Simple Alternating Minimization Provably Solves Complete Dictionary Learning *

Geyu Liang
Industrial and Operations Engineering
University of Michigan
lianggy@umich.edu

Gavin Zhang
Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
jialun2@illinois.edu

Salar Fattahi
Industrial and Operations Engineering
University of Michigan
fattahi@umich.edu

Richard Y. Zhang
Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
ryz@illinois.edu

October 23, 2022

Abstract

This paper focuses on *complete dictionary learning problem*, where the goal is to reparametrize a set of given signals as linear combinations of atoms from a learned dictionary. There are two main challenges faced by theoretical and practical studies of dictionary learning: the lack of theoretical guarantees for practically-used heuristic algorithms, and their poor scalability when dealing with huge-scale datasets. Towards addressing these issues, we show that when the dictionary to be learned is orthogonal, that an alternating minimization method directly applied to the nonconvex and discrete formulation of the problem exactly recovers the ground truth. For the huge-scale, potentially on-line setting, we propose a minibatch version of our algorithm, which can provably learn a complete dictionary from a huge-scale dataset with minimal sample complexity, linear sparsity level, and linear convergence rate, thereby negating the need for any convex relaxation for the problem. Our numerical experiments showcase the superiority of our method compared with the existing techniques when applied to tasks on real data.

1 Introduction

We consider the following optimization problem:

$$\min_{\boldsymbol{X},\boldsymbol{D}} \|\boldsymbol{Y}-\boldsymbol{D}\boldsymbol{X}\|_F^2+\zeta\|\boldsymbol{X}\|_0, \tag{DL}$$

where $\boldsymbol{Y}\in\mathbb{R}^{n\times p}$ commonly denotes a data matrix whose columns are observed signals. Given $\boldsymbol{Y}$, our goal is to find a dictionary $\boldsymbol{D}\in\mathbb{R}^{n\times k}$ and the corresponding code $\boldsymbol{X}\in\mathbb{R}^{k\times p}$ such that: (1) each signal in $\boldsymbol{Y}$ is approximately represented as a linear combination of columns (also called atoms) of

*Corresponding author: Salar Fattahi, fattahi@umich.edu. Financial support for this work was provided by NSF Award DMS-2152776, ONR Award N00014-22-1-2127, NSF CAREER Award ECCS-2047462, and in part by the C3.ai Digital Transformation Institute.

$\boldsymbol{D}$; and (2) such representation uses as few atoms as possible. In other words, we require columns of $\boldsymbol{X}$ to be sparse, which means a large portion of entries in $\boldsymbol{X}$ should be zero. Here, the pseudo-norm $\|\cdot\|_0$ counts the number of non-zero entries in $\boldsymbol{X}$ and is being used as a regularizer to promote sparsity.

Problem DL is widely known as the *dictionary learning* problem. The optimal dictionary $\boldsymbol{D}^*$ computed from DL gives an optimally-sparse representation of the data (Donoho and Elad (2003)), and its columns have a natural interpretation as a set of feature vectors that "best explain the data." As such, Problem DL arises ubiquitously across numerous application domains, ranging from clustering and classification (Ramirez et al. (2010); Tošić and Frossard (2011)), to image, audio, and video processing (Mairal et al. (2007); Liu et al. (2018); Grosse et al. (2012)), to facial recognition (Xu et al. (2017)), to medical imaging (Zhao et al. (2021)), and to many others.

Perhaps the most widely-used heuristic for solving DL is based on alternating minimization: to solve for the sparse code by fixing the dictionary, and then use the resulting code to update the dictionary. Examples of this method are the Method of Optimal Direction (MOD, Engan et al. (1999)) and KSVD (Aharon et al. (2006)). However, despite their simplicity, the methods based on alternating minimization are faced with two major challenges:

Lack of theoretical guarantees: Due to the nonconvexity of DL stemming from the nonlinear term $\boldsymbol{DX}$ and discrete regularizer $\|\boldsymbol{X}\|_0$, methods like MOD and KSVD are poorly understood on the theoretical side. As a result, their recovered solution may be arbitrarily far from the sparsest representation of the signal, which is a serious concern in applications where efficient and provable sparse representation of the data is crucial.

Limited scalability and incompatibility with on-line learning: Existing methods become intractable for modern "big data" applications, such as high-resolution image processing (Ayas and Ekinci (2020)) and video processing (Haq et al. (2020)), whose datasets are huge-scale and often dynamically generated, and for which processing must be performed essentially in real-time. Consider the task of image processing as an example. In order to learn a dictionary from more than 10000 images, the existing state-of-the-art (namely, the KSVD method) has to first reduce the sample size to a few hundreds through down-sampling and then divide the images into block patches of 8×8 pixels (Aharon et al. (2006)). Moreover, when newly-generated data arrive, there is no mechanism within the algorithm that allows for the dictionary to be incrementally improved, other than to run the algorithm from scratch again. As a result, the lack of scalability of the algorithm severely undermines the quality of dictionary and may lead to serious ramifications in applications like medical imaging.

1.1 Our Contribution

In this work, we assume the data matrix $\boldsymbol{Y}$ is generated as $\boldsymbol{Y} = \boldsymbol{D}^*\boldsymbol{X}^*$ where $\boldsymbol{D}^* \in \mathbb{R}^{n\times n}$ is a square dictionary whose columns are the true atoms and $\boldsymbol{X}^*$ is a sparse code matrix. We consider scenarios where the true dictionary $\boldsymbol{D}^*$ is assumed to be orthogonal or a general non-singular matrix. The former is commonly referred to as *orthonormal dictionary learning (ODL)*, while the later is known as *complete dictionary learning (CDL)* in the literature (Sun et al., 2015; Zhai et al., 2020). Specifically, towards solving DL in such settings, we make the following contributions:

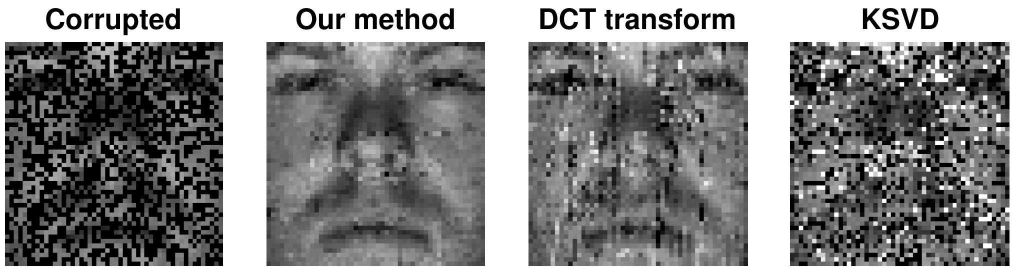

Figure 1: **A comparison of image denosing using our proposed method (Algorithm 3), KSVD and DCT**. We choose a random image and artificially corrupt 50% of the pixels. Reconstruction is done via orthogonal matching pursuit, using the learned dictionaries. Here we use 35 atoms, which corresponds to a sparsity level of 35%. The corrupted original image is plotted in the left hand side, and the three reconstructed images are shown on the right. We see the our method achieves a much better denoising result than both KSVD and DCT.

1. **Provable convergence**: We first consider the simplest and most intuitive alternating minimization scheme on DL, where we alternatingly fix one variable and optimize DL with respect to the other. We prove that for ODL, this simple algorithm provably recovers both $\boldsymbol{X}^*$ and $\boldsymbol{D}^*$. Then, we propose an extension to CDL through a carefully designed preconditioner based on Cholesky factorization. We show the provable scalability of our method in both huge-scale and online settings.

2. **Statistical guarantee**: Our methods enjoy promising statistical guarantees compared to the existing work. In particular, our algorithm converges to the ground truth under the assumption that a constant fraction of the entries in the sparse code matrix is non-zero, a sparsity level which is not reached by previous alternating minimization methods to the best of our knowledge.

3. **Practical performance**: We showcase the effectiveness and scalability of our algorithms through practical tasks of image denoising and image reconstruction on large dataset. We show that our method beats the state-of-the-art methods in both performance and efficiency, as illustrated in Figure 1 and Section 4.

1.2 Notation

In this paper, we use a or α to denote scalars, $\boldsymbol{a}$ to denote vectors and $\boldsymbol{A}$ do denote matrices. We use $\boldsymbol{A}_{(i,\cdot)}$ to denote the ith row of $\boldsymbol{A}$ and $\boldsymbol{A}_{(\cdot,j)}$ to denote the jth column of $\boldsymbol{A}$. We use $\|\boldsymbol{A}\|_2$ to denote the spectral norm of $\boldsymbol{A}$, $\|\boldsymbol{A}\|_F$ to denote Frobenius norm of $\boldsymbol{A}$, $\|\boldsymbol{A}\|_{2,\infty}$ to denote the maximum column norm of $\boldsymbol{A}$, $\|\boldsymbol{A}\|_0$ to denote the total number of non-zero entries in $\boldsymbol{A}$, and $\|\boldsymbol{A}\|_1$ to denote the entry-wise ℓ_1 norm of $\boldsymbol{A}$. We use $\boldsymbol{I}_d$ to denote the $d \times d$ identity matrix, and $\mathbb{O}(n)$ to denote the orthogonal group in dimension n. We use $\operatorname{supp}(\boldsymbol{A})$ to denote the set of indices of non-zero entries of

$\boldsymbol{A}$. We define the hard-thresholding operator $\mathrm{HT}_\zeta(\cdot)$ at level ζ as:

$$(\mathrm{HT}_\zeta(\boldsymbol{A}))_{(i,j)} = \begin{cases} \boldsymbol{A}_{(i,j)} & \text{if} \quad |\boldsymbol{A}_{(i,j)}| \geq \zeta \\ 0 & \text{if} \quad |\boldsymbol{A}_{(i,j)}| < \zeta \end{cases}. \tag{1}$$

We use $\sigma_i(\boldsymbol{A})$ to denote the ith largest singular value of $\boldsymbol{A}$ and $\kappa(\boldsymbol{A})$ to denote the condition number of $\boldsymbol{A}$. We define the operator $\mathrm{Polar}(\boldsymbol{A}) = \boldsymbol{U}_{\boldsymbol{A}}\boldsymbol{V}_{\boldsymbol{A}}^\top$, where $\boldsymbol{U}_{\boldsymbol{A}}\boldsymbol{\Sigma}_{\boldsymbol{A}}\boldsymbol{V}_{\boldsymbol{A}}^\top$ is the Singular Value Decomposition (SVD) of $\boldsymbol{A}$. We also define the operator $\mathcal{L}(\boldsymbol{A}) = \boldsymbol{L}_{\boldsymbol{A}}$, where $\boldsymbol{L}_{\boldsymbol{A}}\boldsymbol{L}_{\boldsymbol{A}}^\top$ is the Cholesky factorization of a positive semidefinite matrix $\boldsymbol{A}$. We use $f(x) = O(g(x))$ when $f(x) \leq Cg(x)$ for all large enough x and some universal constant C. We use $f(x) = \Omega(g(x))$ if $g(x) = O(f(x))$. To streamline the presentation, we say an event happens *with high probability* if it occurs with probability of at least $1 - n^{\Omega(1)}$ with respect to all the randomness in the problem, provided that both the sample size and the number of iterations of the algorithm are upper bounded by some polynomial function of n.

1.3 Organization

The paper is organized as follows. In Section 2, we provide a survey of related research work on sparse dictionary learning problem. Then, we introduce our main results in Section 3. We conclude our paper with numerical experiments in Section 4. All the detailed proofs are deferred to the appendix.

2 Related Work

Since the seminal work of Olshausen and Field (1996), there have been many exciting and inspirational research outcomes in the study of dictionary learning. Here, we provide a brief review on results that are most related to this paper.

The ground truth as the globally-optimal solution: As the first question, we must first verify whether the ground truth $(\boldsymbol{D}^*, \boldsymbol{X}^*)$ is indeed the global minimizer of DL for the generative model $\boldsymbol{Y} = \boldsymbol{D}^*\boldsymbol{X}^*$. Of course, if this were not the case, then solving DL—even to global optimality—may not be enough to actually recover the ground truth $(\boldsymbol{D}^*, \boldsymbol{X}^*)$. A series of work by Gribonval and Schnass (2010); Geng and Wright (2014); Jenatton et al. (2012); Schnass (2015) studied the local optimality of the ground truth for DL by replacing ℓ_0 norm with ℓ_1 norm. Spielman et al. (2012) elegantly showed that the ground truth is the unique global minimizer to DL when $\zeta \to 0$. Specifically, when $p = \Omega(n \log n)$, they showed that $(\boldsymbol{X}^*, \boldsymbol{D}^*)$ is the global minimizer to $\min_{\boldsymbol{X},\boldsymbol{D}} \|\boldsymbol{X}\|_0$ subject to the constraint $\boldsymbol{Y} = \boldsymbol{D}\boldsymbol{X}$. The question remains to be answered is how to find a provable algorithm to recover $\boldsymbol{X}^*$ and $\boldsymbol{D}^*$.

Provable guarantee for alternating minimization: The empirical success achieved by methods like MOD and KSVD has encouraged the emergence of many alternating minimization algorithms (Lee et al. (2006); Mairal et al. (2009); Bao et al. (2013, 2014)). However, theoretical gaps remain in our understanding of why such algorithms can work so well. Currently, the best guarantees that have been obtained for the aforementioned algorithms are limited to the convergence to a critical point. Towards providing a provable guarantee for alternating minimization, Agarwal et al. (2016)

give a theoretical analysis for a specific type of alternating minimization method by replacing the ℓ_0 pseudo-norm with its convex surrogate ℓ_1 norm. Their proposed algorithm solves a LASSO problem in each iteration and relies on a restricted isometry property (RIP) assumption. Arora et al. (2015) considers a general framework where sparse coding can be solved by hard thresholding with a predetermined threshold and dictionary is updated by a gradient descent step with the sparse code fixed. However, their proof breaks down when the sparsity level exceeds $O(\sqrt{n})$. Ravishankar et al. (2020) propose an alternating scheme based on the assumption that each sparse code has equal non-zero entries. Despite their strong provable guarantees, none of these papers provide any experiment on real dataset.

Other method with provable guarantee: Barak et al. (2015) adopt and analyze the Sum of Squares semidefinite programming hierarchy to solve DL. Recent years have also witnessed the trend of analyzing the dictionary learning problem via the Riemann manifold perspective (Sun et al. (2015); Qu et al. (2014); Zhai et al. (2020)). However, the applicability of such methods to real data has remained elusive since little empirical evidence is provided in their work. Inspired by recent results on the benign landscape of matrix factorization problems (Ge et al., 2016, 2017; Fattahi and Sojoudi, 2020), Sun et al. (2016) have shown that a smoothed variant of DL is devoid of spurious local solutions.

3 Our Method

As fundamental as DL is to data analysis and signal processing, it is dauntingly non-convex. There are two sources of non-convexity in the formulation of DL and they are dealt with in different ways on the algorithmic side: the first difficulty is the bi-linearity induced by the term $\boldsymbol{D}\boldsymbol{X}$, which inspires the strategy of alternating minimization. The second difficulty originates from the pseudo-norm $\|\cdot\|_0$, which is non-convex and discrete. To circumvent this issue, the common approach is to replace $\|\cdot\|_0$ with its convex surrogate $\|\cdot\|_1$ (Mairal et al. (2009)Agarwal et al. (2016)), which in turn leads to inferior statistical guarantees (Fan and Li (2001)Zhang (2010)). Our main result is to show that such relaxation is not needed, since a tailored alternating minimization can provably recover the true solution of DL at scales that were not possible before. We first introduce a simple algorithm for the orthogonal dictionary learning. Then, we extend our algorithm to the more general case of complete dictionary learning and its online variant, where the data is processed dynamically over time.

3.1 Full-batch Orthogonal Dictionary Learning

We first assume the dictionary to be an orthogonal matrix $\boldsymbol{D}^* \in \mathbb{O}(n)$, in which case the problem is termed as *Orthogonal Dictionary Learning (ODL)*:

$$\min_{\boldsymbol{D}\in\mathbb{O}(n),\boldsymbol{X}} \|\boldsymbol{Y}-\boldsymbol{D}\boldsymbol{X}\|_F^2+\zeta\|\boldsymbol{X}\|_0. \tag{ODL}$$

Towards solving ODL, we consider the following alternating minimization scheme:

- **Step 1**: Fix $\boldsymbol{D}^{(t)}$, update $\boldsymbol{X}^{(t)} = \arg\min_{\boldsymbol{X}\in\mathbb{R}^{n\times p}} \|\boldsymbol{Y}-\boldsymbol{D}^{(t)}\boldsymbol{X}\|_F^2+\zeta\|\boldsymbol{X}\|_0$.
- **Step 2**: Fix $\boldsymbol{X}^{(t)}$, update $\boldsymbol{D}^{(t+1)} = \arg\min_{\boldsymbol{D}\in\mathbb{O}(n)} \|\boldsymbol{Y}-\boldsymbol{D}\boldsymbol{X}^{(t)}\|_F^2+\zeta\|\boldsymbol{X}^{(t)}\|_0$.

It is easy to see that both steps of the algorithm have closed-form solutions that can be efficiently calculated. In particular, **Step 1** is equivalent to finding the proximal operator of ℓ_0 norm at the reference point $\boldsymbol{D}^{(t)\top}\boldsymbol{Y}$, which can be obtained by hard-thresholding the entries of $\boldsymbol{D}^{(t)\top}\boldsymbol{Y}$ at level ζ. **Step 2** reduces to the famous Procrustes problem, for which the optimal solution is obtained via the polar decomposition $\boldsymbol{D}^{(t+1)} = \text{Polar}\left(\boldsymbol{Y}\boldsymbol{X}^{(t)\top}\right)$. This leads to the following alternating minimization algorithm:

Algorithm 1 Alternating minimization for ODL

```
Input: Y, D^(0), ζ
for t = 0, 1, ..., T do
    Set X^(t) = HT_ζ (D^(t)⊤ Y)
    Set D^(t+1) = Polar (Y X^(t)⊤)
end for
return D^(T), X^(T)
```

We note that Algorithm 1 has been studied before. Bao et al. (2013) report their empirical observation that Algorithm 1 achieves considerable efficiency in image restoration. In Ravishankar et al. (2020), authors provide a theoretical analysis for a variant of Algorithm 1 based on sorted thresholding. One may even argue that Algorithm 1 is similar to the Method of Optimal Directions (MOD). However, the existing theoretical guarantees for Algorithm 1 are very restrictive.

To bridge this knowledge gap, we first introduce our assumption on the sparse matrix $\boldsymbol{X}$.

Assumption 1 (Model for Sparse Code Matrix)**.** *The sparsity pattern of ground truth $\boldsymbol{X}^*$ is drawn from an independent Bernoulli distribution with parameter $\theta = (0, 1]$. In particular, for each entry $1 \le i \le n$, $1 \le j \le p$,*

$$\mathbb{1}_{\boldsymbol{X}^*_{ij} \neq 0} = B_{ij} \quad \textit{where} \quad B_{ij} \overset{i.i.d.}{\sim} \mathcal{B}(\theta). \tag{2}$$

Moreover, the non-zero values of $\boldsymbol{X}^*$ are drawn from an i.i.d zero-mean and finite-variance sub-Gaussian random variables. We also assume the magnitudes of non-zero entries of $\boldsymbol{X}^*$ are lower bounded by some constant Γ. More specifically, for every $(i, j) \in \text{supp}(\boldsymbol{X}^*)$, we have

$$|\boldsymbol{X}^*_{ij}| \ge \Gamma, \quad \mathbb{E}(\boldsymbol{X}^*_{ij}) = 0, \quad \textit{and} \quad \mathbb{E}(\boldsymbol{X}^{*2}_{ij}) = \sigma^2. \tag{3}$$

Now, we are ready to show the convergence of Algorithm 1:

Theorem 1. *Suppose that Assumption 1 holds with sparsity level $\theta \le \frac{1}{144}$, and the initial dictionary $\boldsymbol{D}^{(0)}$ satisfies $\|\boldsymbol{D}^{(0)} - \boldsymbol{D}^*\|_F \le O(1/\log n)$. Moreover, suppose that the sample size satisfies $p \ge \Omega\left(n/\theta^2\right)$. Then, with high probability, the iterations of Algorithm 1 with $\zeta = \Gamma/2$ satisfy:*

$$\|\boldsymbol{D}^{(t)} - \boldsymbol{D}^*\|_F \le (1/2)^t \|\boldsymbol{D}^{(0)} - \boldsymbol{D}^*\|_F \tag{4}$$

$$\|\boldsymbol{X}^{(t)} - \boldsymbol{X}^*\|_F \le (1/2)^t \|\boldsymbol{X}^{(0)} - \boldsymbol{X}^*\|_F \tag{5}$$

Theorem 1 improves upon the existing results on two fronts:

Linear sparsity level: We allow a constant fraction of entries in $\boldsymbol{X}^*$ to be non-zero, thereby improving upon the best known sparsity level of $O(\sqrt{n}/\log n)$ for alternating minimization (Arora et al., 2015).

Linear sample complexity: In order to recover $\boldsymbol{D}^*$ exactly, we only need to observe $O(n)$ many samples, which is even one $\log(n)$ factor smaller than the sample complexity required for the uniqueness of the solution when $\xi \to 0$ (Spielman et al. (2012)). Note that this sample complexity is optimal (modulo constant factors), since it is impossible to recover the true dictionary with sublinear number of samples even if X^* is known.

Even though our result is built on a quite generous initialization requirement $\|\boldsymbol{D}^{(0)} - \boldsymbol{D}^*\|_F \leq O(1/\log n)$, we suspect that it can be potentially improved to $\|\boldsymbol{D}^{(0)} - \boldsymbol{D}^*\|_{2,\infty} \leq O(1/\log n)$, which is the best known convergence radius for alternating minimization (Arora et al., 2015).

Despite its desirable sample complexity and convergence rate, Algorithm 1 suffers from two fundamental limitations. First, its convergence is contingent upon the orthogonality of the true dictionary, which is not satisfied in a lot of applications. Second, it does not readily extend to huge-scale or online settings, where it is prohibitive or impossible to process all samples at once. To address these challenges, we extend our algorithm to complete (non-orthogonal) dictionary learning with mini-batch and online data in the next two subsections.

3.2 Mini-batch Complete Dictionary Learning

We move on to generalize Algorithm 1 to CDL with huge sample size. To distinguish from ODL, we denote the ground truth dictionary as $\boldsymbol{A}^*$ (i.e., $\boldsymbol{Y} = \boldsymbol{A}^*\boldsymbol{X}^*$). Towards dealing with large p, the strategy adopted here is to sub-sample from the columns of $\boldsymbol{Y}$. To address the non-orthogonality of the dictionary, we consider a preconditioner based on Cholesky factorization. In particular, we define a preconditioner $\boldsymbol{P}$ as

$$\boldsymbol{P} = \mathcal{L}\left(\left(\frac{1}{p\theta\sigma^2}\boldsymbol{Y}\boldsymbol{Y}^\top\right)^{-1}\right)^\top, \tag{6}$$

and obtain a new (preconditioned) data matrix $\tilde{\boldsymbol{Y}} = \boldsymbol{P}\boldsymbol{Y}$. To explain the intuition behind this choice of preconditioner, note that $\frac{1}{p\theta\sigma^2}\boldsymbol{Y}\boldsymbol{Y}^\top \approx \boldsymbol{A}^*\boldsymbol{A}^{*\top}$ for large enough p, which allows us to write $\tilde{\boldsymbol{Y}} \approx \mathcal{L}\left(\left(\boldsymbol{A}^*\boldsymbol{A}^{*\top}\right)^{-1}\right)^\top \boldsymbol{A}^*\boldsymbol{X}^*$. Upon defining $\boldsymbol{D}^* = \mathcal{L}\left(\left(\boldsymbol{A}^*\boldsymbol{A}^{*\top}\right)^{-1}\right)^\top \boldsymbol{A}^*$, one can check that $\boldsymbol{D}^* \in \mathbb{O}(n)$ and $\tilde{\boldsymbol{Y}} = \boldsymbol{D}^*\boldsymbol{X}^*$. Indeed, this is an instance of ODL and can be solved using Algorithm 1. Therefore, we propose the following algorithm for solving CDL:

Algorithm 2 Alternating minimization for mini-batch CDL

- **Input:** $\boldsymbol{Y}$, $\boldsymbol{A}^{(0)}$, ζ
- Set $\boldsymbol{P} = \mathcal{L}\left(\left(\frac{1}{p\theta\sigma^2}\boldsymbol{Y}\boldsymbol{Y}^\top\right)^{-1}\right)^\top$.
- Set $\tilde{\boldsymbol{Y}} = \boldsymbol{P}\boldsymbol{Y}$.
- Set $\boldsymbol{D}^{(0)} = \mathcal{L}\left(\left(\boldsymbol{A}^{(0)}\boldsymbol{A}^{0\top}\right)^{-1}\right)^\top \boldsymbol{A}^{(0)}$.
- **for** $t = 0, 1, \ldots, T-1$ **do**
  - Sample $\tilde{p}$ many columns from $\tilde{\boldsymbol{Y}}$ to be $\tilde{\boldsymbol{Y}}^{(t)}$.
  - Set $\tilde{\boldsymbol{X}}^{(t)} = \mathrm{HT}_\zeta\left(\boldsymbol{D}^{(t)\top}\tilde{\boldsymbol{Y}}^{(t)}\right)$.
  - Set $\boldsymbol{D}^{(t+1)} = \mathrm{Polar}\left(\tilde{\boldsymbol{Y}}^{(t)}\tilde{\boldsymbol{X}}^{(t)\top}\right)$.
- **end for**
- **return** $\boldsymbol{P}^{-1}\boldsymbol{D}^{(T)}$ as an approximation to $\boldsymbol{A}^*$

Our next theorem characterizes the performance of Algorithm 2. To better present our result, we consider $\boldsymbol{A}^{(t)}$ to be the output of the algorithm if it stops at tth iteration. Without loss of generality, we assume that $\|\boldsymbol{A}^*\|_2 = 1$ and set $\hat{\kappa}$ to be the condition number of $\boldsymbol{A}^*$.

Theorem 2. *Suppose Assumption 1 holds with sparsity level $\theta \leq \frac{1}{5600}$, and the initial dictionary $\boldsymbol{A}^{(0)}$ satisfies $\|\boldsymbol{A}^{(0)} - \boldsymbol{A}^*\|_F \leq O(1/(\hat{\kappa}^6 \log n))$. Moreover, suppose that the sample size and batch size satisfy $p \geq \Omega(n^2 \log n \sigma^2 \hat{\kappa}^{12}/(\theta^2 \Gamma))$ and $\tilde{p} \geq \Omega\left(n/\theta^2\right)$. Then, with high probability and for some $0 < \rho < 1$, the iterations of Algorithm 2 with $\zeta = \Gamma/2$ satisfy:*

$$\left\|\boldsymbol{A}^{(t)} - \boldsymbol{A}^*\right\|_F \leq \rho^t \|\boldsymbol{A}^{(0)} - \boldsymbol{A}^*\|_F + O\left(\frac{n\hat{\kappa}^6}{\theta\sqrt{p}}\right). \tag{7}$$

Theorem 2 states that Algorithm 2 converges linearly to the true dictionary up to a statistical error of $O\left(\frac{n\hat{\kappa}^6}{\theta\sqrt{p}}\right)$. This statistical error is due to the deviation of the preconditioner from its expectation, which diminishes with p. A noticeable difference between our result and other results is that we do not impose any incoherency requirement or restricted isometry property on $\boldsymbol{A}^*$, which are prevalent in existing results, especially for algorithms that use ℓ_1 relaxation.

Finally, note that Algorithm 2 does not return the code matrix $\boldsymbol{X}^*$. This is due to the fact that our proposed algorithm may not see all the samples. More precisely, it will see $\tilde{p}T$ samples by iteration T, which can be significantly smaller than the full sample size p. Therefore, Algorithm 2 cannot recover the full code matrix. However, given an approximation $\boldsymbol{A}^{(t)}$ of $\boldsymbol{A}^*$, the sparse code corresponding to any sample $\boldsymbol{Y}^{(s)}$ can be readily recovered by computing $\boldsymbol{X}^{(s)} = \mathrm{HT}_\zeta\left(\left(\boldsymbol{P}\boldsymbol{A}^{(t)}\right)^\top \boldsymbol{P}\boldsymbol{Y}^{(s)}\right)$.

3.3 Online Dictionary Learning

Even though the computational efficiency of Algorithm 2 is largely improved compared to its full-batch counterpart, we still may face the following challenges in reality. First, the sample size p is so large that even a one-time calculation of the preconditioner $\boldsymbol{P}$ is unrealistic. Second, samples (columns of $\boldsymbol{Y}$) may arrive dynamically over time. Both situations call for an efficient method to update the preconditioner more efficiently in an online setting.

To update $\boldsymbol{P}$ with a new sample $\boldsymbol{y}$, a naive approach would be to re-compute it from scratch which would cost $O(n^3)$ operations. However, we show that $\boldsymbol{P}$ can be updated more efficiently in $O(n^2)$ operations by taking advantage of the more efficient rank-one updates on matrix inversion and Cholesky factorization. To this goal, we first use the Sherman-Morrison formula to update $\left(\boldsymbol{Y}\boldsymbol{Y}^\top\right)^{-1}$ as

$$\left(\boldsymbol{Y}\boldsymbol{Y}^\top + \boldsymbol{y}\boldsymbol{y}^\top\right)^{-1} = \left(\boldsymbol{Y}\boldsymbol{Y}^\top\right)^{-1} - \boldsymbol{v}\boldsymbol{v}^\top, \quad \text{where} \quad \boldsymbol{v} = \left(1 + \boldsymbol{y}^\top \left(\boldsymbol{Y}\boldsymbol{Y}^\top\right)^{-1} \boldsymbol{y}\right)^{-1/2} \left(\boldsymbol{Y}\boldsymbol{Y}^\top\right)^{-1} \boldsymbol{y},$$

which, given $\left(\boldsymbol{Y}\boldsymbol{Y}^\top\right)^{-1}$, can be obtained in $O(n^2)$ operations. Given the above rank-one update for the inverse, the Cholesky factor $\mathcal{L}\left(\left(\boldsymbol{Y}\boldsymbol{Y}^\top + \boldsymbol{y}\boldsymbol{y}^\top\right)^{-1}\right)$ can be obtained within $O(n^2)$ operations by performing triangular rank-one updates on $\mathcal{L}\left(\left(\boldsymbol{Y}\boldsymbol{Y}^\top\right)^{-1}\right)$ (Krause and Igel, 2015). We explain the precise implementation of these updates in Appendix D.

Inspired by the above update, we propose Algorithm 3 for solving online dictionary learning. We start the algorithm by initializing the inverse of the data matrix (denoted as $\boldsymbol{Z}_{\mathrm{inv}}^{(t)}$) and the preconditioner using p_1 samples. Moreover, we use p_2 samples for our initial data set. When a new sample arrives, we update the preconditioner $\boldsymbol{P}^{(t)}$ and $\boldsymbol{Z}_{\text{inv}}^{(t)}$ via the triangular rank-one update explained in Appendix D, and update the data set accordingly. Then, we perform the alternating minimization as presented before using the new data set and the new preconditioner.

Algorithm 3 Alternating minimization for online dictionary learning

1: **Input:** $\boldsymbol{A}_0^{(0)}$, ζ, p_1,p_2
2: Construct the data matrix $\boldsymbol{Y}$ with p_1 samples and set $\boldsymbol{Z}_{\text{inv}}^{(0)} = \left(\boldsymbol{Y}\boldsymbol{Y}^\top\right)^{-1}$ and $\boldsymbol{P}^{(0)} = \mathcal{L}\left(p_1\theta\sigma^2\boldsymbol{Z}_{\text{inv}}^{(0)}\right)^\top$.
3: Initialize $\boldsymbol{Y}^{(0)}$ with p_2 samples.
4: **for** $t = 0, 1, \ldots, T-1$ **do**
5: Get new vector $\boldsymbol{y}$.
6: Set $\boldsymbol{Y}^{(t)} = [\boldsymbol{Y}^{(t-1)} \quad \boldsymbol{y}]$.
7: Remove first column of $\boldsymbol{Y}^{(t)}$.
8: Update $\boldsymbol{P}^{(t)}$ and $\boldsymbol{Z}_{\text{inv}}^{(t)}$ with $\boldsymbol{P}^{(t-1)}$, $\boldsymbol{Z}_{\text{inv}}^{(t-1)}$, and new $\boldsymbol{y}$ using Algorithm 5 in Appendix D.
9: Set $\tilde{\boldsymbol{Y}}^{(t)} = \boldsymbol{P}^{(t)}\boldsymbol{Y}^{(t)}$
10: Set $\tilde{\boldsymbol{X}}^{(t)} = \text{HT}_\zeta\left(\boldsymbol{D}^{(t)\top}\tilde{\boldsymbol{Y}}^{(t)}\right)$.
11: Set $\boldsymbol{D}^{(t+1)} = \text{Polar}\left(\tilde{\boldsymbol{Y}}^{(t)}\tilde{\boldsymbol{X}}^{(t)\top}\right)$.
12: **end for**
13: **return** $\left(\boldsymbol{P}^{(T-1)}\right)^{-1}\boldsymbol{D}^{(T)}$ as an approximation to $\boldsymbol{A}^*$

Our next theorem establishes the convergence of Algorithm 3 for the online dictionary learning.

Theorem 3. *Suppose that Assumption 1 holds with the sparsity level* $\theta \leq \frac{1}{5600}$*, and the initial dictionary* $\boldsymbol{A}^{(0)}$ *satisfies* $\|\boldsymbol{A}^{(0)} - \boldsymbol{A}^*\|_F \leq O(1/(\hat{\kappa}^6 \log n))$*. Moreover, suppose that we have* $p_1 \geq \Omega(n^2 \log n\sigma^2\hat{\kappa}^{12}/(\theta^2\Gamma))$ *and* $p_2 \geq \Omega\left(n/\theta^2\right)$*. Then, with high probability and for some* $0 < \rho < 1$*, the iterations of Algorithm 3 with* $\zeta = \Gamma/2$ *satisfy:*

$$\left\|\boldsymbol{A}^{(t)} - \boldsymbol{A}^*\right\|_F \leq \rho^t\|\boldsymbol{A}^{(0)} - \boldsymbol{A}^*\|_F + O\left(\frac{n\hat{\kappa}^6}{\theta\sqrt{p_1 + t}}\right). \tag{8}$$

The convergence result of Algorithm 3 is similar to that of Algorithm 2, with a key difference that the statistical error term $O\left(\frac{n\hat{\kappa}^6}{\theta\sqrt{p_1+t}}\right)$ now decreases with t, which is due to the fact that the preconditioner is updated with a new sample at every iteration.

3.4 Discussion

3.4.1 Initialization

The theoretical success of Algorithms 1-3 requires, at least in theory, a good initialization with $O(1/\log n)$ distance to the ground truth. Such initialization can be provided by the initialization scheme introduced in Agarwal et al. (2016) and Arora et al. (2015), albeit with a slightly more restrictive conditions on the sparsity level and sample size. However, we note that these tailored initialization scheme are not easily implementable especially in the huge-scale regime.[1]. Instead, we

[1]We are not aware of any practical implementation of these tailored initialization schemes.

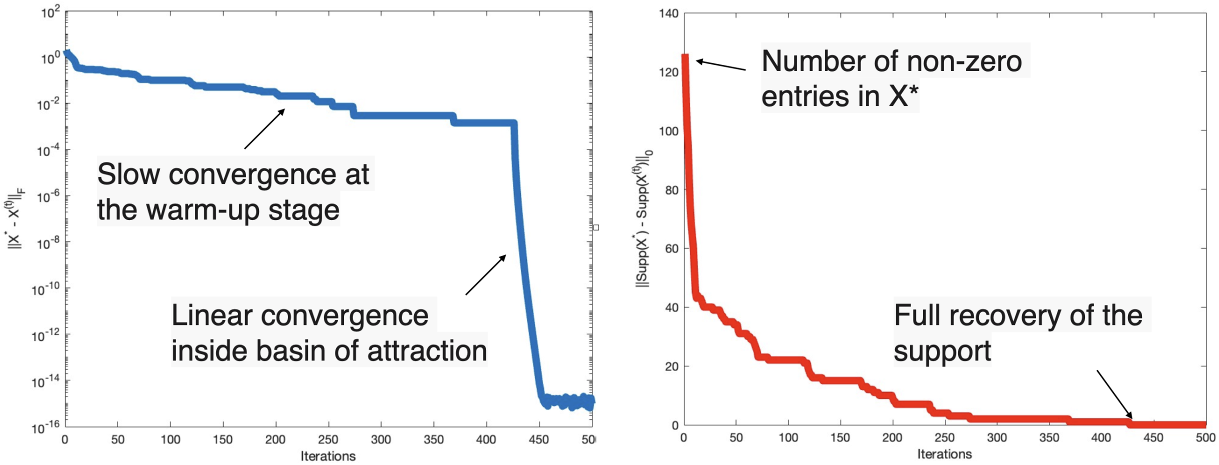

Figure 2: The plots above show a typical run of Algorithm 1 where $p = 100$, $n = 5$ and $\theta = 0.3$. The figure on the left shows the Frobenius norm of $\boldsymbol{X}^* - \boldsymbol{X}^{(t)}$ per iteration. The figure on the right shows the number of non-zero entries of $\text{Supp}(\boldsymbol{X}^*) - \text{Supp}(\boldsymbol{X}^{(t)})$. The number is 126 at the beginning, which is the total number of non-zero entries in $\boldsymbol{X}^*$, and 0 in the end, which indicates the full recovery of the support of $\boldsymbol{X}^*$.

observed that such tailored initialization schemes are not required in practice, as a simple warm-up algorithm with a diminishing threshold would yield a similar performance. In particular, consider the following warm-up algorithm for the initial dictionary for ODL (the warm-up algorithm for CDL is similar and omitted for brevity):

Algorithm 4 Warm-up stage for Algorithm 1

Input: $\boldsymbol{Y}$, ζ_0, $\beta \in (0, 1)$
Set $\boldsymbol{D}^{(0)} = \boldsymbol{I}_n$
for $t = 0, 1, \ldots, T_0$ **do**
  $\zeta_{t+1} = \beta \zeta_t$
  Set $\boldsymbol{X}^{(t)} = \text{HT}_{\zeta_t}\left(\boldsymbol{D}^{(t)\top}\boldsymbol{Y}\right)$
  Set $\boldsymbol{D}^{(t+1)} = \boldsymbol{I}_n$ if $\boldsymbol{X}^{(t)} = \boldsymbol{0}_n$, and $\boldsymbol{D}^{(t+1)} = \text{Polar}\left(\boldsymbol{Y}\boldsymbol{X}^{(t)\top}\right)$ if $\boldsymbol{X}^{(t)} \neq \boldsymbol{0}_n$.
end for
return $\boldsymbol{D}^{(T_0)}$

Notice that a large ζ_0 will force $\boldsymbol{X}^{(t)}$ to be an all-zero matrix and make $\boldsymbol{D}^{(t+1)}$ an identity matrix. As we shrink the threshold ζ_t in Algorithm 4, the matrix $\boldsymbol{X}^{(t)}$ eventually becomes nonzero, and the iterates $\boldsymbol{D}^{(0)}$ start to gradually move towards the basin of attraction of $\boldsymbol{D}^*$, from which the linear convergence of Algorithm 1 is guaranteed according to Theorem 1. We also note that in practice, the movement of iterates from $\boldsymbol{I}_d$ to the basin does not enjoy the linear convergence property. Figure 2 illustrates that our proposed warm-up algorithm eventually puts the initial dictionary close enough to the true dictionary, from which the algorithm converges linearly. The theoretical analysis of this warm-start algorithm is left as an enticing challenge for future research.

3.4.2 Further Speed-up for Online Dictionary Learning

Even though Algorithm 3 works in an online setting, its computational cost is bottlenecked by the matrix-matrix multiplication and the SVD step, leading to a per-iteration cost of $O(n^3)$. Here, we propose a possible way to further speed up the algorithm by considering $\tilde{\boldsymbol{Y}}^{(t)}\tilde{\boldsymbol{X}}^{(t)\top}$ as the sum of rank-one matrices $\tilde{\boldsymbol{y}}\tilde{\boldsymbol{x}}^\top$. More specifically, after receiving $\boldsymbol{y}$ at each iteration, we use $\boldsymbol{y}$ to calculate only one pair of $\tilde{\boldsymbol{y}}$ and $\tilde{\boldsymbol{x}}$ with the current preconditioner $\boldsymbol{P}^{(t)}$ and the current dictionary $\boldsymbol{D}^{(t)}$. Then we can perform rank-one update on Polar$(\cdot)$ which can be performed in $O(n^2)$ theoretically (Stange (2008)). The per-iteration cost of $O(n^2)$ must be optimal since it is the same cost of forming and storing an $n \times n$ dictionary. However, as delineated by Stange (2008), the proposed rank-one update of the polar decomposition cannot be efficiently implemented in practice despite its desirable computational cost in theory. Therefore, we leave the practical implementation of this technique to future research.

4 Numerical Experiments

In this section, we evaluate the performance of our algorithm in practice. In particular, we consider the task of learning a dictionary for the Yale Face Databse [2], which consists of 60000 greyscale images for individual faces. Experiments in this section are performed on a MacBook Pro 2021 with the Apple M1 Pro chip and a 16GB unified memory for a serial implementation in MATLAB 2022a. Our experiments demonstrate that our method significantly outperforms KSVD in terms of running time and image reconstruction quality.

For a fair comparison, we randomly sample 1000 grey scale images and compare the learned dictionary using Algorithm 3, KSVD, and the standard discrete fourier transform. For large sample sizes, previous methods like full-batch alternating minimization and KSVD have prohibitively expensive runtimes, while the speed and convergence of our mini-batch algorithm 3 is almost unaffected, since its per-iteration cost is independent of the sample size p. The only reason we keep the number of training data relatively small is that KSVD is extremely slow compared to Algorithm 3 and running KSVD with 60000 greyscale images will take multiple days in our current setup. In contrast, Algorithm 3 only takes a few minutes.

Using the learned dictionary, we compare the results of two common tasks: (1) image reconstruction and (2) image denoising using the learned dictionary. In both of these tasks, we can clearly see that the dictionary learned via Algorithm 3 achieves a better reconstruction using much less time.

Image Reconstruction A pair of the reconstruction results are shown in Figure 3. To generate this plot we randomly sample 1000 figures of individual faces of size 50×50. Instead of directly learning a dictionary for the dataset, we follow the procedure in Aharon et al. (2006) and divide each figure into 25 tiles of 10×10 patches. These patches are then reshaped into a 100×1 vector and stacked together into a matrix Y of size 100×25000. This corresponds to $n = 100$ and $p = 25000$ in our theoretical results. Unfortunately, while such a large p poses no issue for our proposed algorithm, it is prohibitive for KSVD. Therefore we sub-sample 5000 columns of Y, so that now Y is of size 100×5000. Our goal now is to learn a dictionary A of size 100×100 for this data matrix.

For a fair comparison, we select a termination time of 50 seconds for both algorithms. For reference, we also plot the image constructed using a discrete cosine transform (DCT), assuming the

[2] Yale Face Database. http://vision.ucsd.edu/content/yale-face-database

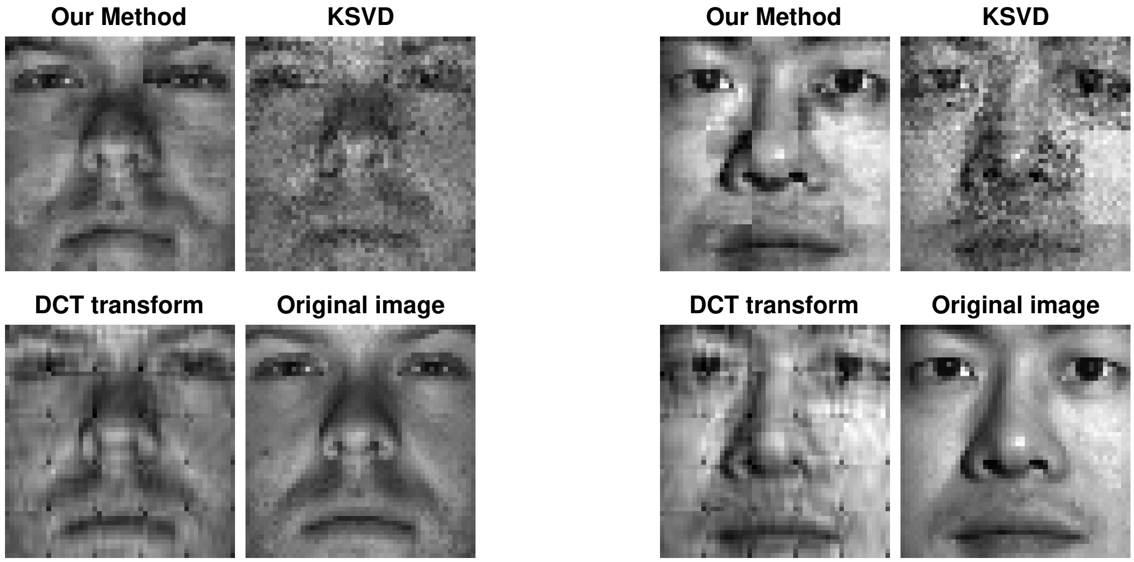

Figure 3: **A comparison of image reconstruction using our proposed method (Algorithm 3), KSVD and DCT**. Given a dataset of multiple greyscale images, we divided each image into patches and stack them into a large data matrix. A dictionary is learned using our minibatch alternating minimization algorithm, KSVD and DCT. For fair comparison, we cap the runtime of all algorithms at 50 seconds. Reconstruction results for two random images are shown in the upper and lower blocks. We use 35 atoms corresponding a sparsity level of 35%. We see that our method achieves a much better reconstruction than both KSVD and DCT.

same sparsity level. Here the implementation of KSVD is done using a standard sparse learning library in Matlab (Mairal et al., 2007) and the discrete cosine transform is done using the Matlab function `dct2`.

Similar to the setting in Aharon et al. (2006), we choose a sparsity level of approximately 35%, which simply corresponds to using 35 atoms for reconstruction. With a given dictionary, the reconstruction is done using a standard implementation of orthogonal matching pursuit (OMP) found in the SPAMs library in Matlab (Mairal et al., 2007). In Figure 3, we see that our proposed method achieves a better reconstruction than both KSVD and DCT.

Image Denoising In addition to image reconstruction, we also gauge the quality of our learned dictionary using a common task in image processing which is called *denoising*. Here, an image is corrupted with 50% missing pixels (in our implementation this corresponds to setting the greyscale value to 0). Our goal is to learn a dictionary and use it to denoise this image by filling in the missing pixels. The setting in which we learn the dictionary is exactly the same as before: we divided each image into patches and stack them into a large, wide matrix. The running time for both Algorithm 3 and KSVD are capped at 50 seconds.

Once a dictionary is learned, the reconstruction is done using orthogonal matching pursuit. Here the sparsity level is the same as in the reconstruction case, i.e., 35%. A comparison of denoised results using dictionaries learned with Algorithm 3 and KSVD are shown in Figure 1. For reference, we also include results using a dictionary learned via DCT. Here we also see that the dictionary learned by Algorithm 3 outperforms both KSVD and DCT, achieving a much better reconstruction of the original image despite the fact that half of the pixels are missing.

5 Conclusion

In this paper, we study dictionary learning problem, where the goal is to represent a given set of data samples as linear combinations of a few atoms from a learned dictionary. The existing algorithms for dictionary learning often lack scalability or provable guarantees. In this paper, we show that a simple alternating minimization algorithm provably solves both orthogonal and complete dictionary learning problems. Unlike other provably convergent algorithms for dictionary learning, our proposed method does not rely on any convex relaxation of the problem, and can be easily implemented in realistic scales. Through realistic case studies on image reconstruction and image denoising, we showcase the superiority of our proposed algorithm compared with the most commonly-used algorithms for dictionary learning.

References

Agarwal, A., Anandkumar, A., Jain, P., and Netrapalli, P. (2016). Learning sparsely used overcomplete dictionaries via alternating minimization. *SIAM Journal on Optimization*, 26(4):2775–2799.

Aharon, M., Elad, M., and Bruckstein, A. (2006). K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311–4322.

Arora, S., Ge, R., Ma, T., and Moitra, A. (2015). Simple, efficient, and neural algorithms for sparse coding. In *Conference on learning theory*, pages 113–149. PMLR.

Ayas, S. and Ekinci, M. (2020). Single image super resolution using dictionary learning and sparse coding with multi-scale and multi-directional gabor feature representation. *Information Sciences*, 512:1264–1278.

Bao, C., Cai, J.-F., and Ji, H. (2013). Fast sparsity-based orthogonal dictionary learning for image restoration. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3384–3391.

Bao, C., Ji, H., Quan, Y., and Shen, Z. (2014). L0 norm based dictionary learning by proximal methods with global convergence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3858–3865.

Barak, B., Kelner, J. A., and Steurer, D. (2015). Dictionary learning and tensor decomposition via the sum-of-squares method. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 143–151.

Bhatia, R. (1994). Matrix factorizations and their perturbations. *Linear Algebra and its applications*, 197:245–276.

Donoho, D. L. and Elad, M. (2003). Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization. *Proceedings of the National Academy of Sciences*, 100(5):2197–2202.

Engan, K., Aase, S. O., and Husoy, J. H. (1999). Method of optimal directions for frame design. In *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258)*, volume 5, pages 2443–2446. IEEE.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96(456):1348–1360.

Fattahi, S. and Sojoudi, S. (2020). Exact guarantees on the absence of spurious local minima for non-negative rank-1 robust principal component analysis. *Journal of machine learning research*.

Ge, R., Jin, C., and Zheng, Y. (2017). No spurious local minima in nonconvex low rank problems: A unified geometric analysis. In *International Conference on Machine Learning*, pages 1233–1242. PMLR.

Ge, R., Lee, J. D., and Ma, T. (2016). Matrix completion has no spurious local minimum. *Advances in neural information processing systems*, 29.

Geng, Q. and Wright, J. (2014). On the local correctness of ℓ_1-minimization for dictionary learning. In *2014 IEEE International Symposium on Information Theory*, pages 3180–3184. IEEE.

Gribonval, R. and Schnass, K. (2010). Dictionary identification—sparse matrix-factorization via ℓ_1-minimization. *IEEE Transactions on Information Theory*, 56(7):3523–3539.

Grosse, R., Raina, R., Kwong, H., and Ng, A. Y. (2012). Shift-invariance sparse coding for audio classification. *arXiv preprint arXiv:1206.5241*.

Haq, I. U., Fujii, K., and Kawahara, Y. (2020). Dynamic mode decomposition via dictionary learning for foreground modeling in videos. *Computer Vision and Image Understanding*, 199:103022.

Jenatton, R., Gribonval, R., and Bach, F. (2012). Local stability and robustness of sparse dictionary learning in the presence of noise. *arXiv preprint arXiv:1210.0685*.

Krause, O. and Igel, C. (2015). A more efficient rank-one covariance matrix update for evolution strategies. In *Proceedings of the 2015 ACM Conference on Foundations of Genetic Algorithms XIII*, pages 129–136.

Lee, H., Battle, A., Raina, R., and Ng, A. (2006). Efficient sparse coding algorithms. *Advances in neural information processing systems*, 19.

Li, R.-C. (1995). New perturbation bounds for the unitary polar factor. *SIAM Journal on Matrix Analysis and Applications*, 16(1):327–332.

Liu, M., Nie, L., Wang, X., Tian, Q., and Chen, B. (2018). Online data organizer: micro-video categorization by structure-guided multimodal dictionary learning. *IEEE Transactions on Image Processing*, 28(3):1235–1247.

Mairal, J., Bach, F., Ponce, J., and Sapiro, G. (2009). Online dictionary learning for sparse coding. In *Proceedings of the 26th annual international conference on machine learning*, pages 689–696.

Mairal, J., Elad, M., and Sapiro, G. (2007). Sparse representation for color image restoration. *IEEE Transactions on image processing*, 17(1):53–69.

Olshausen, B. A. and Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609.

Qu, Q., Sun, J., and Wright, J. (2014). Finding a sparse vector in a subspace: Linear sparsity using alternating directions. *Advances in Neural Information Processing Systems*, 27.

Ramirez, I., Sprechmann, P., and Sapiro, G. (2010). Classification and clustering via dictionary learning with structured incoherence and shared features. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3501–3508. IEEE.

Ravishankar, S., Ma, A., and Needell, D. (2020). Analysis of fast structured dictionary learning. *Information and Inference: A Journal of the IMA*, 9(4):785–811.

Schnass, K. (2015). Local identification of overcomplete dictionaries. *J. Mach. Learn. Res.*, 16(Jun):1211–1242.

Spielman, D. A., Wang, H., and Wright, J. (2012). Exact recovery of sparsely-used dictionaries. In *Conference on Learning Theory*, pages 37–1. JMLR Workshop and Conference Proceedings.

Stange, P. (2008). On the efficient update of the singular value decomposition. In *PAMM: Proceedings in Applied Mathematics and Mechanics*, volume 8, pages 10827–10828. Wiley Online Library.

Sun, J., Qu, Q., and Wright, J. (2015). Complete dictionary recovery over the sphere. In *2015 International Conference on Sampling Theory and Applications (SampTA)*, pages 407–410. IEEE.

Sun, J., Qu, Q., and Wright, J. (2016). Complete dictionary recovery over the sphere i: Overview and the geometric picture. *IEEE Transactions on Information Theory*, 63(2):853–884.

Tošić, I. and Frossard, P. (2011). Dictionary learning. *IEEE Signal Processing Magazine*, 28(2):27–38.

Vershynin, R. (2018). *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press.

Xu, Y., Li, Z., Yang, J., and Zhang, D. (2017). A survey of dictionary learning algorithms for face recognition. *IEEE access*, 5:8502–8514.

Zhai, Y., Yang, Z., Liao, Z., Wright, J., and Ma, Y. (2020). Complete dictionary learning via l4-norm maximization over the orthogonal group. *J. Mach. Learn. Res.*, 21(165):1–68.

Zhang, C.-H. (2010). Nearly unbiased variable selection under minimax concave penalty. *The Annals of statistics*, 38(2):894–942.

Zhao, R., Li, H., and Liu, X. (2021). A survey of dictionary learning in medical image analysis and its application for glaucoma diagnosis. *Archives of Computational Methods in Engineering*, 28(2):463–471.

A Preliminaries

Before presenting the proofs of our main theorems, in this section we present some preliminary results in high dimensional statistics and matrix perturbation theory which will be useful in the later sections. We start with a classic result in covariance estimation for sub-Gaussian distributions. We denote the sub-Gaussian norm and L^2 norm of a random variable with $\|\cdot\|_{\psi_2}$ and $\|\cdot\|_{L^2}$ respectively.

Theorem 4 (Tail Bound for Covariance Estimation (Vershynin, 2018))**.** *Let $\boldsymbol{x}$ be a sub-Gaussian random vector in $\mathbb{R}^n$ with covariance matrix $\boldsymbol{\Sigma}$, such that*

$$\|\langle \boldsymbol{x}, \boldsymbol{z}\rangle\|_{\psi_2} \le C_{ce}\|\langle \boldsymbol{x}, \boldsymbol{z}\rangle\|_{L^2} \quad \text{for any } \boldsymbol{z} \in \mathbb{R}^n, \tag{9}$$

for some $C_{ce} \geq 1$. Let $\boldsymbol{X} \in \mathbb{R}^{n\times p}$ be a matrix whose columns have the identical and independent distribution as $\boldsymbol{x}$. Then, we have for any $u \geq 0$,

$$\left\|\frac{1}{p}\boldsymbol{X}\boldsymbol{X}^\top - \boldsymbol{\Sigma}\right\|_2 \lesssim C_{ce}^2\left(\sqrt{\frac{n+u}{p}} + \frac{n+u}{p}\right)\|\boldsymbol{\Sigma}\|_2 \tag{10}$$

with probability at least $1 - 2\exp(-u)$.

The next theorem describes the concentration of the norm of a random vector whose entries are sub-Gaussian random variables.

Theorem 5 (Concentration of the norm (Vershynin, 2018))**.** *Let $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) \in \mathbb{R}^n$ be a random vector with independent, sub-Gaussian coordinates $\boldsymbol{x}_i$ that satisfy $\mathbb{E}\boldsymbol{x}_i^2 = 1$. Then*

$$\big\|\|\boldsymbol{x}\|_2 - \sqrt{n}\big\|_{\psi_2} \le CK^2, \tag{11}$$

where $K = \max_i \|\boldsymbol{x}_i\|_{\psi_2}$ and C is an absolute constant.

Finally we introduce the perturbation bound for the polar factorization.

Theorem 6 (Perturbation Bound for the Unitary Polar Factor (Li, 1995)))**.** *Let $\boldsymbol{A}$ and $\tilde{\boldsymbol{A}}$ be two full rank $n \times n$ matrices with polar decompositions $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{P}$ and $\tilde{\boldsymbol{A}} = \tilde{\boldsymbol{U}}\tilde{\boldsymbol{P}}$. Then*

$$\|\boldsymbol{U} - \tilde{\boldsymbol{U}}\|_F \le \frac{2}{\sigma_n(\boldsymbol{A}) + \sigma_n(\tilde{\boldsymbol{A}})}\|\boldsymbol{A} - \tilde{\boldsymbol{A}}\|_F. \tag{12}$$

B Proof of Main Theorems

In this section we present the proofs of our main theorems. First, we present the proof of Theorem 1. Then, we will present the proof of Theorem 3. The proof of Theorem 2 immediately follows from the proof of Theorem 3 as a simpler case.

B.1 Proof of Theorem 1

We use induction to prove Theorem 1. The induction hypothesis we need is:

$$\|\boldsymbol{D}^{(s)} - \boldsymbol{D}^*\|_F \le O\left(\frac{1}{\log n}\right) \qquad \text{at iteration } s. \tag{13}$$

The base case is easily verified given the initial condition. Since $\|\boldsymbol{D}^{(0)} - \boldsymbol{D}^*\|_{2,\infty} \le \|\boldsymbol{D}^{(0)} - \boldsymbol{D}^*\|_F$, we can invoke the following lemma—which is a special case of (Arora et al., 2015, Lemma 16)—to show exact support recovery.

Lemma 1 (Exact Support Recovery for Sparse Coding). *Consider $\boldsymbol{Y} = \boldsymbol{D}^* \boldsymbol{X}^*$ where $\boldsymbol{D}^* \in \mathbb{O}^n$ and $\boldsymbol{X}^*$ are drawn from distributions satisfying Assumption 1. Suppose that $\boldsymbol{D}$ satisfies*

$$\|\boldsymbol{D} - \boldsymbol{D}^*\|_{2,\infty} \leq O\left(\frac{1}{\log n}\right), \tag{14}$$

Then, with probability at least $1 - 2\exp\left\{\log(pn) - C\log^2 n\right\}$ for some universal constant C, we have

$$\operatorname{supp}\left(\mathrm{HT}_{\Gamma/2}\left(\boldsymbol{D}^\top \boldsymbol{Y}\right)\right) = \operatorname{supp}\left(\boldsymbol{X}^*\right). \tag{15}$$

Lemma 1 together with our induction hypothesis (13) implies that

$$\operatorname{supp}\left(\boldsymbol{X}^{(s)}\right) = \operatorname{supp}\left(\boldsymbol{X}^*\right). \tag{16}$$

We use the exact recovery of support to bound the error on the sparse code:

$$\begin{aligned}
\|\boldsymbol{X}^{(s)} - \boldsymbol{X}^*\|_F &= \|\mathrm{HT}_{C/2}(\boldsymbol{D}^{(s)\top}\boldsymbol{Y}) - \boldsymbol{X}^*\|_F \\
&\overset{(a)}{\leq} \|\boldsymbol{D}^{(s)\top}\boldsymbol{Y} - \boldsymbol{D}^{*\top}\boldsymbol{Y}\|_F \\
&\leq \|\boldsymbol{D}^{(s)} - \boldsymbol{D}^*\|_F \|\boldsymbol{Y}\|_2,
\end{aligned} \tag{17}$$

where we used (16) for inequality (a). The above inequality implies that

$$\begin{aligned}
\frac{\|\boldsymbol{X}^{(s)} - \boldsymbol{X}^*\|_F}{\|\boldsymbol{X}^*\|_2} &= \frac{\|\boldsymbol{X}^{(s)} - \boldsymbol{X}^*\|_F}{\|\boldsymbol{D}^*\boldsymbol{X}^*\|_2} \\
&= \frac{\|\boldsymbol{X}^{(s)} - \boldsymbol{X}^*\|_F}{\|\boldsymbol{Y}\|_2} \\
&\leq \|\boldsymbol{D}^{(s)} - \boldsymbol{D}^*\|_F \\
&\leq O\left(\frac{1}{\log n}\right)
\end{aligned} \tag{18}$$

To proceed, we need the following lemma, the proof of which can be found in Appendix C.

Lemma 2 (Guaranteed Improvement on Polar Decomposition). *Suppose $\boldsymbol{X}^*$ is drawn from distribution satisfying Assumption 1, $\boldsymbol{Y} = \boldsymbol{D}^*\boldsymbol{X}^*$, $p \geq \Omega(n/\theta^2)$, and we have an approximation $\boldsymbol{X}$ of $\boldsymbol{X}^*$ such that $\frac{\|\boldsymbol{X} - \boldsymbol{X}^*\|_F}{\|\boldsymbol{X}^*\|_2} \leq O(\sqrt{\theta})$, then with high probability we have*

$$\|\mathrm{Polar}(\boldsymbol{Y}\boldsymbol{X}^\top) - \boldsymbol{D}^*\|_F < \frac{6\sqrt{\theta}\|\boldsymbol{X} - \boldsymbol{X}^*\|_F}{\|\boldsymbol{X}^*\|_2}. \tag{19}$$

Invoking Lemma 2 with $\boldsymbol{X} = \boldsymbol{X}^{(s)}$ together with (18) leads to

$$\|\boldsymbol{D}^{(s+1)} - \boldsymbol{D}^*\|_F < \frac{6\sqrt{\theta}\|\boldsymbol{X}^{(s)} - \boldsymbol{X}^*\|_F}{\|\boldsymbol{X}^*\|_2}. \tag{20}$$

As a result, we successfully show that for $\theta < 1/144$, we have

$$\|\boldsymbol{D}^{(s+1)} - \boldsymbol{D}^*\|_F < \frac{\|\boldsymbol{X}^{(s)} - \boldsymbol{X}^*\|_F}{2\|\boldsymbol{X}^*\|_2} \leq \frac{1}{2}\|\boldsymbol{D}^{(s)} - \boldsymbol{D}^*\|_F. \tag{21}$$

Therefore, the induction hypothesis (13) holds for $t = s + 1$. Consequently, the linear convergence for $\frac{\|\boldsymbol{X}^{(s)}-\boldsymbol{X}^*\|_F}{\|\boldsymbol{X}^*\|_2}$ follows from (18) and (21):

$$\frac{\|\boldsymbol{X}^{(s+1)} - \boldsymbol{X}^*\|_F}{\|\boldsymbol{X}^*\|_2} \leq \|\boldsymbol{D}^{(s+1)} - \boldsymbol{D}^*\|_F < \frac{\|\boldsymbol{X}^{(s)} - \boldsymbol{X}^*\|_F}{2\|\boldsymbol{X}^*\|_2}.$$

This completes the proof. □

B.2 Proof of Theorem 3

The linear convergence of $\boldsymbol{A}^{(t)}$ hinges on the linear convergence of $\boldsymbol{D}^{(t)}$ towards

$$\boldsymbol{D}^* = \mathcal{L}((\boldsymbol{A}^*\boldsymbol{A}^{*\top})^{-1})^\top \boldsymbol{A}^*. \tag{22}$$

Specifically, the main challenge in the proof is to show the following linear convergence:

$$\|\boldsymbol{D}^{(s+1)} - \boldsymbol{D}^*\|_F \leq \rho\|\boldsymbol{D}^{(s)} - \boldsymbol{D}^*\|_F + O\left(\|\boldsymbol{P}^{(s)}\boldsymbol{A}^* - \boldsymbol{D}^*\|_F\right), \tag{23}$$

for some $0 < \rho < 1$. To this goal, we introduce the following intermediate lemmas, the proofs of which can be found in Appendix C.

Lemma 3 (Preconditioner Approximation)**.** *Under Assumption 1, with high probability we have:*

$$\left\|\boldsymbol{P}^{(t)} - \mathcal{L}\left(\left(\boldsymbol{A}^*\boldsymbol{A}^{*\top}\right)^{-1}\right)^\top\right\|_2 \leq O\left(\frac{\hat{\kappa}^6}{\theta}\sqrt{\frac{n}{p_1 + t}}\right). \tag{24}$$

Lemma 4 (Spectral Property for Sparse Code Matrix)**.** *For a random matrix $\boldsymbol{X}$ that is drawn from distribution satisfying Assumption 1 and $p \geq \Omega(n/\theta^2)$, we have*

$$\|\boldsymbol{X}\|_2 \leq \frac{11}{10}\sqrt{p\theta}\sigma, \quad \sigma_n(\boldsymbol{X}) \geq \frac{9}{10}\sqrt{p\theta}\sigma, \quad \kappa(\boldsymbol{X}) \leq \frac{11}{9}, \tag{25}$$

with high probability.

For the remainder of this section, we abuse the notation and use $\boldsymbol{X}^*$ to denote the sparse coding matrix that generates $\boldsymbol{Y}^{(s)} = \boldsymbol{A}^*\boldsymbol{X}^*$, where $\boldsymbol{Y}^{(s)}$ is obtained by adding the latest sample as the last column and removing the oldest sample from its first column (see Steps 6 and 7 in Algorithm 3). Indeed, $\boldsymbol{X}^*$ is different from iteration to iteration, but it plays a similar role in the proof to the ground truth matrix in the full-batch case. We also consider $\tilde{\boldsymbol{X}}$ to be $\tilde{\boldsymbol{X}} = \boldsymbol{D}^{*\top}\tilde{\boldsymbol{Y}}^{(s)}$, which is again a different sparse coding matrix for different iterations since $\tilde{\boldsymbol{Y}}^{(s)} = \boldsymbol{P}^{(s)}\boldsymbol{Y}^{(s)}$ and both $\boldsymbol{P}^{(s)}$ and $\boldsymbol{Y}^{(s)}$ change from iteration to iteration.

Towards proving (23), we first show that $\tilde{\boldsymbol{X}}^{(s)}$ recovers the sparsity pattern of $\boldsymbol{X}^*$ if we use enough number of samples to construct the preconditioner. We consider one entry $\left(\boldsymbol{D}^{(s)\top}\tilde{\boldsymbol{Y}}^{(s)}\right)_{ij}$ of $\boldsymbol{D}^{(s)\top}\tilde{\boldsymbol{Y}}^{(s)}$ and write

$$\begin{aligned}\left(\boldsymbol{D}^{(s)\top}\tilde{\boldsymbol{Y}}^{(s)}\right)_{ij} &= \left(\boldsymbol{D}^{(s)\top}\boldsymbol{D}^*\tilde{\boldsymbol{X}}\right)_{ij} \\ &= \left(\boldsymbol{D}^{(s)\top}\boldsymbol{D}^*(\tilde{\boldsymbol{X}} - \boldsymbol{X}^*)\right)_{ij} + \left(\boldsymbol{D}^{(s)\top}\boldsymbol{D}^*\boldsymbol{X}^*\right)_{ij}.\end{aligned} \tag{26}$$

The first term can be bounded as

$$
\begin{aligned}
\left|\left(\boldsymbol{D}^{(s)\top}\boldsymbol{D}^*(\tilde{\boldsymbol{X}}-\boldsymbol{X}^*)\right)_{ij}\right| &\leq \|\boldsymbol{D}^{(s)\top}\boldsymbol{D}^*\|_2\|\tilde{\boldsymbol{X}}-\boldsymbol{X}^*\|_{2,\inf} \\
&= \|\tilde{\boldsymbol{X}}-\boldsymbol{X}^*\|_{2,\inf} \\
&\leq \|\boldsymbol{D}^{*\top}\boldsymbol{P}^{(s)}\boldsymbol{A}^*\boldsymbol{X}^*-\boldsymbol{X}^*\|_{2,\inf} \\
&\leq \|\boldsymbol{P}^{(s)}\boldsymbol{A}^*-\boldsymbol{D}^*\|_2\|\boldsymbol{X}^*\|_{2,\inf} \\
&\leq \left\|\boldsymbol{P}^{(s)}-\mathcal{L}\left(\left(\boldsymbol{A}^*\boldsymbol{A}^{*\top}\right)^{-1}\right)\right\|_2\|\boldsymbol{A}^*\|_2\|\boldsymbol{X}^*\|_{2,\inf} \\
&\lesssim \frac{\hat{\kappa}^6}{\theta}\sqrt{\frac{n}{p_1+t}}\|\boldsymbol{X}^*\|_{2,\inf}. \qquad (27)
\end{aligned}
$$

The last inequality is a result of Lemma 3. As will be clear in its proof, Lemma 3 holds with probability of at least $1-2\exp(-\log^2 n)$, which allows us to invoke it for every $\boldsymbol{P}^{(s)}$. The term $\|\boldsymbol{X}^*\|_{2,\inf}$ denotes the largest column norm of matrix $\boldsymbol{X}^*$. The expectation for the norm of each column, based on Assumption 1, is $\sigma\sqrt{\theta n}$. The random variable $\|\boldsymbol{X}^*_{(\cdot,k)}\|_2-\sigma\sqrt{\theta n}$ is sub-Gaussian with sub-Gaussian norm $O(\sqrt{\theta}\sigma)$ by Theorem 5. As a result, we can bound the norm of the kth column of $\boldsymbol{X}^*$ as

$$
\mathbb{P}\left(\|\boldsymbol{X}^*_{(\cdot,k)}\|_2 \geq 2\sigma\sqrt{\theta n}\right) \leq 2\exp\left(-Cn\right), \qquad (28)
$$

for some constant C. After taking the union bound, we have

$$
\|\boldsymbol{X}^*\|_{2,\inf} = \max_{1\leq k\leq p_2}\|\boldsymbol{X}^*_{(\cdot,k)}\|_2 \leq 2\sigma\sqrt{\theta n\log p_2} \qquad (29)
$$

with probability of at least $1-2T\exp\left(-Cn\right)$. Therefore, we have that with high probability,

$$
\left|\left(\boldsymbol{D}^{(s)\top}\boldsymbol{D}^*(\tilde{\boldsymbol{X}}-\boldsymbol{X}^*)\right)_{ij}\right| \lesssim \frac{\hat{\kappa}^6\sigma}{\sqrt{\theta}}\sqrt{\frac{n^2\log p_2}{p_1+t}}, \qquad (30)
$$

for every (i,j). Based on the above inequality and with the choice of $p_1 \geq \Omega(n^2\log n\sigma^2\hat{\kappa}^{12}/(\theta^2\Gamma))$, we have $\left|\left(\boldsymbol{D}^{(s)\top}\boldsymbol{D}^*(\tilde{\boldsymbol{X}}-\boldsymbol{X}^*)\right)_{ij}\right| < \Gamma/4$. Now, recalling (26), we have bounded the first term by $\Gamma/4$. The deviation of the second term from X^*_{ij} can also be bounded by $\Gamma/4$ with high probability, by the proof of Lemma 1 which will not be repeated here. To sum up, we have $|\left(\boldsymbol{D}^{(s)\top}\tilde{\boldsymbol{Y}}^{(s)}\right)_{ij}| < \Gamma/2$ when $X^*_{ij}=0$ and $|\left(\boldsymbol{D}^{(s)\top}\tilde{\boldsymbol{Y}}^{(s)}\right)_{ij}| > \Gamma/2$ when $X^*_{ij}\neq 0$, which leads to

$$
\mathrm{supp}\left(\tilde{\boldsymbol{X}}^{(s)}\right) = \mathrm{supp}(\boldsymbol{X}^*). \qquad (31)
$$

From now on, we use $\mathcal{P}(\cdot)$ to denote a projection of a matrix onto $\mathrm{supp}(\boldsymbol{X}^*)$. Specifically, for any matrix $\boldsymbol{A}\in\mathbb{R}^{n\times p_2}$, we have

$$
\left(\mathcal{P}(\boldsymbol{A})\right)_{ij} = \begin{cases} \boldsymbol{A}_{ij} & \text{if} \quad (i,j)\in\mathrm{supp}(\boldsymbol{X}^*) \\ 0 & \text{if} \quad (i,j)\notin\mathrm{supp}(\boldsymbol{X}^*) \end{cases}. \qquad (32)
$$

Given the exact support recovery, we have

$$
\begin{aligned}
\|\tilde{\boldsymbol{X}}^{(s)} - \mathcal{P}(\tilde{\boldsymbol{X}})\|_F &= \|\mathcal{P}(\boldsymbol{D}^{(s)\top}\tilde{\boldsymbol{Y}}^{(s)}) - \mathcal{P}(\tilde{\boldsymbol{X}})\|_F \\
&\leq \|\boldsymbol{D}^{(s)\top}\tilde{\boldsymbol{Y}}^{(s)} - \tilde{\boldsymbol{X}}\|_F \\
&= \|\boldsymbol{D}^{(s)\top}\tilde{\boldsymbol{Y}}^{(s)} - \boldsymbol{D}^{*\top}\tilde{\boldsymbol{Y}}^{(s)}\|_F \\
&\leq \|\boldsymbol{D}^{(s)} - \boldsymbol{D}^*\|_F\|\tilde{\boldsymbol{Y}}^{(s)}\|_2 \\
&= \|\boldsymbol{D}^{(s)} - \boldsymbol{D}^*\|_F\|\tilde{\boldsymbol{X}}\|_2.
\end{aligned} \tag{33}
$$

As a result we have

$$
\begin{aligned}
&\frac{\|\tilde{\boldsymbol{X}}^{(s)} - \mathcal{P}(\tilde{\boldsymbol{X}})\|_F}{\|\mathcal{P}(\tilde{\boldsymbol{X}})\|_2} \\
&= \frac{\|\tilde{\boldsymbol{X}}^{(s)} - \mathcal{P}(\tilde{\boldsymbol{X}})\|_F}{\|\tilde{\boldsymbol{X}}\|_2}\frac{\|\tilde{\boldsymbol{X}}\|_2}{\|\mathcal{P}(\tilde{\boldsymbol{X}})\|_2} \\
&\leq \|\boldsymbol{D}^{(s)} - \boldsymbol{D}^*\|_F\frac{\|\tilde{\boldsymbol{X}}\|_2}{\|\tilde{\boldsymbol{X}}\|_2 - \|\tilde{\boldsymbol{X}} - \mathcal{P}(\tilde{\boldsymbol{X}})\|_2} \\
&\leq \|\boldsymbol{D}^{(s)} - \boldsymbol{D}^*\|_F\frac{\|\tilde{\boldsymbol{X}}\|_2}{\|\tilde{\boldsymbol{X}}\|_2 - \|\tilde{\boldsymbol{X}} - \mathcal{P}(\tilde{\boldsymbol{X}})\|_F} \\
&\overset{(a)}{\leq} \|\boldsymbol{D}^{(s)} - \boldsymbol{D}^*\|_F\frac{\|\tilde{\boldsymbol{X}}\|_2}{\|\tilde{\boldsymbol{X}}\|_2 - \|\tilde{\boldsymbol{X}} - \boldsymbol{X}^*\|_F} \\
&\overset{(b)}{\leq} \|\boldsymbol{D}^{(s)} - \boldsymbol{D}^*\|_F\left(1 + \frac{\|\tilde{\boldsymbol{X}} - \boldsymbol{X}^*\|_F}{\|\boldsymbol{X}^*\|_2 - 2\|\tilde{\boldsymbol{X}} - \boldsymbol{X}^*\|_F}\right) \\
&\leq \|\boldsymbol{D}^{(s)} - \boldsymbol{D}^*\|_F\left(1 + \frac{\frac{\|\tilde{\boldsymbol{X}} - \boldsymbol{X}^*\|_F}{\|\boldsymbol{X}^*\|_2}}{1 - \frac{2\|\tilde{\boldsymbol{X}} - \boldsymbol{X}^*\|_F}{\|\boldsymbol{X}^*\|_2}}\right) \\
&\overset{(c)}{\leq} \|\boldsymbol{D}^{(s)} - \boldsymbol{D}^*\|_F\left(1 + \frac{\frac{1}{10}}{1 - 2\times\frac{1}{10}}\right) \\
&= \frac{9}{8}\|\boldsymbol{D}^{(s)} - \boldsymbol{D}^*\|_F.
\end{aligned} \tag{34}
$$

Here inequality (a) is due to the decomposition of $\tilde{\boldsymbol{X}} - \boldsymbol{X}^*$ onto and outside the support of $\boldsymbol{X}^*$. Specifically, consider $\mathcal{P}_\perp(\cdot)$ to be the projection orthogonal to $\boldsymbol{P}(\cdot)$. We have $\mathcal{P}_\perp(\tilde{\boldsymbol{X}} - \mathcal{P}(\tilde{\boldsymbol{X}})) = \mathcal{P}_\perp(\tilde{\boldsymbol{X}} - \boldsymbol{X}^*)$ while $\mathcal{P}(\tilde{\boldsymbol{X}} - \mathcal{P}(\tilde{\boldsymbol{X}}))$ is an all zero matrix. So we can conclude $\|\tilde{\boldsymbol{X}} - \mathcal{P}(\tilde{\boldsymbol{X}})\|_F \leq \|\tilde{\boldsymbol{X}} - \boldsymbol{X}^*\|_F$. The inequality (b) is from $\|\boldsymbol{X}^*\|_2 - \|\tilde{\boldsymbol{X}} - \boldsymbol{X}^*\|_F \leq \|\boldsymbol{X}^*\|_2 - \|\tilde{\boldsymbol{X}} - \boldsymbol{X}^*\|_2 \leq \|\tilde{\boldsymbol{X}}\|_2$. The inequality (c) holds due to $\frac{\|\tilde{\boldsymbol{X}} - \boldsymbol{X}^*\|_F}{\|\boldsymbol{X}^*\|_2} \leq 1/10$, which is a result from Lemma 3:

$$
\begin{aligned}
\|\tilde{\boldsymbol{X}} - \boldsymbol{X}^*\|_F &= \|\boldsymbol{D}^{*\top}\boldsymbol{P}^{(s)}\boldsymbol{A}^*\boldsymbol{X}^* - \boldsymbol{X}^*\|_F \\
&\leq \|\boldsymbol{P}^{(s)}\boldsymbol{A}^* - \boldsymbol{D}^*\|_F\|\boldsymbol{X}^*\|_2 \\
&\leq \sqrt{n}\|\boldsymbol{P}^{(s)}\boldsymbol{A}^* - \boldsymbol{D}^*\|_2\|\boldsymbol{X}^*\|_2 \\
&\lesssim \frac{\hat{\kappa}^6}{\theta}\sqrt{\frac{n^2}{p_1 + t}}\|\boldsymbol{X}^*\|_2,
\end{aligned} \tag{35}
$$

and $p_1 \geq \Omega(n^2 \log n \sigma^2 \hat{\kappa}^{12}/(\theta^2 \Gamma))$.

So far, we have bounded $\frac{\|\tilde{\boldsymbol{X}}^{(s)} - \mathcal{P}(\tilde{\boldsymbol{X}})\|_F}{\|\mathcal{P}(\tilde{\boldsymbol{X}})\|_2}$ with $\|\boldsymbol{D}^{(s)} - \boldsymbol{D}^*\|_F$, the next step is to bound $\|\boldsymbol{D}^{(s+1)} - \boldsymbol{D}^*\|_F$ with $\frac{\|\tilde{\boldsymbol{X}}^{(s)} - \mathcal{P}(\tilde{\boldsymbol{X}})\|_F}{\|\mathcal{P}(\tilde{\boldsymbol{X}})\|_2}$. For $\|\boldsymbol{D}^{(s+1)} - \boldsymbol{D}^*\|_F$, we have

$$\begin{aligned}
&\|\boldsymbol{D}^{(s+1)} - \boldsymbol{D}^*\|_F \\
&= \|\text{Polar}(\tilde{\boldsymbol{Y}}^{(s)} \tilde{\boldsymbol{X}}^{(s)\top}) - \boldsymbol{D}^*\|_F \\
&= \|\text{Polar}(\boldsymbol{D}^* \tilde{\boldsymbol{X}} \tilde{\boldsymbol{X}}^{(s)\top}) - \boldsymbol{D}^*\|_F \\
&= \|\text{Polar}(\boldsymbol{D}^* \tilde{\boldsymbol{X}} \tilde{\boldsymbol{X}}^{(s)\top}) - \text{Polar}(\boldsymbol{D}^* \mathcal{P}(\tilde{\boldsymbol{X}}) \tilde{\boldsymbol{X}}^{(s)\top}) \\
&\quad + \text{Polar}(\boldsymbol{D}^* \mathcal{P}(\tilde{\boldsymbol{X}}) \tilde{\boldsymbol{X}}^{(s)\top}) - \boldsymbol{D}^*\|_F \\
&= \|\text{Polar}(\boldsymbol{D}^* \tilde{\boldsymbol{X}} \tilde{\boldsymbol{X}}^{(s)\top}) - \text{Polar}(\boldsymbol{D}^* \mathcal{P}(\tilde{\boldsymbol{X}}) \tilde{\boldsymbol{X}}^{(s)\top})\|_F \\
&\quad + \|\text{Polar}(\boldsymbol{D}^* \mathcal{P}(\tilde{\boldsymbol{X}}) \tilde{\boldsymbol{X}}^{(s)\top}) - \boldsymbol{D}^*\|_F
\end{aligned} \tag{36}$$

To bound the first term $\|\text{Polar}(\boldsymbol{D}^* \tilde{\boldsymbol{X}} \tilde{\boldsymbol{X}}^{(s)\top}) - \text{Polar}(\boldsymbol{D}^* \mathcal{P}(\tilde{\boldsymbol{X}}) \tilde{\boldsymbol{X}}^{(s)\top})\|_F$, we invoke Theorem 6 and get

$$\begin{aligned}
&\|\text{Polar}(\boldsymbol{D}^* \tilde{\boldsymbol{X}} \tilde{\boldsymbol{X}}^{(s)\top}) - \text{Polar}(\boldsymbol{D}^* \mathcal{P}(\tilde{\boldsymbol{X}}) \tilde{\boldsymbol{X}}^{(s)\top})\|_F \\
\leq\ & \frac{2\|\boldsymbol{D}^* \tilde{\boldsymbol{X}} \tilde{\boldsymbol{X}}^{(s)\top} - \boldsymbol{D}^* \mathcal{P}(\tilde{\boldsymbol{X}}) \tilde{\boldsymbol{X}}^{(s)\top}\|_F}{\sigma_n(\boldsymbol{D}^* \tilde{\boldsymbol{X}} \tilde{\boldsymbol{X}}^{(s)\top}) + \sigma_n(\boldsymbol{D}^* \mathcal{P}(\tilde{\boldsymbol{X}}) \tilde{\boldsymbol{X}}^{(s)\top})} \\
\overset{(a)}{\leq}\ & \frac{2\kappa(\tilde{\boldsymbol{X}}^{(s)}) \|\tilde{\boldsymbol{X}} - \mathcal{P}(\tilde{\boldsymbol{X}})\|_F}{2\sigma_n(\boldsymbol{X}^*) - \|\boldsymbol{X}^* - \tilde{\boldsymbol{X}}\|_2 - \|\boldsymbol{X}^* - \mathcal{P}(\tilde{\boldsymbol{X}})\|_2} \\
=\ & \frac{2\kappa(\tilde{\boldsymbol{X}}^{(s)}) \|\tilde{\boldsymbol{X}} - \mathcal{P}(\tilde{\boldsymbol{X}})\|_F}{\|\boldsymbol{X}^*\|_2} \cdot \frac{1}{\frac{2}{\kappa(\boldsymbol{X}^*)} - \frac{\|\boldsymbol{X}^* - \tilde{\boldsymbol{X}}\|_2}{\|\boldsymbol{X}^*\|_2} - \frac{\|\boldsymbol{X}^* - \mathcal{P}(\tilde{\boldsymbol{X}})\|_2}{\|\boldsymbol{X}^*\|_2}} \\
\overset{(b)}{\leq}\ & \frac{2\kappa(\tilde{\boldsymbol{X}}^{(s)}) \|\tilde{\boldsymbol{X}} - \mathcal{P}(\tilde{\boldsymbol{X}})\|_F}{\|\boldsymbol{X}^*\|_2} \frac{1}{18/11 - 1/10 - 1/10} \\
\leq\ & \frac{3\kappa(\tilde{\boldsymbol{X}}^{(s)}) \|\tilde{\boldsymbol{X}} - \mathcal{P}(\tilde{\boldsymbol{X}})\|_F}{\|\boldsymbol{X}^*\|_2}
\end{aligned} \tag{37}$$

Inequality (a) is due to normalizing both the numerator and the denominator with $\|\tilde{\boldsymbol{X}}^{(s)}\|_2$. In inequality (b) we use the result of Lemma 4. Next we aim to find a bound on $\kappa(\tilde{\boldsymbol{X}}^{(s)})$, and we do so by first bounding $\kappa(\mathcal{P}(\tilde{\boldsymbol{X}}))$. By Lemma 4 and Lemma 3, we have

$$\begin{aligned}
\kappa(\mathcal{P}(\tilde{\boldsymbol{X}})) &= \frac{\|\mathcal{P}(\tilde{\boldsymbol{X}})\|_2}{\sigma_n(\mathcal{P}(\tilde{\boldsymbol{X}}))} \\
&\leq \frac{\|\boldsymbol{X}^*\|_2 + \|\mathcal{P}(\tilde{\boldsymbol{X}}) - \boldsymbol{X}^*\|_2}{\sigma_n(\boldsymbol{X}^*) - \|\mathcal{P}(\tilde{\boldsymbol{X}}) - \boldsymbol{X}^*\|_2} \\
&= \frac{\|\boldsymbol{X}^*\|_2 + \|\mathcal{P}(\tilde{\boldsymbol{X}}) - \boldsymbol{X}^*\|_2}{\|\boldsymbol{X}^*\|_2/\kappa(\boldsymbol{X}^*) - \|\mathcal{P}(\tilde{\boldsymbol{X}}) - \boldsymbol{X}^*\|_2} \\
&\leq \frac{1 + \|\mathcal{P}(\tilde{\boldsymbol{X}}) - \boldsymbol{X}^*\|_F/\|\boldsymbol{X}^*\|_2}{1/\kappa(\boldsymbol{X}^*) - \|\mathcal{P}(\tilde{\boldsymbol{X}}) - \boldsymbol{X}^*\|_F/\|\boldsymbol{X}^*\|_2} \\
&\leq \frac{1 + 1/10}{9/11 - 1/10} \\
&= \frac{121}{79}.
\end{aligned} \tag{38}$$

By a similar argument we have

$$\begin{aligned}
\kappa(\tilde{\boldsymbol{X}}^{(s)}) &\leq \frac{\|\mathcal{P}(\tilde{\boldsymbol{X}})\|_2 + \|\mathcal{P}(\tilde{\boldsymbol{X}}) - \tilde{\boldsymbol{X}}^{(s)}\|_2}{\sigma_n(\mathcal{P}(\tilde{\boldsymbol{X}})) - \|\mathcal{P}(\tilde{\boldsymbol{X}}) - \tilde{\boldsymbol{X}}^{(s)}\|_2} \\
&\leq \frac{1 + \|\mathcal{P}(\tilde{\boldsymbol{X}}) - \tilde{\boldsymbol{X}}^{(s)}\|_F/\|\mathcal{P}(\tilde{\boldsymbol{X}})\|_2}{1/\kappa(\mathcal{P}(\tilde{\boldsymbol{X}})) - \|\mathcal{P}(\tilde{\boldsymbol{X}}) - \tilde{\boldsymbol{X}}^{(s)}\|_F/\|\mathcal{P}(\tilde{\boldsymbol{X}})\|_2} \\
&\overset{(a)}{\leq} \frac{1 + 9/32}{79/121 - 9/32} \\
&< 4.
\end{aligned} \tag{39}$$

For inequality (a) we used (34) and the fact that $\|\boldsymbol{D}^{(s)} - \boldsymbol{D}^*\|_F \leq O(\frac{1}{\log n}) \leq 1/4$. To conclude we have the first term of the right hand side of (36) bounded as

$$\begin{aligned}
\|\text{Polar}(\boldsymbol{D}^*\tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}^{(s)\top}) - \text{Polar}(\boldsymbol{D}^*\mathcal{P}(\tilde{\boldsymbol{X}})\tilde{\boldsymbol{X}}^{(s)\top})\|_F &< \frac{12\|\tilde{\boldsymbol{X}} - \mathcal{P}(\tilde{\boldsymbol{X}})\|_F}{\|\boldsymbol{X}^*\|_2} \\
&\leq \frac{12\|\boldsymbol{X}^* - \mathcal{P}(\tilde{\boldsymbol{X}})\|_F}{\|\boldsymbol{X}^*\|_2}.
\end{aligned} \tag{40}$$

Now we turn our focus to bounding the second term of (36), which is $\|\text{Polar}(\boldsymbol{D}^*\mathcal{P}(\tilde{\boldsymbol{X}})\tilde{\boldsymbol{X}}^{(s)\top}) - \boldsymbol{D}^*\|_F$. We first introduce the following lemma by Ravishankar et al. (2020):

Lemma 5. *For any matrix $\boldsymbol{X}^* \in \mathbb{R}^{n\times p}$ and $\boldsymbol{Y} = \boldsymbol{D}^*\boldsymbol{X}^*$ where $\boldsymbol{D}^* \in \mathbb{O}^n$, consider an approximation $\boldsymbol{X}$ to $\boldsymbol{X}^*$ such that the normalized error matrix $\boldsymbol{E} := (\boldsymbol{X} - \boldsymbol{X}^*)/\|\boldsymbol{X}^*\|_2$ satisfies:*

$$\|\boldsymbol{E}\|_F < \frac{1}{\kappa^2(\boldsymbol{X}^*)}. \tag{41}$$

Then an approximation to $\boldsymbol{D}^*$, obtained by $\boldsymbol{D} = \text{Polar}(\boldsymbol{Y}\boldsymbol{X}^\top)$, satisfies:

$$\|\boldsymbol{D} - \boldsymbol{D}^*\|_F \leq \frac{\kappa(\boldsymbol{X}^*)^4}{2}\frac{\left\|\boldsymbol{X}^*\boldsymbol{E}^\top - \boldsymbol{E}\boldsymbol{X}^{*\top}\right\|_F}{\|\boldsymbol{X}^*\|_2} + O(\|\boldsymbol{E}\|_F^2). \tag{42}$$

Now we can set

$$\tilde{\boldsymbol{E}}^{(s)} = \frac{\tilde{\boldsymbol{X}}^{(s)} - \mathcal{P}(\tilde{\boldsymbol{X}})}{\|\mathcal{P}(\tilde{\boldsymbol{X}})\|_2}, \tag{43}$$

and notice that

$$\|\tilde{\boldsymbol{E}}^{(s)}\|_F \overset{Eq\ 34}{\leq} \frac{9}{8}\|\boldsymbol{D}^{(s)} - \boldsymbol{D}^*\|_F \leq O(1/\log n) \leq \left(\frac{79}{121}\right)^2 \overset{Eq\ 38}{\leq} \frac{1}{\kappa^2(\mathcal{P}(\tilde{\boldsymbol{X}}))}. \tag{44}$$

Upon invoking Lemma 5 we get

$$\|\text{Polar}(\boldsymbol{D}^*\mathcal{P}(\tilde{\boldsymbol{X}})\tilde{\boldsymbol{X}}^{(s)\top}) - \boldsymbol{D}^*\|_F \leq \frac{\kappa(\mathcal{P}(\tilde{\boldsymbol{X}}))^4}{2}\frac{\left\|\mathcal{P}(\tilde{\boldsymbol{X}})\tilde{\boldsymbol{E}}^{(s)\top} - \tilde{\boldsymbol{E}}^{(s)}\mathcal{P}(\tilde{\boldsymbol{X}})^\top\right\|_F}{\|\mathcal{P}(\tilde{\boldsymbol{X}})\|_2} + O(\|\tilde{\boldsymbol{E}}^{(s)}\|_F^2). \tag{45}$$

By recalling that $\text{supp}(\tilde{\boldsymbol{X}}^{(s)}) = \text{supp}(\boldsymbol{X}^*)$, we follow the same argument in the proof of Lemma 2 and conclude that

$$\frac{\left\|\mathcal{P}(\tilde{\boldsymbol{X}})\tilde{\boldsymbol{E}}^{(s)\top} - \tilde{\boldsymbol{E}}^{(s)}\mathcal{P}(\tilde{\boldsymbol{X}})^\top\right\|_F}{2\|\mathcal{P}(\tilde{\boldsymbol{X}})\|_2} \leq \frac{\max_{1\leq k\leq n}\left\|\boldsymbol{M}_k\mathcal{P}(\tilde{\boldsymbol{X}})\tilde{\boldsymbol{M}}_k\right\|_2\|\tilde{\boldsymbol{E}}^{(s)}\|_F}{\|\mathcal{P}(\tilde{\boldsymbol{X}})\|_2}, \tag{46}$$

where $\boldsymbol{M}_k$ and $\tilde{\boldsymbol{M}}_k$ are defined in the same way as in the proof of Lemma 2. Since both $\boldsymbol{M}_k$ and $\tilde{\boldsymbol{M}}_k$ are linear operators, we have

$$\begin{aligned}
\frac{\max_{1\leq k\leq n}\left\|\boldsymbol{M}_k\mathcal{P}(\tilde{\boldsymbol{X}})\tilde{\boldsymbol{M}}_k\right\|_2}{\|\mathcal{P}(\tilde{\boldsymbol{X}})\|_2} &\leq \frac{\max_{1\leq k\leq n}\left\|\boldsymbol{M}_k\boldsymbol{X}^*\tilde{\boldsymbol{M}}_k\right\|_2 + \|\mathcal{P}(\tilde{\boldsymbol{X}}) - \boldsymbol{X}^*\|_2}{\|\boldsymbol{X}^*\|_2 - \|\mathcal{P}(\tilde{\boldsymbol{X}}) - \boldsymbol{X}^*\|_2} \\
&\leq \frac{\max_{1\leq k\leq n}\left\|\boldsymbol{M}_k\boldsymbol{X}^*\tilde{\boldsymbol{M}}_k\right\|_2 + \|\mathcal{P}(\tilde{\boldsymbol{X}}) - \boldsymbol{X}^*\|_2}{\|\boldsymbol{X}^*\|_2(1 - \|\mathcal{P}(\tilde{\boldsymbol{X}}) - \boldsymbol{X}^*\|_F/\|\boldsymbol{X}^*\|_2)} \\
&\leq \frac{10}{9}\left(\frac{\max_{1\leq k\leq n}\left\|\boldsymbol{M}_k\boldsymbol{X}^*\tilde{\boldsymbol{M}}_k\right\|_2 + \|\mathcal{P}(\tilde{\boldsymbol{X}}) - \boldsymbol{X}^*\|_2}{\|\boldsymbol{X}^*\|_2}\right) \\
&\leq \frac{10}{9}\left(\frac{\max_{1\leq k\leq n}\left\|\boldsymbol{M}_k\boldsymbol{X}^*\tilde{\boldsymbol{M}}_k\right\|_2}{\|\boldsymbol{X}^*\|_2} + \frac{\|\mathcal{P}(\tilde{\boldsymbol{X}}) - \boldsymbol{X}^*\|_F}{\|\boldsymbol{X}^*\|_2}\right) \\
&\leq \frac{10}{9}\left(\frac{\max_{1\leq k\leq n}\left\|\boldsymbol{M}_k\boldsymbol{X}^*\tilde{\boldsymbol{M}}_k\right\|_2}{\|\boldsymbol{X}^*\|_2} + \frac{1}{10}\right)
\end{aligned} \tag{47}$$

In the proof of Lemma 2, we have that with probability $1 - 2\exp(\log n - \log^2 n)$ and the condition $p_2 \geq \Omega(n/\theta^2)$,

$$\frac{\max_{1\leq k\leq n}\left\|\boldsymbol{M}_k\boldsymbol{X}^*\tilde{\boldsymbol{M}}_k\right\|_2}{\|\boldsymbol{X}^*\|_2} \leq \frac{11}{9}\sqrt{\theta}. \tag{48}$$

Combined with (45), we have

$$\|\text{Polar}(\boldsymbol{D}^*\mathcal{P}(\tilde{\boldsymbol{X}})\tilde{\boldsymbol{X}}^{(s)\top}) - \boldsymbol{D}^*\|_F \leq \frac{\kappa(\mathcal{P}(\tilde{\boldsymbol{X}}))^4}{2} \frac{\left\|\mathcal{P}(\tilde{\boldsymbol{X}})\tilde{\boldsymbol{E}}^{(s)\top} - \tilde{\boldsymbol{E}}^{(s)}\mathcal{P}(\tilde{\boldsymbol{X}})^\top\right\|_F}{\|\mathcal{P}(\tilde{\boldsymbol{X}})\|_2} + O(\|\tilde{\boldsymbol{E}}^{(s)}\|_F^2)$$
$$\leq (\frac{10}{9}\left(\frac{11}{9}\sqrt{\theta} + \frac{1}{10}\right)\kappa(\mathcal{P}(\tilde{\boldsymbol{X}}))^4 + O(\|\tilde{\boldsymbol{E}}^{(s)}\|_F))\|\tilde{\boldsymbol{E}}^{(s)}\|_F \tag{49}$$

The above inequality combined with $\kappa(\mathcal{P}(\tilde{\boldsymbol{X}})) \leq 121/79$ (due to (38)), $\theta < 1/5600$, and $\|\tilde{\boldsymbol{E}}^{(s)}\|_F \leq O(1/\log n)$ leads to

$$\|\text{Polar}(\boldsymbol{D}^*\mathcal{P}(\tilde{\boldsymbol{X}})\tilde{\boldsymbol{X}}^{(s)\top}) - \boldsymbol{D}^*\|_F \leq \frac{41}{50}\|\tilde{\boldsymbol{E}}^{(s)}\|_F, \tag{50}$$

with probability of at least $1 - 2\exp(\log n - \log^2 n)$. Therefore, we have a bound for the second term of (36). As a result, (36) can be rewritten as

$$\|\boldsymbol{D}^{(s+1)} - \boldsymbol{D}^*\|_F \leq \frac{41}{50}\|\tilde{\boldsymbol{E}}^{(s)}\|_F + \frac{12\|\boldsymbol{X}^* - \mathcal{P}(\tilde{\boldsymbol{X}})\|_F}{\|\boldsymbol{X}^*\|_2}$$
$$\overset{\text{Eq34}}{\leq} \frac{369}{400}\|\boldsymbol{D}^{(s)} - \boldsymbol{D}^*\|_F + \frac{12\|\boldsymbol{X}^* - \mathcal{P}(\tilde{\boldsymbol{X}})\|_F}{\|\boldsymbol{X}^*\|_2}$$
$$\leq \frac{369}{400}\|\boldsymbol{D}^{(s)} - \boldsymbol{D}^*\|_F + 12\|\boldsymbol{P}^{(s)}\boldsymbol{A}^* - \boldsymbol{D}^*\|_F \tag{51}$$

This completes the proof of the induction step (23). Finally, we use (51) to complete the proof of Theorem 3. This goal, notice that

$$\|\boldsymbol{A}^{(s+1)} - \boldsymbol{A}^*\|_F = \|\boldsymbol{P}^{(s+1)-1}\boldsymbol{D}^{(s+1)} - \left(\mathcal{L}((\boldsymbol{A}^*\boldsymbol{A}^{*\top})^{-1})^\top\right)^{-1}\boldsymbol{D}^*\|_F$$
$$\leq \|\boldsymbol{D}^{(s+1)} - \boldsymbol{D}^*\|_F + \|\boldsymbol{P}^{(s+1)-1} - \left(\mathcal{L}((\boldsymbol{A}^*\boldsymbol{A}^{*\top})^{-1})^\top\right)^{-1}\|_F$$
$$\leq \|\boldsymbol{D}^{(s+1)} - \boldsymbol{D}^*\|_F + O\left(\|\boldsymbol{P}^{(s+1)} - \mathcal{L}\left(\left(\boldsymbol{A}^*\boldsymbol{A}^{*\top}\right)^{-1}\right)^\top\|_F\right). \tag{52}$$

Via a similar argument, we can write

$$\|\boldsymbol{D}^{(s)} - \boldsymbol{D}^*\|_F \leq \|\boldsymbol{A}^{(s)} - \boldsymbol{A}^*\|_F + O\left(\|\boldsymbol{P}^{(s)} - \mathcal{L}\left(\left(\boldsymbol{A}^*\boldsymbol{A}^{*\top}\right)^{-1}\right)^\top\|_F\right). \tag{53}$$

Finally, we can use Lemma 3 to bound $\|\boldsymbol{P}^{(s)}\boldsymbol{A}^* - \boldsymbol{D}^*\|_F$, $\left\|\boldsymbol{P}^{(s)} - \mathcal{L}\left(\left(\boldsymbol{A}^*\boldsymbol{A}^{*\top}\right)^{-1}\right)^\top\right\|_F$, and $\left\|\boldsymbol{P}^{(s+1)} - \mathcal{L}\left(\left(\boldsymbol{A}^*\boldsymbol{A}^{*\top}\right)^{-1}\right)^\top\right\|_F$ with $O\left(\frac{\hat{\kappa}^6}{\theta}\sqrt{\frac{n}{p_1+t}}\right)$. This completes the proof. □

B.3 Proof of Theorem 2

It is easy to see that Theorem 2 is a special case of Theorem 3, where we do not perform preconditioner update and use one pre-calculated preconditioner throughout. By recalling the proof of Theorem 3 we immediately have

$$\|\boldsymbol{D}^{(s+1)} - \boldsymbol{D}^*\|_F \leq \frac{369}{400}\|\boldsymbol{D}^{(s)} - \boldsymbol{D}^*\|_F + 12\|\boldsymbol{P}\boldsymbol{A}^* - \boldsymbol{D}^*\|_F, \tag{54}$$

for iterates of Algorithm 2. Based on our induction, we have

$$\begin{aligned}
\|\boldsymbol{D}^{(t+1)}-\boldsymbol{D}^*\|_F &\leq \frac{369}{400}\|\boldsymbol{D}^{(t)}-\boldsymbol{D}^*\|_F+12\|\boldsymbol{P}\boldsymbol{A}^*-\boldsymbol{D}^*\|_F \\
&\leq \left(\frac{369}{400}\right)^t\|\boldsymbol{D}^{(0)}-\boldsymbol{D}^*\|_F+\left(\frac{369}{400}+\ldots+\left(\frac{369}{400}\right)^t\right)12\|\boldsymbol{P}\boldsymbol{A}^*-\boldsymbol{D}^*\|_F \\
&\leq \left(\frac{369}{400}\right)^t\|\boldsymbol{D}^{(0)}-\boldsymbol{D}^*\|_F+\frac{12}{1-\frac{369}{400}}\|\boldsymbol{P}\boldsymbol{A}^*-\boldsymbol{D}^*\|_F \\
&= \left(\frac{369}{400}\right)^t\|\boldsymbol{D}^{(0)}-\boldsymbol{D}^*\|_F+\frac{4800\|\boldsymbol{P}\boldsymbol{A}^*-\boldsymbol{D}^*\|_F}{31}. \qquad (55)
\end{aligned}$$

Based on the above inequality, one can use the same argument as in (52) and (53) to establish the linear convergence of $\boldsymbol{A}^{(t)}$. This completes the proof. □

C Proof of Lemmas

In this section we provide the proofs of the lemmas we have used.

C.1 Proof of Lemma 4

Consider each column vector of $\boldsymbol{X}$ as a random vector. Upon defining $\Sigma_{\boldsymbol{X}}$ as the covariance matrix of $\boldsymbol{x}$, it can be verified that

$$\Sigma_{\boldsymbol{X}}=\theta\sigma^2\boldsymbol{I}_n. \qquad (56)$$

The diagonal entries of $\Sigma_{\boldsymbol{X}}$ are $\mathbb{E}(\boldsymbol{X}_{ij}^2)$ which are respectively equal to the product of $\mathbb{E}(B_{ij}^2)=\theta$ and σ^2. The off-diagonal entries of $\Sigma_{\boldsymbol{X}}$ are the pair-wise covariance between different entries of $\boldsymbol{X}$, which are equal 0 by the independence assumption. We now try to invoke Theorem 4 to prove Lemma 4. More precisely, for any unit-norm $\boldsymbol{z}\in\mathbb{R}^n$ and any ith column of $\boldsymbol{X}$, we have

$$\left\|\langle\boldsymbol{X}_{(\cdot,i)},\boldsymbol{z}\rangle\right\|_{\psi_2}^2=\left\|\sum_{j=1}^n\boldsymbol{X}_{(j,i)}z_j\right\|_{\psi_2}^2\lesssim\sum_{j=1}^n z_j^2\left\|\boldsymbol{X}_{(j,i)}\right\|_{\psi_2}^2\lesssim\sigma^2, \qquad (57)$$

which implies that $\left\|\langle\boldsymbol{X}_{(\cdot,i)},\boldsymbol{z}\rangle\right\|_{\psi_2}^2\leq C_1\sigma^2$ for some C_1 and

$$\left\|\langle\boldsymbol{X}_{(\cdot,i)},\boldsymbol{z}\rangle\right\|_{L^2}^2=\mathbb{E}\langle\boldsymbol{X}_{(\cdot,i)},\boldsymbol{z}\rangle^2=\boldsymbol{z}^\top\Sigma_{\boldsymbol{X}}\boldsymbol{z}=\theta\sigma^2. \qquad (58)$$

Now we are ready to invoke Theorem 4. By choosing $C_{ce}=C_1/\sqrt{\theta}$, we have for some constant C_2:

$$\left\|\frac{1}{p}\boldsymbol{X}\boldsymbol{X}^\top-\theta\sigma^2\boldsymbol{I}_n\right\|_2\leq C_2\left(\sqrt{\frac{n+u}{p}}+\frac{n+u}{p}\right)\sigma^2, \qquad (59)$$

with probability $1-2\exp(-u)$. Consider $C_p=40000C_2^2$. Upon assuming $p\geq n+u$, one can bound the right hand side by

$$C_2\left(\sqrt{\frac{n+u}{p}}+\frac{n+u}{p}\right)\sigma^2\leq 2C_2\sqrt{\frac{n+u}{p}}\sigma^2\leq\frac{\theta\sigma^2}{100}. \qquad (60)$$

Then, due to $\sqrt{a^2+b^2} \le a+b$, we have

$$\|\boldsymbol{X}\|_2 \le \sqrt{p\theta}\sigma + \frac{1}{10}\sqrt{p\theta}\sigma = \frac{11}{10}\sqrt{p\theta}\sigma. \tag{61}$$

Moreover, by $\sqrt{a^2-b^2} \le a-b$ for $a \ge b$, we have

$$\sigma_n(\boldsymbol{X}) \ge \sqrt{p\theta}\sigma - \frac{1}{10}\sqrt{p\theta}\sigma = \frac{9}{10}\sqrt{p\theta}\sigma. \tag{62}$$

The proof is complete after setting $u = \log^2 n$. □

C.2 Proof of Lemma 3

We start by noticing that

$$\frac{1}{p\theta\sigma^2}\boldsymbol{Y}\boldsymbol{Y}^\top = \boldsymbol{A}^*\left(\frac{1}{p\theta\sigma^2}\boldsymbol{X}^*\boldsymbol{X}^{*\top}\right)\boldsymbol{A}^{*\top} \approx \boldsymbol{A}^*\boldsymbol{A}^{*\top}. \tag{63}$$

The approximation $\frac{1}{p\theta\sigma^2}\boldsymbol{X}^*\boldsymbol{X}^{*\top} \approx \boldsymbol{I}_{n\times n}$ is a result of our sub-Gaussian Bernoulli model for $\boldsymbol{X}^*$, and is a direct result of Theorem 4. In particular, we have

$$\left\|\frac{1}{p}\boldsymbol{X}^*\boldsymbol{X}^{*\top} - \theta\sigma^2\boldsymbol{I}_{n\times n}\right\|_2 \lesssim \sigma^2\sqrt{\frac{n+u}{p}} \tag{64}$$

with probability at least $1-2\exp(-u)$. With the same probability, we have

$$\begin{aligned}\left\|\frac{1}{p\theta\sigma^2}\boldsymbol{Y}\boldsymbol{Y}^\top - \boldsymbol{A}^*\boldsymbol{A}^{*\top}\right\|_2 &= \left\|\boldsymbol{A}^*\left(\frac{1}{p\theta\sigma^2}\boldsymbol{X}^*\boldsymbol{X}^{*\top} - \boldsymbol{I}_{n\times n}\right)\boldsymbol{A}^{*\top}\right\|_2 \\ &\lesssim \frac{1}{\theta}\sqrt{\frac{n+u}{p}} \end{aligned}\tag{65}$$

For simplicity we set $\boldsymbol{\Delta}_1 = \frac{1}{p\theta\sigma^2}\boldsymbol{Y}\boldsymbol{Y}^\top - \boldsymbol{A}^*\boldsymbol{A}^{*\top}$. Using the Taylor expansion of the matrix inverse, we have

$$\begin{aligned}\left(\frac{1}{p\theta\sigma^2}\boldsymbol{Y}\boldsymbol{Y}^\top\right)^{-1} &= \left(\boldsymbol{A}^*\boldsymbol{A}^{*\top} + \boldsymbol{\Delta}_1\right)^{-1} \\ &= \left(\boldsymbol{A}^*\boldsymbol{A}^{*\top}\left(I_{r\times r} + \left(\boldsymbol{A}^*\boldsymbol{A}^{*\top}\right)^{-1}\boldsymbol{\Delta}_1\right)\right)^{-1} \\ &= \left(\boldsymbol{A}^*\boldsymbol{A}^{*\top}\right)^{-1} - \left(\boldsymbol{A}^*\boldsymbol{A}^{*\top}\right)^{-1}\boldsymbol{\Delta}_1\left(\boldsymbol{A}^*\boldsymbol{A}^{*\top}\right)^{-1} + O(\boldsymbol{\Delta}_1^2)\end{aligned}\tag{66}$$

As a result we have

$$\left\|\left(\frac{1}{p\theta\sigma^2}\boldsymbol{Y}\boldsymbol{Y}^\top\right)^{-1} - \left(\boldsymbol{A}^*\boldsymbol{A}^{*\top}\right)^{-1}\right\|_2 \lesssim \frac{\hat{\kappa}^4}{\theta}\sqrt{\frac{n+u}{p}}. \tag{67}$$

Similarly, by Corollary 4.8 from Bhatia (1994), we have

$$\begin{aligned}
&\left\|\mathcal{L}\left(\left(\frac{1}{p\theta\sigma^2}\boldsymbol{Y}\boldsymbol{Y}^\top\right)^{-1}\right)-\mathcal{L}\left(\left(\boldsymbol{A}^*\boldsymbol{A}^{*\top}\right)^{-1}\right)\right\|_2\\
&\lesssim \kappa\left(\left(\boldsymbol{A}^*\boldsymbol{A}^{*\top}\right)^{-1}\right)\left\|\left(\frac{1}{p\theta\sigma^2}\boldsymbol{Y}\boldsymbol{Y}^\top\right)^{-1}-\left(\boldsymbol{A}^*\boldsymbol{A}^{*\top}\right)^{-1}\right\|_2\\
&\le \frac{\hat{\kappa}^6}{\theta}\sqrt{\frac{n+u}{p}}. \qquad (68)
\end{aligned}$$

We conclude the proof by setting $u = \log^2 n$. □

C.3 Proof of Lemma 2

First we invoke Lemma 4 and get:

$$\kappa\left(\boldsymbol{X}^*\right) < \frac{11}{9} \qquad (69)$$

with probability $1-2\exp\left\{-\log^2 n\right\}$. Then, by setting $\boldsymbol{E} = \left(\boldsymbol{X}-\boldsymbol{X}^*\right)/\|\boldsymbol{X}^*\|_2$ and $\widehat{\boldsymbol{X}}^* = \boldsymbol{X}^*/\|\boldsymbol{X}^*\|_2$, we immediately have

$$\|\boldsymbol{D}-\boldsymbol{D}^*\|_F \le \frac{\kappa(\boldsymbol{X}^*)^4}{2}\left\|\widehat{\boldsymbol{X}}^*\boldsymbol{E}^\top-\boldsymbol{E}\widehat{\boldsymbol{X}}^{*\top}\right\|_F + O(\|\boldsymbol{E}\|_F^2) \qquad (70)$$

after invoking Lemma 5. Now define $\mathcal{T}(\cdot)$ to be the operator that replaces all the diagonal entries of a matrix with zeros. It is easy to see that

$$\frac{1}{2}\left\|\widehat{\boldsymbol{X}}^*\boldsymbol{E}^\top-\boldsymbol{E}\widehat{\boldsymbol{X}}^{*\top}\right\|_F \le \left\|\mathcal{T}(\widehat{\boldsymbol{X}}^*\boldsymbol{E}^\top)\right\|_F. \qquad (71)$$

The above inequality is due to the fact that the matrices on both sides have zero diagonal and for the off-diagonal entries, the bound can be verified by elementary inequality $(a-b)^2 \le 2(a^2+b^2)$. To further investigate this bound, we introduce two matrices. Let matrix $\boldsymbol{M}_k$ denote an $n\times n$ diagonal matrix of ones and a zero at location (k,k). Left multiplying $\widehat{\boldsymbol{X}}^*$ by $\boldsymbol{M}_k$ corresponds to replacing the kth row of $\widehat{\boldsymbol{X}}^*$ with zeros. Let $\tilde{\boldsymbol{M}}_k$ denote an $p\times p$ diagonal matrix that has ones at entries (i,i) for $i \in \operatorname{supp}\left(\widehat{\boldsymbol{X}}^*_{(k,\cdot)}\right)$ and zeros elsewhere. Right multiplying $\tilde{\boldsymbol{X}}^*$ by $\tilde{\boldsymbol{M}}_k$ corresponds to replacing all the columns that are zero at kth row with zeros. Now, we make the following observation:

$$\begin{aligned}
\left\|\mathcal{T}(\widehat{\boldsymbol{X}}^*\boldsymbol{E}^\top)\right\|_F &= \sqrt{\sum_{k=1}^n \left\|(\mathcal{T}(\widehat{\boldsymbol{X}}^*\boldsymbol{E}^\top))_{(\cdot,k)}\right\|_2^2}\\
&= \sqrt{\sum_{k=1}^n \left\|\left(\boldsymbol{M}_k\widehat{\boldsymbol{X}}^*\tilde{\boldsymbol{M}}_k\right)\boldsymbol{E}_{(k,\cdot)}\right\|_2^2}\\
&\le \sqrt{\sum_{k=1}^n \left\|\boldsymbol{M}_k\widehat{\boldsymbol{X}}^*\tilde{\boldsymbol{M}}_k\right\|_2^2\left\|\boldsymbol{E}_{(k,\cdot)}\right\|_2^2}\\
&\le \max_{1\le k\le n}\left\|\boldsymbol{M}_k\widehat{\boldsymbol{X}}^*\tilde{\boldsymbol{M}}_k\right\|_2\|\boldsymbol{E}\|_F. \qquad (72)
\end{aligned}$$

Notice that the second equation is based on the exact recovery of the support $\text{supp}(\boldsymbol{E}) \subseteq \text{supp}(\widehat{\boldsymbol{X}}^*)$, and that the normalization step $\widehat{\boldsymbol{X}}^* = \boldsymbol{X}^*/\|\boldsymbol{X}^*\|_2$ does not change its support. We first turn our focus onto the statistical property of $\max_{1\le k\le n}\left\|\boldsymbol{M}_k\boldsymbol{X}^*\tilde{\boldsymbol{M}}_k\right\|_2$. Define $\boldsymbol{G}^k \in \mathbb{R}^{(n-1)\times p}$ as the matrix $\boldsymbol{M}_k\boldsymbol{X}^*\tilde{\boldsymbol{M}}_k$ after removing its kth row. One can immediately see that $\|\boldsymbol{G}^k\|_2 = \|\boldsymbol{M}_k\boldsymbol{X}^*\tilde{\boldsymbol{M}}_k\|_2$. To reduce the ambiguity of notations, we assume $k = n$, which means we remove the last row. Then one should immediately notice that for $1 \le i \le n-1$, $1 \le j \le p$, each entry of $\boldsymbol{G}^k$ as a random variable has following property:

$$\mathbb{1}_{\boldsymbol{G}^k_{ij}\neq 0} = B_{kj}B_{ij} \quad \text{where} \quad B_{kj}, B_{ij} \overset{i.i.d.}{\sim} \mathcal{B}(\theta). \tag{73}$$

In short, $\boldsymbol{G}^k$ is a matrix that satisfies Assumption 1 with parameter θ^2. This means that we can invoke Lemma 4 to bound $\|\boldsymbol{G}^k\|_2$ for each k. Given $p \ge C_p(n+u)/\theta^2$, we have that for some specific k:

$$\|\boldsymbol{G}^k\|_2 \le \frac{11}{10}\sqrt{p}\theta\sigma, \tag{74}$$

with probability $1 - 2\exp(-\log^2 n)$. To bound maximal $\|\boldsymbol{G}^k\|_2$ for $1 \le k \le n$, we take the union bound and obtain

$$\max_{1\le k\le n}\left\|\boldsymbol{M}_k\boldsymbol{X}^*\tilde{\boldsymbol{M}}_k\right\|_2 = \max_{1\le k\le n}\|\boldsymbol{G}^k\|_2 \le \frac{11}{10}\sqrt{p}\theta\sigma, \tag{75}$$

with probability $1 - 2\exp(\log n - \log^2 n)$ as long as $p \gtrsim (n + \log^2 n)/\theta^2$. Combining this result with (70), we have

$$\begin{aligned}
\|\boldsymbol{D} - \boldsymbol{D}^*\|_F \le& \frac{\kappa(\boldsymbol{X}^*)^4}{2}\left\|\widehat{\boldsymbol{X}}^*\boldsymbol{E}^\top - \boldsymbol{E}\widehat{\boldsymbol{X}}^{*\top}\right\|_F + O(\|\boldsymbol{E}\|_F^2)\\
\le& \frac{\kappa(\boldsymbol{X}^*)^4 \max_{1\le k\le n}\left\|\boldsymbol{M}_k\boldsymbol{X}^*\tilde{\boldsymbol{M}}_k\right\|_2}{\|\boldsymbol{X}^*\|_2}\|\boldsymbol{E}\|_F + O(\|\boldsymbol{E}\|_F^2)\\
\overset{(a)}{\le}& \frac{\left(\frac{11}{9}\right)^4 \frac{11}{10}\sqrt{p}\theta\sigma}{\frac{9}{10}\sqrt{p\theta}\sigma}\|\boldsymbol{E}\|_F + O(\|\boldsymbol{E}\|_F^2)\\
\le& \left(\left(\frac{11}{9}\right)^5\sqrt{\theta} + O(\|\boldsymbol{E}\|_F)\right)\|\boldsymbol{E}\|_F\\
<& 2\left(\frac{11}{9}\right)^5\sqrt{\theta}\|\boldsymbol{E}\|_F\\
<& 6\sqrt{\theta}\|\boldsymbol{E}\|_F,
\end{aligned} \tag{76}$$

with probability $1 - 2\exp(\log(n+1) - \log^2 n)$. Inequality (a) is due to Lemma 4 and (75). This completes the proof. □

C.4 Proof of Lemma 1

We consider one entry $\left(\boldsymbol{D}^\top\boldsymbol{Y}\right)_{ij}$ of $\boldsymbol{D}^\top\boldsymbol{Y}$, which can be written as

$$\left(\boldsymbol{D}^\top\boldsymbol{Y}\right)_{ij} = \langle \boldsymbol{D}_{(\cdot,i)}, \boldsymbol{Y}_{(\cdot,j)}\rangle = \langle \boldsymbol{D}_{(\cdot,i)}, \boldsymbol{D}^*_{(\cdot,i)}\rangle \boldsymbol{X}^*_{ij} + \sum_{k\neq i}\langle \boldsymbol{D}_{(\cdot,i)}, \boldsymbol{D}^*_{(\cdot,k)}\rangle \boldsymbol{X}^*_{kj}. \tag{77}$$

The first term can be decomposed as

$$\begin{aligned}\langle \boldsymbol{D}_{(\cdot,i)}, \boldsymbol{D}^*_{(\cdot,i)}\rangle \boldsymbol{X}^*_{ij} &= \langle \boldsymbol{D}_{(\cdot,i)}, \boldsymbol{D}_{(\cdot,i)}\rangle \boldsymbol{X}^*_{ij} + \langle \boldsymbol{D}_{(\cdot,i)}, \boldsymbol{D}^*_{(\cdot,i)} - \boldsymbol{D}_{(\cdot,i)}\rangle \boldsymbol{X}^*_{ij} \\ &= \boldsymbol{X}^*_{ij} + \langle \boldsymbol{D}_{(\cdot,i)}, \boldsymbol{D}^*_{(\cdot,i)} - \boldsymbol{D}_{(\cdot,i)}\rangle \boldsymbol{X}^*_{ij}. \end{aligned} \tag{78}$$

As a result when $\boldsymbol{X}^*_{ij} = 0$, the first term is equal to zero. When $\boldsymbol{X}^*_{ij} \neq 0$, the absolute value of the first term is lower bounded by $(1 - O(\frac{1}{\log n}))\Gamma \geq \frac{3\Gamma}{4}$. As for the second term, it is a sum of sub-Gaussian random variable. Its mean is zero since $\boldsymbol{X}^*_{kj}$ has zero mean. On the other hand, the variance of the second term can be bounded as:

$$\begin{aligned}\mathrm{Var}\left(\sum_{k\neq i}\langle \boldsymbol{D}_{(\cdot,i)}, \boldsymbol{D}^*_{(\cdot,k)}\rangle \boldsymbol{X}^*_{kj}\right) &\overset{(a)}{=} \theta \sum_{k\neq i}\langle \boldsymbol{D}_{(\cdot,i)}, \boldsymbol{D}^*_{(\cdot,k)}\rangle^2\sigma^2 \\ &\overset{(b)}{=} \theta \sum_{k\neq i} 2\left(\langle \boldsymbol{D}_{(\cdot,i)}, \boldsymbol{D}^*_{(\cdot,k)}\rangle^2 + \langle \boldsymbol{D}^*_{(\cdot,i)} - \boldsymbol{D}_{(\cdot,i)}, \boldsymbol{D}^*_{(\cdot,k)}\rangle^2\right)\sigma^2 \\ &\leq 2\theta \|\boldsymbol{D}^*\|_2^2 \|\boldsymbol{D}^*_{(\cdot,i)} - \boldsymbol{D}_{(\cdot,i)}\|_2^2\sigma^2 \\ &\leq 2\theta \|\boldsymbol{D}^*\|_2^2 \|\boldsymbol{D}^* - \boldsymbol{D}\|_{2,\infty}^2\sigma^2 \\ &\leq \frac{\theta\sigma^2}{8\log^2 n}. \end{aligned} \tag{79}$$

Here (a) is due to the fact that $\boldsymbol{X}^*_{kj}$ has variance $\theta\sigma^2$ and (b) is due to elementary inequality $(a+b)^2 \leq 2(a^2+b^2)$. As a result we can see that the term $\sum_{k\neq i}\langle \boldsymbol{D}_i, \boldsymbol{D}^*_k\rangle \boldsymbol{X}^*_{kj}$ is a sub-Gaussian random variable with $O(1/\log n)$ variance. To ensure exact support recovery, we need to make sure $\left|\sum_{k\neq i}\langle \boldsymbol{D}_{(\cdot,i)}, \boldsymbol{D}^*_{(\cdot,k)}\rangle \boldsymbol{X}^*_{kj}\right|$ never exceeds $\frac{\Gamma}{4}$ for every entry of $\boldsymbol{X}^*$. This can be done by noting that

$$\begin{aligned}&\mathbb{P}\left(\left|\sum_{k\neq i}\langle \boldsymbol{D}_{(\cdot,i)}, \boldsymbol{D}^*_{(\cdot,k)}\rangle \boldsymbol{X}^*_{kj}\right| \leq \Gamma/4 \text{ for all } (i,j)\right) \\ &\geq 1 - 2pn\exp\left\{-\frac{C(\log n)^2}{\theta\sigma^2\Gamma^2}\right\} \\ &= 1 - 2\exp\left\{\log(pn) - \frac{C(\log n)^2}{\theta\sigma^2\Gamma^2}\right\}. \end{aligned} \tag{80}$$

Therefore, with probability $1 - 2\exp\left\{\log(pn) - \frac{C(\log n)^2}{\theta\sigma^2\Gamma^2}\right\}$, we have that $\left|\left(\boldsymbol{D}^\top \boldsymbol{Y}\right)_{ij}\right| < \frac{\Gamma}{4}$ when $\boldsymbol{X}^*_{ij} = 0$. Similarly, when $\boldsymbol{X}^*_{ij} \neq 0$, we have $\left|\left(\boldsymbol{D}^\top \boldsymbol{Y}\right)_{ij}\right| \geq \frac{3\Gamma}{4} - \frac{\Gamma}{4} = \frac{\Gamma}{2}$. Therefore, hard-thresholding the elements of $\boldsymbol{D}^\top \boldsymbol{Y}$ at the level $\frac{\Gamma}{2}$ will recover the sparsity pattern of $\boldsymbol{X}^*$.

D Rank-one Updates for the Preconditioner

Our goal in this section is to explain how to obtain the preconditioner $\boldsymbol{P}^{(t)} = \mathcal{L}\left(\left(\boldsymbol{Y}\boldsymbol{Y}^\top + \boldsymbol{y}\boldsymbol{y}^\top\right)^{-1}\right)$ based on $\boldsymbol{P}^{(t-1)}\mathcal{L}\left(\left(\boldsymbol{Y}\boldsymbol{Y}^\top\right)^{-1}\right)$ and $\left(\boldsymbol{Y}\boldsymbol{Y}^\top\right)^{-1}$ within $O(n^2)$ operations. Recall the Sherman-Morrison formula:

$$\left(\boldsymbol{Y}\boldsymbol{Y}^\top + \boldsymbol{y}\boldsymbol{y}^\top\right)^{-1} = \left(\boldsymbol{Y}\boldsymbol{Y}^\top\right)^{-1} - \boldsymbol{v}\boldsymbol{v}^\top, \quad \text{where} \quad \boldsymbol{v} = \left(1 + \boldsymbol{y}^\top\left(\boldsymbol{Y}\boldsymbol{Y}^\top\right)^{-1}\boldsymbol{y}\right)^{-1/2}\left(\boldsymbol{Y}\boldsymbol{Y}^\top\right)^{-1}\boldsymbol{y}.$$

Therefore, given $\left(\boldsymbol{Y}\boldsymbol{Y}^\top\right)^{-1}$, $\left(\boldsymbol{Y}\boldsymbol{Y}^\top + \boldsymbol{y}\boldsymbol{y}^\top\right)^{-1}$ can be obtained in $O(n^2)$ operations. Given the above rank-one update, Krause and Igel (2015) introduced a triangular rank-one update to obtain the Cholesky factorization of $\left(\boldsymbol{Y}\boldsymbol{Y}^\top + \boldsymbol{y}\boldsymbol{y}^\top\right)^{-1}$. We adapt this triangular rank-one to our setting and present an algorithm that produces $\boldsymbol{P}^{(t)}$ and $\left(\boldsymbol{Y}\boldsymbol{Y}^\top + \boldsymbol{y}\boldsymbol{y}^\top\right)^{-1}$ based on $\boldsymbol{P}^{(t-1)}$ and $\left(\boldsymbol{Y}\boldsymbol{Y}^\top\right)^{-1}$ within $O(n^2)$ operations. For simplicity, we define

$$\boldsymbol{A} = \left(\boldsymbol{Y}\boldsymbol{Y}^\top\right)^{-1}, \quad \boldsymbol{L} = \boldsymbol{P}^{(t-1)}, \quad \boldsymbol{A}' = \left(\boldsymbol{Y}\boldsymbol{Y}^\top + \boldsymbol{y}\boldsymbol{y}^\top\right)^{-1}, \quad \boldsymbol{L}' = \boldsymbol{P}^{(t)}.$$

Algorithm 5 Rank-one update for the preconditioner

Input: $\boldsymbol{L}$, $\boldsymbol{A}$, $\boldsymbol{y}$.
Calculate $\boldsymbol{v} = \frac{\boldsymbol{A}\boldsymbol{y}}{\sqrt{1+\boldsymbol{y}^\top \boldsymbol{A}\boldsymbol{y}}}$.
Set $\boldsymbol{A}' = \boldsymbol{A} - \boldsymbol{v}\boldsymbol{v}^\top$.
Set $\boldsymbol{L} = \frac{\boldsymbol{L}}{\sqrt{(p_1+t)\theta\sigma^2}}$.
Set $\boldsymbol{w} = \boldsymbol{v}$.
Set $b = 1$.
for $j = 1, \ldots, n$ **do**
    Set $\boldsymbol{L}'_{jj} = \sqrt{\boldsymbol{L}_{jj}^2 - \frac{1}{b}\boldsymbol{w}_j}$.
    Set $\gamma = \boldsymbol{L}_{jj}^2 b - \boldsymbol{w}_j^2$.
    for $k = j+1, \ldots, n$ **do**
        Set $\boldsymbol{w}_k = \boldsymbol{w}_k - \frac{\boldsymbol{w}_j}{\boldsymbol{L}_{jj}}\boldsymbol{L}_{kj}$.
        Set $\boldsymbol{L}'_{kj} = \frac{\boldsymbol{L}'_{jj}}{\boldsymbol{L}_{jj}}\boldsymbol{L}_{kj} - \frac{\boldsymbol{L}'_{jj}\boldsymbol{w}_j}{\gamma}\boldsymbol{w}_k$.
    end for
    Set $b = b - \frac{\boldsymbol{w}_j^2}{\boldsymbol{L}_{jj}^2}$
end for
Set $\boldsymbol{L}' = \sqrt{(p_1+t+1)\theta\sigma^2}\boldsymbol{L}'$.
return $\boldsymbol{L}'$ and $\boldsymbol{A}'$.

Finally, we note that the above rank-one update algorithm for Cholesky decomposition is already implemented in MATLAB function `cholupdate`.