 \\
&\overset{a_1}{\leq} pr\|\widetilde{x_i}\|_\infty^4\|M\|_2 \overset{a_2}{\leq} C_3 pr\log^2(mp)\|M\|_2,
\end{aligned}$$

above, a_1 holds due to rotational invariance discussed above, and the relation between ℓ_2 and ℓ_∞ norms. Also, a_2 is due to applying bound in (22). Now, we can calculate R as: $\|\widetilde{S_i} - \mathbb{E}\widetilde{S_i}\|_2 \leq \|\widetilde{S_i}\|_2 + \mathbb{E}\|\widetilde{S_i}\|_2 \leq 2\|\widetilde{S_i}\|_2 \leq C_4 pr\log^2(mp)\|M\|_2 = R$, where we have used both the triangle inequality and Jensen's inequality in the first inequality above. For σ, we define $\widetilde{S}$ as the truncated version of S, independent copy of S_i's. Hence:

$$\begin{aligned}
&\sigma = \left\|\mathbb{E}\widetilde{S}^2 - (\mathbb{E}\widetilde{S})^2\right\|_2 \overset{a_1}{\leq} \|\mathbb{E}\widetilde{S}^2\|_2 \\
&= \left\|\mathbb{E}\left(\widetilde{x}\widetilde{x}^T M\widetilde{x}\widetilde{x}^T\widetilde{x}\widetilde{x}^T M\widetilde{x}\widetilde{x}^T\right)\right\|_2 \\
&= \left\|\mathbb{E}\left(\|\widetilde{x}\|_2^2\left(\widetilde{x}^T M\widetilde{x}\right)^2\widetilde{x}\widetilde{x}^T\right)\right\|_2 \\
&\overset{a_2}{\leq} C_5 pr^2\log^3(pm)\|M\|_2^2\left\|\mathbb{E}\left(\widetilde{x}\widetilde{x}^T\right)\right\|_2 \overset{a_3}{\leq} C_5 pr^2\log^3(pm)\|M\|_2^2,
\end{aligned}$$

where a_1 is followed as $(\mathbb{E}\widetilde{S})^2$ is a positive semidefinite matrix. In addition, a_2 holds due to the upper bound on $\left(\widetilde{x}^T M\widetilde{x}\right)^2\|\widetilde{x}\|_2^2$, i.e., $\left(\widetilde{x}^T M\widetilde{x}\right)^2\|\widetilde{x}\|_2^2 = \left(\widetilde{x}^T U_M\Sigma V_M^T\widetilde{x}\right)^2\|\widetilde{x}\|_2^2 \leq \|U_M^T\widetilde{x}\|_2^2\|V_M^T\widetilde{x}\|_2^2\|M\|_2^2\|\widetilde{x}\|_2^2 \leq pr^2\|\widetilde{x}\|_\infty^6\|M\|_2 \leq C_6 pr^2\log^3(mp)\|M\|_2$, where we have again used the same argument of rotational invariance. Finally, a_3 holds due to the fact that $\mathbb{E}\left(\widetilde{x_i}\widetilde{x_i}^T\right) \preceq I$. Now, we can use the matrix Bernstein inequality for bounding $\left\|\frac{1}{m}\sum_{i=1}^m\widetilde{S_i} - \mathbb{E}\widetilde{S_i}\right\|_2$:

$$\begin{aligned}
&\mathbb{P}\left(\left\|\frac{1}{m}\sum_{i=1}^{m}(\widetilde{S_i} - \mathbb{E}\widetilde{S_i})\right\|_2 \geq t\right) \leq 2p\exp\left(\frac{-mt^2}{\sigma + Rt/3}\right) \\
&\leq 2p\exp\left(\frac{-mt^2}{C_5 pr^2\log^3(pm)\|M\|_2^2 + C_4 pr\log^2(mp)\|M\|_2 t/3}\right) \\
&\overset{a_1}{\leq} 2p\exp\left(\frac{-mt^2}{C_7 pr^2\log^3(pm)\|M\|_2^2}\right),
\end{aligned} \tag{23}$$

where a_1 holds by choosing constant C_7 to be sufficiently large. Now choose $t \geq \|M\|_2\sqrt{C_8\frac{pr^2\log^3(pm)}{m}\log(\frac{p}{\xi_2'})}$. Thus with probability at least $1-\xi_2'$, we have, $\left\|\frac{1}{m}\sum_{i=1}^m(\widetilde{S_i} - \mathbb{E}\widetilde{S_i})\right\|_2 \leq \sqrt{C_8\frac{pr^2\log^3(pm)}{m}\log(\frac{p}{\xi_2'})}\|M\|_2$, This bound shows that by taking $m = \mathcal{O}(\frac{1}{\theta^2}pr^2\log^3 p\log(\frac{p}{\xi_2'}))$ for some $\theta > 0$, we can bound the LHS of the above inequality. Actually, this choice of m determines the sample complexity of Alg. 1. Recall that $\widetilde{S_i}$ includes the truncated random variables, i.e, $\widetilde{S_i} = \widetilde{x_i}\widetilde{x_i}^T M\widetilde{x_i}\widetilde{x_i}^T$. Also, $\mathbb{P}\left(x_i^{(j)} = \widetilde{x_i}^{(j)}\right) \geq 1 - \frac{1}{(mp)^{C_2}} \geq 1 - \frac{1}{(p)^{C_9}}$. Hence, we need to extend our result to the original x_i. By definition of $\widetilde{x_i}$ in (22) and choosing constant C_9 sufficiently large, we have $\mathbb{P}\left(\|S_i - \widetilde{S_i}\|_2 = 0\right) = \mathbb{P}\left(\|x_i x_i^T - \widetilde{x_i}\widetilde{x_i}^T\|_2 = 0\right) \geq 1 - \frac{1}{(p)^{C_{10}}}$. Here, we have used the union bound over p^2

variables. Since we have m random matrices S_i, we need to take another union bound. As a result, with probability $1-\xi_2$ where $\xi_2 = \frac{1}{(p)^{C_{11}}}$, we have, $\left\|\frac{1}{m}\sum_{i=1}^m (S_i - \mathbb{E}S_i)\right\|_2 \leq \sqrt{C_8 \frac{pr^2\log^3 p}{m}\log(\frac{p}{\xi_2})}\|M\|_2$. □

Proof of Theorem 7. Let L_t be the estimation of the algorithm in iteration t, and L_* denotes the ground truth matrix. Then for constants $C, C'C'' > 0$,

$$\begin{aligned}
&\big\|L_t - L_* - \frac{1}{2m}\mathcal{A}^*\mathcal{A}(L_t - L_*) + (\frac{1}{2m}\mathbf{1}^T\mathcal{A}(L_t) \\
&\qquad - \frac{1}{2m}\mathbf{1}^T\mathcal{A}(L_*))I\big\|_2 \\
&\overset{a_1}{\leq} \big\|\frac{1}{2m}\mathcal{A}^*\mathcal{A}(L_t - L_*) - (L_t - L_*) - \frac{1}{2}Tr(L_t - L_*) \\
&\qquad - \frac{1}{2m}\mathbf{1}^T\mathcal{A}(L_t - L_*)I + \frac{1}{2}Tr(L_t - L_*)I\big\|_2 \\
&\overset{a_2}{\leq} \big\|\frac{1}{2m}\mathcal{A}^*\mathcal{A}(L_t - L_*) - (L_t - L_*) - \frac{1}{2}Tr(L_t - L_*)I\big\|_2 \\
&\qquad + \big\|\frac{1}{2m}\mathbf{1}^T\mathcal{A}(L_t - L_*)I - \frac{1}{2}Tr(L_t - L_*)I\big\|_2 \\
&\overset{a_3}{\leq} \left(C'\sqrt{\frac{pr^2\log^3 p}{m}\log(\frac{p}{\xi_2})}\right)\big\|L_t - L_*\big\|_2 \\
&\qquad + C\sqrt{\frac{1}{m}\log(\frac{p}{\xi_1})}\big\|L_t - L_*\big\|_2 \\
&\overset{a_4}{\leq} C''\delta\big\|L_t - L_*\big\|_2 = \rho\big\|L_t - L_*\big\|_2, \qquad (24)
\end{aligned}$$

where a_1 is followed by adding and subtracting of $Tr(L_t - L_*)I$, inequality a_2 follows from triangle inequality, a_3 holds with probability $1-\xi_1-\xi_2 = 1-\xi$ by invoking Lemma 13, and Lemma 14 (by fixed matrix $L_t - L_*$ with rank $2r$), and finally a_4 is followed by choosing $m = \mathcal{O}\left(\frac{1}{\delta^2}pr^2\log^3 p\log(\frac{p}{\xi})\right)$ for some $\delta > 0$. Choose δ sufficiently small to conclude. □

Corollary 15. *From Theorem 7 we have:*

1) *Let U be the bases for the column space of fixed matrices L_1 and L_2 such that rank$(L_i) \leq r$ for $i = 1, 2$ and $\mathcal{P}_U$ is the projection onto it. Also consider all the assumptions of Theorem 7. Then $\big\|L_1 - L_2 - \frac{1}{2m}\mathcal{P}_U\mathcal{A}^*\mathcal{A}(L_1 - L_2) + \mathcal{P}_U\frac{1}{2}Tr(L_1 - \bar{y})I\big\|_2 \leq \rho\|L_1 - L_2\|_2$.*
2) *$\big\|\frac{1}{2m}\mathcal{A}^*\mathcal{A}(L_1 - L_2) - \frac{1}{2}Tr(L_1 - \bar{y})I\big\|_2 \geq (1-\rho)\|L_1 - L_2\|_2$.*

Proof of Theorem 6. The proof is very similar to the proof of Lemma 14 and we only give a brief sketch. The idea is again to use the matrix Bernstein inequality; to do this, we have to use the truncation trick both on the random vector x_i and the noise vector e. We introduce $\widetilde{x}_i$ as (22) and similarly $\widetilde{e}$ as ($j = 1, \ldots, m$): $\widetilde{e}^{(j)} = e^{(j)}$ if $|e^{(j)}| \leq c_1'\sqrt{\log m}$; otherwise, $\widetilde{e}^{(j)} = 0$.

In the following expressions, $c_l' > 0$ for $l = 1, 4$ are absolute constants and $c_l' > 0$ for $l = 2, 3, 5, 6, 7$ are some constants which depend on τ. Let $W_i = \widetilde{e}_i\widetilde{x}_i\widetilde{x}_i^T$ for $i = 1, \ldots, m$ and $W = \widetilde{e_r}\widetilde{x}\widetilde{x}^T$ be a independent copy of W_i's (i.e, $\widetilde{e_r}$ and $\widetilde{x}$ are independent copies of e_i and x_i, respectively). Hence, $\mathbb{E}\frac{\mathcal{A}^*e}{m} = \frac{1}{m}\sum_{i=1}^m \mathbb{E}\widetilde{e}_i\widetilde{x}_i\widetilde{x}_i^T = \mathbb{E}S_i = 0$ and $\mathbb{P}(\widetilde{e}_i = e_i) \geq 1 - \frac{1}{m^{c_2'}}$ by assumptions on e. Now, parameters R and σ in the matrix Bernstein inequality can be calculated as $\sigma = \|\mathbb{E}WW^T\|_2 = \left\|\mathbb{E}\widetilde{e_r}^2\mathbb{E}(\|\widetilde{x}\|_2^2\widetilde{x}\widetilde{x}^T)\right\|_2 \leq c_3'p\log(m)\log(mp)$, and $R = \|\widetilde{e_r}\widetilde{x}\widetilde{x}^T\|_2 \leq c_4'p\sqrt{\log m}\log(mp)$. As a result, for all $t_3 \geq 0$, we have $\mathbb{P}\left(\left\|\frac{1}{m}\sum_{i=1}^m W_i\right\|_2 \geq t_3\right) \leq 2p\exp\left(\frac{mt_3^2}{\sigma + Rt_3/3}\right) \leq 2p\exp\left(\frac{mt_3^2}{c_5'p\log(m)\log(mp)}\right)$, where the last inequality holds by sufficiently large c_5'. Now, similar to Lemma 14 by choosing $t_3 \geq \sqrt{c_6'\frac{p\log^2 p}{m}\log(\frac{p}{\xi_3})}$ and the union bound, we obtain $\left\|\frac{1}{m}\mathcal{A}^*e\right\|_2 \leq \sqrt{c_6'\frac{p\log^2 p}{m}\log(\frac{p}{\xi_3})}$. with probability at least $1-\xi_3$. On the other hand, since e_i's are subgaussian random variables, by simple application of the Hoeffding inequality [36], we have, $|\frac{1}{m}\mathbf{1}^Te| \leq \sqrt{\frac{c_7'}{m}\log(\frac{1}{\xi_4})}$ with probability at least $1-\xi_4$: Combining the above results together and letting $\gamma = \xi_3 + \xi_4$, we obtained the claim bound in the theorem. □

Proof of Theorem 11. Let V^t, V^{t+1}, and V^* denote the bases for the column space of L_t, L_{t+1}, and L_*, respectively. Assume $\nu = 1 + \frac{2}{\sqrt{1-\varepsilon}\sqrt{\kappa-1}}$. Also, let $V^t \cup V^{t+1} \cup V^* \subseteq \Omega_t := \Omega$. Hence, Ω_t is the set of matrices with $rank \leq 2\kappa r + r^*$. Define $b = L_t - \eta\mathcal{P}_\Omega\partial F(L_t)$, $\alpha = \alpha_{2\kappa r + R^*}$, and $\beta = \beta_{2\kappa r + r^*}$. Thus:

$$\begin{aligned}
\|L_{t+1} - L_*\|_F^2 &\overset{a_1}{\leq} \nu\|b - L_*\|_F^2 = \nu\|L_t - L_* - \eta_t\mathcal{P}_\Omega\partial F(L_t)\|_F^2 \\
&= \nu\|L_t - L_*\|_F^2 - 2\eta_t\nu\langle L_t - L_*, \mathcal{P}_\Omega\frac{1}{m}\mathcal{B}^*\mathrm{sgn}(\mathcal{B}(L_t) - y)\rangle \\
&\qquad + \nu\eta_t^2\|\mathcal{P}_\Omega\partial F(L_t)\|_F^2 \\
&= \nu\|L_t - L_*\|_F^2 - 2\frac{\eta_t\nu}{m}\langle\mathcal{B}(L_t - L_*), \mathrm{sgn}(\mathcal{B}(L_t - L_*))\rangle \\
&\qquad + \nu\eta_t^2\|\mathcal{P}_\Omega\partial F(L_t)\|_F^2 \\
&= \nu\|L_t - L_*\|_F^2 - 2\frac{\eta_t\nu}{m}\|\mathcal{B}(L_t - L_*)\|_1 + \nu\eta_t^2\|\mathcal{P}_\Omega\partial F(L_t)\|_F^2,
\end{aligned}$$

where a_1 holds by applying lemma 10, and due to the fact that L_{t+1} is the best low-rank approximation to b, it also happens to be the best low-rank approximation to b. Now we can bound the third term, $\|\mathcal{P}_\Omega\partial F(L_t)\|_F^2$ as follows:

$$\begin{aligned}
\|\mathcal{P}_\Omega\partial F(L_t)\|_F^2 &= \frac{1}{m}\|\mathcal{P}_\Omega\mathcal{B}^*\mathrm{sgn}(\mathcal{B}(L_t) - y)\|_F^2 \\
&= \frac{1}{m}\langle\mathcal{P}_\Omega\mathcal{B}^*\mathrm{sgn}(\mathcal{B}(L_t) - y), \mathcal{P}_\Omega\mathcal{B}^*\mathrm{sgn}(\mathcal{B}(L_t) - y)\rangle \\
&= \frac{1}{m}\langle\mathrm{sgn}(\mathcal{B}(L_t) - y), \mathcal{B}\mathcal{P}_\Omega\mathcal{B}^*\mathrm{sgn}(\mathcal{B}(L_t) - y)\rangle \\
&\overset{a_1}{\leq} \frac{1}{m}\|\mathcal{B}\mathcal{P}_\Omega\mathcal{B}^*\mathrm{sgn}(\mathcal{B}(L_t) - y)\|_1 \overset{a_2}{\leq} \beta\|\mathcal{P}_\Omega\partial F(L_t)\|_F,
\end{aligned}$$

where a_1 holds by Hölder's inequality, and a_2 is due to applying RIP(ℓ_1, ℓ_2) property. Hence, we obtain, $\|\mathcal{P}_\Omega\partial F(L_t)\|_F \leq \beta$. Now we have the error bound as $\|L_{t+1} - L_*\|_F^2 \leq \nu\|L_t - L_*\|_F^2 - 2\frac{\eta_t\nu}{m}\|\mathcal{B}(L_t - L_*)\|_1 + \nu\eta^2\beta^2$, Now let $\eta_t = \frac{\|\mathcal{B}(L_t - L_*)\|_1}{\beta^2}$. By using the RIP($\ell_1, \ell_2$) property, we have

$$\|L_{t+1} - L_*\|_F^2 \leq \nu(1 - \frac{\alpha^2}{\beta^2})\|L_t - L_*\|_F^2. \qquad (25)$$

In order to have linear convergence, we need to have $\sqrt{\nu(1 - \frac{\alpha^2}{\beta^2})} < 1$. If we simplify this condition together with the condition on κ, stated in Lemma 10, we obtain: $\kappa > 1 + \max\left\{\frac{4\left((\frac{\alpha^2}{\beta^2}) - 1\right)^2}{1-\varepsilon}, \frac{1}{1-\varepsilon}\right\}$. This completes the proof. □