SIAM J. MATH. DATA SCI.
Vol. 7, No. 3, pp. 855–883

Simple Alternating Minimization Provably Solves Complete Dictionary Learning*

Geyu Liang[†], Gavin Zhang[‡], Salar Fattahi[†§], and Richard Y. Zhang[‡]

Abstract. This paper focuses on the *noiseless complete dictionary learning problem,* where the goal is to represent a set of given signals as linear combinations of a small number 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 propose a simple and efficient algorithm that provably recovers the ground truth when applied to the nonconvex and discrete formulation of the problem in the noiseless setting. We also extend our proposed method to mini-batch and online settings where the data is huge-scale or arrives continuously over time. At the core of our proposed method lies an efficient preconditioning technique that transforms the unknown dictionary to a near-orthonormal one, for which we prove a simple alternating minimization technique converges linearly to the ground truth under minimal conditions. Our numerical experiments on synthetic and real datasets showcase the superiority of our method compared with the existing techniques.

Key words. nonconvex optimization, dictionary learning, online algorithms

MSC codes. 90C26, 90C06, 90C30, 68W27

DOI. 10.1137/23M1568120

1. Introduction.

The dictionary learning problem seeks to represent data as a linear combination of a small number of basis elements, known as *atoms,* which are learned from the data itself. The resulting collection of atoms, known as the *dictionary,* can then be used for a variety of signal processing tasks [28, 25], often as a plug-in replacement for classical bases based on cosines, wavelets, or Gabor filters. The advantage of a learned dictionary over these classical bases is that it is tuned to the input dataset and can therefore provide a sparser representation for each input signal that uses fewer atoms.

The problem of learning a dictionary that provides the *sparsest* representation is nonconvex and highly difficult to solve, both in theory and in practice. In this paper, we focus on learning a *complete* dictionary, whose number of atoms matches the dimensionality of the data, via the following optimization problem:

*Received by the editors April 24, 2023; accepted for publication (in revised form) February 25, 2025; published electronically July 1, 2025.
https://doi.org/10.1137/23M1568120

Funding: Financial support for this work was provided by NSF CAREER Award CCF-2337776, 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.

[†]Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI 48109 USA (lianggy@umich.edu, fattahi@umich.edu).
[‡]Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA (jialun2@illinois.edu, ryz@illinois.edu).
[§]Corresponding author.

$$\text{(CDL)} \qquad \min_{\boldsymbol{X},\boldsymbol{D}} \|\boldsymbol{Y}-\boldsymbol{D}\boldsymbol{X}\|_F^2+\zeta\|\boldsymbol{X}\|_0.$$

Here, $\boldsymbol{Y}\in\mathbb{R}^{n\times p}$ is a data matrix whose columns are observed signals, and our goal is to find a *square* dictionary matrix $\boldsymbol{D}\in\mathbb{R}^{n\times n}$ and corresponding sparse code $\boldsymbol{X}\in\mathbb{R}^{n\times p}$ to approximately represent $\boldsymbol{Y}$ while minimizing its ℓ_0-pseudonorm (denoted as $\|\cdot\|_0$) which counts its number of nonzero elements.

The ability to solve (CDL) to optimality would allow us to strike the best-possible trade-off between the sparsity of the representation and its quality of fit (according to the preference expressed by the parameter $\zeta>0$). Unfortunately, the ℓ_0-pseudonorm presents a significant difficulty, as it is not only nonconvex, but also discrete and combinatorial. Even rigorously verifying that a learned dictionary is first-order locally optimal could require exhaustively enumerating all 2^{np} possible sparsity patterns. It is therefore understandable that the most widely used heuristics like KSVD [2] and MOD [12] do not actually guarantee convergence to an optimal solution, not even when provided with a near-optimal initial guess. Instead, the state-of-the-art for provable optimality relies on relaxing the combinatorial ℓ_0-pseudonorm into the convex ℓ_1-norm [1] or the convex ℓ_4-norm [43]. Under assumptions like mutual incoherency [4], restricted isometry property (RIP) [1], or large sample complexity [40, 43], the relaxation is exact and an optimal solution to the original combinatorial problem can be recovered. However, when the relaxation is inexact, the recovered solution may be drastically different from the solution to the original (CDL).

In this paper, we revisit the combinatorial ℓ_0-pseudonorm in (CDL), motivated by the fact that the *orthogonal* instance (in which the dictionary $\boldsymbol{D}$ is constrained to be orthonormal) is significantly easier to analyze. In the orthogonal setting, alternating minimization is *simple,* in that the minimizations over $\boldsymbol{D}$ and $\boldsymbol{X}$ at each iteration have simple closed-form solutions. We give the first proof that the resulting sequence converges to the ground truth model under minimal conditions, provided that the algorithm is initialized properly. To extend to general complete dictionaries, which are not necessarily orthogonal, our key idea is to use a data-driven preconditioning step to "orthogonalize" the data. Surprisingly, this preconditioning step allows us to extend our linear convergence guarantee to general complete dictionary learning without significant modifications. As shown in Figure 1, the resulting preconditioned algorithm learns more powerful dictionaries that strike a better balance between the sparsity of representation and the quality of fit than the commonly used heuristic KSVD.

1.1. Summary of results.

At the heart of our method to recover a pair $(\boldsymbol{D}^*,\boldsymbol{X}^*)$ is an efficient alternating minimization algorithm for the *orthonormal* dictionary learning problem, which reads

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

where $\mathbb{O}(n)$ denotes the set of $n\times n$ orthonormal matrices. While the orthogonality constraint adds another nonconvex component to the original formulation, we point out that the minimization of $\boldsymbol{X}$ for a fixed orthonormal $\boldsymbol{D}\in\mathbb{O}(n)$ has a cheap closed-form solution,

$$\operatorname*{argmin}_{\boldsymbol{X}} \|\boldsymbol{Y}-\boldsymbol{D}\boldsymbol{X}\|_F^2+\zeta\|\boldsymbol{X}\|_0=\mathrm{HT}_\zeta(\boldsymbol{D}^T\boldsymbol{Y}),$$

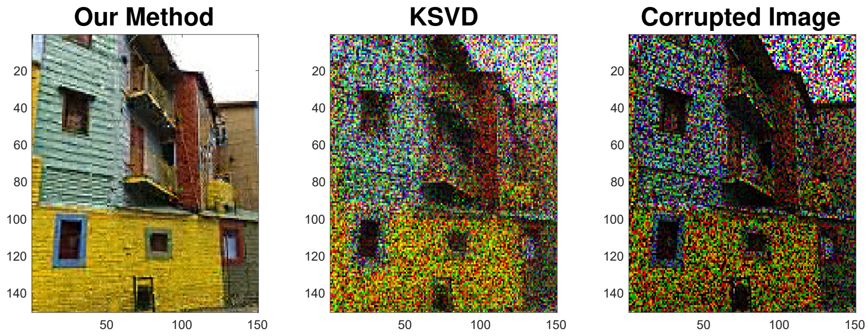

Figure 1. A comparison of image denoising using dictionaries learned via our proposed method (Algorithm 2.3) and via KSVD. *We choose a random landscape image and artificially corrupt* 50% *of the pixels. Reconstruction is done via orthogonal matching pursuit with the learned dictionaries. The corrupted original image is shown on the right, and the two reconstructed images are shown on the left. We see that the dictionary learned via our method achieves a much better denoising result than one learned via KSVD. We refer the reader to section* 4.2 *for the details of setup.*

via a hard-thresholding operator $\mathrm{HT}_\zeta(\cdot)$ at level ζ, which is defined as

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

Similarly, the minimization of an orthonormal $\boldsymbol{D} \in \mathbb{O}(n)$ for a fixed $\boldsymbol{X}$ also has a well-known closed-form solution via the so-called orthogonal Procrustes problem

$$\underset{\boldsymbol{D}\in\mathbb{O}(n)}{\operatorname{argmin}} \|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{X}\|_F^2 + \zeta\|\boldsymbol{X}\|_0 = \operatorname{Polar}(\boldsymbol{Y}\boldsymbol{X}^\top),$$

where $\operatorname{Polar}(\boldsymbol{A}) = \boldsymbol{U}_{\boldsymbol{A}}\boldsymbol{V}_{\boldsymbol{A}}^\top$ and $\boldsymbol{U}_{\boldsymbol{A}}\boldsymbol{\Sigma}_{\boldsymbol{A}}\boldsymbol{V}_{\boldsymbol{A}}^\top$ is the singular value decomposition (SVD) of $\boldsymbol{A}$. Therefore, the iterations

$$\boldsymbol{X}^{(t+1)} = \mathrm{HT}_\zeta(\boldsymbol{D}^{(t)\top}\boldsymbol{Y}), \qquad \boldsymbol{D}^{(t+1)} = \operatorname{Polar}(\boldsymbol{Y}\boldsymbol{X}^{(t+1)\top}) \tag{1.1}$$

correspond exactly to an alternating minimization solution of problem (ODL). While (1.1) has been derived before, we give the first rigorous proof that the sequence locally converges at a linear rate with minimal conditions on the sparsity rate of $\boldsymbol{X}^*$. Concretely, we prove that when the data are generated as $\boldsymbol{Y} = \boldsymbol{D}^*\boldsymbol{X}^*$, with $\boldsymbol{D}^*$ an orthogonal dictionary and $\boldsymbol{X}^*$ a sparse random matrix, (1.1) converges linearly to a ground truth $(\boldsymbol{D}^*, \boldsymbol{X}^*)$, when it is initialized within an $O(\frac{1}{\sigma\sqrt{\theta n}})$ Frobenius neighborhood of this solution.

We use a data-driven preconditioning step to generalize (1.1) to general instances of (CDL), for which the square dictionary matrix $\boldsymbol{D}^*$ is not necessarily orthonormal. The basic idea is to learn an orthonormal dictionary $\tilde{\boldsymbol{D}}$ with respect to the preconditioned data matrix $\tilde{\boldsymbol{Y}} = (\boldsymbol{Y}\boldsymbol{Y}^\top)^{-1/2}\boldsymbol{Y}$, and then to output a complete dictionary $\boldsymbol{D} = (\boldsymbol{Y}\boldsymbol{Y}^\top)^{+1/2}\tilde{\boldsymbol{D}}$ by reversing the preconditioning. Indeed, if the data are generated as $\boldsymbol{Y} = \boldsymbol{D}^*\boldsymbol{X}^*$ with $\boldsymbol{D}^*$ deterministic and $\boldsymbol{X}^*$ a sparse random matrix satisfying $\mathbb{E}[\boldsymbol{X}^*\boldsymbol{X}^{*\top}] = I_n$, then $\mathbb{E}[\boldsymbol{Y}\boldsymbol{Y}^\top] = \boldsymbol{D}^*\mathbb{E}[\boldsymbol{X}^*\boldsymbol{X}^{*\top}]\boldsymbol{D}^{*\top} = \boldsymbol{D}^*\boldsymbol{D}^{*\top}$. Therefore, the preconditioned data are generated as $\tilde{\boldsymbol{Y}} \approx \tilde{\boldsymbol{D}}^*\boldsymbol{X}^*$ with respect to an orthogonal dictionary $\tilde{\boldsymbol{D}}^* = (\boldsymbol{D}^*\boldsymbol{D}^{*\top})^{+1/2}\boldsymbol{D}^*$ and the same sparse code $\boldsymbol{X}^*$. We show that such preconditioning can be efficiently implemented via low-rank updates of the corresponding Cholesky factors. One of our main contributions is to show that this simple preconditioning step allows us to extend our linear convergence guarantee from orthogonal dictionaries (ODL) to general complete dictionaries (CDL), even in the online and mini-batch settings.

A key strength of our convergence guarantee is that it does not rely on any incoherency/RIP assumptions. In fact, we can recover *very ill-conditioned* dictionaries that are highly coherent, although at the expense of a higher sample complexity. Moreover, we can recover codes with sparsity levels in the order of $O(n)$; in contrast, existing methods based on alternating minimization can handle sparsity levels of at most $O(n^\alpha)$ for some $\alpha \leq 1/2$. Another technique that achieves a near-linear sparsity level $O(n^{1-\gamma})$ for $\gamma > 0$ is based on *sum-of-squares hierarchy* [8], which is indeed inefficient in practice. Using Riemannian optimization techniques, [39] was the first to show that (CDL) with linear sparsity level can be solved in polynomial time. However, the Riemannian optimization algorithm can be expensive, and the associated sample complexity has a dependency of $\Omega(n^5)$ for orthogonal dictionaries and $\Omega(n^7)$ for complete dictionaries [39].

In practice, the iterations (1.1) admit highly efficient implementations, which can fully take advantage of the massive parallelism inherent in modern hardware. The hard-thresholding operator is embarrassingly parallel, while the SVD operation that constitutes the polar operator can be implemented using hardware-optimized LAPACK implementations. The most expensive part of (1.1) is actually its need to iterate and sum over all p columns of the data matrix $\boldsymbol{Y}$ at every iteration. To address potential scalability issues with a very large p, we propose a mini-batch version,

$$\boldsymbol{X}^{(t+1)}_{(\cdot,i)} = \mathrm{HT}_\zeta(\boldsymbol{D}^{(t)\top}\boldsymbol{Y}_{(\cdot,i)}) \quad \forall i \in \Omega, \qquad \boldsymbol{D}^{(t+1)} = \mathrm{Polar}\left(\sum_{i\in\Omega}\boldsymbol{Y}_{(\cdot,i)}\boldsymbol{X}^{(t+1)\top}_{(\cdot,i)}\right), \tag{1.2}$$

that approximates $\boldsymbol{Y}\boldsymbol{X}^{(t+1)\top} \approx \sum_{i\in\Omega}\boldsymbol{Y}_{(\cdot,i)}\boldsymbol{X}^{(t+1)\top}_{(\cdot,i)}$ over a mini-batch $\Omega \subseteq \{1,2,\ldots,p\}$ and updates only the sparse codes $\boldsymbol{X}^{(t+1)}_{(\cdot,i)}$ associated with the mini-batch samples $i \in \Omega$. We show that the preconditioner can also be incrementally updated via an efficient low-rank update formula for the Cholesky factor, which is comparable to the classical Sherman–Morrison–Woodbury formula for updating the determinant/matrix inverse. We show that the accuracy of the recovered dictionary is explicitly lower bounded by the statistical error of the preconditioner. We also show that such an error will diminish in the online setting as more samples come in. This is the first time that efficient updating is applied to methods that use the preconditioning for (CDL).

Despite its mini-batch nature, we show that our method enjoys the same sharp linear convergence to the true solution. Finally, we extend our technique to the *online dictionary learning,* where the data samples arrive sequentially over time. To the best of our knowledge, existing methods for online dictionary learning work with batch sizes of at least $O(n)$ (see, e.g., [32, Theorem 1] and [3, Theorem 2]). In contrast, our proposed preconditioner admits an efficient triangular rank-one update with as few as one new sample, thereby making it particularly appealing in the online setting.

1.2. Related work.

Optimality conditions. As the first question, we must first verify whether a ground truth $(\boldsymbol{D}^*, \boldsymbol{X}^*)$ is a global minimizer of (CDL) for the generative model $\boldsymbol{Y} = \boldsymbol{D}^*\boldsymbol{X}^*$. Of course, if this were not the case, solving (CDL)—even to global optimality—may not be enough to recover a ground truth $(\boldsymbol{D}^*, \boldsymbol{X}^*)$. A series of work [18, 14, 20, 17, 34, 11, 19] studied the local optimality of the ground truth for (CDL) by replacing ℓ_0-norm with ℓ_1-norm. The work [38] elegantly showed that the ground truth is the unique global minimizer to (CDL) when $\zeta \to 0$. Specifically, when $p = \Omega(n \log n)$, they showed that any ground truth $(\boldsymbol{D}^*, \boldsymbol{X}^*)$ 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: how to design a provable algorithm for recovering $(\boldsymbol{X}^*, \boldsymbol{D}^*)$.

Alternating minimization. The empirical success achieved by methods like MOD and KSVD has encouraged the emergence of many alternating minimization algorithms [23, 27, 7]. Towards providing a provable guarantee for alternating minimization, it is common to replace the ℓ_0-pseudonorm with its convex surrogate ℓ_1-norm [1, 10, 29, 41]. Such a compromise is due to the prohibitive computational cost and formidable analytical challenges of the ℓ_0- pseudonorm. However, ℓ_1 relaxation is biased towards solutions with smaller entries, thereby leading to an inferior sparsity level [44]. On the algorithmic side, methods based on ℓ_1-relaxation need to solve variants of the LASSO problem at each iteration either exactly [1, 10] or approximately via an automatic differentiation with backpropagation [29, 41]. In this work, we focus on the formulation of (CDL) without any convex relaxation, which is the original intention of sparse dictionary learning. Unlike the ℓ_1 method, the theoretical underpinnings of the alternating algorithms for the ℓ_0 formulation are far less explored. In [35, 19, 36, 30], the authors study the convergence behavior of a variant of KSVD. Specifically, the dictionary update step is performed by maximizing the absolute norm of the S-largest responses where S is the number of nonzero entries in each signal. In [33], authors propose an alternating scheme based on sorting the nonzero entries of sparse codes. However, the success of these algorithms relies on restrictive generative models like symmetric decaying [34] and fixed sparsity S [33].

Riemannian manifold optimization for complete dictionary learning. One line of work focuses on solving (CDL) via Riemannian manifold optimization techniques [31, 43]. Notably, the work [40] was the first to show that a smoothed variant of (CDL) based on ℓ_1-relaxation is devoid of spurious local solutions. Similar benign landscape results have also been independently discovered in the analysis of other problems like matrix factorization [13]. As a result, one can provably recover one dictionary atom at a time via the Riemannian trust-region method. The work [16] showed that similar results can be achieved by first-order methods on the Riemannian manifold. In a different approach, the work [43] studied the ℓ_4-maximization on the Steifel manifold, which is further generalized in [37] to ℓ_p-maximization. Akin to our own work, preconditioning the data matrix plays an important role in these algorithms.

Online dictionary learning. Another line of research has focused on the online variants of (CDL). The work [27] first proposed an online algorithm that solves ℓ_1-relaxation of (CDL), but it only guarantees convergence to a critical point. The work [26] enhanced the usability and practicality of ℓ_1-relaxation but provided no theoretical guarantees. The work [3] proposed an alternating algorithm that can recover the ground truth $(\boldsymbol{X}^*, \boldsymbol{D}^*)$, which is further improved in [32]. However, both variants have a specific requirement on the size of the input for each iteration and cannot deal with situations where samples are received one by one.

1.3. Notation. We use $\boldsymbol{A}_{(i,\cdot)}$, $\boldsymbol{A}_{(\cdot,j)}$, and $\boldsymbol{A}_{ij}$ to denote the ith row, jth column and (i,j)th entry of $\boldsymbol{A}$, respectively. We use $\|\boldsymbol{A}\|_2$ to denote the spectral norm of $\boldsymbol{A}$, $\|\boldsymbol{A}\|_F$ to denote the Frobenius norm of $\boldsymbol{A}$, $\|\boldsymbol{A}\|_{1,2}$ to denote the maximum 2-norm of the columns of $\boldsymbol{A}$, $\|\boldsymbol{A}\|_0$ to denote the total number of nonzero entries in $\boldsymbol{A}$, and $\|\boldsymbol{A}\|_1$ to denote the entrywise ℓ_1 -norm of $\boldsymbol{A}$. The symbol $\boldsymbol{I}_d$ denotes the $d\times d$ identity matrix, and $\mathbb{O}(n)$ denotes the orthogonal group in dimension n. We use $\mathrm{supp}(\boldsymbol{A})$ to refer to the set of indices of nonzero entries of $\boldsymbol{A}$. The symbol $\sigma_i(\boldsymbol{A})$ denotes the ith largest singular value of $\boldsymbol{A}$, and $\kappa(\boldsymbol{A})$ denotes the condition number of $\boldsymbol{A}$. Given a matrix $\boldsymbol{A}$, its polar decomposition is defined as $\mathrm{Polar}(\boldsymbol{A}) = \boldsymbol{U_A}\boldsymbol{V_A}^\top$, where $\boldsymbol{U_A}\boldsymbol{\Sigma_A}\boldsymbol{V_A}^\top$ is the SVD of $\boldsymbol{A}$. We also define the operator $\mathcal{L}(\boldsymbol{A}) = \boldsymbol{L_A}^\top$, where $\boldsymbol{L_A}\boldsymbol{L_A}^\top$ is the Cholesky factorization of a positive semidefinite matrix $\boldsymbol{A}$. For an event $\mathcal{E}$, its indicator function is denoted as $\mathbb{1}_{\mathcal{E}}$. Given two sequences $f(n)$ and $g(n)$, the notations $f(n)\lesssim g(n)$ and $f(n) = O(g(n))$ imply that there exists a universal constant $C>0$ satisfying $f(n)\le Cg(n)$ for all large enough n. Similarly, the notations $f(n)\gtrsim g(n)$ and $f(n)=\Omega(g(n))$ imply that there exists a universal constant $C>0$ satisfying $f(n)\ge Cg(n)$ for all large enough n. We use $f(n)=\omega(g(n))$ if for all constants $C>0$ we have $f(n)\ge Cg(n)$ for all large enough n. We say an event happens *with high probability* if it occurs with probability of at least $1-n^{-\omega(1)}$ with respect to all randomness in the problem.

2. Our method. We consider a noiseless model, where the data matrix is generated according to $\boldsymbol{Y} = \boldsymbol{D}^*\boldsymbol{X}^*$. For any signed permutation matrix Π, the pair $(\boldsymbol{D}^*\Pi, \Pi^\top\boldsymbol{X}^*)$ is also a valid ground truth that satisfies $\|\Pi^\top\boldsymbol{X}^*\|_0 = \|\boldsymbol{X}^*\|_0$.[1] Let $\mathcal{M} = \{(\boldsymbol{D}^*\Pi, \Pi^\top\boldsymbol{X}^*) : \Pi \text{ is a signed permutation}\}$. Our goal is to recover an arbitrary element $(\boldsymbol{D}^*\Pi, \Pi^\top\boldsymbol{X}^*)\in\mathcal{M}$.

2.1. Orthogonal dictionary learning. We first assume that the dictionary is an orthogonal matrix $\boldsymbol{D}^*\in\mathbb{O}(n)$. This leads to Algorithm 2.1, an alternating minimization algorithm based on the closed-form solutions (1.1).

We note that Algorithm 2.1 has been studied before. The paper [6] reports the empirical performance of Algorithm 2.1 on an image restoration task. The paper [33] provides a theoretical analysis for a variant of Algorithm 2.1 based on sorted thresholding. One may even argue that Algorithm 2.1 is similar to the method of optimal directions (MOD). However, the existing theoretical guarantees for Algorithm 2.1 are indeed restrictive (see the discussion on alternating minimization in section 1.2). To bridge this knowledge gap, we first introduce our assumption on the sparse matrix $\boldsymbol{X}$.

Algorithm 2.1. Alternating minimization for (ODL).

1: **Input:** $\boldsymbol{Y}$, $\boldsymbol{D}^{(0)}$, ζ
2: **for** $t=0,1,\dots,T$ **do**
3: Set $\boldsymbol{X}^{(t)} = \mathrm{HT}_\zeta(\boldsymbol{D}^{(t)\top}\boldsymbol{Y})$
4: Set $\boldsymbol{D}^{(t+1)} = \mathrm{Polar}(\boldsymbol{Y}\boldsymbol{X}^{(t)\top})$
5: **end for**
6: **return** $\boldsymbol{D}^{(T)}, \boldsymbol{X}^{(T)}$

[1] A signed permutation matrix is a generalized permutation matrix whose nonzero entries are ± 1.

Assumption 2.1 (model for sparse code). The sparsity pattern of ground truth $\boldsymbol{X}^*$ follows a Bernoulli distribution with parameter $\theta = (0, 1]$. In particular,

$$\mathbb{1}_{\boldsymbol{X}^*_{ij} \neq 0} = B_{ij}, \quad \text{where} \quad B_{ij} \overset{i.i.d.}{\sim} \mathcal{B}(\theta) \text{ for all } 1 \le i \le n,\ 1 \le j \le p.$$

The nonzero values of $\boldsymbol{X}^*$ are independently drawn from a sub-Gaussian distribution with mean zero and constant variance σ^2. Moreover, the magnitudes of nonzero entries of $\boldsymbol{X}^*$ are lower bounded by some constant Γ. More specifically, we have

$$|\boldsymbol{X}^*_{ij}| \ge \Gamma, \quad \mathbb{E}(\boldsymbol{X}^*_{ij}) = 0, \quad \text{and} \quad \mathbb{E}(\boldsymbol{X}^{*2}_{ij}) = \sigma^2 \text{ for every } (i,j) \in \operatorname{supp}(\boldsymbol{X}^*).$$

Now, we are ready to show the convergence of Algorithm 2.1.

Theorem 2.2. *Suppose that $\boldsymbol{Y} = \boldsymbol{D}^*\boldsymbol{X}^*$, where $\boldsymbol{D}^*$ is orthogonal and $\boldsymbol{X}^*$ satisfies Assumption 2.1 with sparsity level $0 < \theta < 1/2$. Suppose that the initial dictionary $\boldsymbol{D}^{(0)}$ satisfies $\|\boldsymbol{D}^{(0)} - \boldsymbol{D}^*\|_F \lesssim \frac{1}{\sigma\sqrt{\theta n}}$ for some $(\boldsymbol{D}^*, \boldsymbol{X}^*) \in \mathcal{M}$. Moreover, suppose that $n/\theta^2 \lesssim p \lesssim n^\gamma$ for some constant $\gamma > 0$. Then, with probability at least $1 - n^{-\omega(1)}$, for any $T \ge 1$, the output of Algorithm 2.1 with $\zeta = \Gamma/2$ satisfies*

$$\|\boldsymbol{D}^{(T)} - \boldsymbol{D}^*\|_F \le 0.9^T \|\boldsymbol{D}^{(0)} - \boldsymbol{D}^*\|_F, \quad \|\boldsymbol{X}^{(T)} - \boldsymbol{X}^*\|_F \le 0.9^T \|\boldsymbol{X}^{(0)} - \boldsymbol{X}^*\|_F$$

Theorem 2.2 improves upon the existing results on two fronts.

Linear sparsity level. We allow a constant fraction of entries in $\boldsymbol{X}^*$ to be nonzero, thereby improving upon the best-known sparsity level of $O(\sqrt{n}/\log n)$ for alternating minimization [3]. Moreover, the imposed upper bound on the sample size asserts that it must be bounded by a polynomial function of n. This mild assumption is only included to simplify the presentation of our main result.

Linear sample complexity. In order to recover $\boldsymbol{D}^*$ exactly, we only need to observe $O(n)$ many samples, which is $\log(n)$ factor smaller than the sample complexity required for the uniqueness of the solution when $\xi \to 0$ [38]. This sample complexity is optimal (modulo constant factors) since it is impossible to recover the true dictionary with a sublinear number of samples even if $\boldsymbol{X}^*$ is known. Note that the theoretical convergence of Algorithm 2.1 is contingent upon a good initial dictionary. As shown in Lemma 3.1, the imposed condition on the initial dictionary automatically guarantees the recovery of the support for any $\boldsymbol{X}^{(t)}$ with $t \ge 0$. In subsection 2.4, we discuss possible ways to obtain such an initial dictionary in theory and practice. We also suspect that our initialization requirement can be improved to $\|\boldsymbol{D}^{(0)} - \boldsymbol{D}^*\|_{1,2} \le O(1/\log n)$ with fresh samples at every iteration, which is the best-known radius for alternating minimization [3].

Despite its desirable properties, Algorithm 2.1 suffers from two fundamental limitations. First, its convergence depends on the orthogonality of the true dictionary, which may not be satisfied in many applications. Second, it does not readily extend to huge-scale or online settings, where it is prohibitive or even impossible to process all data samples together. To address these challenges, we next extend our algorithm to complete (nonorthogonal) dictionary learning with mini-batch and online data.

2.2. Mini-batch complete dictionary learning. 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, we subsample from the columns of $\boldsymbol{Y}$. To address the nonorthogonality of the dictionary, we consider a preconditioner $\boldsymbol{P}$ defined as

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

Using this preconditioner, we 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 $\tilde{\boldsymbol{Y}}\approx\mathcal{L}((\boldsymbol{A}^*\boldsymbol{A}^{*\top})^{-1})\boldsymbol{A}^*\boldsymbol{X}^*$. Upon defining $\boldsymbol{D}^*=\mathcal{L}((\boldsymbol{A}^*\boldsymbol{A}^{*\top})^{-1})\boldsymbol{A}^*$, one can check that $\boldsymbol{D}^*\in\mathbb{O}(n)$ and $\tilde{\boldsymbol{Y}}\approx\boldsymbol{D}^*\boldsymbol{X}^*$. Indeed, this is an instance of (ODL) and can be solved via Algorithm 2.1. The detailed implementation is provided in Algorithm 2.2. We note that the dependency of the preconditioner on θ and σ^2 is purely to simplify the presentation and can be replaced by a simple normalization step in practice. Our next theorem characterizes the performance of Algorithm 2.2. Without loss of generality, we assume that $\|\boldsymbol{A}^*\|_2=1$ and set $\hat{\kappa}$ to be the condition number of $\boldsymbol{A}^*$.

Theorem 2.3. *Suppose that $\boldsymbol{Y}=\boldsymbol{A}^*\boldsymbol{X}^*$, where $\boldsymbol{X}^*$ satisfies Assumption 2.1 with sparsity level $0<\theta<1/2$. Suppose that the initial dictionary $\boldsymbol{A}^{(0)}$ satisfies $\|\boldsymbol{A}^{(0)}-\boldsymbol{A}^*\|_F\lesssim\frac{1}{\hat{\kappa}\sigma\sqrt{\theta n}}$ for some $(\boldsymbol{A}^*,\boldsymbol{X}^*)\in\mathcal{M}$. Moreover, suppose that $n/\theta^2\lesssim\tilde{p}\lesssim n^\gamma$, $\hat{\kappa}^{12}n^3\log^2\tilde{p}/\theta\lesssim p$, $T\lesssim n^\beta$, and $\log n\gtrsim\beta+\gamma$ for some constants $\gamma,\beta>0$. Then with probability at least $1-n^{-\omega(1)}$, the output of Algorithm 2.2 with $\zeta=\Gamma/2$ satisfies*

$$\left\|\boldsymbol{A}^{(T)}-\boldsymbol{A}^*\right\|_F\le 0.9^T\|\boldsymbol{A}^{(0)}-\boldsymbol{A}^*\|_F+O\left(\frac{n\hat{\kappa}^6}{\theta\sqrt{p}}\right). \tag{2.1}$$

Theorem 2.3 states that Algorithm 2.2 converges linearly to the true dictionary up to a statistical error $O(\frac{n\hat{\kappa}^6}{\theta\sqrt{p}})$. This statistical error is due to the deviation of the preconditioner from its expectation, which diminishes with p. A key distinction of our result is that we do not impose any incoherency requirement or RIP on $\boldsymbol{A}^*$, which are common assumptions in existing results. We also highlight that the upper bound on T is indeed very mild.

Algorithm 2.2. Alternating minimization for mini-batch (CDL).

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

To demonstrate this, note that, in order to guarantee $0.9^T\|\mathbf{A}^{(0)}-\mathbf{A}^*\|_F = O(\frac{n\hat{\kappa}^6}{\theta\sqrt{p}})$, it suffices to have $T=\Omega(\frac{\theta\sqrt{p}}{\hat{\kappa}^6 n\log n})$, a condition that satisfies $T=O(n^\beta)$ for some $\beta>0$ for all practical purposes.

2.3. Online dictionary learning. Although the computational efficiency of Algorithm 2.2 is largely improved compared to its full-batch counterpart, it may not be implementable in huge-scale or online settings due to the dependency of the preconditioner on the entire dataset with size p. Both situations call for a cheap method to update the preconditioner more efficiently in an online fashion.

To update $\boldsymbol{P}$ with a new sample $\boldsymbol{y}$, a naive approach would be to recompute 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 $(\boldsymbol{Y}\boldsymbol{Y}^\top)^{-1}$ as

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

which, given $(\boldsymbol{Y}\boldsymbol{Y}^\top)^{-1}$, can be obtained in $O(n^2)$ operations. Given the above rank-one update for the inverse, the Cholesky factor $\mathcal{L}((\boldsymbol{Y}\boldsymbol{Y}^\top+\boldsymbol{y}\boldsymbol{y}^\top)^{-1})$ can be obtained within $O(n^2)$ operations by performing triangular rank-one updates on $\mathcal{L}((\boldsymbol{Y}\boldsymbol{Y}^\top)^{-1})$ [22]. We explain the implementation of this method in Appendix C.

Inspired by the above update, we propose an online variant of Algorithm 2.3. We start the algorithm by initializing the preconditioner and the data matrix using p_1 and p_2 samples, respectively, both potentially significantly smaller than p. When a new sample arrives, we update the preconditioner $\boldsymbol{P}^{(t)}$ via the triangular rank-one update and update the data set accordingly. Our next theorem establishes the convergence of Algorithm 2.3.

Algorithm 2.3. Alternating minimization for online dictionary learning.

Input: $\boldsymbol{A}^{(0)}$, ζ, p_1, p_2
Set $\boldsymbol{Z}_{\text{inv}}^{(0)}=(\bar{\boldsymbol{Y}}\bar{\boldsymbol{Y}}^\top)^{-1}$, $\boldsymbol{P}^{(0)}=\mathcal{L}(p_1\theta\sigma^2\boldsymbol{Z}_{\text{inv}}^{(0)})^\top$, where $\bar{\boldsymbol{Y}}$ is constructed with p_1 samples and $\boldsymbol{D}^{(0)}=\boldsymbol{P}^{(0)}\boldsymbol{A}^{(0)}$.
Initialize $\boldsymbol{Y}^{(0)}$ with p_2 samples.
for $t=0,1,\dots,T-1$ **do**
  Given a new $\boldsymbol{y}$, set $\boldsymbol{Y}^{(t)}=[\boldsymbol{Y}^{(t-1)}\ \boldsymbol{y}]$ and remove the first column of $\boldsymbol{Y}^{(t)}$.
  Update $\boldsymbol{P}^{(t)}$ and $\boldsymbol{Z}_{\text{inv}}^{(t)}$ using $\boldsymbol{P}^{(t-1)}$, $\boldsymbol{Z}_{\text{inv}}^{(t-1)}$, and $\boldsymbol{y}$ via Algorithm B.1.
  Set $\tilde{\boldsymbol{Y}}^{(t)}=\boldsymbol{P}^{(t)}\boldsymbol{Y}^{(t)}$.
  Set $\tilde{\boldsymbol{X}}^{(t)}=\mathrm{HT}_\zeta(\boldsymbol{D}^{(t)\top}\tilde{\boldsymbol{Y}}^{(t)})$.
  Set $\boldsymbol{D}^{(t+1)}=\mathrm{Polar}(\tilde{\boldsymbol{Y}}^{(t)}\tilde{\boldsymbol{X}}^{(t)\top})$.
end for
return $(\boldsymbol{P}^{(T-1)})^{-1}\boldsymbol{D}^{(T)}$ as an approximation to $\boldsymbol{A}^*$.

Theorem 2.4. *Suppose that $\boldsymbol{Y} = \boldsymbol{A}^*\boldsymbol{X}^*$, where $\boldsymbol{X}^*$ satisfies Assumption 2.1 with sparsity level $0 < \theta < 1/2$. Suppose that the initial dictionary $\boldsymbol{A}^{(0)}$ satisfies $\|\boldsymbol{A}^{(0)} - \boldsymbol{A}^*\|_F \lesssim \frac{1}{\hat{\kappa}\sigma\sqrt{\theta n}}$ for some $(\boldsymbol{A}^*, \boldsymbol{X}^*) \in \mathcal{M}$. Moreover, suppose that $n/\theta^2 \lesssim p_2 \lesssim n^\gamma$, $\hat{\kappa}^{12} n^3 \log^2 p_2/\theta \lesssim p_1$, and $T \lesssim n^\beta$ for some constants $\gamma, \beta > 0$. Then, with probability at least $1 - n^{-\omega(1)}$, the output of Algorithm 2.3 with $\zeta = \Gamma/2$ satisfies*

$$\left\|\boldsymbol{A}^{(T)} - \boldsymbol{A}^*\right\|_F \le 0.9^T \|\boldsymbol{A}^{(0)} - \boldsymbol{A}^*\|_F + O\left(\frac{n\hat{\kappa}^6}{\theta\sqrt{p_1 + T}}\right). \tag{2.2}$$

The convergence result of Algorithm 2.3 is similar to that of Algorithm 2.2, with a key difference that the statistical error $O(\frac{n\hat{\kappa}^6}{\theta\sqrt{p_1+T}})$ now decreases with T, which is due to the fact that the preconditioner becomes progressively more accurate as new samples continue to arrive.

2.4. Initialization. The theoretical success of Algorithms 2.1–2.3 requires, at least in theory, a good initialization with $O(1/\sqrt{\theta n})$ distance to the ground truth. Such initialization can be provided by the initialization scheme introduced in [1] and [3], albeit with slightly more restrictive conditions on the sparsity level. For the same generative model as described in Assumption 2.1, computationally intensive and data-intensive algorithms, such as the Riemannian trust-region method (RTR) [39], can also be employed to design initial dictionaries that satisfy the conditions outlined in Theorems 2.2–2.4. Specifically, for (ODL), [39, Theorem 3.1] demonstrates that the RTR can obtain an initial point with a distance of $O(1/\sqrt{\theta n})$ from the ground truth, albeit with an increased sample complexity of $p = \Omega(n^5)$. Similarly, for (CDL), [39, Theorem 3.2] illustrates that RTR can yield an initial point with a distance of $O(1/\sqrt{\theta n})$ from the ground truth with a sample complexity of $p = \Omega(n^7)$. Combining these results with Theorems 2.2–2.4 allows for an end-to-end global guarantee at the cost of increasing the sample complexity to $\Omega(n^5)$ for (ODL) and $\Omega(n^7)$ for (CDL).

However, we note that these tailored initialization schemes are not easily implementable, especially in a huge-scale regime. In practice, we observed that a simple warm-start algorithm would yield similar performance. Consider a variant of Algorithm 2.1 with a diminishing threshold presented in Algorithm 2.4 (the warm-start schemes for Algorithms 2.2 and 2.3 are similar and hence omitted for brevity).

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, the matrix $\boldsymbol{X}^{(t)}$ eventually becomes nonzero, and

Algorithm 2.4. Algorithm 2.1 with warm-start.

1: **Input:** $\boldsymbol{Y}$, ζ_0, ζ, $\beta \in (0,1)$
2: Set $\boldsymbol{D}^{(0)} = \boldsymbol{I}_n$.
3: **for** $t = 0, 1, \ldots, T_0$ **do**
4: $\quad \zeta_{t+1} = \max\{\beta\zeta_t, \zeta\}$.
5: $\quad$ Set $\boldsymbol{X}^{(t)} = \mathrm{HT}_{\zeta_t}(\boldsymbol{D}^{(t)\top}\boldsymbol{Y})$.
6: $\quad$ Set $\boldsymbol{D}^{(t+1)} = \boldsymbol{I}_n$ if $\boldsymbol{X}^{(t)} = \boldsymbol{0}_n$, and $\boldsymbol{D}^{(t+1)} = \mathrm{Polar}(\boldsymbol{Y}\boldsymbol{X}^{(t)\top})$ if $\boldsymbol{X}^{(t)} \neq \boldsymbol{0}_n$.
7: **end for**
8: **return** $\boldsymbol{D}^{(T_0)}$

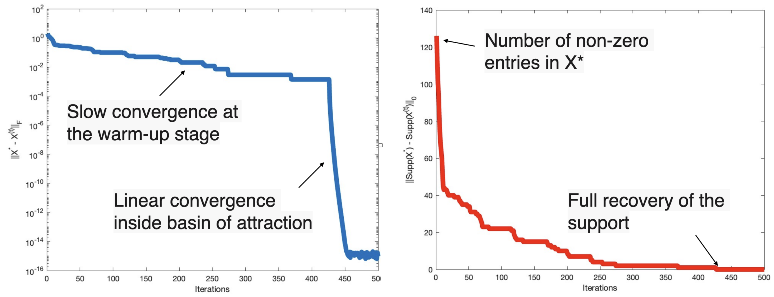

Figure 2. *The plots above show the iterates of Algorithm 2.4 with $p = 100$, $n = 5$, and $\theta = 0.3$. The left figure shows the error in the sparse code. The right figure shows the number of nonzero entries of* $\mathrm{Supp}(\boldsymbol{X}^*) - \mathrm{Supp}(\boldsymbol{X}^{(t)})$. *The number is 126 at the beginning, which is the total number of nonzero entries in $\boldsymbol{X}^*$, and 0 in the end, which indicates the full recovery of the support of $\boldsymbol{X}^*$.*

the iterates $\boldsymbol{D}^{(0)}$ start to gradually move towards the basin of attraction of $\boldsymbol{D}^*$. Figure 2 demonstrates how our proposed warm-start algorithm generates an initial dictionary that is sufficiently close to the true dictionary. We also observe that, in practice, the convergence of iterates from $\boldsymbol{I}_d$ to the basin of local convergence is not linear. However, once the iterates reach this basin, they exhibit linear convergence to the ground truth, which is theoretically guaranteed by Theorem 2.2. We also note that our proposed warm-start algorithm can serve as an effective initialization scheme for other DL algorithms, including those discussed in section 1.2. The theoretical analysis of this warm-start algorithm is left as an enticing challenge for future research.

3. Proofs.

In this section, we present the proofs of our main theorems.

3.1. Proof of Theorem 2.2.

We use induction to prove Theorem 2.2. We consider the following induction hypothesis:

$$\|\boldsymbol{D}^{(s)} - \boldsymbol{D}^*\|_F = O\left(\frac{1}{\sigma\sqrt{\theta n}}\right) \qquad \text{at iteration } s. \tag{3.1}$$

As will be shown later, this induction hypothesis will lead to the desired linear convergence result. The base case is verified, given the initial condition. We next present the following two key lemmas, the proofs of which can be found in Appendix B.

Lemma 3.1 (exact support recovery). *Consider $\boldsymbol{Y} = \boldsymbol{D}^*\boldsymbol{X}^*$, where $\boldsymbol{D}^* \in \mathbb{O}^n$ and $\boldsymbol{X}^*$ satisfies Assumption 2.1. For all $\boldsymbol{D}$ satisfying $\|\boldsymbol{D} - \boldsymbol{D}^*\|_{1,2} = O(\frac{1}{\sigma\sqrt{\theta n}})$, with probability at least $1 - 2\exp(\log p - Cn)$, we have $\mathrm{supp}(\mathrm{HT}_{\Gamma/2}(\boldsymbol{D}^\top\boldsymbol{Y})) = \mathrm{supp}(\boldsymbol{X}^*)$.*

Lemma 3.2 (guaranteed improvement on polar decomposition). *Consider $\boldsymbol{Y} = \boldsymbol{D}^*\boldsymbol{X}^*$, where $\boldsymbol{D}^* \in \mathbb{O}^n$ and $\boldsymbol{X}^*$ satisfies Assumption 2.1 with sparsity level $0 < \theta < 1/2$, and suppose that $p = \Omega(n/\theta^2)$. Then, with probability at least $1 - 2\exp(\log n - n)$, the following inequality holds for every approximation $\boldsymbol{X}$ of $\boldsymbol{X}^*$ such that $\frac{\|\boldsymbol{X}-\boldsymbol{X}^*\|_F}{\|\boldsymbol{X}^*\|_2} = O(1)$ and $\mathrm{supp}(\boldsymbol{X}) = \mathrm{supp}(\boldsymbol{X}^*)$:*

$$\|\mathrm{Polar}(\boldsymbol{Y}\boldsymbol{X}^\top) - \boldsymbol{D}^*\|_F < 0.9\frac{\|\boldsymbol{X} - \boldsymbol{X}^*\|_F}{\|\boldsymbol{X}^*\|_2}.$$

Lemma 3.1 together with our induction hypothesis (3.1) implies that

$$\operatorname{supp}\left(\boldsymbol{X}^{(s)}\right) = \operatorname{supp}\left(\mathrm{HT}_{\Gamma/2}(\boldsymbol{D}^{(s)\top}\boldsymbol{Y})\right) = \operatorname{supp}\left(\boldsymbol{X}^{*}\right). \tag{3.2}$$

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

$$\|\boldsymbol{X}^{(s)} - \boldsymbol{X}^{*}\|_F = \|\mathrm{HT}_{\Gamma/2}(\boldsymbol{D}^{(s)\top}\boldsymbol{Y}) - \boldsymbol{X}^{*}\|_F \overset{\text{(a)}}{\leq} \|\boldsymbol{D}^{(s)} - \boldsymbol{D}^{*}\|_F\|\boldsymbol{Y}\|_2,$$

where we used (3.2) for (a). The above inequality implies that

$$\frac{\|\boldsymbol{X}^{(s)} - \boldsymbol{X}^{*}\|_F}{\|\boldsymbol{X}^{*}\|_2} = \frac{\|\boldsymbol{X}^{(s)} - \boldsymbol{X}^{*}\|_F}{\|\boldsymbol{D}^{*}\boldsymbol{X}^{*}\|_2} = \frac{\|\boldsymbol{X}^{(s)} - \boldsymbol{X}^{*}\|_F}{\|\boldsymbol{Y}\|_2} \leq \|\boldsymbol{D}^{(s)} - \boldsymbol{D}^{*}\|_F = O\left(\frac{1}{\sigma\sqrt{\theta n}}\right). \tag{3.3}$$

Invoking Lemma 3.2 with $\boldsymbol{X} = \boldsymbol{X}^{(s)}$ together with (3.3) leads to

$$\|\boldsymbol{D}^{(s+1)} - \boldsymbol{D}^{*}\|_F < \frac{0.9\|\boldsymbol{X}^{(s)} - \boldsymbol{X}^{*}\|_F}{\|\boldsymbol{X}^{*}\|_2} < 0.9\|\boldsymbol{D}^{(s)} - \boldsymbol{D}^{*}\|_F.$$

Therefore, the induction hypothesis (3.1) holds for $t = s+1$. Consequently, the linear convergence for $\frac{\|\boldsymbol{X}^{(s)} - \boldsymbol{X}^{*}\|_F}{\|\boldsymbol{X}^{*}\|_2}$ follows from (3.3) and the above inequality:

$$\frac{\|\boldsymbol{X}^{(s+1)} - \boldsymbol{X}^{*}\|_F}{\|\boldsymbol{X}^{*}\|_2} \leq \|\boldsymbol{D}^{(s+1)} - \boldsymbol{D}^{*}\|_F < 0.9\frac{\|\boldsymbol{X}^{(s)} - \boldsymbol{X}^{*}\|_F}{\|\boldsymbol{X}^{*}\|_2}.$$

The proof is complete by noticing a lower bound of the probability that both Lemma 3.1 and Lemma 3.2 hold is $1 - 2\exp(\log p - Cn) - 2\exp(\log n - n) = 1 - n^{-\omega(1)}$, which follows from our assumption $p = O(n^{\gamma})$. This completes the proof.

Next, we provide the proof of Theorem 2.4. As will be shown later, the proof of Theorem 2.3 readily follows from this proof.

3.2. Proof of Theorem 2.4. The linear convergence of $\boldsymbol{A}^{(T)}$ hinges on the linear convergence of the preconditioned dictionary $\boldsymbol{D}^{(T)}$ towards

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

To establish this fact, we first prove the following inequality:

$$\|\boldsymbol{D}^{(s+1)} - \boldsymbol{D}^{*}\|_F \leq 0.9\|\boldsymbol{D}^{(s)} - \boldsymbol{D}^{*}\|_F + O\left(\|\boldsymbol{P}^{(s)}\boldsymbol{A}^{*} - \boldsymbol{D}^{*}\|_F\right), \quad s = 0, 2, \ldots, T-1. \tag{3.4}$$

We present the following intermediate lemmas, the proofs of which are in Appendix B.

Lemma 3.3 (bounding preconditioner error). *Suppose that $\boldsymbol{X}^{*}$ satisfies Assumption 2.1. Then, with probability at least $1 - 2\exp(-n)$, we have*

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

Lemma 3.4 (spectral property for sparse code matrix). *Suppose that $\boldsymbol{X}^*$ satisfies Assumption 2.1 and $p=\Omega(\tau n/\theta^2)$ for an arbitrary $\tau>0$. Then, with probability at least $1-2\exp(-n)$, we have*

$$\|\boldsymbol{X}^*\|_2 \le \left(1+\tau^{-\frac{1}{4}}\right)\sqrt{p\theta}\sigma, \quad \sigma_n\left(\boldsymbol{X}^*\right) \ge \left(1-\tau^{-\frac{1}{4}}\right)\sqrt{p\theta}\sigma, \quad \kappa\left(\boldsymbol{X}^*\right) \le \frac{1+\tau^{-\frac{1}{4}}}{1-\tau^{-\frac{1}{4}}}.$$

As with the proof of Theorem 2.3, we will use the induction hypothesis $\|\boldsymbol{D}^{(s)}-\boldsymbol{D}^*\|_F \le O(\frac{1}{\sigma\sqrt{\theta n}})$. The base case is easy to verify given the initial condition and Lemma 3.3:

$$\begin{aligned}
\|\boldsymbol{D}^{(0)}-\boldsymbol{D}^*\|_F &= \|\boldsymbol{P}^{(0)}\boldsymbol{A}^{(0)} - \mathcal{L}((\boldsymbol{A}^*\boldsymbol{A}^{*\top})^{-1})^\top \boldsymbol{A}^*\|_F \\
&= \|\boldsymbol{P}^{(0)}\boldsymbol{A}^{(0)} - \mathcal{L}((\boldsymbol{A}^*\boldsymbol{A}^{*\top})^{-1})^\top \boldsymbol{A}^{(0)} + \mathcal{L}((\boldsymbol{A}^*\boldsymbol{A}^{*\top})^{-1})^\top \boldsymbol{A}^{(0)} - \mathcal{L}((\boldsymbol{A}^*\boldsymbol{A}^{*\top})^{-1})^\top \boldsymbol{A}^*\|_F \\
&\le \|\boldsymbol{P}^{(0)} - \mathcal{L}((\boldsymbol{A}^*\boldsymbol{A}^{*\top})^{-1})^\top\|_2 \|\boldsymbol{A}^{(0)}\|_F + \|\mathcal{L}((\boldsymbol{A}^*\boldsymbol{A}^{*\top})^{-1})^\top\|_2 \|\boldsymbol{A}^{(0)}-\boldsymbol{A}^*\|_F \\
&\le O\left(\frac{\hat{\kappa}^6}{\theta}\sqrt{\frac{n}{p_1}}\cdot\sqrt{n}\right) + O\left(\hat{\kappa}\cdot\frac{1}{\hat{\kappa}\sigma\sqrt{\theta n}}\right) = O\left(\frac{1}{\sigma\sqrt{\theta n}}\right).
\end{aligned}$$

Moreover, after achieving (3.4) for each s, we have $\|\boldsymbol{D}^{(s+1)}-\boldsymbol{D}^*\|_F = O(\frac{1}{\sigma\sqrt{\theta n}})$ since

$$\|\boldsymbol{P}^{(s)}\boldsymbol{A}^* - \boldsymbol{D}^*\|_F \le \left\|\boldsymbol{P}^{(s)} - \mathcal{L}\left(\left(\boldsymbol{A}^*\boldsymbol{A}^{*\top}\right)^{-1}\right)^\top\right\|_2 \|\boldsymbol{D}^*\|_F = O\left(\frac{1}{\sigma\sqrt{\theta n}}\right).$$

For the remainder of this section, we abuse the notation and use $\boldsymbol{X}^*$ to denote the sparse code 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 2.3). 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 define $\tilde{\boldsymbol{X}}$ as $\tilde{\boldsymbol{X}} = \boldsymbol{D}^{*\top}\tilde{\boldsymbol{Y}}^{(s)}$, which is again a different sparse code 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. Let us assume that the initial sample size and batch size satisfy $p_1 = \Omega(\tau_1 n^3 \log^2 p_2 \sigma^2 \hat{\kappa}^{12}/(\theta\Gamma^2))$ and $p_2 = \Omega(\tau_2^4 n/\theta^2)$ for parameters $\tau_1, \tau_2 > 0$ to be defined later.

Towards proving (3.4), we first show that $\tilde{\boldsymbol{X}}^{(s)}$ recovers the sparsity pattern of $\boldsymbol{X}^*$ if we use a large enough number of samples to construct the preconditioner. To show this, we consider one entry $(\boldsymbol{D}^{(s)\top}\tilde{\boldsymbol{Y}}^{(s)})_{ij}$ of $\boldsymbol{D}^{(s)\top}\tilde{\boldsymbol{Y}}^{(s)}$ and write

$$\left(\boldsymbol{D}^{(s)\top}\tilde{\boldsymbol{Y}}^{(s)}\right)_{ij} = \underbrace{\left(\boldsymbol{D}^{(s)\top}\boldsymbol{D}^*(\tilde{\boldsymbol{X}}-\boldsymbol{X}^*)\right)_{ij}}_{:=\mathcal{A}_{ij}} + \underbrace{\left(\boldsymbol{D}^{(s)\top}\boldsymbol{D}^*\boldsymbol{X}^*\right)_{ij}}_{:=\mathcal{B}_{ij}}. \tag{3.5}$$

The first term in the right-hand side can be bounded as

$$\begin{aligned}
|\mathcal{A}_{ij}| &\le \|\boldsymbol{D}^{(s)\top}\boldsymbol{D}^*\|_2 \|\tilde{\boldsymbol{X}}-\boldsymbol{X}^*\|_{1,2} = \|\tilde{\boldsymbol{X}}-\boldsymbol{X}^*\|_{1,2} \\
&\le \|\boldsymbol{D}^{*\top}\boldsymbol{P}^{(s)}\boldsymbol{A}^*\boldsymbol{X}^* - \boldsymbol{X}^*\|_{1,2} \le \|\boldsymbol{P}^{(s)}\boldsymbol{A}^*-\boldsymbol{D}^*\|_2\|\boldsymbol{X}^*\|_{1,2} \\
&\le \left\|\boldsymbol{P}^{(s)} - \mathcal{L}\left(\left(\boldsymbol{A}^*\boldsymbol{A}^{*\top}\right)^{-1}\right)\right\|_2 \|\boldsymbol{A}^*\|_2\|\boldsymbol{X}^*\|_{1,2} \lesssim \frac{\hat{\kappa}^6}{\theta}\sqrt{\frac{n}{p_1+T}}\|\boldsymbol{X}^*\|_{1,2} \qquad (3.6)
\end{aligned}$$

with probability at least $1-2\exp(-n)$, where the last inequality is due to Lemma 3.3. The expectation for the norm of each column, based on Assumption 2.1, is $\sigma\sqrt{\theta n}$. Therefore, according to Theorem A.2 (see Appendix A), the random variable $\|\boldsymbol{X}^*_{(\cdot,k)}\|_2 - \sigma\sqrt{\theta n}$ is sub-Gaussian with sub-Gaussian norm $O(\sqrt{\theta}\sigma)$. As a result, the kth column of $\boldsymbol{X}^*$ satisfies

$$\mathbb{P}\left(\|\boldsymbol{X}^*_{(\cdot,k)}\|_2 \geq 2\sigma\sqrt{\theta n}\right) \leq 2\exp\left(-Cn\right) \tag{3.7}$$

for some constant C. Upon taking a union bound over different columns, we have

$$\|\boldsymbol{X}^*\|_{1,2} = \max_{1\leq k\leq p_2} \|\boldsymbol{X}^*_{(\cdot,k)}\|_2 \leq 2\sigma\sqrt{\theta n \log p_2} \tag{3.8}$$

with probability at least $1-2\exp(-Cn)$. Combining this inequality with (3.6), we have

$$|\mathcal{A}_{ij}| \lesssim \frac{\hat{\kappa}^6\sigma}{\sqrt{\theta}}\sqrt{\frac{n^2\log p_2}{p_1+T}}$$

for every (i,j). Based on the above inequality and with the choice of $p_1 \gtrsim \frac{n^2\log p_2 \sigma^2\hat{\kappa}^{12}}{\theta\Gamma^2}$, we have $|\mathcal{A}_{ij}| < \Gamma/4$. Following the proof of Lemma 3.1, the deviation of the second term $\mathcal{B}_{ij}$ from $\boldsymbol{X}^*_{ij}$ can also be upper bounded by $\Gamma/4$ with high probability, which will not be repeated here. Therefore, we have $|(\boldsymbol{D}^{(s)\top}\tilde{\boldsymbol{Y}}^{(s)})_{ij}| < \Gamma/2$ when $\boldsymbol{X}^*_{ij}=0$ and $|(\boldsymbol{D}^{(s)\top}\tilde{\boldsymbol{Y}}^{(s)})_{ij}| > \Gamma/2$ when $\boldsymbol{X}^*_{ij}\neq 0$, which leads to $\mathrm{supp}(\tilde{\boldsymbol{X}}^{(s)}) = \mathrm{supp}(\boldsymbol{X}^*)$ with probability at least $1-2\exp(-n)-2\exp(-Cn)-2\exp(\log p_2 - Cn)$. For any matrix $\boldsymbol{M}\in\mathbb{R}^{n\times p_2}$, we define the projection operator $\mathcal{P}$ as

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

Given the exact support recovery, we have

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

As a result we have

$$\begin{aligned} &\frac{\|\tilde{\boldsymbol{X}}^{(s)} - \mathcal{P}(\tilde{\boldsymbol{X}})\|_F}{\|\mathcal{P}(\tilde{\boldsymbol{X}})\|_2} = \frac{\|\tilde{\boldsymbol{X}}^{(s)} - \mathcal{P}(\tilde{\boldsymbol{X}})\|_F}{\|\tilde{\boldsymbol{X}}\|_2}\frac{\|\tilde{\boldsymbol{X}}\|_2}{\|\mathcal{P}(\tilde{\boldsymbol{X}})\|_2} \leq \|\boldsymbol{D}^{(s)}-\boldsymbol{D}^*\|_F\frac{\|\tilde{\boldsymbol{X}}\|_2}{\|\tilde{\boldsymbol{X}}\|_2 - \|\tilde{\boldsymbol{X}}-\mathcal{P}(\tilde{\boldsymbol{X}})\|_2} \\ &\leq \|\boldsymbol{D}^{(s)}-\boldsymbol{D}^*\|_F\frac{\|\tilde{\boldsymbol{X}}\|_2}{\|\tilde{\boldsymbol{X}}\|_2 - \|\tilde{\boldsymbol{X}}-\mathcal{P}(\tilde{\boldsymbol{X}})\|_F} \overset{\text{(a)}}{\leq} \|\boldsymbol{D}^{(s)}-\boldsymbol{D}^*\|_F\frac{\|\tilde{\boldsymbol{X}}\|_2}{\|\tilde{\boldsymbol{X}}\|_2 - \|\tilde{\boldsymbol{X}}-\boldsymbol{X}^*\|_F} \\ &\overset{\text{(b)}}{\leq} \|\boldsymbol{D}^{(s)}-\boldsymbol{D}^*\|_F\left(1+\frac{\|\tilde{\boldsymbol{X}}-\boldsymbol{X}^*\|_F}{\|\boldsymbol{X}^*\|_2 - 2\|\tilde{\boldsymbol{X}}-\boldsymbol{X}^*\|_F}\right) \leq \|\boldsymbol{D}^{(s)}-\boldsymbol{D}^*\|_F\left(1+\frac{\frac{\|\tilde{\boldsymbol{X}}-\boldsymbol{X}^*\|_F}{\|\boldsymbol{X}^*\|_2}}{1-\frac{2\|\tilde{\boldsymbol{X}}-\boldsymbol{X}^*\|_F}{\|\boldsymbol{X}^*\|_2}}\right) \\ &\overset{\text{(c)}}{\leq} \|\boldsymbol{D}^{(s)}-\boldsymbol{D}^*\|_F\left(1+\frac{\frac{1}{\tau_1}}{1-2\times\frac{1}{\tau_1}}\right) = \frac{\tau_1-1}{\tau_1-2}\|\boldsymbol{D}^{(s)}-\boldsymbol{D}^*\|_F. \end{aligned} \tag{3.9}$$

Here inequality (a) is due to the decomposition of $\tilde{\boldsymbol{X}} - \boldsymbol{X}^*$ onto and outside the support of $\boldsymbol{X}^*$. Specifically, define $\mathcal{P}_\perp(\cdot)$ as the projection onto the complement of $\mathcal{P}(\cdot)$. We have $\mathcal{P}_\perp(\tilde{\boldsymbol{X}} - \mathcal{P}(\tilde{\boldsymbol{X}})) = \mathcal{P}_\perp(\tilde{\boldsymbol{X}} - \boldsymbol{X}^*)$ and $\mathcal{P}(\tilde{\boldsymbol{X}} - \mathcal{P}(\tilde{\boldsymbol{X}})) = 0$, which in turn implies $\|\tilde{\boldsymbol{X}} - \mathcal{P}(\tilde{\boldsymbol{X}})\|_F \leq \|\tilde{\boldsymbol{X}} - \boldsymbol{X}^*\|_F$. The inequality (b) is due to $\|\boldsymbol{X}^*\|_2 - \|\tilde{\boldsymbol{X}} - \boldsymbol{X}^*\|_F \leq \|\boldsymbol{X}^*\|_2 - \|\tilde{\boldsymbol{X}} - \boldsymbol{X}^*\|_2 \leq \|\tilde{\boldsymbol{X}}\|_2$. The inequality (c) holds due to $\frac{\|\tilde{\boldsymbol{X}} - \boldsymbol{X}^*\|_F}{\|\boldsymbol{X}^*\|_2} \leq 1/\tau_1$, which is a result of Lemma 3.3:

$$\begin{aligned}\|\tilde{\boldsymbol{X}} - \boldsymbol{X}^*\|_F &= \|\boldsymbol{D}^{*\top}\boldsymbol{P}^{(s)}\boldsymbol{A}^*\boldsymbol{X}^* - \boldsymbol{X}^*\|_F \leq \|\boldsymbol{P}^{(s)}\boldsymbol{A}^* - \boldsymbol{D}^*\|_F\|\boldsymbol{X}^*\|_2 \\ &\leq \sqrt{n}\|\boldsymbol{P}^{(s)}\boldsymbol{A}^* - \boldsymbol{D}^*\|_2\|\boldsymbol{X}^*\|_2 \lesssim \frac{\hat{\kappa}^6}{\theta}\sqrt{\frac{n^2}{p_1+T}}\|\boldsymbol{X}^*\|_2,\end{aligned} \tag{3.10}$$

and $p_1 = \Omega(\tau_1 n^3 \log^2 p_2 \sigma^2 \hat{\kappa}^{12}/(\theta\Gamma^2))$.

So far, we have upper bounded $\frac{\|\tilde{\boldsymbol{X}}^{(s)} - \mathcal{P}(\tilde{\boldsymbol{X}})\|_F}{\|\mathcal{P}(\tilde{\boldsymbol{X}})\|_2}$ with $\|\boldsymbol{D}^{(s)} - \boldsymbol{D}^*\|_F$. Next, we provide an upper bound on $\|\boldsymbol{D}^{(s+1)} - \boldsymbol{D}^*\|_F$ in terms of $\frac{\|\tilde{\boldsymbol{X}}^{(s)} - \mathcal{P}(\tilde{\boldsymbol{X}})\|_F}{\|\mathcal{P}(\tilde{\boldsymbol{X}})\|_2}$. We have

$$\begin{aligned}&\|\boldsymbol{D}^{(s+1)} - \boldsymbol{D}^*\|_F = \|\mathrm{Polar}(\tilde{\boldsymbol{Y}}^{(s)}\tilde{\boldsymbol{X}}^{(s)\top}) - \boldsymbol{D}^*\|_F = \|\mathrm{Polar}(\boldsymbol{D}^*\tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}^{(s)\top}) - \boldsymbol{D}^*\|_F \\ &\quad = \|\mathrm{Polar}(\boldsymbol{D}^*\tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}^{(s)\top}) - \mathrm{Polar}(\boldsymbol{D}^*\mathcal{P}(\tilde{\boldsymbol{X}})\tilde{\boldsymbol{X}}^{(s)\top}) + \mathrm{Polar}(\boldsymbol{D}^*\mathcal{P}(\tilde{\boldsymbol{X}})\tilde{\boldsymbol{X}}^{(s)\top}) - \boldsymbol{D}^*\|_F \\ &\quad = \|\mathrm{Polar}(\boldsymbol{D}^*\tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}^{(s)\top}) - \mathrm{Polar}(\boldsymbol{D}^*\mathcal{P}(\tilde{\boldsymbol{X}})\tilde{\boldsymbol{X}}^{(s)\top})\|_F + \|\mathrm{Polar}(\boldsymbol{D}^*\mathcal{P}(\tilde{\boldsymbol{X}})\tilde{\boldsymbol{X}}^{(s)\top}) - \boldsymbol{D}^*\|_F.\end{aligned} \tag{3.11}$$

To control the first term on the right-hand side, we invoke Theorem A.3 (see Appendix A):

$$\begin{aligned}&\|\mathrm{Polar}(\boldsymbol{D}^*\tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}^{(s)\top}) - \mathrm{Polar}(\boldsymbol{D}^*\mathcal{P}(\tilde{\boldsymbol{X}})\tilde{\boldsymbol{X}}^{(s)\top})\|_F \\ &\quad \leq \frac{2\|\boldsymbol{D}^*\tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}^{(s)\top} - \boldsymbol{D}^*\mathcal{P}(\tilde{\boldsymbol{X}})\tilde{\boldsymbol{X}}^{(s)\top}\|_F}{\sigma_n(\boldsymbol{D}^*\tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}^{(s)\top}) + \sigma_n(\boldsymbol{D}^*\mathcal{P}(\tilde{\boldsymbol{X}})\tilde{\boldsymbol{X}}^{(s)\top})} \\ &\quad \overset{\text{(a)}}{\leq} \frac{2\kappa(\tilde{\boldsymbol{X}}^{(s)})\|\tilde{\boldsymbol{X}} - \mathcal{P}(\tilde{\boldsymbol{X}})\|_F}{2\sigma_n(\boldsymbol{X}^*) - \|\boldsymbol{X}^* - \tilde{\boldsymbol{X}}\|_2 - \|\boldsymbol{X}^* - \mathcal{P}(\tilde{\boldsymbol{X}})\|_2} \\ &\quad = \frac{2\kappa(\tilde{\boldsymbol{X}}^{(s)})\|\tilde{\boldsymbol{X}} - \mathcal{P}(\tilde{\boldsymbol{X}})\|_F}{\|\boldsymbol{X}^*\|_2} \cdot \frac{1}{\frac{2}{\kappa(\boldsymbol{X}^*)} - \frac{\|\boldsymbol{X}^* - \tilde{\boldsymbol{X}}\|_2}{\|\boldsymbol{X}^*\|_2} - \frac{\|\boldsymbol{X}^* - \mathcal{P}(\tilde{\boldsymbol{X}})\|_2}{\|\boldsymbol{X}^*\|_2}} \\ &\quad \overset{\text{(b)}}{\leq} \frac{\tau_1(\tau_2 - 1)}{\tau_1(\tau_2+1) - \tau_2 + 1}\frac{\kappa(\tilde{\boldsymbol{X}}^{(s)})\|\tilde{\boldsymbol{X}} - \mathcal{P}(\tilde{\boldsymbol{X}})\|_F}{\|\boldsymbol{X}^*\|_2},\end{aligned}$$

where (a) follows after normalizing both the numerator and the denominator with $\|\tilde{\boldsymbol{X}}^{(s)}\|_2$. Moreover, (b) follows from (3.10), Lemma 3.4, and $p_2 = \Omega(\tau_2^4 n/\theta^2)$. Next, we provide a bound on $\kappa(\tilde{\boldsymbol{X}}^{(s)})$, and we do so by first bounding $\kappa(\mathcal{P}(\tilde{\boldsymbol{X}}))$. By Lemmas 3.4 and 3.3, we have

$$\begin{aligned}\kappa(\mathcal{P}(\tilde{\boldsymbol{X}})) &= \frac{\|\mathcal{P}(\tilde{\boldsymbol{X}})\|_2}{\sigma_n(\mathcal{P}(\tilde{\boldsymbol{X}}))} \leq \frac{\|\boldsymbol{X}^*\|_2 + \|\mathcal{P}(\tilde{\boldsymbol{X}}) - \boldsymbol{X}^*\|_2}{\sigma_n(\boldsymbol{X}^*) - \|\mathcal{P}(\tilde{\boldsymbol{X}}) - \boldsymbol{X}^*\|_2} \\ &= \frac{\|\boldsymbol{X}^*\|_2 + \|\mathcal{P}(\tilde{\boldsymbol{X}}) - \boldsymbol{X}^*\|_2}{\|\boldsymbol{X}^*\|_2/\kappa(\boldsymbol{X}^*) - \|\mathcal{P}(\tilde{\boldsymbol{X}}) - \boldsymbol{X}^*\|_2} \leq \frac{\tau_1 + 1}{\frac{\tau_1(\tau_2-1)}{\tau_2+1} - 1}.\end{aligned} \tag{3.12}$$

Upon assuming $\tau_1 \geq 10$ and $\tau_2 \geq 10$, we have

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

For (a), we used (3.9) and the fact that $\|\boldsymbol{D}^{(s)} - \boldsymbol{D}^*\|_F \leq 1/4$. To conclude, we have

$$\|\mathrm{Polar}(\boldsymbol{D}^* \tilde{\boldsymbol{X}} \tilde{\boldsymbol{X}}^{(s)\top}) - \mathrm{Polar}(\boldsymbol{D}^* \mathcal{P}(\tilde{\boldsymbol{X}}) \tilde{\boldsymbol{X}}^{(s)\top})\|_F < \frac{12\|\tilde{\boldsymbol{X}} - \mathcal{P}(\tilde{\boldsymbol{X}})\|_F}{\|\boldsymbol{X}^*\|_2} \leq \frac{12\|\boldsymbol{X}^* - \mathcal{P}(\tilde{\boldsymbol{X}})\|_F}{\|\boldsymbol{X}^*\|_2}.$$

Next, we turn our focus to $\|\mathrm{Polar}(\boldsymbol{D}^* \mathcal{P}(\tilde{\boldsymbol{X}}) \tilde{\boldsymbol{X}}^{(s)\top}) - \boldsymbol{D}^*\|_F$, which is the second term of (3.11). We first introduce the following lemma, which is borrowed from [33].

Lemma 3.5 (approximation error for dictionary (see the proof of Lemma 2.4 in [33])). *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 $\boldsymbol{X}^*$ with $\boldsymbol{E} := (\boldsymbol{X} - \boldsymbol{X}^*)/\|\boldsymbol{X}^*\|_2$ satisfying $\|\boldsymbol{E}\|_F < 1/\kappa^2(\boldsymbol{X}^*)$. Then, for some constant $C_T > 0$, $\boldsymbol{D} = \mathrm{Polar}(\boldsymbol{Y}\boldsymbol{X}^\top)$ satisfies:*

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

To invoke Lemma 3.5, let $\tilde{\boldsymbol{E}}^{(s)} = \frac{\tilde{\boldsymbol{X}}^{(s)} - \mathcal{P}(\tilde{\boldsymbol{X}})}{\|\mathcal{P}(\tilde{\boldsymbol{X}})\|_2}$, and notice that

$$\|\tilde{\boldsymbol{E}}^{(s)}\|_F \overset{\text{(a)}}{\leq} \frac{9}{8} \|\boldsymbol{D}^{(s)} - \boldsymbol{D}^*\|_F \leq \frac{9}{32} \leq \left(\frac{79}{121}\right)^2 \overset{\text{(b)}}{\leq} \frac{1}{\kappa^2(\mathcal{P}(\tilde{\boldsymbol{X}}))},$$

where (a) and (b) follow from (3.9) and (3.12), respectively. Therefore, Lemma 3.5 can be invoked:

(3.13)
$$\|\mathrm{Polar}(\boldsymbol{D}^* \mathcal{P}(\tilde{\boldsymbol{X}}) \tilde{\boldsymbol{X}}^{(s)\top}) - \boldsymbol{D}^*\|_F \leq \frac{\kappa(\mathcal{P}(\tilde{\boldsymbol{X}}))^4}{2} \frac{\left\|\mathcal{P}(\tilde{\boldsymbol{X}}) \tilde{\boldsymbol{E}}^{(s)\top} - \tilde{\boldsymbol{E}}^{(s)} \mathcal{P}(\tilde{\boldsymbol{X}})^\top\right\|_F}{\|\mathcal{P}(\tilde{\boldsymbol{X}})\|_2} + C_T \|\boldsymbol{E}\|_F^2.$$

By recalling $\mathrm{supp}(\tilde{\boldsymbol{X}}^{(s)}) = \mathrm{supp}(\boldsymbol{X}^*)$, we follow the same argument in the proof of Lemma 3.2 and conclude that

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

where $\boldsymbol{M}_k$ and $\tilde{\boldsymbol{M}}_k$ are defined in the same way as in the proof of Lemma 3.2. Since both $\boldsymbol{M}_k$ and $\tilde{\boldsymbol{M}}_k$ are linear operators, we have

$$\frac{\max_{1 \leq k \leq n} \left\|\boldsymbol{M}_k \mathcal{P}(\tilde{\boldsymbol{X}}) \tilde{\boldsymbol{M}}_k\right\|_2}{\|\mathcal{P}(\tilde{\boldsymbol{X}})\|_2} \leq \frac{\max_{1 \leq k \leq n} \left\|\boldsymbol{M}_k \boldsymbol{X}^* \tilde{\boldsymbol{M}}_k\right\|_2 + \|\mathcal{P}(\tilde{\boldsymbol{X}}) - \boldsymbol{X}^*\|_2}{\|\boldsymbol{X}^*\|_2 (1 - \|\mathcal{P}(\tilde{\boldsymbol{X}}) - \boldsymbol{X}^*\|_F / \|\boldsymbol{X}^*\|_2)}$$

$$\leq \frac{\tau_1}{\tau_1 - 1} \left(\frac{\max_{1 \leq k \leq n} \left\|\boldsymbol{M}_k \boldsymbol{X}^* \tilde{\boldsymbol{M}}_k\right\|_2}{\|\boldsymbol{X}^*\|_2} + \frac{1}{\tau_1} \right). \tag{3.14}$$

Following the proof of Lemma 3.2, we have that with probability $1-2\exp(\log n - n)$ and $p_2 = \Omega(\tau_2^4 n/\theta^2)$

$$\frac{\max_{1\le k\le n}\left\|\boldsymbol{M}_k\boldsymbol{X}^*\tilde{\boldsymbol{M}}_k\right\|_2}{\|\boldsymbol{X}^*\|_2} \le \frac{\tau_2+1}{\tau_2-1}\sqrt{\theta}. \tag{3.15}$$

Combined with (3.13), we have

$$\begin{aligned}\|\mathrm{Polar}(\boldsymbol{D}^*\mathcal{P}(\tilde{\boldsymbol{X}})\tilde{\boldsymbol{X}}^{(s)\top}) - \boldsymbol{D}^*\|_F &\le \frac{\kappa(\mathcal{P}(\tilde{\boldsymbol{X}}))^4}{2}\frac{\left\|\mathcal{P}(\tilde{\boldsymbol{X}})\tilde{\boldsymbol{E}}^{(s)\top} - \tilde{\boldsymbol{E}}^{(s)}\mathcal{P}(\tilde{\boldsymbol{X}})^\top\right\|_F}{\|\mathcal{P}(\tilde{\boldsymbol{X}})\|_2} + C_T\|\tilde{\boldsymbol{E}}^{(s)}\|_F^2\\ &\le \left(\frac{\tau_1}{\tau_1-1}\left(\frac{\tau_2+1}{\tau_2-1}\sqrt{\theta}+\frac{1}{\tau_1}\right)\kappa(\mathcal{P}(\tilde{\boldsymbol{X}}))^4 + C_T\|\tilde{\boldsymbol{E}}^{(s)}\|_F\right)\|\tilde{\boldsymbol{E}}^{(s)}\|_F.\end{aligned}$$

Therefore, we have a bound for the second term on the right-hand side of (3.11). Recall that $\kappa(\mathcal{P}(\tilde{\boldsymbol{X}})) \le \frac{\tau_1+1}{\frac{\tau_1(\tau_2-1)}{\tau_2+1}-1}$ due to (3.12) and $0<\theta<1/2$. Therefore, similar to the proof of Lemma 3.2, for sufficiently large τ_1 and τ_2 and $\|\tilde{\boldsymbol{E}}^{(s)}\|_F \le \frac{1}{10C_T}$, (3.11) can be rewritten as

$$\begin{aligned}\|\boldsymbol{D}^{(s+1)} - \boldsymbol{D}^*\|_F &\overset{(a)}{\le} 0.9\|\boldsymbol{D}^{(s)} - \boldsymbol{D}^*\|_F + \frac{12\|\boldsymbol{X}^* - \mathcal{P}(\tilde{\boldsymbol{X}})\|_F}{\|\boldsymbol{X}^*\|_2}\\ &\le 0.9\|\boldsymbol{D}^{(s)} - \boldsymbol{D}^*\|_F + 12\|\boldsymbol{P}^{(s)}\boldsymbol{A}^* - \boldsymbol{D}^*\|_F,\end{aligned} \tag{3.16}$$

where (a) follows from (3.9). This completes the proof of the induction step (3.4). For each induction step, our analysis holds with probability at least $1-O(\exp(\log p_2 - n))$. After taking the union bound, the first T steps of the induction hold with probability at least $1-O(\exp(\log T + \log p_2 - n))$, which is $1-n^{-\omega(1)}$ when $T=O(n^\beta)$ and $p_2=O(n^\gamma)$. Finally, we use (3.16) to complete the proof of Theorem 2.4. To this goal, notice that

$$\begin{aligned}\|\boldsymbol{A}^{(s+1)} - \boldsymbol{A}^*\|_F &= \left\|\boldsymbol{P}^{(s+1)-1}\boldsymbol{D}^{(s+1)} - \left(\mathcal{L}((\boldsymbol{A}^*\boldsymbol{A}^{*\top})^{-1})^\top\right)^{-1}\boldsymbol{D}^*\right\|_F\\ &\le \|\boldsymbol{D}^{(s+1)} - \boldsymbol{D}^*\|_F + \left\|\boldsymbol{P}^{(s+1)-1} - \left(\mathcal{L}((\boldsymbol{A}^*\boldsymbol{A}^{*\top})^{-1})^\top\right)^{-1}\right\|_F\\ &\le 0.9\|\boldsymbol{D}^{(s)} - \boldsymbol{D}^*\|_F + O\left(\|\boldsymbol{P}^{(s)}\boldsymbol{A}^* - \boldsymbol{D}^*\|_F\right) + O\left(\left\|\boldsymbol{P}^{(s+1)} - \mathcal{L}\left(\left(\boldsymbol{A}^*\boldsymbol{A}^{*\top}\right)^{-1}\right)^\top\right\|_F\right).\end{aligned} \tag{3.17}$$

Via a similar argument, we can write

$$\|\boldsymbol{D}^{(s)} - \boldsymbol{D}^*\|_F \le \|\boldsymbol{A}^{(s)} - \boldsymbol{A}^*\|_F + O\left(\left\|\boldsymbol{P}^{(s)} - \mathcal{L}\left(\left(\boldsymbol{A}^*\boldsymbol{A}^{*\top}\right)^{-1}\right)^\top\right\|_F\right). \tag{3.18}$$

Combining (3.17) and (3.18), we have

$$\begin{aligned}&\|\boldsymbol{A}^{(s+1)} - \boldsymbol{A}^*\|_F \le 0.9\|\boldsymbol{A}^{(s)} - \boldsymbol{A}^*\|_F\\ &+ O\left(\left\|\boldsymbol{P}^{(s)} - \mathcal{L}\left(\left(\boldsymbol{A}^*\boldsymbol{A}^{*\top}\right)^{-1}\right)^\top\right\|_F + \|\boldsymbol{P}^{(s)}\boldsymbol{A}^* - \boldsymbol{D}^*\|_F + \left\|\boldsymbol{P}^{(s+1)} - \mathcal{L}\left(\left(\boldsymbol{A}^*\boldsymbol{A}^{*\top}\right)^{-1}\right)^\top\right\|_F\right).\end{aligned}$$

By invoking Lemma 3.3, we can bound $\|\boldsymbol{P}^{(s)}\boldsymbol{A}^* - \boldsymbol{D}^*\|_F$, $\|\boldsymbol{P}^{(s)} - \mathcal{L}((\boldsymbol{A}^*\boldsymbol{A}^{*\top})^{-1})^\top\|_F$, and $\|\boldsymbol{P}^{(s+1)} - \mathcal{L}((\boldsymbol{A}^*\boldsymbol{A}^{*\top})^{-1})^\top\|_F$ with $O(\frac{\hat{\kappa}^6}{\theta}\sqrt{\frac{n}{p_1+T}})$. This completes the proof. ■

3.3. Proof of Theorem 2.3. Theorem 2.3 is a special case of Theorem 2.4, where we use one precalculated preconditioner throughout. In particular, it is easy to see that (3.16) in the proof of Theorem 2.4 holds after replacing $\boldsymbol{P}^{(s)}$ with $\boldsymbol{P}$:

$$\|\boldsymbol{D}^{(s+1)} - \boldsymbol{D}^*\|_F \le 0.9\|\boldsymbol{D}^{(s)} - \boldsymbol{D}^*\|_F + 12\|\boldsymbol{P}\boldsymbol{A}^* - \boldsymbol{D}^*\|_F. \tag{3.19}$$

Therefore, we have

$$\begin{aligned}\|\boldsymbol{D}^{(s+1)} - \boldsymbol{D}^*\|_F &\le 0.9\|\boldsymbol{D}^{(s)} - \boldsymbol{D}^*\|_F + 12\|\boldsymbol{P}\boldsymbol{A}^* - \boldsymbol{D}^*\|_F \\ &\le 0.9^s\|\boldsymbol{D}^{(0)} - \boldsymbol{D}^*\|_F + (0.9 + \cdots + 0.9^s)\, 12\|\boldsymbol{P}\boldsymbol{A}^* - \boldsymbol{D}^*\|_F \\ &= 0.9^s\|\boldsymbol{D}^{(0)} - \boldsymbol{D}^*\|_F + O(\|\boldsymbol{P}\boldsymbol{A}^* - \boldsymbol{D}^*\|_F).\end{aligned}$$

Given the above inequality, one can use the same argument as in (3.17) and (3.18) to establish the linear convergence of $\boldsymbol{A}^{(s)}$. The details are omitted for brevity. ■

4. Numerical experiments.

In this section, we validate our theoretical results using synthetic and real data. All experiments are performed on a MacBook Pro 2021 with the Apple M1 Pro chip and a 16 GB unified memory for a serial implementation in MATLAB 2022a. The code is available at: https://github.com/lianggeyuleo/CompleteDL.git.

4.1. Synthetic dataset.

We validate our theoretical results using a synthetic dataset. We consider the generative model $\boldsymbol{Y} = \boldsymbol{A}^*\boldsymbol{X}^*$ where $\boldsymbol{A}^*$ is a randomly generated orthogonal or full-rank dictionary. Moreover, $\boldsymbol{X}^*$ is generated from a Gaussian–Bernoulli distribution followed by a truncation step, where the entries $\boldsymbol{X}^*_{ij}$ with $|\boldsymbol{X}^*_{ij}| < \Gamma = 0.3$ are replaced by $\boldsymbol{X}^*_{ij} = \mathrm{sgn}(\boldsymbol{X}^*_{ij})\Gamma$. Consequently, the threshold ζ in each algorithm is set to $\Gamma/2 = 0.15$.

First, we compare the efficiency of our method in solving (ODL) to other state-of-the-art techniques. In this experiment, we vary the signal dimension n from 50 to 800 and set the sample size $p = 100n$, which corresponds to the linear sample size condition in Theorem 2.2. We invoke the warm-start method in Algorithm 2.4 to generate the initial points. For each trial, we stop the algorithm when the consecutive iterates are close to each other ($\|\boldsymbol{D}^{(t-1)} - \boldsymbol{D}^{(t-1)}\|_2 \le 10^{-7}$) and record the running time and final error ($\|\boldsymbol{D}^{(T)} - \boldsymbol{D}^*\|_F$).

Many algorithms for dictionary learning exhibit poor scalability in this setting. For instance, the convergence time of the *KSVD method* [2], which is perhaps the most well-known alternating minimization algorithm for dictionary learning, exceeds 300 s even for our smallest instance $n = 50$. Another example is the alternating minimization based on ℓ_1-regularization [1], which suffers from a similar scalability issue. Instead, in Figure 3, we consider two other candidate algorithms that have comparable scalability with ours: (1) *Gradient-based method* [3]: In this method, the dictionary update step (step 4 in Algorithm 2.1) is replaced by a gradient step $\boldsymbol{D}^{(t+1)} = \boldsymbol{D}^{(t)} - \eta\nabla_{\boldsymbol{D}^{(t)}} L$ where L is the objective of (ODL). Here, we pick $\eta = 10^{-5}$ after fine-tuning, which is the largest step-size to guarantee convergence in practice. (2) *ℓ_4-maximization–based method* [43]: The work [43] introduces a projected gradient ascent for maximizing the objective function $f(\boldsymbol{D}) = \|\boldsymbol{D}^\top\boldsymbol{Y}\|_4^4$. This method is shown to have superior performance compared to SPAMS [21] and the subgradient descent method [5]. As can be

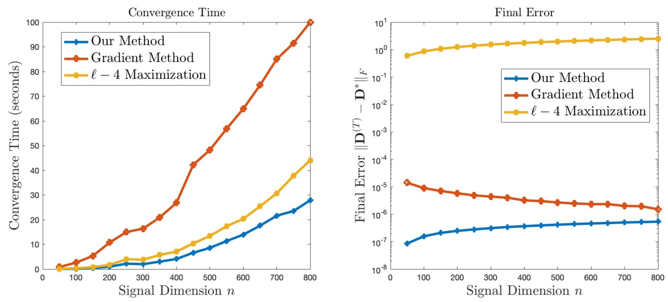

Figure 3. *We compare three different dictionary learning methods with their running time and final error until convergence. The results above are averaged over* 5 *independent trials. All methods use the same initial point. The stopping criterion is when consecutive iterates are close to each other* ($\|\boldsymbol{D}^{(t-1)}-\boldsymbol{D}^{(t-1)}\|_2\leq 10^{-7}$)*.*

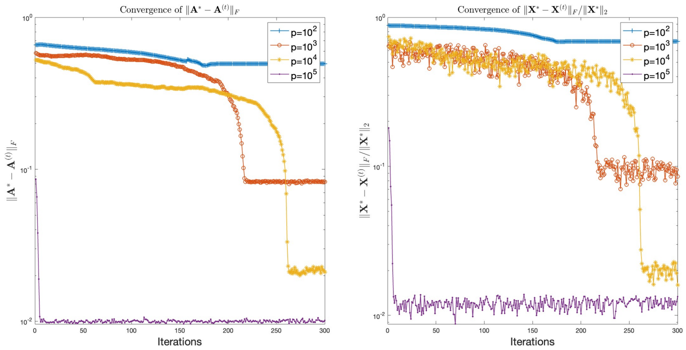

Figure 4. *Results of Algorithm* 2.2 *with* $n=5$*,* $\theta=0.3$*,* $\tilde{p}=100$*, and varying* p*. The specific number of iterations to reach convergence varies with the sample size since it depends on the distance between the ground truth and initialization.*

seen in Figure 3, our method converges faster while achieving a smaller error. Particularly, the gradient-based method will require more time to converge, while ℓ_4-maximization is restricted by its poor accuracy.

In Figure 4, we report the performance of Algorithm 2.2 with warm-start for varying sample sizes. The true dictionary $\boldsymbol{A}^*$ is a randomly generated full-rank $n\times n$ matrix, and we choose $n=5$ and $\theta=0.3$. We set $\tilde{p}=10^2$ and vary p. Our experiments support our theoretical results in the following aspects: (1) We observe the fast convergence of our algorithm for different p, which is in line with Theorem 2.3. Similar to Figure 2, the effect of the warm-up phase is evident in the early stages of the iterations. (2) We see clear improvement in the accuracy of the final solution with a larger sample size. Such improvement is characterized by Theorem 2.3 with the additional error term, which diminishes to zero as $p\to\infty$.

Finally, we numerically test the stability and sample complexity of our algorithm as $\boldsymbol{A}^*$ becomes increasingly ill-conditioned. In Figure 5 (right), we present the results, illustrating the sample size required to achieve a final error of $\frac{\|\boldsymbol{A}^{(T)}-\boldsymbol{A}^*\|_F}{\|\boldsymbol{A}^*\|_F}\leq 0.1$ as the condition number of $\boldsymbol{A}^*$ varies from 1 to 1000. Surprisingly, our practical findings indicate that the relationship

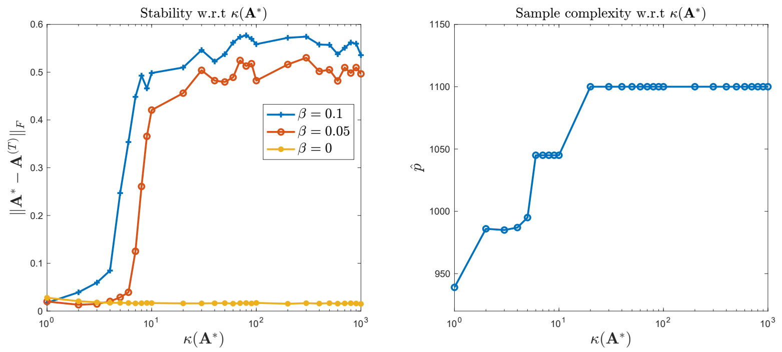

Figure 5. *Left: Final error of Algorithm 2.2 in relation to condition number $\kappa(\boldsymbol{A}^*)$ and varied noise levels β for a fixed sample size of $\tilde{p} = p = 10^5$. Right: Required sample size to achieve $\frac{\|\boldsymbol{A}^{(T)} - \boldsymbol{A}^*\|_F}{\|\boldsymbol{A}^*\|_F} \le 0.1$ as a function of the condition number $\kappa(\boldsymbol{A}^*)$. In both settings, we fix $n = 5$, $\theta = 0.3$, and $T = 1000$.*

between sample size and the condition number differs from what our theoretical framework suggests, showcasing a more favorable scenario in practice.

To further underscore the robustness of our algorithm, we investigate its performance in a noisy setting, where $\boldsymbol{Y} = \boldsymbol{A}^*\boldsymbol{X}^* + \mathcal{E}$, with $\mathcal{E}$ representing a noise matrix whose elements follow a Gaussian distribution with zero mean and variance β^2. As illustrated in Figure 5 (left), our algorithm remains unaffected by the increasing condition number $\kappa(\boldsymbol{A}^*)$ in the noiseless scenario. However, it encounters difficulties in recovering $\boldsymbol{A}^*$ when both noise and ill-conditioning are present. We acknowledge that a theoretical explanation for this phenomenon in the noisy setting remains a topic for future investigation.

4.2. Real dataset. In this section, we showcase the performance of Algorithm 2.3 in learning a dictionary for the Landscape Dataset [15] consisting of 7,000 colored images of different landscapes, 20 of which are set aside as a test dataset. To gauge the quality of our learned dictionary, we use our dictionary to perform a denoising task on these images.

Dataset. Each figure in our dataset is a colored image of size $150 \times 150 \times 3$. Instead of directly learning a dictionary for the whole dataset, we follow the procedure in [2] and divide each figure into 225 patches of size $10 \times 10 \times 3$. Each patch is then reshaped into a 300×1 vector. Collecting all 225 patches over all 7000 images results in 1,575,000 patch samples. We downsample this dataset to 10,000 patch samples so that KSVD can also be applied to this dataset. This results in a data matrix $\boldsymbol{Y}$ of size $300 \times 10{,}000$ with $n = 300$ and $p = 10{,}000$. Our goal is to learn a dictionary $\boldsymbol{A}$ of size 300×300 for this data matrix.

Image denoising. We gauge the quality of our learned dictionary by using it to denoise noisy images. Here, we corrupt each image with 50% missing pixels. In other words, we select 50% of the 150×150 pixels in each image uniformly at random and then set the pixel values across all three RBG channels to 0. Our goal is to learn a dictionary and use it to denoise this image by filling in the missing pixels. For reconstruction, we choose a sparsity level of approximately 10%, which simply corresponds to 30 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 [28].

For both the image denoising examples in Figures 1 and 7, we set $p_1 = p_2 = 500$ in Algorithm 2.3 (corresponding to a batch size of 500) and run it for a total of $T = 2000$ iterations, which took 125 seconds for the constructed data matrix $\boldsymbol{Y}$. We randomly sample 20 images from the dataset, which are then corrupted with noise. The images before and after the denoising are shown in the first two rows of Figure 7. Here we see that our learned dictionary reconstructs the original image almost exactly.

Additionally, we compare our learned dictionary with that of KSVD, which is implemented using a standard sparse learning library [28] in MATLAB. We allow KSVD a total running time of 1100 seconds. A comparison of denoised results using dictionaries learned with Algorithm 2.3 and KSVD is shown in Figure 1. Here, we also see that the dictionary learned by Algorithm 2.3 greatly outperforms KSVD, achieving a much better reconstruction of the original image despite being 10 times faster. Furthermore, we plot the similarity score between the reconstructed image and the original image versus the running time of both algorithms. In particular, at each iteration of both algorithms, we use the dictionary at that iteration to reconstruct a denoised image and compare it with the original. We see in Figure 6 that our algorithm achieves a similarity score of 0.9 in 120 s, while KSVD can only achieve a similarity score of 0.28 in 2500 s.

Image inpainting. To further gauge the efficacy of our learned dictionary, here we consider the more challenging task of *inpainting.* Here instead of corrupting each image with 50% of missing pixels, we opt to black out entire pixel blocks. We segment each image into patches with dimensions of 25×25 to facilitate inpainting for larger areas of obscured pixels. Much like our previous approach, we execute Algorithm 2.3 for a total of $T = 2{,}000$, which consumed approximately 2,100 s. For the reconstruction process, we maintain a sparsity level of approximately 10%, as we did previously.

Our reconstruction results are depicted in Figure 7. We randomly black out blocks of dimensions 3×3, 5×5, and 10×10. In all three scenarios, the total count of obscured blocks equals the total number of patches. It is evident that for smaller blacked-out blocks, our dictionary performs well in reconstructing the images. However, when dealing with larger

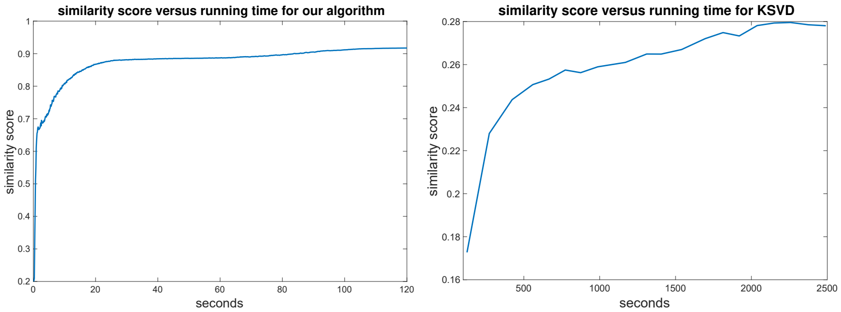

Figure 6. Similarity score between reconstructed image and original image. *We display the similarity score between the denoised image and the original one for our algorithm (left) and KSVD (right). As time progresses, the dictionary obtained from our algorithm improves, leading to increasingly better reconstructions. On the other hand, while the dictionary obtained from KSVD improves with time, the reconstruction quality exhibits a slower rate of improvement.*

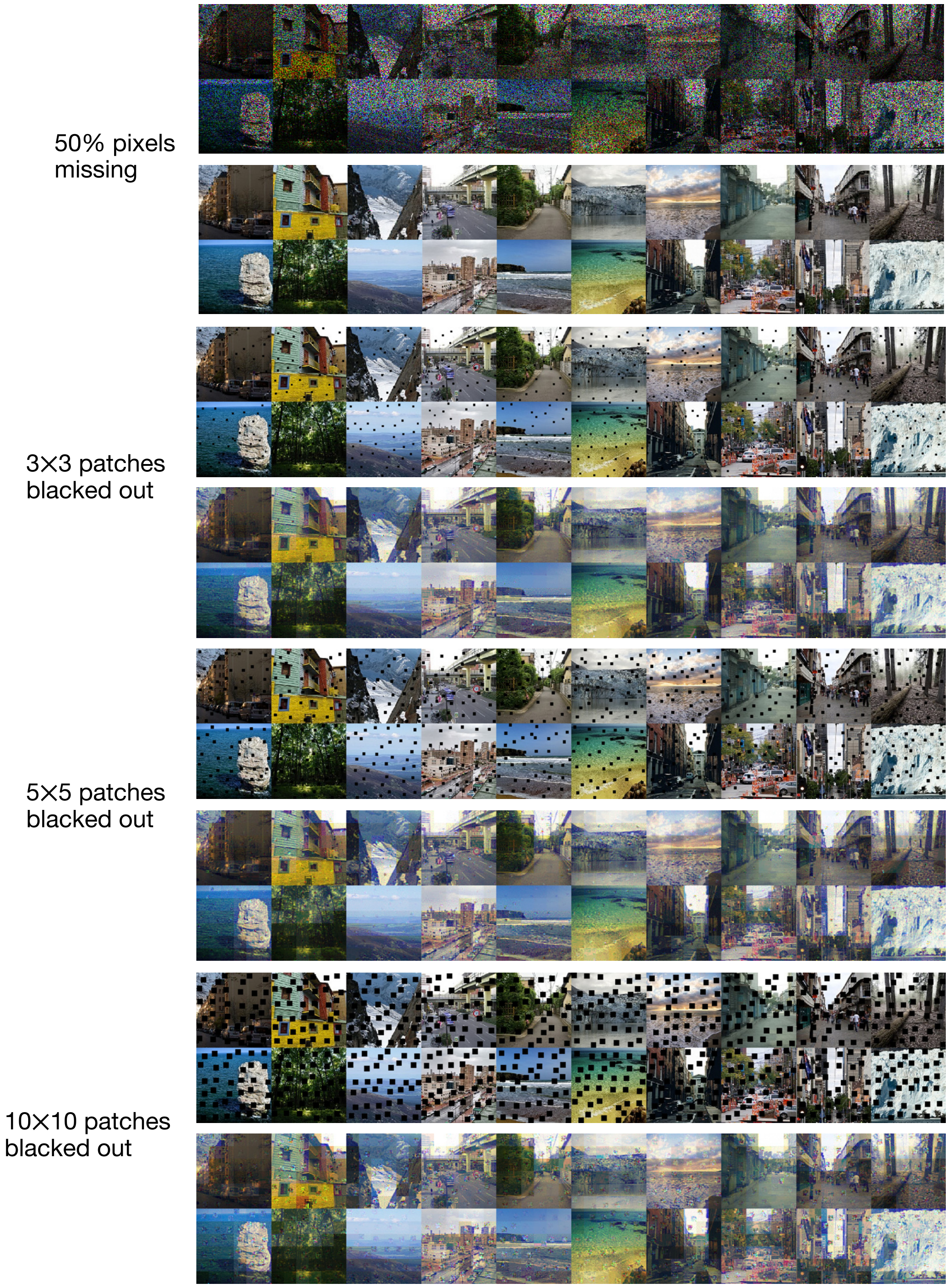

Figure 7. Image denoising and inpainting. *Given a dataset of multiple landscape images, we divided each image into patches and combined them to form a large data matrix. A dictionary is learned using our mini-batch alternating minimization algorithm. A total of* 20 *noisy images are shown with different noise patterns, alongside their respective reconstructed versions.*

10×10 blocks, the inpainting task becomes more challenging. In this case, we observe that the reconstructed images exhibit some imperfections within the blacked-out regions.

5. Conclusion.

In this paper, we study the 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. This paper shows 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 synthetic and realistic experiments on image denoising, we showcase the superiority of our proposed algorithm compared to the most commonly used algorithms for dictionary learning.

Appendix A. Preliminary.

To prove our main theorems, we will rely on several preliminary results from high-dimensional statistics and matrix perturbation theory, which will be essential for our arguments. We denote the sub-Gaussian norm and L^2-norm of a random variable with $\|\cdot\|_{\psi_2}$ and $\|\cdot\|_{L^2}$, respectively.

Theorem A.1 (concentration of sample covariance matrix [42]). *Let $\boldsymbol{x}$ be a sub-Gaussian random vector in $\mathbb{R}^n$ with covariance matrix $\boldsymbol{\Sigma}$, such that*

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

for some $C_{ce} \ge 1$. Let $\boldsymbol{X} \in \mathbb{R}^{n\times p}$ be a matrix whose columns have identical and independent distribution as $\boldsymbol{x}$. Then, for any $u \ge 0$ and with probability at least $1 - 2\exp(-u)$, we have

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

Theorem A.2 (concentration of norm [42]). *Let $\boldsymbol{x} \in \mathbb{R}^n$ be a random vector with independent, sub-Gaussian coordinates $\boldsymbol{x}_i$ that satisfy $\mathbb{E}\boldsymbol{x}_i^2 = 1$. Then, $\|\|\boldsymbol{x}\|_2 - \sqrt{n}\|_{\psi_2} \le CK^2$, where $K = \max_i \|\boldsymbol{x}_i\|_{\psi_2}$ and C is an absolute constant.*

We next introduce a perturbation bound for the polar decomposition.

Theorem A.3 (perturbation bound for polar decomposition [24]). *Let $\boldsymbol{A}, \tilde{\boldsymbol{A}} \in \mathbb{R}^{n\times n}$ be full-rank matrices with polar decompositions $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{P}$ and $\tilde{\boldsymbol{A}} = \tilde{\boldsymbol{U}}\tilde{\boldsymbol{P}}$. Then*

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

Appendix B. Proof of auxiliary lemmas.

B.1. Proof of Lemma 3.4.

Consider each column vector of $\boldsymbol{X}^*$ as a random vector. Upon defining $\Sigma_{\boldsymbol{X}^*}$ as the covariance matrix of $\boldsymbol{x}$, we have $\Sigma_{\boldsymbol{X}} = \theta\sigma^2\boldsymbol{I}_n$. We now use Theorem A.1 to prove Lemma 3.4. For any unit-norm $\boldsymbol{z} \in \mathbb{R}^n$ and any ith column of $\boldsymbol{X}^*$, we have

$$\left\|\langle \boldsymbol{X}^*_{(\cdot,i)}, \boldsymbol{z}\rangle\right\|_{\psi_2}^2 = \left\|\sum_{j=1}^n \boldsymbol{X}^*_{(j,i)} z_j\right\|_{\psi_2}^2 \le \sum_{j=1}^n z_j^2\left\|\boldsymbol{X}^*_{(j,i)}\right\|_{\psi_2}^2 \le C_1\sigma^2$$

for some constant $C_1 > 0$. The last inequality follows from the fact that, for a sub-Gaussian random variable, the sub-Gaussian norm is always bounded by its variance. Moreover,

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

We are now ready to invoke Theorem A.1. By choosing $C_{ce} = C_1/\sqrt{\theta}$, we have for some constant $C_2 > 0$

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

with probability $1 - 2\exp(-u)$. Upon setting $u = n$ and assuming $p \geq 8C_2^2 \tau n/\theta^2$, one can bound the right-hand side by

$$C_2 \left(\sqrt{\frac{n+u}{p}} + \frac{n+u}{p} \right) \sigma^2 \leq 2C_2 \sqrt{\frac{2n}{p}} \sigma^2 \leq \frac{\theta\sigma^2}{\tau^{1/2}}.$$

Combining the above inequality with (B.1) leads to the desired inequalities. ∎

B.2. Proof of Lemma 3.3. Let us define $p = p_1 + t$. We start by noting that

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

Moreover, according to Theorem A.1, we have

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

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

$$\left\| \frac{1}{p\theta\sigma^2} \boldsymbol{Y}\boldsymbol{Y}^\top - \boldsymbol{A}^*\boldsymbol{A}^{*\top} \right\|_2 = \left\| \boldsymbol{A}^* \left(\frac{1}{p\theta\sigma^2} \boldsymbol{X}^* \boldsymbol{X}^{*\top} - \boldsymbol{I}_{n\times n} \right) \boldsymbol{A}^{*\top} \right\|_2 \leq \frac{C_2}{\theta} \sqrt{\frac{n+u}{p}}$$

where in the last inequality, we used $\|\boldsymbol{A}^*\|_2 = 1$. We define $\boldsymbol{\Delta}_1 = \frac{1}{p\theta\sigma^2} \boldsymbol{Y}\boldsymbol{Y}^\top - \boldsymbol{A}^*\boldsymbol{A}^{*\top}$. Using the Taylor expansion, we have

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

As a result, we have

$$\left\| \left(\frac{1}{p\theta\sigma^2} \boldsymbol{Y}\boldsymbol{Y}^\top \right)^{-1} - \left(\boldsymbol{A}^*\boldsymbol{A}^{*\top} \right)^{-1} \right\|_2 C_3 \frac{\hat{\kappa}^4}{\theta} \sqrt{\frac{n+u}{p}}$$

for some constant $C_3 > 0$. Similarly, by Corollary 4.8 from [9], we have

$$\left\| \mathcal{L}\left(\left(\frac{1}{p\theta\sigma^2}\boldsymbol{Y}\boldsymbol{Y}^\top\right)^{-1}\right) - \mathcal{L}\left(\left(\boldsymbol{A}^*\boldsymbol{A}^{*\top}\right)^{-1}\right)\right\|_2$$
$$\leq C_3\kappa\left(\left(\boldsymbol{A}^*\boldsymbol{A}^{*\top}\right)^{-1}\right)\left\|\left(\frac{1}{p\theta\sigma^2}\boldsymbol{Y}\boldsymbol{Y}^\top\right)^{-1} - \left(\boldsymbol{A}^*\boldsymbol{A}^{*\top}\right)^{-1}\right\|_2 \leq C_3\frac{\hat{\kappa}^6}{\theta}\sqrt{\frac{n+u}{p}}.$$

We conclude the proof by setting $u = n$. ■

B.3. Proof of Lemma 3.2. Upon choosing $\tau = 1$ in Lemma 3.4, we have $\kappa(\boldsymbol{X}^*) < 11/9$ with probability $1-2\exp\{-n\}$. Then, by setting $\boldsymbol{E} = (\boldsymbol{X}-\boldsymbol{X}^*)/\|\boldsymbol{X}^*\|_2$ and $\widehat{\boldsymbol{X}}^* = \boldsymbol{X}^*/\|\boldsymbol{X}^*\|_2$, we immediately have

$$\|\boldsymbol{D}-\boldsymbol{D}^*\|_F \leq \frac{\kappa(\boldsymbol{X}^*)^4}{2}\left\|\widehat{\boldsymbol{X}}^*\boldsymbol{E}^\top - \boldsymbol{E}\widehat{\boldsymbol{X}}^{*\top}\right\|_F + C_T\|\boldsymbol{E}\|_F^2 \tag{B.3}$$

after invoking Lemma 3.5. 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 $\frac{1}{2}\|\widehat{\boldsymbol{X}}^*\boldsymbol{E}^\top - \boldsymbol{E}\widehat{\boldsymbol{X}}^{*\top}\|_F \leq \|\mathcal{T}(\widehat{\boldsymbol{X}}^*\boldsymbol{E}^\top)\|_F$. 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 $i\in \operatorname{supp}(\widehat{\boldsymbol{X}}^*_{(k,\cdot)})$ and zeros elsewhere. Right multiplying $\tilde{\boldsymbol{X}}^*$ by $\tilde{\boldsymbol{M}}_k$ corresponds to replacing all the columns that are zero at the kth row with zeros. Now, we make the following observation:

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

The second equality is due to $\operatorname{supp}(\boldsymbol{E}) \subseteq \operatorname{supp}(\widehat{\boldsymbol{X}}^*)$ and that the normalization step $\widehat{\boldsymbol{X}}^* = \boldsymbol{X}^*/\|\boldsymbol{X}^*\|_2$ does not change the support of $\widehat{\boldsymbol{X}}^*$. We first focus on $\max_{1\leq k\leq n}\|\boldsymbol{M}_k\boldsymbol{X}^*\tilde{\boldsymbol{M}}_k\|_2$. 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 see that $\|\boldsymbol{G}^k\|_2 = \|\boldsymbol{M}_k\boldsymbol{X}^*\tilde{\boldsymbol{M}}_k\|_2$. Without loss of generality, we assume $k=n$, which means that we remove the last row. Recall that right multiplying $\boldsymbol{X}^*$ by $\tilde{\boldsymbol{M}}_k$ will replace each column of $\boldsymbol{X}^*$ whose kth row is zero by an all-zero column vector. Therefore, $\boldsymbol{G}^k$ has the following property:

$$\mathbb{1}_{\boldsymbol{G}^k_{ij}\neq 0} = B_{kj}B_{ij}, \quad \text{where} \quad B_{kj}, B_{ij} \overset{\text{i.i.d.}}{\sim} \mathcal{B}(\theta) \text{ for } 1\leq i\leq n-1,\ 1\leq j\leq p\ .$$

In short, $\boldsymbol{G}^k$ is a matrix that satisfies Assumption 2.1 with parameter θ^2. Therefore, we can invoke Lemma 3.4 to bound $\|\boldsymbol{G}^k\|_2$ for each k. Given $p\gtrsim \tau(n+u)/\theta^2$, we have that $\|\boldsymbol{G}^k\|_2 \leq (1+\tau^{-1/4})\sqrt{p}\theta\sigma$ for some specific k, with probability $1-2\exp(-u)$. To bound maximal $\|\boldsymbol{G}^k\|_2$ for $1\leq k\leq n$, we take the union bound and obtain

$$\max_{1\leq k\leq n}\left\|\boldsymbol{M}_k\boldsymbol{X}^*\tilde{\boldsymbol{M}}_k\right\|_2 = \max_{1\leq k\leq n}\|\boldsymbol{G}^k\|_2 \leq \left(1+\tau^{-1/4}\right)\sqrt{p}\theta\sigma, \tag{B.4}$$

with probability $1-2\exp(\log n - u)$ as long as $p \gtrsim \tau(n+u)/\theta^2$. Combining this result with (B.3), we have

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

with probability $1-2\exp(\log n - u)$. Inequality (a) is due to Lemma 3.4 and (B.4). Recall that $\theta \le 1/2$. Therefore, for sufficiently large constant τ, we have $(\frac{1+\tau^{-1/4}}{1-\tau^{-1/4}})^5\sqrt{\theta} \le 0.8$. Moreover, $C_T\|\boldsymbol{E}\|_F \le 0.1$, provided that $\|\boldsymbol{E}\|_F \le 1/(10C_T)$. Therefore, we have $\|\boldsymbol{D}-\boldsymbol{D}^*\|_F \le 0.9\|\boldsymbol{E}\|_F$. The proof is complete upon choosing $u = n$. ∎

B.4. Proof of Lemma 3.1. The (i,j)th entry $(\boldsymbol{D}^\top\boldsymbol{Y})_{ij}$ of $\boldsymbol{D}^\top\boldsymbol{Y}$ can be written as

$$\text{(B.5)}\qquad \left(\boldsymbol{D}^\top\boldsymbol{Y}\right)_{ij} = \langle \boldsymbol{D}_{(\cdot,i)}, \boldsymbol{Y}_{(\cdot,j)}\rangle = \underbrace{\langle \boldsymbol{D}_{(\cdot,i)}, \boldsymbol{D}^*_{(\cdot,i)}\rangle \boldsymbol{X}^*_{ij}}_{:=\mathcal{A}_{ij}} + \underbrace{\sum_{k\ne i}\langle \boldsymbol{D}_{(\cdot,i)}, \boldsymbol{D}^*_{(\cdot,k)}\rangle \boldsymbol{X}^*_{kj}}_{:=\mathcal{B}_{ij}}.$$

The first term $\mathcal{A}_{ij}$ can be decomposed as

$$\mathcal{A}_{ij} = \langle \boldsymbol{D}_{(\cdot,i)}, \boldsymbol{D}_{(\cdot,i)}\rangle \boldsymbol{X}^*_{ij} + \langle \boldsymbol{D}_{(\cdot,i)}, \boldsymbol{D}^*_{(\cdot,i)} - \boldsymbol{D}_{(\cdot,i)}\rangle \boldsymbol{X}^*_{ij} = \boldsymbol{X}^*_{ij} + \langle \boldsymbol{D}_{(\cdot,i)}, \boldsymbol{D}^*_{(\cdot,i)} - \boldsymbol{D}_{(\cdot,i)}\rangle \boldsymbol{X}^*_{ij}.$$

As a result, when $\boldsymbol{X}^*_{ij} = 0$, we have $\mathcal{A}_{ij} = 0$. Moreover, when $\boldsymbol{X}^*_{ij} \ne 0$, we have $|\mathcal{A}_{ij}| \ge (1-\|\boldsymbol{D}^*-\boldsymbol{D}\|_{1,2})\Gamma \ge \frac{3\Gamma}{4}$, where the last inequality holds for every $n \ge \frac{16}{\sigma^2\theta}$. The choice of the constant $3/4$ is to streamline the proof and can be replaced by any constant in $(\frac{1}{2}, 1)$. To prove the lemma it suffices to show that, for all $\boldsymbol{D}$ such that $\|\boldsymbol{D}-\boldsymbol{D}^*\|_{1,2} \le C_1/n$ for some C_1 to be defined later, we have $|\mathcal{B}_{ij}| \le \Gamma/4$ with high probability. Note that this will lead to $|(\boldsymbol{D}^\top\boldsymbol{Y})_{ij}| \ge |\mathcal{A}_{ij}| - |\mathcal{B}_{ij}| \ge \Gamma/2$, thereby proving the statement. We have

$$\text{(B.6)}\qquad \begin{aligned}|\mathcal{B}_{ij}| &= \left|\left\langle \boldsymbol{D}_{(\cdot,i)}, \sum_{k\ne i}\boldsymbol{D}^*_{(\cdot,k)}\boldsymbol{X}^*_{kj}\right\rangle\right|\\ &= \left|\left\langle \boldsymbol{D}_{(\cdot,i)} - \boldsymbol{D}^*_{(\cdot,i)}, \sum_{k\ne i}\boldsymbol{D}^*_{(\cdot,k)}\boldsymbol{X}^*_{kj}\right\rangle\right|\\ &\le \left\|\boldsymbol{D}_{(\cdot,i)} - \boldsymbol{D}^*_{(\cdot,i)}\right\|_2 \left\|\sum_{k\ne i}\boldsymbol{D}^*_{(\cdot,k)}\boldsymbol{X}^*_{kj}\right\|_2\end{aligned}$$

Algorithm B.1. Rank-one update for the preconditioner.

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

$$\text{(B.7)} \qquad \begin{aligned} &\le \left\|\boldsymbol{D}_{(\cdot,i)} - \boldsymbol{D}^*_{(\cdot,i)}\right\|_2 \left\|\sum_{k=1}^n \boldsymbol{D}^*_{(\cdot,k)}\boldsymbol{X}^*_{kj}\right\|_2 \\ &= \left\|\boldsymbol{D}_{(\cdot,i)} - \boldsymbol{D}^*_{(\cdot,i)}\right\|_2 \left\|\boldsymbol{Y}_{(\cdot,j)}\right\|_2 \\ &= \left\|\boldsymbol{D}_{(\cdot,i)} - \boldsymbol{D}^*_{(\cdot,i)}\right\|_2 \left\|\boldsymbol{X}_{(\cdot,j)}\right\|_2. \end{aligned}$$

Following the argument of (3.7), we have $\|\boldsymbol{X}_{(\cdot,j)}\|_2 \le 2\sigma\sqrt{\theta n}$ with probability at least $1 - 2\exp(-Cn)$. With the same probability, we have the following for any $\boldsymbol{D}$ that satisfies $\|\boldsymbol{D} - \boldsymbol{D}^*\|_{1,2} \le \frac{\Gamma}{8\sigma\sqrt{\theta n}}$:

$$|\mathcal{B}_{ij}| \le \left\|\boldsymbol{D}_{(\cdot,i)} - \boldsymbol{D}^*_{(\cdot,i)}\right\|_2 \left\|\boldsymbol{X}_{(\cdot,j)}\right\|_2 \le \Gamma/4.$$

Upon taking the union bound over all the entries of the jth column of $\boldsymbol{X}$, we have $|\mathcal{B}_{ij}| < \Gamma/4$ for all (i,j) with probability $1 - 2\exp(\log p - Cn)$. This completes the proof of the second statement. ■

Appendix C. Rank-one updates for the preconditioner.

We define

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

The above rank-one update algorithm for Cholesky decomposition (Algorithm B.1) is already implemented in the MATLAB function `cholupdate`.

REFERENCES

[1] A. Agarwal, A. Anandkumar, P. Jain, and P. Netrapalli, *Learning sparsely used overcomplete dictionaries via alternating minimization*, SIAM J. Optim., 26 (2016), pp. 2775–2799, https://doi.org/10.1137/140979861.
[2] M. Aharon, M. Elad, and A. Bruckstein, *K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation*, IEEE Trans. Signal Process., 54 (2006), pp. 4311–4322, https://doi.org/10.1109/TSP.2006.881199.

[3] S. Arora, R. Ge, T. Ma, and A. Moitra, *Simple, efficient, and neural algorithms for sparse coding*, in Conference on Learning Theory, PMLR, 2015, pp. 113–149.
[4] S. Arora, R. Ge, and A. Moitra, *New algorithms for learning incoherent and overcomplete dictionaries*, in Conference on Learning Theory, PMLR, 2014, pp. 779–806.
[5] Y. Bai, Q. Jiang, and J. Sun, *Subgradient Descent Learns Orthogonal Dictionaries*, preprint, arXiv:1810.10702, 2018.
[6] C. Bao, J.-F. Cai, and H. Ji, *Fast sparsity-based orthogonal dictionary learning for image restoration*, in Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 3384–3391.
[7] C. Bao, H. Ji, Y. Quan, and Z. Shen, *ℓ_0 norm based dictionary learning by proximal methods with global convergence*, in Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 3858–3865.
[8] B. Barak, J. A. Kelner, and D. Steurer, *Dictionary learning and tensor decomposition via the sum-of-squares method*, in Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing, 2015, pp. 143–151.
[9] R. Bhatia, *Matrix factorizations and their perturbations*, Linear Algebra Appl., 197 (1994), pp. 245–276.
[10] N. Chatterji and P. L. Bartlett, *Alternating minimization for dictionary learning with random initialization*, in Advances in Neural Information Processing Systems 30, 2017.
[11] J. E. Cohen and N. Gillis, *Identifiability of complete dictionary learning*, SIAM J. Math. Data Sci., 1 (2019), pp. 518–536, https://doi.org/10.1137/18M1233339.
[12] K. Engan, S. O. Aase, and J. H. Husoy, *Method of optimal directions for frame design*, in 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing, IEEE, 1999, pp. 2443–2446.
[13] R. Ge, C. Jin, and Y. Zheng, *No spurious local minima in nonconvex low rank problems: A unified geometric analysis*, in International Conference on Machine Learning, PMLR, 2017, pp. 1233–1242.
[14] Q. Geng and J. Wright, *On the local correctness of ℓ_1-minimization for dictionary learning*, in 2014 IEEE International Symposium on Information Theory, IEEE, 2014, pp. 3180–3184.
[15] B. Ghimire, *Landscape Color and Grayscale Images*, https://www.kaggle.com/datasets/theblackmamba31/landscape-image-colorization (accessed 2023).
[16] D. Gilboa, S. Buchanan, and J. Wright, *Efficient dictionary learning with gradient descent*, in International Conference on Machine Learning, PMLR, 2019, pp. 2252–2259.
[17] R. Gribonval, R. Jenatton, and F. Bach, *Sparse and spurious: Dictionary learning with noise and outliers*, IEEE Trans. Inform. Theory, 61 (2015), pp. 6298–6319, https://doi.org/10.1109/TIT.2015.2472522.
[18] R. Gribonval and K. Schnass, *Dictionary identification—sparse matrix-factorization via ℓ_1-minimization*, IEEE Trans. Inform. Theory, 56 (2010), pp. 3523–3539, https://doi.org/10.1109/TIT.2010.2048466.
[19] J. Hu and K. Huang, *Global identifiability of ℓ_1-based dictionary learning via matrix volume optimization*, in Advances in Neural Information Processing Systems 36, 2023, pp. 36165–36186.
[20] R. Jenatton, R. Gribonval, and F. Bach, *Local Stability and Robustness of Sparse Dictionary Learning in the Presence of Noise*, preprint, arXiv:1210.0685, 2012.
[21] R. Jenatton, J. Mairal, G. Obozinski, and F. R. Bach, *Proximal methods for sparse hierarchical dictionary learning*, in Proceedings of the 27th International Conference on Machine Learning, 2010, pp. 487–494.
[22] O. Krause and C. Igel, *A more efficient rank-one covariance matrix update for evolution strategies*, in Proceedings of the 2015 ACM Conference on Foundations of Genetic Algorithms XIII, 2015, pp. 129–136.
[23] H. Lee, A. Battle, R. Raina, and A. Ng, *Efficient sparse coding algorithms*, in Advances in Neural Information Processing Systems 19, 2006.
[24] R.-C. Li, *New perturbation bounds for the unitary polar factor*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 327–332, https://doi.org/10.1137/S0895479893256359.
[25] M. Liu, L. Nie, X. Wang, Q. Tian, and B. Chen, *Online data organizer: Micro-video categorization by structure-guided multimodal dictionary learning*, IEEE Trans. Image Process., 28 (2018), pp. 1235–1247, https://doi.org/10.1109/TIP.2018.2875363.
[26] C. Lu, J. Shi, and J. Jia, *Online robust dictionary learning*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 415–422.

[27] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, *Online dictionary learning for sparse coding*, in Proceedings of the 26th Annual International Conference on Machine Learning, 2009, pp. 689–696.
[28] J. Mairal, M. Elad, and G. Sapiro, *Sparse representation for color image restoration*, IEEE Trans. Image Process., 17 (2007), pp. 53–69, https://doi.org/10.1109/TIP.2007.911828.
[29] B. Malézieux, T. Moreau, and M. Kowalski, *Understanding approximate and unrolled dictionary learning for pattern recovery*, in International Conference on Learning Representations, 2021.
[30] M.-C. Pali and K. Schnass, *Dictionary learning—from local towards global and adaptiv*, Inf. Inference, 12 (2023), iaad008.
[31] Q. Qu, J. Sun, and J. Wright, *Finding a sparse vector in a subspace: Linear sparsity using alternating directions*, in Advances in Neural Information Processing Systems 27, 2014.
[32] S. Rambhatla, X. Li, and J. Haupt, *NOODL: Provable Online Dictionary Learning and Sparse Coding*, preprint, arXiv:1902.11261, 2019.
[33] S. Ravishankar, A. Ma, and D. Needell, *Analysis of fast structured dictionary learning*, Inf. Inference, 9 (2020), pp. 785–811.
[34] K. Schnass, *Local identification of overcomplete dictionaries*, J. Mach. Learn. Res., 16 (2015), pp. 1211–1242.
[35] K. Schnass, *Convergence radius and sample complexity of ITKM algorithms for dictionary learning*, Appl. Comput. Harmon. Anal., 45 (2018), pp. 22–58, https://doi.org/10.1016/j.acha.2016.08.002.
[36] K. Schnass and F. Teixeira, *Compressed dictionary learning*, J. Fourier Anal. Appl., 26 (2020), 33, https://doi.org/10.1007/s00041-020-09738-6.
[37] Y. Shen, Y. Xue, J. Zhang, K. Letaief, and V. Lau, *Complete dictionary learning via ℓ_p-norm maximization*, in Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence, PMLR 124, 2020, pp. 280–289.
[38] D. A. Spielman, H. Wang, and J. Wright, *Exact recovery of sparsely-used dictionaries*, in Conference on Learning Theory, JMLR Workshop and Conference Proceedings, 2012, pp. 37–31.
[39] J. Sun, Q. Qu, and J. Wright, *Complete dictionary recovery over the sphere*, in 2015 International Conference on Sampling Theory and Applications (SampTA), IEEE, 2015, pp. 407–410.
[40] J. Sun, Q. Qu, and J. Wright, *Complete dictionary recovery over the sphere* I*: Overview and the geometric picture*, IEEE Trans. Inform. Theory, 63 (2017), pp. 853–884, https://doi.org/10.1109/TIT.2016.2632162.
[41] B. Tolooshams and D. Ba, *Stable and Interpretable Unrolled Dictionary Learning*, preprint, arXiv:2106.00058, 2021.
[42] R. Vershynin, *High-Dimensional Probability: An Introduction with Applications in Data Science*, Camb. Ser. Stat. Probab. Math. 47, Cambridge University Press, 2018.
[43] Y. Zhai, Z. Yang, Z. Liao, J. Wright, and Y. Ma, *Complete dictionary learning via ℓ^4-norm maximization over the orthogonal group*, J. Mach. Learn. Res., 21 (2020), 165.
[44] C.-H. Zhang, *Nearly unbiased variable selection under minimax concave penalty*, Ann. Statist., 38 (2010), pp. 894–942, https://doi.org/10.1214/09-AOS729.