

Efficient Sparse Representation Learning with Applications in Personalized and Explainable AI

by

Geyu Liang

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Industrial and Operations Engineering)
in the University of Michigan
2025

Doctoral Committee:

Professor Salar Fattahi, Chair
Professor Raed Al Kontar
Professor Ruiwei Jiang
Professor Viswanath Nagarajan
Professor Qing Qu

Geyu Liang

lianggy@umich.edu

ORCID iD: 0000-0002-5763-5985

© Geyu Liang 2025

To my parents, whose unconditional love makes me who I am.

ACKNOWLEDGEMENTS

For the past four years, I was extremely lucky to work with several excellent researchers including my advisor, collaborators and fellow PhD students. I also received tremendous support from my friends and family. They are the ones that make this thesis possible.

First and foremost, I would like to extend my utmost gratitude to Professor Salar Fattahi, who recruited me as a PhD student, introduced me to the exciting research field of machine learning and trained me as a professional researcher. His strong knowledge, insightful vision and versatile skill-set make him an outstanding role model for me as a researcher. Many of my research breakthroughs are credited to numerous motivating discussions with him and inspirational suggestions from him. He also provided valuable advice for me outside research, in areas including presentation, networking, task management, work life balance and so on, which I will inevitably benefit from in my future career. For the guidance he provided, I cannot express my gratitude enough.

I could not ask for a better committee than the one that I have. I am grateful for the generous assistance and feedback provided by Prof Qing Qu, Prof Raed Al Kontar, Prof Ruiwei Jiang and Prof Viswanath Nagarajan. I would like to thank my other collaborators — Jianhao Ma, Gavin Zhang, Richard Zhang, Naichen Shi and Senne Michielssen — for their stellar expertise and selfless collaboration. I thank NSF and ONR for funding my research. I would also like to thank the University of Michigan, especially the Department of IOE, for fostering a friendly and collaborative environment in which I have been very fortunate to work every day for the past four years. I would like to thank Prof Marina Epelman, Prof Albert Berahas and Prof Siqian Shen for making it possible. I joined IOE with a very high expectation, yet this big family still managed to exceed it.

I consider myself extremely lucky to meet a lot of new friends during my PhD life. I would like to thank Jianhao Ma for sharing his remarkable research insight and being a constant motivation, Qiyuan Chen for those wonderful dinners at his place and investment advice for the steel industry, Jaeshin Park for showing me how to be a caring and responsible family man and Jiacheng Liu for encouraging me during our job hunting season and sharing those hilarious celebrity gossips during our dinner at Bursley. This list would not be complete without mentioning Aaresh Bhathena, Guixian Chen, Naichen Shi, Zhengjia Zhuo, Xinyu Fei,

Zhongzhu Chen, Jingwen Tang, Kevin Smith, Cheoljoon Jeong and Yushan Qu. A special shoutout goes to Jiahao Shi, with whom I shared an apartment, an office and countless memorable experiences together over the past four years. I have to mention the other members of *the Traver Gang* — Mingke Wang (*Wang San*), Boxuan Chang (*CB*), Yifan Dong (*DG*), Zhiyi Wang (*Miss Wang*) and Ziyang Wang (*Mini-Boss*). Thanks to these people, my memory about Ann Arbor will always be cheerful and warm. I hope we will reunite soon in the future.

There are no words that can describe my gratitude to my parents — Yan Liang and Junyi Liang. I am forever indebted to them for their support and sacrifice. Specifically, I would like to thank my mother Yan for her endless health advice and my father Junyi for all the time he spent on Google translate (and ChatGPT recently) trying to understand my paper. My thankfulness extends to my whole big family back in China. I can still vividly remember the encouragement they showed during my last visit before my PhD began.

I was fortunate to keep in touch with many old friends and receive their support during my PhD. To these people who are pursuing their dreams around the world, I am grateful for the memories we shared and wish them the best of luck.

There are so many other people I encountered during my PhD that this acknowledgment has no capacity to mention. My heartfelt appreciation goes to all of them.

TABLE OF CONTENTS

DEDICATION	ii
ACKNOWLEDGEMENTS	iii
LIST OF FIGURES	viii
LIST OF TABLES	xi
ABSTRACT	xii
CHAPTER	
1 Introduction	1
1.1 Background	2
1.2 Organization	6
1.3 Notations	7
2 Efficient Alternating Minimization for Complete Dictionary Learning . .	9
2.1 Introduction	9
2.1.1 Summary of results	11
2.1.2 Related work	13
2.2 Simple Alternating Minimization	15
2.2.1 Orthogonal Dictionary Learning	15
2.2.2 Mini-batch Complete Dictionary Learning	17
2.2.3 Online Dictionary Learning	18
2.2.4 Initialization	19
2.3 Main Proof	20
2.3.1 Proof of Theorem 1	21
2.3.2 Proof of Theorem 3	22
2.3.3 Proof of Theorem 2	29
2.4 Additional Proof	29
2.4.1 Preliminary	29
2.4.2 Proof of Auxiliary Lemmas	30
2.4.3 Rank-one Updates for the Preconditioner	35
2.5 Numerical Experiments	35
2.5.1 Synthetic Dataset	35
2.5.2 Real Dataset	37

2.6	Concluding Remarks	40
3	Dictionary Learning for Heterogeneous Distribution	42
3.1	Introduction	42
3.1.1	Related Works	44
3.2	PerDL: Personalized Dictionary Learning	45
3.3	PerMA: Meta-algorithm of Solving PerDL	47
3.3.1	Global Matching and Local Updates	48
3.4	Theoretical Guarantees	51
3.5	Further Discussion on Linearly Convergent Algorithms	52
3.6	Proof of Theorems	52
3.6.1	Proof of Theorem 7	53
3.6.2	Proof of Theorem 8	55
3.7	Experiments Overview	57
3.7.1	Synthetic Dataset	57
3.7.2	Training with Imbalanced Data	58
3.7.3	Surveillance Video Dataset	59
3.8	Further Details on the Experiments	60
3.8.1	Details of Section 3.7.1	61
3.8.2	Details of Section 3.7.2	61
3.8.3	Details of Section 3.7.3	62
3.8.4	Additional Experiments	63
3.9	Concluding Remarks	66
4	Sparse Representation Learning and Explainable AI	68
4.1	Introduction	68
4.2	Related Literature	70
4.3	Revisiting IP-OMP	72
4.3.1	A Major Hurdle: Inaccurate Query Features	74
4.4	Constrained Concept Refinement	76
4.4.1	Accuracy and Interpretability Guarantees	78
4.4.2	Further Discussion	80
4.5	Proof of Theorems	81
4.5.1	Proof of Theorem 11	81
4.5.2	Proof of Theorem 12	86
4.5.3	Proof of Theorem 13	92
4.5.4	Proof of Lemmas	95
4.6	Experiments on Generative Model	102
4.7	Application: Interpretable Image Classification	104
4.7.1	More Details on Algorithm 14	105
4.7.2	Performance	106
4.7.3	Interpretability	109
4.7.4	More Experiments on Interpretability	109
4.8	Concluding Remarks	117

BIBLIOGRAPHY	118
------------------------	-----

LIST OF FIGURES

FIGURE

2.1	A comparison of image denoising using dictionaries learned via our proposed method (Algorithm 5) 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. Here we use 30 atoms, which corresponds to a sparsity level of 10%. 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 readers to section 4.2 for the details of setup.	10
2.2	The plots above show the iterates of Algorithm 6 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 non-zero entries of $\text{Supp}(\mathbf{X}^*) - \text{Supp}(\mathbf{X}^{(t)})$. The number is 126 at the beginning, which is the total number of non-zero entries in $\mathbf{X}^*$, and 0 in the end, which indicates the full recovery of the support of $\mathbf{X}^*$	21
2.3	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 criteria is when consecutive iterates are close to each other ($\ \mathbf{D}^{(t-1)} - \mathbf{D}^{(t-2)}\ _2 \leq 10^{-7}$).	37
2.4	Results of Algorithm 4 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.	38
2.5	(left) Final error of Algorithm 2.2 in relation to condition number $\kappa(\mathbf{A}^*)$ and varied noise levels β for a fixed sample size of $\tilde{p} = p = 10^5$. (right) Required sample size to achieve $\frac{\ \mathbf{A}^{(T)} - \mathbf{A}^*\ _F}{\ \mathbf{A}^*\ _F} \leq 0.1$ as a function of the condition number $\kappa(\mathbf{A}^*)$. In both settings, we fix $n = 5$, $\theta = 0.3$, and $T = 1000$	38
2.6	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.	40

2.7	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.	41
3.1	A schematic diagram for <code>global_matching</code> (Algorithm 9). At each iteration, we find the shortest path from s to t (highlighted with red), estimate one atom of $\mathbf{D}^{g*}$ using all passed vertices and remove the path (including the vertices) from $\mathcal{G}$	49
3.2	<code>PerMA</code> improves the accuracy of the recovered global dictionary for <i>all</i> clients, even if some (three out of ten) are weak learners.	58
3.3	<code>PerMA</code> improves training with imbalanced datasets. We consider the image reconstruction task on the imbalanced MNIST dataset using only five atoms from a learned global dictionary. The first row corresponds to the original images. The second row is based on the dictionary learned on a single client with an imbalanced dataset. The third row shows the improved performance of the learned dictionary using our proposed method on the same client.	59
3.4	<code>PerMA</code> effectively separates the background from moving objects in video frames. Here we reconstruct the surveillance video frames using global and local dictionaries learned by <code>PerMA</code> . We reconstruct the frames using only 50 atoms from the combined dictionaries.	60
3.5	Comparisons of different methods on synthetic datasets. In the first row, clients are provided with heterogeneous datasets with similar sizes; in the second row, we consider the special cases in which one of the clients has an insufficient sample size and evaluate the performance of the dictionary learned by that specific client. The first column corresponds to final errors with varying d ; the second column corresponds to total running times with varying d ; the third column corresponds to final errors with varying N ; and the forth column corresponds to final errors with varying sparsity level. All the results are averaged over 3 independent trials.	64
3.6	Objects and background separation by <code>PerPCA</code>	65
3.7	Increase of path length for different n^g . Here the x -axis denotes the number of iterations for Algorithm 9 and the x -axis denotes the distance of the $\mathcal{P}$ for each iteration.	66
4.1	The red arrows represent the backpropagation training process for classic explainable AI models. The algorithm introduced in this chapter extends the training process to refine concept embeddings with constraints on their deviation from initial embeddings, represented by green arrows and box.	69
4.2	Results on synthetic dataset.	103
4.3	Prediction accuracy of CCR and its baseline across iterations, with the final test accuracy of CLIP-IP-OMP and lf-CBM indicated for reference.	107

4.4	The first example illustrates a <i>simple case</i> where CCR successfully learns the correct concepts. The second example represents a <i>misleading case</i> , where the image contains concepts like “a flag” that, while relevant and visually apparent, could potentially mislead classification. However, CCR effectively extracts both useful and misleading information, assigning appropriate weights to ensure the correct prediction.	110
4.5	CIFAR-10 (a)	112
4.6	CIFAR-10 (b)	112
4.7	CIFAR-10 (c)	112
4.8	CIFAR-100 (a)	113
4.9	CIFAR-100 (b)	113
4.10	CIFAR-100 (c)	113
4.11	CUB-200 (a)	114
4.12	CUB-200 (b)	114
4.13	CUB-200 (c)	114
4.14	ImageNet (a)	115
4.15	ImageNet (b)	115
4.16	ImageNet (c)	115
4.17	Places365 (a)	116
4.18	Places365 (b)	116
4.19	Places365 (c)	116

LIST OF TABLES

TABLE

- | | | |
|-----|---|-----|
| 3.1 | Comparisons of different methods on MNIST datasets. A smaller MSE, a larger PSNR, and an SSIM closer to 1 indicate better image reconstruction quality. Here k denotes the number of atoms used to reconstruct the image. | 66 |
| 4.1 | Average explanation length (AEL), average sparsity ratio(ASR) and average concept embedding deviation (ACED) for CCR. | 108 |

ABSTRACT

Sparse representation learning plays a crucial role in modern machine learning and signal processing, offering a powerful framework for extracting structured and interpretable representations from high-dimensional data. This thesis explores key theoretical and practical advancements in sparse representation learning, focusing on efficient dictionary learning algorithms, their personalized counterparts for heterogeneous data, and the role of sparse codes in demonstrating model interpretability.

A central challenge in sparse representation learning is the trade-off between efficiency, scalability, and theoretical guarantees. This thesis develops provable and computationally efficient methods for learning structured dictionaries, addressing fundamental issues in scalability and performance. By leveraging novel optimization techniques, it introduces algorithms that not only recover underlying sparse structures with theoretical guarantees but also scale effectively to large datasets and streaming settings.

Beyond efficiency, this thesis extends sparse representation learning to personalized settings, where data exhibits both shared and unique structures. A new framework is introduced to disentangle these components, enabling more adaptive and robust representations across diverse datasets. This approach has broad implications, from improving generalization in imbalanced learning scenarios to enhancing multi-source data analysis.

Finally, this work explores the intersection of sparse representations and explainable AI, addressing the long-standing challenge of balancing interpretability and predictive performance. By refining concept representations in structured ways, this thesis demonstrates how sparse codes can be leveraged to enhance both accuracy and transparency in machine learning models. The proposed methods achieve state-of-the-art performance in interpretable learning tasks while maintaining computational efficiency.

Together, these contributions advance the theoretical foundations and practical applications of sparse representation learning. By bridging efficiency, personalization, and interpretability, this thesis provides new insights and methodologies that extend the impact of sparse learning across a wide range of domains, from signal processing to explainable machine learning.

CHAPTER 1

Introduction

Representation Learning is a foundational paradigm in modern machine learning, facilitating the extraction of structured and meaningful features from high-dimensional data. Its applications span a wide range of domains, including natural language processing and computer vision, where effective representations enhance predictive accuracy, improve generalization, and support downstream interpretability. However, as datasets increase in both size and complexity, the challenge of learning efficient and interpretable representations becomes more pronounced. Traditional machine learning approaches often contend with the well-documented *curse of dimensionality*, wherein the computational and storage demands associated with high-dimensional data significantly hinder optimization and scalability. Additionally, while deep learning models have demonstrated remarkable success, their overparameterized and opaque nature poses challenges for interpretability and reliable generalization beyond training distributions.

Sparse Representation Learning offers a compelling alternative by encoding data as a linear combination of a small subset of basis elements, rather than relying on dense or redundant representations. The imposition of sparsity naturally enhances computational efficiency by reducing both storage and processing requirements while simultaneously mitigating the risk of overfitting. Beyond computational benefits, sparsity fosters interpretability, as the learned sparse codes often correspond to meaningful latent structures within the data, rendering them particularly useful for tasks such as feature selection, signal recovery, and explainable AI. Consequently, sparse representations have found extensive applications in fields such as compressed sensing, denoising, neuroscience, and model compression.

Despite these advantages, a fundamental trade-off persists between efficiency and performance in sparse representation learning. Several existing sparse coding techniques enforce sparsity through sequential selection rules; however, such approaches become computationally expensive, particularly when the representation basis exhibits high cardinality. Moreover, numerous challenges remain unresolved: many existing methods lack rigorous theoretical guarantees, the adaptation of sparse learning to heterogeneous data remains underex-

plored, and its integration into interpretable machine learning is still in its nascent stages. This thesis seeks to address these challenges by introducing novel algorithmic frameworks that effectively balance efficiency, adaptability, and interpretability in sparse representation learning.

1.1 Background

In this section, we provide a high-level overview of *Dictionary Learning*, which is arguably one of the most fundamental and widely studied problems in sparse representation learning.

The dictionary learning problem has its origins in the broader framework of *Sparse Approximation*. Sparse approximation is a fundamental technique in representation learning that aims to represent data using a sparse linear combination of basis elements. Given a data point represented as a vector $\mathbf{y} \in \mathbb{R}^d$ and a dictionary $\mathbf{D} \in \mathbb{R}^{d \times n}$, where each column $\mathbf{D}_{(:,i)} \in \mathbb{R}^d$ for $1 \leq i \leq n$ corresponds to a dictionary atom, the objective of sparse approximation is to determine a sparse coefficient vector $\mathbf{x} \in \mathbb{R}^n$ such that the given data sample can be approximated as:

$$\mathbf{y} \approx \mathbf{D}\mathbf{x}, \quad \text{where } \mathbf{x} \text{ is sparse.}$$

Mathematically, the sparse approximation problem is typically formulated as one of the following two closely related optimization problems:

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_0, \tag{1.1}$$

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2 \quad \text{subject to} \quad \|\mathbf{x}\|_0 \leq s. \tag{1.2}$$

In both formulations, $\|\cdot\|_0$ denotes the ℓ_0 pseudo-norm, which counts the number of nonzero elements in a vector. The regularization parameter λ in Equation (1.1) and the predefined sparsity level s in Equation (1.2) govern the trade-off between sparsity and reconstruction accuracy in the resulting solution $\mathbf{x}$.

Due to the non-convexity of the objective function in both formulations, solving the sparse approximation problem is computationally challenging. Existing approaches for addressing this challenge can be broadly categorized into two main types:

- *Convex Relaxation*: One approach involves convexifying the ℓ_0 norm to transform Equation (1.1) into a convex optimization problem. A common relaxation replaces the ℓ_0 norm with the ℓ_1 norm, which also promotes sparsity and leads to the well-known Least Absolute Shrinkage and Selection Operator (LASSO) regression problem.

- *Greedy Heuristic Methods*: Another class of solutions relies on iterative selection strategies, where entries of $\mathbf{x}$ are sequentially added to the support set of the final solution. A notable example is *Orthogonal Matching Pursuit (OMP)*, a widely used greedy algorithm for sparse approximation. Given that OMP will be frequently referred to and compared with in this thesis, we provide its pseudo-code for completeness:

Algorithm 1 Orthogonal Matching Pursuit (OMP)

```

1: Input: Dictionary  $\mathbf{D} \in \mathbb{R}^{d \times k}$ , target signal  $\mathbf{y} \in \mathbb{R}^d$ , sparsity level  $s$ .
2: Initialize residual:  $\mathbf{r} \leftarrow \mathbf{y}$ 
3: Initialize support set:  $\mathcal{S} \leftarrow \emptyset$ 
4: Initialize sparse code:  $\mathbf{x} \leftarrow \mathbf{0}$ 
5: for  $t = 1$  to  $s$  do
6:   Compute correlations:  $\mathbf{c} \leftarrow \mathbf{D}^\top \mathbf{r}$ 
7:   Select best atom:  $i^* \leftarrow \arg \max_i |\mathbf{c}_i|$ 
8:   Update support set:  $\mathcal{S} \leftarrow \mathcal{S} \cup \{i^*\}$ 
9:   Solve least squares:

$$\mathbf{x}_{\mathcal{S}} \leftarrow \arg \min_{\mathbf{z}} \|\mathbf{y} - \mathbf{D}_{(:,\mathcal{S})}\mathbf{z}\|_2^2$$

10:  Compute new residual:  $\mathbf{r} \leftarrow \mathbf{y} - \mathbf{D}_{(:,\mathcal{S})}\mathbf{x}_{\mathcal{S}}$ 
11:  if  $\|\mathbf{r}\|_2 < \epsilon$  then
12:    break
13:  end if
14: end for
15: return sparse vector  $\mathbf{x}$ 

```

On a high level, OMP iteratively selects dictionary atoms that are most correlated with the current residual $\mathbf{r}$ and then performs an orthogonal projection to update the estimated sparse coefficients. The algorithm stops when a predefined stopping criterion is met, such as reaching a fixed sparsity level s or when the residual norm falls below a given threshold.

With a solid understanding of the sparse approximation problem, we now introduce the dictionary learning problem. In contrast to sparse approximation, dictionary learning extends the setting by considering multiple $(\mathbf{y}, \mathbf{x})$ pairs and treating the dictionary $\mathbf{D}$ as an additional optimization variable. Specifically, given a dataset represented as a matrix $\mathbf{Y} \in \mathbb{R}^{d \times p}$, where each column $\mathbf{Y}_{(:,i)} \in \mathbb{R}^d$ for $1 \leq i \leq p$ corresponds to a data sample, the objective of dictionary learning is to determine a dictionary $\mathbf{D} \in \mathbb{R}^{d \times n}$ and a sparse coefficient matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ such that:

$$\mathbf{Y} \approx \mathbf{D}\mathbf{X}, \quad \text{where } \mathbf{X} \text{ is sparse.}$$

Building upon Equation (1.1), the dictionary learning problem is formulated as:

$$\min_{\mathbf{X}, \mathbf{D}} \|\mathbf{Y} - \mathbf{DX}\|_F^2 + \lambda \|\mathbf{X}\|_0. \quad (1.3)$$

Compared to Equation (1.1), which is already non-convex, the formulation in Equation (1.3) introduces an additional layer of non-convexity due to the presence of both $\mathbf{X}$ and $\mathbf{D}$ as optimization variables. A widely adopted approach for solving Equation (1.3) is alternating minimization, where the algorithm iteratively updates:

- The dictionary $\mathbf{D}$ while keeping $\mathbf{X}$ fixed.
- The sparse coefficients $\mathbf{X}$ while keeping $\mathbf{D}$ fixed.

One of the most fundamental algorithms for dictionary learning is *K-SVD*, which employs this alternating minimization framework to iteratively refine both $\mathbf{X}$ and $\mathbf{D}$. As a key representative of dictionary learning algorithms and an essential benchmark, we present the pseudo-code for the K-SVD algorithm in Algorithm 2.

As illustrated in Algorithm 2, K-SVD is an iterative dictionary learning algorithm that jointly optimizes both the dictionary and the sparse representations of data. It alternates between a sparse coding step, which is solved using OMP, and a dictionary update step, where individual dictionary atoms are refined through *Singular Value Decomposition (SVD)*. At each iteration, the contribution of a specific dictionary atom is isolated by computing the residual error, followed by the application of SVD to obtain the best rank-1 approximation, thereby updating both the dictionary atom and its corresponding sparse coefficients.

It is important to note that K-SVD operates in a sequential manner for both the sparse coding and dictionary update steps, with each sub-step involving computationally intensive operations such as least squares optimization and SVD. Consequently, K-SVD exhibits limited scalability, particularly when applied to large-scale problems where signal vectors correspond to tokens in deep learning models or high-resolution images. This raises the following fundamental question:

Q1: *Does there exist a computationally efficient dictionary learning algorithm that also provides provable guarantees?*

Another significant challenge in dictionary learning, particularly within the context of modern machine learning, pertains to the reliability of its foundational assumption. Specifically, the existence of a dictionary $\mathbf{D}$ with a fixed number of columns n that does not scale with p is predicated on the assumption that all signals in $\mathbf{Y}$ originate from a single underlying distribution. However, when this assumption is violated, it raises the following fundamental question:

Algorithm 2 K-SVD Algorithm

- 1: **Input:** Target signal matrix $\mathbf{Y} \in \mathbb{R}^{d \times p}$, initial dictionary $\mathbf{D} \in \mathbb{R}^{d \times n}$, sparsity level s , number of iterations T .
- 2: **for** $t = 1$ to T **do**
- 3: **Sparse Coding Step:** Solve the sparse approximation problem using OMP (Algorithm 1) to obtain $\mathbf{X}$:

$$\mathbf{X} \leftarrow \arg \min_{\mathbf{X}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 \quad \text{subject to} \quad \|\mathbf{X}_{(:,j)}\|_0 \leq s, \quad \forall j.$$

- 4: **for** $i = 1$ to n **do**
- 5: Identify the data samples using dictionary atom $\mathbf{D}_{(:,i)}$:

$$\Omega_i = \{j \mid \mathbf{X}_{(i,j)} \neq 0\}$$

- 6: **if** $\Omega_i = \emptyset$ **then**
- 7: Continue to next dictionary atom
- 8: **end if**
- 9: Compute the error matrix excluding $\mathbf{D}_{(:,i)}$:

$$\mathbf{E}^i = \mathbf{Y} - \sum_{\substack{l=1 \\ l \neq i}}^n \mathbf{D}_{(:,l)} \mathbf{X}_{(l, \cdot)}$$

- 10: Restrict $\mathbf{E}^i$ to columns in Ω_i :

$$\mathbf{E}^{\Omega_i} = \mathbf{E}_{(:, \Omega_i)}^i$$

- 11: Compute the SVD of $\mathbf{E}^{\Omega_i}$:

$$\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}^\top = \text{SVD}(\mathbf{E}^{\Omega_i})$$

- 12: Update the dictionary atom:

$$\mathbf{D}_{(:,i)} \leftarrow \mathbf{U}_{(:,1)}$$

- 13: Update the sparse coefficients:

$$\mathbf{X}_{(i, \Omega_i)} \leftarrow \mathbf{\Sigma}_{(1,1)} \mathbf{V}_{(:,1)}^\top$$

- 14: **end for**
 - 15: **end for**
 - 16: **return** Updated dictionary $\mathbf{D}$ and sparse code matrix $\mathbf{X}$
-

Q2: *Can existing dictionary learning algorithms effectively and efficiently adapt to data distributions that exhibit heterogeneity?*

Finally, it is essential to recall that the primary objective of dictionary learning is to efficiently represent raw signals (i.e., $\mathbf{y}$ or the columns of $\mathbf{Y}$) using sparse codes (i.e., $\mathbf{x}$ or the columns of $\mathbf{X}$). A question that may be even more fundamental than *How can Equation (1.3) be solved?* is *Why should it be solved?*, which can be formally stated as follows:

Q3: *Beyond the evident advantage of reducing storage requirements, what are the broader benefits of solving the dictionary learning problem?*

A high-level summary of the contributions of this thesis is as follows: this work provides affirmative answers to both **Q1** and **Q2**, while also presenting a compelling case study to investigate **Q3**.

1.2 Organization

This thesis is structured as follows:

In Chapter 2, we examine the *noiseless complete dictionary learning problem*, which seeks to represent a given set of signals as linear combinations of a small number of atoms from a learned dictionary in $\mathbb{R}^{d \times d}$. Two primary challenges arise in both theoretical and practical studies of dictionary learning: the lack of theoretical guarantees for heuristic algorithms commonly used in practice and their poor scalability when applied to large-scale datasets. To address these issues, we propose a simple yet efficient algorithm that provably recovers the ground truth when applied to the nonconvex and discrete formulation of the problem in the noiseless setting. Furthermore, we extend our proposed method to mini-batch and online settings, accommodating scenarios where data is either large-scale or arrives sequentially over time. At the core of our approach is an efficient preconditioning technique that transforms the unknown dictionary into a near-orthonormal one, enabling a simple alternating minimization technique to achieve linear convergence to the ground truth under minimal assumptions. Numerical experiments on both synthetic and real-world datasets demonstrate the superior performance of our method compared to existing techniques.

In Chapter 3, we introduce the problem of *Personalized Dictionary Learning* (PerDL), a challenging yet relevant extension of dictionary learning aimed at learning sparse linear representations from heterogeneous datasets that share common structures. In PerDL, we model the shared and unique features of each dataset as *global* and *local* dictionaries, respectively. The challenges of PerDL extend beyond those of classical *dictionary learning* (DL), as they include the inherent difficulty of disentangling shared and dataset-specific features. In this chapter, we rigorously formulate the PerDL problem and establish conditions under

which the global and local dictionaries can be provably separated. Under these conditions, we introduce a meta-algorithm, *Personalized Matching and Averaging* (PerMA), capable of recovering both global and local dictionaries from heterogeneous datasets. PerMA is highly efficient, converging to the ground truth at a linear rate under suitable assumptions. Additionally, it leverages knowledge from strong learners to enhance the predictive performance of weaker learners. As a general framework for extracting global and local dictionaries, we demonstrate applications of PerDL in various learning tasks, including training with imbalanced datasets and video surveillance.

In Chapter 4, we investigate the relationship between sparse representation learning and explainable AI. The long-standing trade-off between accuracy and interpretability in machine learning (ML) presents a significant challenge, particularly for emerging *interpretable-by-design* methods that aim to enhance interpretability but often do so at the expense of predictive performance. In this chapter, we address this issue by analyzing how deviations in concept representations—an essential component of interpretable models—affect predictive accuracy and propose a novel framework to mitigate these effects. Our framework is designed to optimize concept embeddings while enforcing constraints that preserve interpretability. Using a generative model as a testbed, we rigorously prove that our algorithm attains zero loss while progressively improving the interpretability of the resulting model. Additionally, we evaluate the practical effectiveness of our proposed framework in generating explainable predictions for image classification tasks across multiple benchmarks. Compared to existing explainability methods, our approach enhances predictive accuracy while maintaining model interpretability across large-scale datasets, all while significantly reducing computational cost.

1.3 Notations

Scalars, vectors, matrices, and sets: we use a or α to denote scalars, $\mathbf{a}$ to denote vectors and $\mathbf{A}$ do denote matrices. We use $\mathbf{A}_{(i,\cdot)}$, $\mathbf{A}_{(\cdot,j)}$ and $\mathbf{A}_{ij}$ to denote the i th row, j th column and (i,j) -th entry of $\mathbf{A}$, respectively. We use $\|\mathbf{A}\|_2$ to denote the spectral norm of $\mathbf{A}$, $\|\mathbf{A}\|_F$ to denote Frobenius norm of $\mathbf{A}$, $\|\mathbf{A}\|_{1,2}$ to denote the maximum 2-norm of the columns of $\mathbf{A}$, $\|\mathbf{A}\|_0$ to denote the total number of non-zero entries in $\mathbf{A}$, and $\|\mathbf{A}\|_1$ to denote the entry-wise ℓ_1 norm of $\mathbf{A}$. The symbol $\mathbf{I}_d$ denotes the $d \times d$ identity matrix, and $\mathcal{O}(n)$ denotes the orthogonal group in dimension n . We use $\text{supp}(\mathbf{A})$ to refer to the set of indices of non-zero entries of $\mathbf{A}$. The symbol $\sigma_i(\mathbf{A})$ denotes the i th largest singular value of $\mathbf{A}$ and $\kappa(\mathbf{A})$ denotes the condition number of $\mathbf{A}$. Given a matrix $\mathbf{A}$, its polar decomposition is defined as $\text{Polar}(\mathbf{A}) = \mathbf{U}_\mathbf{A} \mathbf{V}_\mathbf{A}^\top$, where $\mathbf{U}_\mathbf{A} \mathbf{\Sigma}_\mathbf{A} \mathbf{V}_\mathbf{A}^\top$ is the SVD of $\mathbf{A}$. We also define the operator

$\mathcal{L}(\mathbf{A}) = \mathbf{L}_\mathbf{A}^\top$, where $\mathbf{L}_\mathbf{A}\mathbf{L}_\mathbf{A}^\top$ is the Cholesky factorization of a positive semidefinite matrix $\mathbf{A}$.

Probability: for an event E , the notation $\mathbb{P}(E)$ denotes the probability of its occurrence and $\mathbb{1}_E$ denotes its indicator function. For a random variable x , $\mathbb{E}[x]$ represents its expected value and $\text{Var}[x]$ represents its variance. The notation $x \sim \mathcal{N}(\mu, \Sigma)$ indicates that x is a random vector sampled from a Gaussian distribution with mean vector μ and covariance matrix Σ .

Functions: 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 $f(n) \leq Cg(n)$ 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 $f(n) \geq Cg(n)$ for all large enough n . We use $f(n) = \omega(g(n))$ if for all constants $C > 0$ we have $f(n) \geq Cg(n)$ 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.

CHAPTER 2

Efficient Alternating Minimization for Complete Dictionary Learning

This chapter is based on the paper *Simple Alternating Minimization Provably Solves Complete Dictionary Learning* [75] by Geyu Liang, Gavin Zhang, Salar Fattahi, Richard Y. Zhang, which will appear in *SIAM Journal on Mathematics of Data Science (SIMODS)*.

2.1 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 [81, 77], 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 non-convex and highly difficult to solve, both in theory and in practice. In this chapter, we focus on learning a *complete* dictionary, whose number of atoms matches the dimensionality of the data, via the following optimization problem:

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

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

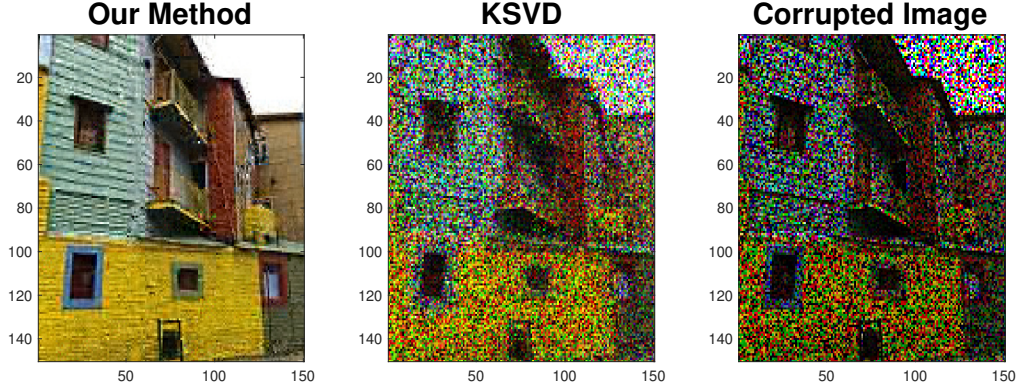


Figure 2.1: **A comparison of image denoising using dictionaries learned via our proposed method (Algorithm 5) and via KSVD.** We choose a random landscape image and artificially corrupt 50% of the pixels. Reconstruction is done via orthogonal matching pursuit with the learned dictionaries. Here we use 30 atoms, which corresponds to a sparsity level of 10%. 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 readers to section 4.2 for the details of setup.

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 pseudo-norm presents a significant difficulty, as it is not only non-convex, 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 [4] and MOD [34] 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 pseudo-norm into the convex ℓ_1 norm [2] or the convex ℓ_4 norm [139]. Under assumptions like mutual incoherency [8], restricted isometry property (RIP) [2] or large sample complexity [116, 139], 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 than the solution to the original CDL.

In this chapter, we revisit the combinatorial ℓ_0 pseudo-norm in CDL, motivated by the fact that the *orthogonal* instance (in which the dictionary $\mathbf{D}$ is constrained to be orthonormal) is significantly easier to analyze. In the orthogonal setting, alternating minimization is *simple*, in that the minimizations over $\mathbf{D}$ and $\mathbf{X}$ 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 2.1, 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.

2.1.1 Summary of results

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

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

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 $\mathbf{X}$ for a fixed orthonormal $\mathbf{D} \in \mathbb{O}(n)$ has a cheap closed-form solution:

$$\arg \min_{\mathbf{X}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 + \zeta \|\mathbf{X}\|_0 = \text{HT}_\zeta(\mathbf{D}^T \mathbf{Y})$$

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

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

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

$$\arg \min_{\mathbf{D} \in \mathbb{O}(n)} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 + \zeta \|\mathbf{X}\|_0 = \text{Polar}(\mathbf{Y}\mathbf{X}^\top),$$

where $\text{Polar}(\mathbf{A}) = \mathbf{U}_\mathbf{A} \mathbf{V}_\mathbf{A}^\top$, and $\mathbf{U}_\mathbf{A} \mathbf{\Sigma}_\mathbf{A} \mathbf{V}_\mathbf{A}^\top$ is the Singular Value Decomposition (SVD) of $\mathbf{A}$. Therefore, the following iterations

$$\mathbf{X}^{(t+1)} = \text{HT}_\zeta(\mathbf{D}^{(t)\top} \mathbf{Y}), \quad \mathbf{D}^{(t+1)} = \text{Polar}(\mathbf{Y}\mathbf{X}^{(t+1)\top}) \quad (2.1)$$

correspond exactly to an alternating minimization solution of Problem ODL. While (2.1) 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 $\mathbf{X}^*$. Concretely, we prove that when the data are generated as $\mathbf{Y} = \mathbf{D}^*\mathbf{X}^*$, with $\mathbf{D}^*$ an orthogonal dictionary, and $\mathbf{X}^*$ a sparse random matrix, (2.1) converges linearly to a ground truth $(\mathbf{D}^*, \mathbf{X}^*)$, 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 (2.1) to general instances of CDL, for which the square dictionary matrix $\mathbf{D}^*$ is not necessarily orthonormal. The basic idea is to learn an orthonormal dictionary $\tilde{\mathbf{D}}$ with respect to the preconditioned data matrix $\tilde{\mathbf{Y}} = (\mathbf{Y}\mathbf{Y}^\top)^{-1/2}\mathbf{Y}$, and then to output a complete dictionary $\mathbf{D} = (\mathbf{Y}\mathbf{Y}^\top)^{+1/2}\tilde{\mathbf{D}}$ by reversing the preconditioning. Indeed, if the data are generated as $\mathbf{Y} = \mathbf{D}^*\mathbf{X}^*$ with $\mathbf{D}^*$ deterministic, and $\mathbf{X}^*$ a sparse random matrix satisfying $\mathbb{E}[\mathbf{X}^*\mathbf{X}^{*\top}] = \mathbf{I}_n$, then $\mathbb{E}[\mathbf{Y}\mathbf{Y}^\top] = \mathbf{D}^*\mathbb{E}[\mathbf{X}^*\mathbf{X}^{*\top}]\mathbf{D}^{*\top} = \mathbf{D}^*\mathbf{D}^{*\top}$. Therefore, the preconditioned data are generated as $\tilde{\mathbf{Y}} \approx \tilde{\mathbf{D}}^*\mathbf{X}^*$ with respect to an orthogonal dictionary $\tilde{\mathbf{D}}^* = (\mathbf{D}^*\mathbf{D}^{*\top})^{+1/2}\mathbf{D}^*$ and the same sparse code $\mathbf{X}^*$. 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* [13], which is indeed inefficient in practice. Using Riemannian optimization techniques, [115] 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 [115].

In practice, the iterations (2.1) admit highly efficient implementations, that can fully take advantage of the massive parallelism inherent in modern hardware. The hard-threshold 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 (2.1) is actually its need to iterate and sum over all p columns of the data matrix $\mathbf{Y}$ at every iteration. To address potential scalability issues with a very large p ,

we propose a mini-batch version,

$$\mathbf{X}_{(\cdot,i)}^{(t+1)} = \text{HT}_{\zeta}(\mathbf{D}^{(t)\top} \mathbf{Y}_{(\cdot,i)}) \quad \forall i \in \Omega, \quad \mathbf{D}^{(t+1)} = \text{Polar} \left(\sum_{i \in \Omega} \mathbf{Y}_{(\cdot,i)} \mathbf{X}_{(\cdot,i)}^{(t+1)\top} \right). \quad (2.2)$$

that approximates $\mathbf{Y}\mathbf{X}^{(t+1)\top} \approx \sum_{i \in \Omega} \mathbf{Y}_{(\cdot,i)} \mathbf{X}_{(\cdot,i)}^{(t+1)\top}$ over a mini-batch $\Omega \subseteq \{1, 2, \dots, p\}$, and updates only the sparse codes $\mathbf{X}_{(\cdot,i)}^{(t+1)}$ 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. [95, Theorem 1] and [7, 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.

2.1.2 Related work

Optimality conditions As the first question, we must first verify whether a ground truth $(\mathbf{D}^*, \mathbf{X}^*)$ is a global minimizer of CDL for the generative model $\mathbf{Y} = \mathbf{D}^* \mathbf{X}^*$. Of course, if this were not the case, solving CDL—even to global optimality—may not be enough to recover a ground truth $(\mathbf{D}^*, \mathbf{X}^*)$. A series of work [48, 41, 53, 47, 101, 30, 51] studied the local optimality of the ground truth for CDL by replacing ℓ_0 norm with ℓ_1 norm. The work [113] elegantly showed that the ground truth is the unique global minimizer to CDL when $\zeta \rightarrow 0$. Specifically, when $p = \Omega(n \log n)$, they showed that any ground truth $(\mathbf{D}^*, \mathbf{X}^*)$ is the global minimizer to $\min_{\mathbf{X}, \mathbf{D}} \|\mathbf{X}\|_0$ subject to the constraint $\mathbf{Y} = \mathbf{D}\mathbf{X}$. The question remains to be answered: how to design a provable algorithm for recovering $(\mathbf{X}^*, \mathbf{D}^*) \in \mathcal{M}$.

Alternating minimization The empirical success achieved by methods like MOD and KSVD has encouraged the emergence of many alternating minimization algorithms [68, 80, 12]. Towards providing a provable guarantee for alternating minimization, it is common

to replace the ℓ_0 pseudo-norm with its convex surrogate ℓ_1 norm [2, 24, 82, 122]. Such a compromise is due to the prohibitive computational cost and formidable analytical challenges of ℓ_0 pseudo-norm. However, ℓ_1 relaxation is biased towards solutions with smaller entries, thereby leading to an inferior sparsity level [140]. On the algorithmic side, methods based on ℓ_1 relaxation need to solve variants of the LASSO problem at each iteration either exactly [2, 24] or approximately via an automatic differentiation with backpropagation [82, 122]. 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 [102, 51, 103, 90], the authors study the convergence behavior of a variant of K-SVD. Specifically, the dictionary update step is performed by maximizing the absolute norm of the S -largest responses where S is the number of non-zero entries in each signal. In [98], authors propose an alternating scheme based on sorting the non-zero entries of sparse codes. However, the success of these algorithms relies on restrictive generative models like symmetric decaying [101] and fixed sparsity S [98].

Riemannian manifold optimization for complete dictionary learning One line of work focuses on solving CDL via Riemannian manifold optimization techniques [92, 139]. Notably, the work [116] 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 [39]. As a result, one can provably recover one dictionary atom at a time via the Riemannian trust-region method. The work [43] showed that similar results can be achieved by first-order methods on the Riemannian manifold. In a different approach, the work [139] studied the ℓ_4 maximization on the Steifel manifold, which is further generalized in [106] 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 [80] first proposed an online algorithm that solves ℓ_1 relaxation of CDL, but it only guarantees convergence to a critical point. The work [78] enhanced the usability and practicality of ℓ_1 relaxation but provided no theoretical guarantees. The work [7] proposed an alternating algorithm that can recover the ground truth $(\mathbf{X}^*, \mathbf{D}^*)$, which is further improved in [95]. 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.

2.2 Simple Alternating Minimization

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

2.2.1 Orthogonal Dictionary Learning

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

Algorithm 3 Alternating minimization for ODL

```

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

```

We note that Algorithm 3 has been studied before. The paper [11] reports the empirical performance of Algorithm 3 on an image restoration task. The paper [98] provides a theoretical analysis for a variant of Algorithm 3 based on sorted thresholding. One may even argue that Algorithm 3 is similar to the Method of Optimal Directions (MOD). However, the existing theoretical guarantees for Algorithm 3 are indeed restrictive (see the discussion on alternating minimization in Section 2.1.2). To bridge this knowledge gap, we first introduce our assumption on the sparse matrix $\mathbf{X}$.

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

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

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

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

Now, we are ready to show the convergence of Algorithm 3:

Theorem 1. *Suppose that $\mathbf{Y} = \mathbf{D}^* \mathbf{X}^*$ where $\mathbf{D}^*$ is orthogonal and $\mathbf{X}^*$ satisfies Assumption 1 with sparsity level $0 < \theta < 1/2$. Suppose that the initial dictionary $\mathbf{D}^{(0)}$ satisfies $\|\mathbf{D}^{(0)} - \mathbf{D}^*\|_F \lesssim \frac{1}{\sigma\sqrt{\theta n}}$ for some $(\mathbf{D}^*, \mathbf{X}^*) \in \mathcal{M}$. 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 $T \geq 1$, the output of Algorithm 3 with $\zeta = \Gamma/2$ satisfies:*

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

Theorem 1 improves upon the existing results on two fronts:

Linear sparsity level We allow a constant fraction of entries in $\mathbf{X}^*$ to be non-zero, thereby improving upon the best known sparsity level of $O(\sqrt{n}/\log n)$ for alternating minimization [7]. 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 $\mathbf{D}^*$ 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 $\xi \rightarrow 0$ [113]. 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 $\mathbf{X}^*$ is known. Note that the theoretical convergence of Algorithm 3 is contingent upon a good initial dictionary. As shown in Lemma 1, the imposed condition on the initial dictionary automatically guarantees the recovery of the support for any $\mathbf{X}^{(t)}$ with $t \geq 0$. In Subsection 2.2.4, 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 $\|\mathbf{D}^{(0)} - \mathbf{D}^*\|_{1,2} \leq O(1/\log n)$ with fresh samples every iteration, which is the best-known radius for alternating minimization [7].

Despite its desirable properties, Algorithm 3 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 (non-orthogonal) dictionary learning with mini-batch and online data.

2.2.2 Mini-batch Complete Dictionary Learning

To distinguish from ODL, we denote the ground truth dictionary as $\mathbf{A}^*$ (i.e., $\mathbf{Y} = \mathbf{A}^* \mathbf{X}^*$). Towards dealing with large p , we sub-sample from the columns of $\mathbf{Y}$. To address the non-orthogonality of the dictionary, we consider a preconditioner $\mathbf{P}$ defined as

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

Using this preconditioner, we obtain a new (preconditioned) data matrix $\tilde{\mathbf{Y}} = \mathbf{P} \mathbf{Y}$. To explain the intuition behind this choice of preconditioner, note that $\frac{1}{p\theta\sigma^2} \mathbf{Y} \mathbf{Y}^\top \approx \mathbf{A}^* \mathbf{A}^{*\top}$ for large enough p , which allows us to write $\tilde{\mathbf{Y}} \approx \mathcal{L} \left((\mathbf{A}^* \mathbf{A}^{*\top})^{-1} \right) \mathbf{A}^* \mathbf{X}^*$. Upon defining $\mathbf{D}^* = \mathcal{L} \left((\mathbf{A}^* \mathbf{A}^{*\top})^{-1} \right) \mathbf{A}^*$, one can check that $\mathbf{D}^* \in \mathbb{O}(n)$ and $\tilde{\mathbf{Y}} \approx \mathbf{D}^* \mathbf{X}^*$. Indeed, this is an instance of ODL and can be solved via Algorithm 3. The detailed implementation is provided in Algorithm 4. 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 4. Without loss of

Algorithm 4 Alternating minimization for mini-batch CDL

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

generality, we assume that $\|\mathbf{A}^*\|_2 = 1$ and set $\hat{\kappa}$ to be the condition number of $\mathbf{A}^*$.

Theorem 2. *Suppose that $\mathbf{Y} = \mathbf{A}^* \mathbf{X}^*$, where $\mathbf{X}^*$ satisfies Assumption 1 with sparsity level $0 < \theta < 1/2$. Suppose that the initial dictionary $\mathbf{A}^{(0)}$ satisfies $\|\mathbf{A}^{(0)} - \mathbf{A}^*\|_F \lesssim \frac{1}{\hat{\kappa}\sigma\sqrt{\theta n}}$ for some $(\mathbf{A}^*, \mathbf{X}^*) \in \mathcal{M}$. 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 high probability, the output of Algorithm 4 with $\zeta = \Gamma/2$ satisfies:*

$$\|\mathbf{A}^{(T)} - \mathbf{A}^*\|_F \leq 0.9^T \|\mathbf{A}^{(0)} - \mathbf{A}^*\|_F + O \left(\frac{n\hat{\kappa}^6}{\theta\sqrt{p}} \right). \quad (2.3)$$

Theorem 2 states that Algorithm 4 converges linearly to the true dictionary up to a statistical error $O\left(\frac{n\hat{\kappa}^6}{\theta\sqrt{p}}\right)$. This statistical error is due to the deviation of the preconditioner from its expectation, which diminishes with p . A key distinction of our result is that we do not impose any incoherency requirement or restricted isometry property on $\mathbf{A}^*$, which are common assumptions in existing results. We also highlight that the upper bound on T is indeed very mild. To demonstrate this, note that, in order to guarantee $0.9^T \|\mathbf{A}^{(0)} - \mathbf{A}^*\|_F = O\left(\frac{n\hat{\kappa}^6}{\theta\sqrt{p}}\right)$, it suffices to have $T = \Omega\left(\frac{\theta\sqrt{p}}{\hat{\kappa}^6 n \log n}\right)$, a condition that satisfies $T = O(n^\beta)$ for some $\beta > 0$ for all practical purposes.

2.2.3 Online Dictionary Learning

Even though the computational efficiency of Algorithm 4 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 $\mathbf{P}$ with a new sample $\mathbf{y}$, a naive approach would be to re-compute it from scratch which would cost $O(n^3)$ operations. However, we show that $\mathbf{P}$ 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 $(\mathbf{Y}\mathbf{Y}^\top)^{-1}$ as

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

which, given $(\mathbf{Y}\mathbf{Y}^\top)^{-1}$, can be obtained in $O(n^2)$ operations. Given the above rank-one update for the inverse, the Cholesky factor $\mathcal{L}\left((\mathbf{Y}\mathbf{Y}^\top + \mathbf{y}\mathbf{y}^\top)^{-1}\right)$ can be obtained within $O(n^2)$ operations by performing triangular rank-one updates on $\mathcal{L}\left((\mathbf{Y}\mathbf{Y}^\top)^{-1}\right)$ [63]. We explain the implementation of this method in Section 2.4.3.

Inspired by the above update, we propose an online variant of Algorithm 5. 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 $\mathbf{P}^{(t)}$ via the triangular rank-one update and update the data set accordingly. Our next theorem establishes the convergence of Algorithm 5.

Theorem 3. *Suppose that $\mathbf{Y} = \mathbf{A}^*\mathbf{X}^*$, where $\mathbf{X}^*$ satisfies Assumption 1 with sparsity level $0 < \theta < 1/2$. Suppose that the initial dictionary $\mathbf{A}^{(0)}$ satisfies $\|\mathbf{A}^{(0)} - \mathbf{A}^*\|_F \lesssim \frac{1}{\hat{\kappa}\sigma\sqrt{\theta n}}$ for some $(\mathbf{A}^*, \mathbf{X}^*) \in \mathcal{M}$. 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*

Algorithm 5 Alternating minimization for online dictionary learning

- 1: **Input:** $\mathbf{A}^{(0)}$, ζ , p_1, p_2
 - 2: Set $\mathbf{Z}_{\text{inv}}^{(0)} = (\bar{\mathbf{Y}}\bar{\mathbf{Y}}^\top)^{-1}$, $\mathbf{P}^{(0)} = \mathcal{L} \left(p_1 \theta \sigma^2 \mathbf{Z}_{\text{inv}}^{(0)} \right)^\top$, where $\bar{\mathbf{Y}}$ is constructed with p_1 samples and $\mathbf{D}^{(0)} = \mathbf{P}^{(0)} \mathbf{A}^{(0)}$.
 - 3: Initialize $\mathbf{Y}^{(0)}$ with p_2 samples.
 - 4: **for** $t = 0, 1, \dots, T-1$ **do**
 - 5: Given a new $\mathbf{y}$, set $\mathbf{Y}^{(t)} = [\mathbf{Y}^{(t-1)} \ \mathbf{y}]$ and remove the first column of $\mathbf{Y}^{(t)}$.
 - 6: Update $\mathbf{P}^{(t)}$ and $\mathbf{Z}_{\text{inv}}^{(t)}$ using $\mathbf{P}^{(t-1)}$, $\mathbf{Z}_{\text{inv}}^{(t-1)}$, and $\mathbf{y}$ via Algorithm 7.
 - 7: Set $\tilde{\mathbf{Y}}^{(t)} = \mathbf{P}^{(t)} \mathbf{Y}^{(t)}$
 - 8: Set $\tilde{\mathbf{X}}^{(t)} = \text{HT}_\zeta \left(\mathbf{D}^{(t)\top} \tilde{\mathbf{Y}}^{(t)} \right)$.
 - 9: Set $\mathbf{D}^{(t+1)} = \text{Polar} \left(\tilde{\mathbf{Y}}^{(t)} \tilde{\mathbf{X}}^{(t)\top} \right)$.
 - 10: **end for**
 - 11: **return** $(\mathbf{P}^{(T-1)})^{-1} \mathbf{D}^{(T)}$ as an approximation to $\mathbf{A}^*$
-

$T \lesssim n^\beta$, for some constants $\gamma, \beta > 0$. Then, with probability at least $1 - n^{-\omega(1)}$, the output of Algorithm 5 with $\zeta = \Gamma/2$ satisfies:

$$\|\mathbf{A}^{(T)} - \mathbf{A}^*\|_F \leq 0.9^T \|\mathbf{A}^{(0)} - \mathbf{A}^*\|_F + O \left(\frac{n\hat{\kappa}^6}{\theta\sqrt{p_1 + T}} \right). \quad (2.4)$$

The convergence result of Algorithm 5 is similar to that of Algorithm 4, with a key difference that the statistical error $O \left(\frac{n\hat{\kappa}^6}{\theta\sqrt{p_1 + T}} \right)$ now decreases with T , which is due to the fact that the preconditioner becomes progressively more accurate as new samples continue to arrive.

2.2.4 Initialization

The theoretical success of Algorithms 3-5 requires, at least in theory, a good initialization with $O(1/\sqrt{n})$ distance to the ground truth. Such initialization can be provided by the initialization scheme introduced in [2] and [7], albeit with slightly more restrictive conditions on the sparsity level. For the same generative model as described in Assumption 1, computationally intensive and data-intensive algorithms, such as the Riemannian Trust Region method (RTR) [115], can also be employed to design initial dictionaries that satisfy the conditions outlined in Theorems 1-3. Specifically, for ODL, [115, Theorem 3.1] demonstrates that the RTR can obtain an initial point with a distance of $O(1/\sqrt{n})$ from the ground truth, albeit with an increased sample complexity of $p = \Omega(n^5)$. Similarly, for CDL, [115, Theorem 3.2] illustrates that RTR can yield an initial point with a distance of $O(1/\sqrt{n})$ from the ground truth with a sample complexity of $p = \Omega(n^7)$. Combining these results with

Theorems 1-3 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 3 with a diminishing threshold presented in Algorithm 6 (the warm-start schemes for Algorithms 4 and 5 are similar and hence omitted for brevity).

Notice that a large ζ_0 will force $\mathbf{X}^{(t)}$ to be an all-zero matrix and make $\mathbf{D}^{(t+1)}$ an identity matrix. As we shrink the threshold ζ_t , the matrix $\mathbf{X}^{(t)}$ eventually becomes nonzero, and the iterates $\mathbf{D}^{(0)}$ start to gradually move towards the basin of attraction of $\mathbf{D}^*$. Figure 2.2 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 $\mathbf{I}_d$ 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 1. 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 2.1.2. The theoretical analysis of this warm-start algorithm is left as an enticing challenge for future research.

Algorithm 6 Algorithm 3 with warm-start

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

2.3 Main Proof

We will present our complete proofs in two sections for the purpose of clarity. In this section, we present the proofs of our main theorems. In the next section, we present the additional proofs for lemmas used in this section.

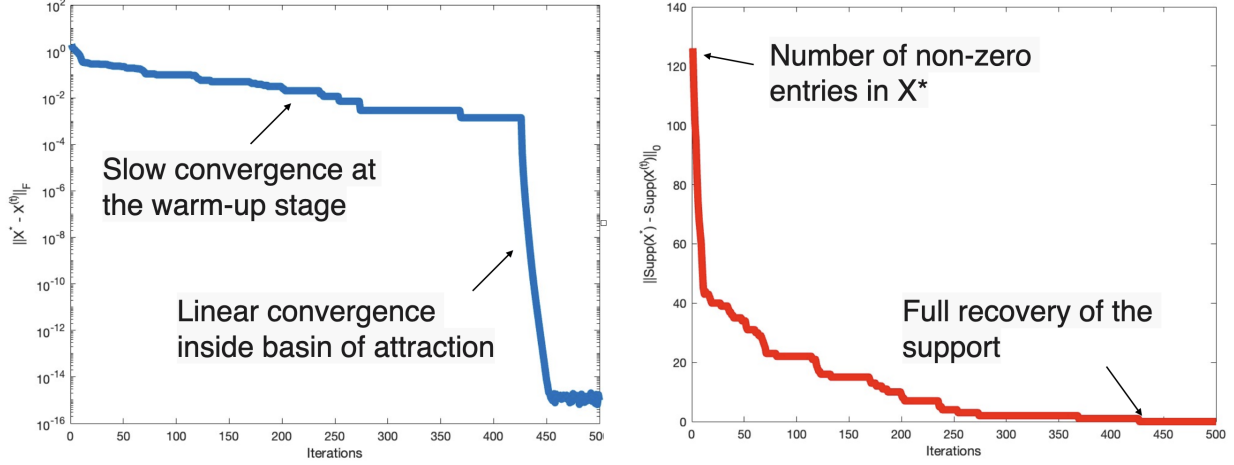


Figure 2.2: The plots above show the iterates of Algorithm 6 with $p = 100$, $n = 5$ and $\theta = 0.3$. The left figure shows the error in the sparse code. The right figure shows the number of non-zero entries of $\text{Supp}(\mathbf{X}^*) - \text{Supp}(\mathbf{X}^t)$. The number is 126 at the beginning, which is the total number of non-zero entries in $\mathbf{X}^*$, and 0 in the end, which indicates the full recovery of the support of $\mathbf{X}^*$.

2.3.1 Proof of Theorem 1

We use induction to prove Theorem 1. We consider the following induction hypothesis:

$$\|\mathbf{D}^{(s)} - \mathbf{D}^*\|_F = O\left(\frac{1}{\sigma\sqrt{\theta n}}\right) \quad \text{at iteration } s. \quad (2.5)$$

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 Section 2.4.2.

Lemma 1 (Exact support recovery). *Consider $\mathbf{Y} = \mathbf{D}^*\mathbf{X}^*$ where $\mathbf{D}^* \in \mathbb{O}^n$ and $\mathbf{X}^*$ satisfies Assumption 1. The following statement holds: for all $\mathbf{D}$ satisfying $\|\mathbf{D} - \mathbf{D}^*\|_{1,2} = O\left(\frac{1}{\sigma\sqrt{\theta n}}\right)$, with probability at least $1 - 2\exp(\log p - Cn)$, we have $\text{supp}(\text{HT}_{\Gamma/2}(\mathbf{D}^\top \mathbf{Y})) = \text{supp}(\mathbf{X}^*)$.*

Lemma 2 (Guaranteed improvement on polar decomposition). *Consider $\mathbf{Y} = \mathbf{D}^*\mathbf{X}^*$ where $\mathbf{D}^* \in \mathbb{O}^n$ and $\mathbf{X}^*$ satisfies Assumption 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 $\mathbf{X}$ of $\mathbf{X}^*$ such that $\frac{\|\mathbf{X} - \mathbf{X}^*\|_F}{\|\mathbf{X}^*\|_2} = O(1)$ and $\text{supp}(\mathbf{X}) = \text{supp}(\mathbf{X}^*)$:*

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

Lemma 1 together with our induction hypothesis (2.5) implies that

$$\text{supp}(\mathbf{X}^{(s)}) = \text{supp}(\text{HT}_{\Gamma/2}(\mathbf{D}^{(s)\top} \mathbf{Y})) = \text{supp}(\mathbf{X}^*). \quad (2.6)$$

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

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

where we used (2.6) for (a). The above inequality implies that

$$\frac{\|\mathbf{X}^{(s)} - \mathbf{X}^*\|_F}{\|\mathbf{X}^*\|_2} = \frac{\|\mathbf{X}^{(s)} - \mathbf{X}^*\|_F}{\|\mathbf{D}^* \mathbf{X}^*\|_2} = \frac{\|\mathbf{X}^{(s)} - \mathbf{X}^*\|_F}{\|\mathbf{Y}\|_2} \leq \|\mathbf{D}^{(s)} - \mathbf{D}^*\|_F = O\left(\frac{1}{\sigma\sqrt{\theta n}}\right) \quad (2.7)$$

Invoking Lemma 2 with $\mathbf{X} = \mathbf{X}^{(s)}$ together with (2.7) leads to

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

Therefore, the induction hypothesis (2.5) holds for $t = s + 1$. Consequently, the linear convergence for $\frac{\|\mathbf{X}^{(s)} - \mathbf{X}^*\|_F}{\|\mathbf{X}^*\|_2}$ follows from (2.7) and the above inequality:

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

The proof is complete by noticing a lower bound of the probability that both Lemma 1 and Lemma 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. $\square$

Next, we provide the proof of Theorem 3. As will be shown later, the proof of Theorem 2 readily follows from this proof.

2.3.2 Proof of Theorem 3

The linear convergence of $\mathbf{A}^{(T)}$ hinges on the linear convergence of the preconditioned dictionary $\mathbf{D}^{(T)}$ towards

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

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

$$\|\mathbf{D}^{(s+1)} - \mathbf{D}^*\|_F \leq 0.9\|\mathbf{D}^{(s)} - \mathbf{D}^*\|_F + O(\|\mathbf{P}^{(s)} \mathbf{A}^* - \mathbf{D}^*\|_F), \quad s = 0, 2, \dots, T-1 \quad (2.8)$$

We present the following intermediate lemmas, the proofs of which are in Section 2.4.2.

Lemma 3 (Bounding preconditioner error). *Suppose that $\mathbf{X}^*$ satisfies Assumption 1. Then, with probability at least $1 - 2\exp(-n)$, we have:*

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

Lemma 4 (Spectral property for sparse code matrix). *Suppose that $\mathbf{X}^*$ satisfies Assumption 1 and $p = \Omega(\tau n / \theta^2)$ for an arbitrary $\tau > 0$. Then, with probability at least $1 - 2\exp(-n)$, we have*

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

Similar with the proof of Theorem 2, we will use the induction hypothesis $\|\mathbf{D}^{(s)} - \mathbf{D}^*\|_F \leq O(\frac{1}{\sigma\sqrt{\theta n}})$. The base case is easy to verify given the initial condition and Lemma 3:

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

Moreover, after achieving (2.8) for each s , we have $\|\mathbf{D}^{(s+1)} - \mathbf{D}^*\|_F = O(\frac{1}{\sigma\sqrt{\theta n}})$ since

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

For the remainder of this section, we abuse the notation and use $\mathbf{X}^*$ to denote the sparse code that generates $\mathbf{Y}^{(s)} = \mathbf{A}^* \mathbf{X}^*$, where $\mathbf{Y}^{(s)}$ 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 5). Indeed, $\mathbf{X}^*$ 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 $\tilde{\mathbf{X}}$ as $\tilde{\mathbf{X}} = \mathbf{D}^{*\top} \tilde{\mathbf{Y}}^{(s)}$, which is again a different sparse code for different iterations since $\tilde{\mathbf{Y}}^{(s)} = \mathbf{P}^{(s)} \mathbf{Y}^{(s)}$ and both $\mathbf{P}^{(s)}$ and $\mathbf{Y}^{(s)}$ 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 (2.8), we first show that $\tilde{\mathbf{X}}^{(s)}$ recovers the sparsity pattern of $\mathbf{X}^*$ if we use enough number of samples to construct the preconditioner. To show this, we consider

one entry $\left(\mathbf{D}^{(s)\top} \tilde{\mathbf{Y}}^{(s)}\right)_{ij}$ of $\mathbf{D}^{(s)\top} \tilde{\mathbf{Y}}^{(s)}$ and write

$$\left(\mathbf{D}^{(s)\top} \tilde{\mathbf{Y}}^{(s)}\right)_{ij} = \underbrace{\left(\mathbf{D}^{(s)\top} \mathbf{D}^* (\tilde{\mathbf{X}} - \mathbf{X}^*)\right)_{ij}}_{:=\mathcal{A}_{ij}} + \underbrace{\left(\mathbf{D}^{(s)\top} \mathbf{D}^* \mathbf{X}^*\right)_{ij}}_{:=\mathcal{B}_{ij}}. \quad (2.9)$$

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

$$\begin{aligned} |\mathcal{A}_{ij}| &\leq \|\mathbf{D}^{(s)\top} \mathbf{D}^*\|_2 \|\tilde{\mathbf{X}} - \mathbf{X}^*\|_{1,2} = \|\tilde{\mathbf{X}} - \mathbf{X}^*\|_{1,2} \\ &\leq \|\mathbf{D}^{*\top} \mathbf{P}^{(s)} \mathbf{A}^* \mathbf{X}^* - \mathbf{X}^*\|_{1,2} \leq \|\mathbf{P}^{(s)} \mathbf{A}^* - \mathbf{D}^*\|_2 \|\mathbf{X}^*\|_{1,2} \\ &\leq \left\| \mathbf{P}^{(s)} - \mathcal{L} \left((\mathbf{A}^* \mathbf{A}^{*\top})^{-1} \right) \right\|_2 \|\mathbf{A}^*\|_2 \|\mathbf{X}^*\|_{1,2} \lesssim \frac{\hat{\kappa}^6}{\theta} \sqrt{\frac{n}{p_1 + T}} \|\mathbf{X}^*\|_{1,2}. \end{aligned} \quad (2.10)$$

with probability at least $1 - 2 \exp(-n)$, where the last inequality is due to Lemma 3. The expectation for the norm of each column, based on Assumption 1, is $\sigma \sqrt{\theta n}$. Therefore, according to Theorem 5 (see Section 2.4.1), the random variable $\|\mathbf{X}_{(\cdot,k)}^*\|_2 - \sigma \sqrt{\theta n}$ is sub-Gaussian with sub-Gaussian norm $O(\sqrt{\theta} \sigma)$. As a result, the k th column of $\mathbf{X}^*$ satisfies

$$\mathbb{P} \left(\|\mathbf{X}_{(\cdot,k)}^*\|_2 \geq 2\sigma \sqrt{\theta n} \right) \leq 2 \exp(-Cn), \quad (2.11)$$

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

$$\|\mathbf{X}^*\|_{1,2} = \max_{1 \leq k \leq p_2} \|\mathbf{X}_{(\cdot,k)}^*\|_2 \leq 2\sigma \sqrt{\theta n \log p_2} \quad (2.12)$$

with probability at least $1 - 2 \exp(-Cn)$. Combining this inequality with (2.10), 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 the first statement of Lemma 1, the deviation of the second term $\mathcal{B}_{ij}$ from $\mathbf{X}_{ij}^*$ can also be upper bounded by $\Gamma/4$ with high probability, which will not be repeated here. Therefore, we have $\left| \left(\mathbf{D}^{(s)\top} \tilde{\mathbf{Y}}^{(s)}\right)_{ij} \right| < \Gamma/2$ when $\mathbf{X}_{ij}^* = 0$ and $\left| \left(\mathbf{D}^{(s)\top} \tilde{\mathbf{Y}}^{(s)}\right)_{ij} \right| > \Gamma/2$ when $\mathbf{X}_{ij}^* \neq 0$, which leads to $\text{supp}(\tilde{\mathbf{X}}^{(s)}) = \text{supp}(\mathbf{X}^*)$ with probability at least $1 - 2 \exp(-n) - 2 \exp(-Cn) - 2 \exp(\log p_2 - Cn)$. For any matrix $\mathbf{M} \in$

$\mathbb{R}^{n \times p_2}$, we define the projection operator $\mathcal{P}$ as

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

Given the exact support recovery, we have

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

As a result we have

$$\begin{aligned} \frac{\|\tilde{\mathbf{X}}^{(s)} - \mathcal{P}(\tilde{\mathbf{X}})\|_F}{\|\mathcal{P}(\tilde{\mathbf{X}})\|_2} &= \frac{\|\tilde{\mathbf{X}}^{(s)} - \mathcal{P}(\tilde{\mathbf{X}})\|_F}{\|\tilde{\mathbf{X}}\|_2} \frac{\|\tilde{\mathbf{X}}\|_2}{\|\mathcal{P}(\tilde{\mathbf{X}})\|_2} \leq \|\mathbf{D}^{(s)} - \mathbf{D}^*\|_F \frac{\|\tilde{\mathbf{X}}\|_2}{\|\tilde{\mathbf{X}}\|_2 - \|\tilde{\mathbf{X}} - \mathcal{P}(\tilde{\mathbf{X}})\|_2} \\ &\leq \|\mathbf{D}^{(s)} - \mathbf{D}^*\|_F \frac{\|\tilde{\mathbf{X}}\|_2}{\|\tilde{\mathbf{X}}\|_2 - \|\tilde{\mathbf{X}} - \mathcal{P}(\tilde{\mathbf{X}})\|_F} \stackrel{(a)}{\leq} \|\mathbf{D}^{(s)} - \mathbf{D}^*\|_F \frac{\|\tilde{\mathbf{X}}\|_2}{\|\tilde{\mathbf{X}}\|_2 - \|\tilde{\mathbf{X}} - \mathbf{X}^*\|_F} \\ &\stackrel{(b)}{\leq} \|\mathbf{D}^{(s)} - \mathbf{D}^*\|_F \left(1 + \frac{\|\tilde{\mathbf{X}} - \mathbf{X}^*\|_F}{\|\mathbf{X}^*\|_2 - 2\|\tilde{\mathbf{X}} - \mathbf{X}^*\|_F} \right) \leq \|\mathbf{D}^{(s)} - \mathbf{D}^*\|_F \left(1 + \frac{\frac{\|\tilde{\mathbf{X}} - \mathbf{X}^*\|_F}{\|\mathbf{X}^*\|_2}}{1 - \frac{2\|\tilde{\mathbf{X}} - \mathbf{X}^*\|_F}{\|\mathbf{X}^*\|_2}} \right) \\ &\stackrel{(c)}{\leq} \|\mathbf{D}^{(s)} - \mathbf{D}^*\|_F \left(1 + \frac{\frac{1}{\tau_1}}{1 - 2 \times \frac{1}{\tau_1}} \right) = \frac{\tau_1 - 1}{\tau_1 - 2} \|\mathbf{D}^{(s)} - \mathbf{D}^*\|_F. \end{aligned} \tag{2.13}$$

Here inequality (a) is due to the decomposition of $\tilde{\mathbf{X}} - \mathbf{X}^*$ onto and outside the support of $\mathbf{X}^*$. Specifically, define $\mathcal{P}_\perp(\cdot)$ as the projection onto the complement of $\mathcal{P}(\cdot)$. We have $\mathcal{P}_\perp(\tilde{\mathbf{X}} - \mathcal{P}(\tilde{\mathbf{X}})) = \mathcal{P}_\perp(\tilde{\mathbf{X}} - \mathbf{X}^*)$ and $\mathcal{P}(\tilde{\mathbf{X}} - \mathcal{P}(\tilde{\mathbf{X}})) = 0$, which in turn implies $\|\tilde{\mathbf{X}} - \mathcal{P}(\tilde{\mathbf{X}})\|_F \leq \|\tilde{\mathbf{X}} - \mathbf{X}^*\|_F$. The inequality (b) is due to $\|\mathbf{X}^*\|_2 - \|\tilde{\mathbf{X}} - \mathbf{X}^*\|_F \leq \|\mathbf{X}^*\|_2 - \|\tilde{\mathbf{X}} - \mathbf{X}^*\|_2 \leq \|\tilde{\mathbf{X}}\|_2$. The inequality (c) holds due to $\frac{\|\tilde{\mathbf{X}} - \mathbf{X}^*\|_F}{\|\mathbf{X}^*\|_2} \leq 1/\tau_1$, which is a result of Lemma 3:

$$\begin{aligned} \|\tilde{\mathbf{X}} - \mathbf{X}^*\|_F &= \|\mathbf{D}^{*\top} \mathbf{P}^{(s)} \mathbf{A}^* \mathbf{X}^* - \mathbf{X}^*\|_F \leq \|\mathbf{P}^{(s)} \mathbf{A}^* - \mathbf{D}^*\|_F \|\mathbf{X}^*\|_2 \\ &\leq \sqrt{n} \|\mathbf{P}^{(s)} \mathbf{A}^* - \mathbf{D}^*\|_2 \|\mathbf{X}^*\|_2 \lesssim \frac{\hat{\kappa}^6}{\theta} \sqrt{\frac{n^2}{p_1 + T}} \|\mathbf{X}^*\|_2, \end{aligned} \tag{2.14}$$

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 $\frac{\|\tilde{\mathbf{X}}^{(s)} - \mathcal{P}(\tilde{\mathbf{X}})\|_F}{\|\mathcal{P}(\tilde{\mathbf{X}})\|_2}$ with $\|\mathbf{D}^{(s)} - \mathbf{D}^*\|_F$. Next, we provide an

upper bound on $\|\mathbf{D}^{(s+1)} - \mathbf{D}^*\|_F$ in terms of $\frac{\|\tilde{\mathbf{X}}^{(s)} - \mathcal{P}(\tilde{\mathbf{X}})\|_F}{\|\mathcal{P}(\tilde{\mathbf{X}})\|_2}$. We have

$$\begin{aligned} \|\mathbf{D}^{(s+1)} - \mathbf{D}^*\|_F &= \|\text{Polar}(\tilde{\mathbf{Y}}^{(s)} \tilde{\mathbf{X}}^{(s)\top}) - \mathbf{D}^*\|_F = \|\text{Polar}(\mathbf{D}^* \tilde{\mathbf{X}} \tilde{\mathbf{X}}^{(s)\top}) - \mathbf{D}^*\|_F \quad (2.15) \\ &= \|\text{Polar}(\mathbf{D}^* \tilde{\mathbf{X}} \tilde{\mathbf{X}}^{(s)\top}) - \text{Polar}(\mathbf{D}^* \mathcal{P}(\tilde{\mathbf{X}}) \tilde{\mathbf{X}}^{(s)\top}) + \text{Polar}(\mathbf{D}^* \mathcal{P}(\tilde{\mathbf{X}}) \tilde{\mathbf{X}}^{(s)\top}) - \mathbf{D}^*\|_F \\ &= \|\text{Polar}(\mathbf{D}^* \tilde{\mathbf{X}} \tilde{\mathbf{X}}^{(s)\top}) - \text{Polar}(\mathbf{D}^* \mathcal{P}(\tilde{\mathbf{X}}) \tilde{\mathbf{X}}^{(s)\top})\|_F + \|\text{Polar}(\mathbf{D}^* \mathcal{P}(\tilde{\mathbf{X}}) \tilde{\mathbf{X}}^{(s)\top}) - \mathbf{D}^*\|_F \end{aligned}$$

To control the first term on the right-hand side, we invoke Theorem 6 (see Section 2.4.1):

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

where (a) follows after normalizing both the numerator and the denominator with $\|\tilde{\mathbf{X}}^{(s)}\|_2$. Moreover, (b) follows from (2.14), Lemma 4 and $p_2 = \Omega(\tau_2^4 n / \theta^2)$. Next, we provide a bound on $\kappa(\tilde{\mathbf{X}}^{(s)})$, and we do so by first bounding $\kappa(\mathcal{P}(\tilde{\mathbf{X}}))$. By Lemma 4 and Lemma 3, we have

$$\begin{aligned} \kappa(\mathcal{P}(\tilde{\mathbf{X}})) &= \frac{\|\mathcal{P}(\tilde{\mathbf{X}})\|_2}{\sigma_n(\mathcal{P}(\tilde{\mathbf{X}}))} \leq \frac{\|\mathbf{X}^*\|_2 + \|\mathcal{P}(\tilde{\mathbf{X}}) - \mathbf{X}^*\|_2}{\sigma_n(\mathbf{X}^*) - \|\mathcal{P}(\tilde{\mathbf{X}}) - \mathbf{X}^*\|_2} \quad (2.16) \\ &= \frac{\|\mathbf{X}^*\|_2 + \|\mathcal{P}(\tilde{\mathbf{X}}) - \mathbf{X}^*\|_2}{\|\mathbf{X}^*\|_2 / \kappa(\mathbf{X}^*) - \|\mathcal{P}(\tilde{\mathbf{X}}) - \mathbf{X}^*\|_2} \leq \frac{\tau_1 + 1}{\frac{\tau_1(\tau_2 - 1)}{\tau_2 + 1} - 1} \end{aligned}$$

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

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

For (a), we used (2.13) and the fact that $\|\mathbf{D}^{(s)} - \mathbf{D}^*\|_F \leq 1/4$. To conclude, we have

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

Next, we turn our focus to $\|\text{Polar}(\mathbf{D}^* \mathcal{P}(\tilde{\mathbf{X}}) \tilde{\mathbf{X}}^{(s)\top}) - \mathbf{D}^*\|_F$, which is the second term of (2.15). We first introduce the following lemma, which is borrowed from [98]:

Lemma 5 (Approximation error for dictionary (See the proof of Lemma 2.4 in [98])). *For any matrix $\mathbf{X}^* \in \mathbb{R}^{n \times p}$ and $\mathbf{Y} = \mathbf{D}^* \mathbf{X}^*$ where $\mathbf{D}^* \in \mathbb{O}^n$, consider an approximation $\mathbf{X}$ to $\mathbf{X}^*$ with $\mathbf{E} := (\mathbf{X} - \mathbf{X}^*) / \|\mathbf{X}^*\|_2$ satisfying $\|\mathbf{E}\|_F < 1/\kappa^2(\mathbf{X}^*)$. Then, for some constant $C_T > 0$, $\mathbf{D} = \text{Polar}(\mathbf{Y} \mathbf{X}^\top)$ satisfies:*

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

To invoke Lemma 5, let $\tilde{\mathbf{E}}^{(s)} = \frac{\tilde{\mathbf{X}}^{(s)} - \mathcal{P}(\tilde{\mathbf{X}})}{\|\mathcal{P}(\tilde{\mathbf{X}})\|_2}$, and notice that

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

where (a) and (b) follow from (2.13) and (2.16), respectively. Therefore, Lemma 5 can be invoked:

$$\|\text{Polar}(\mathbf{D}^* \mathcal{P}(\tilde{\mathbf{X}}) \tilde{\mathbf{X}}^{(s)\top}) - \mathbf{D}^*\|_F \leq \frac{\kappa(\mathcal{P}(\tilde{\mathbf{X}}))^4}{2} \frac{\|\mathcal{P}(\tilde{\mathbf{X}}) \tilde{\mathbf{E}}^{(s)\top} - \tilde{\mathbf{E}}^{(s)} \mathcal{P}(\tilde{\mathbf{X}})^\top\|_F}{\|\mathcal{P}(\tilde{\mathbf{X}})\|_2} + C_T \|\mathbf{E}\|_F^2. \quad (2.17)$$

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

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

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

$$\begin{aligned} \frac{\max_{1 \leq k \leq n} \|\mathbf{M}_k \mathcal{P}(\tilde{\mathbf{X}}) \tilde{\mathbf{M}}_k\|_2}{\|\mathcal{P}(\tilde{\mathbf{X}})\|_2} &\leq \frac{\max_{1 \leq k \leq n} \|\mathbf{M}_k \mathbf{X}^* \tilde{\mathbf{M}}_k\|_2 + \|\mathcal{P}(\tilde{\mathbf{X}}) - \mathbf{X}^*\|_2}{\|\mathbf{X}^*\|_2 (1 - \|\mathcal{P}(\tilde{\mathbf{X}}) - \mathbf{X}^*\|_F / \|\mathbf{X}^*\|_2)} \\ &\leq \frac{\tau_1}{\tau_1 - 1} \left(\frac{\max_{1 \leq k \leq n} \|\mathbf{M}_k \mathbf{X}^* \tilde{\mathbf{M}}_k\|_2}{\|\mathbf{X}^*\|_2} + \frac{1}{\tau_1} \right) \end{aligned} \quad (2.18)$$

Following the proof of Lemma 2, we have that with probability $1 - 2\exp(\log n - n)$ and

$$p_2 = \Omega(\tau_2^4 n / \theta^2)$$

$$\frac{\max_{1 \leq k \leq n} \|\mathbf{M}_k \mathbf{X}^* \tilde{\mathbf{M}}_k\|_2}{\|\mathbf{X}^*\|_2} \leq \frac{\tau_2 + 1}{\tau_2 - 1} \sqrt{\theta}. \quad (2.19)$$

Combined with (2.17), we have

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

Therefore, we have a bound for the second term on the right-hand side of (2.15). Recall that $\kappa(\mathcal{P}(\tilde{\mathbf{X}})) \leq \frac{\tau_1 + 1}{\frac{\tau_1(\tau_2 - 1)}{\tau_2 + 1} - 1}$ due to (2.16) and $0 < \theta < 1/2$. Therefore, similar to the proof of Lemma 2, for sufficiently large τ_1 and τ_2 and $\|\tilde{\mathbf{E}}^{(s)}\|_F \leq \frac{1}{10C_T}$, (2.15) can be rewritten as

$$\begin{aligned} \|\mathbf{D}^{(s+1)} - \mathbf{D}^*\|_F &\stackrel{(a)}{\leq} 0.9 \|\mathbf{D}^{(s)} - \mathbf{D}^*\|_F + \frac{12 \|\mathbf{X}^* - \mathcal{P}(\tilde{\mathbf{X}})\|_F}{\|\mathbf{X}^*\|_2} \\ &\leq 0.9 \|\mathbf{D}^{(s)} - \mathbf{D}^*\|_F + 12 \|\mathbf{P}^{(s)} \mathbf{A}^* - \mathbf{D}^*\|_F, \end{aligned} \quad (2.20)$$

where (a) follows from (2.13). This completes the proof of the induction step (2.8). 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 (2.20) to complete the proof of Theorem 3. To this goal, notice that

$$\begin{aligned} \|\mathbf{A}^{(s+1)} - \mathbf{A}^*\|_F &= \left\| \mathbf{P}^{(s+1)-1} \mathbf{D}^{(s+1)} - (\mathcal{L}((\mathbf{A}^* \mathbf{A}^{*\top})^{-1})^\top)^{-1} \mathbf{D}^* \right\|_F \\ &\leq \|\mathbf{D}^{(s+1)} - \mathbf{D}^*\|_F + \left\| \mathbf{P}^{(s+1)-1} - (\mathcal{L}((\mathbf{A}^* \mathbf{A}^{*\top})^{-1})^\top)^{-1} \right\|_F \\ &\leq 0.9 \|\mathbf{D}^{(s)} - \mathbf{D}^*\|_F + O(\|\mathbf{P}^{(s)} \mathbf{A}^* - \mathbf{D}^*\|_F) + O\left(\left\| \mathbf{P}^{(s+1)} - \mathcal{L}\left((\mathbf{A}^* \mathbf{A}^{*\top})^{-1}\right)^\top \right\|_F\right) \end{aligned} \quad (2.21)$$

Via a similar argument, we can write

$$\|\mathbf{D}^{(s)} - \mathbf{D}^*\|_F \leq \|\mathbf{A}^{(s)} - \mathbf{A}^*\|_F + O\left(\left\| \mathbf{P}^{(s)} - \mathcal{L}\left((\mathbf{A}^* \mathbf{A}^{*\top})^{-1}\right)^\top \right\|_F\right). \quad (2.22)$$

Combining (2.21) and (2.22), we have

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

By invoking Lemma 3, we can bound $\|\mathbf{P}^{(s)} \mathbf{A}^* - \mathbf{D}^*\|_F$, $\left\|\mathbf{P}^{(s)} - \mathcal{L}\left((\mathbf{A}^* \mathbf{A}^{*\top})^{-1}\right)^\top\right\|_F$, and $\left\|\mathbf{P}^{(s+1)} - \mathcal{L}\left((\mathbf{A}^* \mathbf{A}^{*\top})^{-1}\right)^\top\right\|_F$ with $O\left(\frac{\kappa^6}{\theta} \sqrt{\frac{n}{p_1+T}}\right)$. This completes the proof. $\square$

2.3.3 Proof of Theorem 2

Theorem 2 is a special case of Theorem 3, where we use one pre-calculated preconditioner throughout. In particular, it is easy to see that (2.20) in the proof of Theorem 3 holds after replacing $\mathbf{P}^{(s)}$ with $\mathbf{P}$:

$$\|\mathbf{D}^{(s+1)} - \mathbf{D}^*\|_F \leq 0.9\|\mathbf{D}^{(s)} - \mathbf{D}^*\|_F + 12\|\mathbf{P} \mathbf{A}^* - \mathbf{D}^*\|_F, \quad (2.23)$$

Therefore, we have

$$\begin{aligned} \|\mathbf{D}^{(s+1)} - \mathbf{D}^*\|_F &\leq 0.9\|\mathbf{D}^{(s)} - \mathbf{D}^*\|_F + 12\|\mathbf{P} \mathbf{A}^* - \mathbf{D}^*\|_F \\ &\leq 0.9^s \|\mathbf{D}^{(0)} - \mathbf{D}^*\|_F + (0.9 + \dots + 0.9^s) 12\|\mathbf{P} \mathbf{A}^* - \mathbf{D}^*\|_F \\ &= 0.9^s \|\mathbf{D}^{(0)} - \mathbf{D}^*\|_F + O(\|\mathbf{P} \mathbf{A}^* - \mathbf{D}^*\|_F). \end{aligned}$$

Given the above inequality, one can use the same argument as in (2.21) and (2.22) to establish the linear convergence of $\mathbf{A}^{(s)}$. The details are omitted for brevity. $\square$

2.4 Additional Proof

2.4.1 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 4 (Concentration of sample covariance matrix [128]). *Let $\mathbf{x}$ be a sub-Gaussian*

random vector in $\mathbb{R}^n$ with covariance matrix Σ , such that

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

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

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

Theorem 5 (Concentration of norm [128]). *Let $\mathbf{x} \in \mathbb{R}^n$ be a random vector with independent, sub-Gaussian coordinates $\mathbf{x}_i$ that satisfy $\mathbb{E}\mathbf{x}_i^2 = 1$. Then, $\|\mathbf{x}\|_2 - \sqrt{n}\|_{\psi_2} \leq CK^2$, where $K = \max_i \|\mathbf{x}_i\|_{\psi_2}$ and C is an absolute constant.*

We next introduce a perturbation bound for the polar decomposition.

Theorem 6 (Perturbation bound for polar decomposition [70]). *Let $\mathbf{A}, \tilde{\mathbf{A}} \in \mathbb{R}^{n \times n}$ be full rank matrices with polar decompositions $\mathbf{A} = \mathbf{U}\mathbf{P}$ and $\tilde{\mathbf{A}} = \tilde{\mathbf{U}}\tilde{\mathbf{P}}$. Then*

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

2.4.2 Proof of Auxiliary Lemmas

2.4.2.1 Proof of Lemma 4

Consider each column vector of $\mathbf{X}^*$ as a random vector. Upon defining $\Sigma_{\mathbf{X}^*}$ as the covariance matrix of $\mathbf{x}$, we have $\Sigma_{\mathbf{X}} = \theta\sigma^2\mathbf{I}_n$. We now use Theorem 4 to prove Lemma 4. For any unit-norm $\mathbf{z} \in \mathbb{R}^n$ and any i th column of $\mathbf{X}^*$, we have

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

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,

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

We are now ready to invoke Theorem 4. By choosing $C_{ce} = C_1/\sqrt{\theta}$, we have for some constant $C_2 > 0$:

$$\left\| \frac{1}{p} \mathbf{X}^* \mathbf{X}^{*\top} - \theta \sigma^2 \mathbf{I}_n \right\|_2 \leq C_2 \left(\sqrt{\frac{n+u}{p}} + \frac{n+u}{p} \right) \sigma^2, \quad (2.24)$$

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 (2.24) leads to the desired inequalities. $\square$

2.4.2.2 Proof of Lemma 3

Let us define $p = p_1 + t$. We start by noting that

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

Moreover, according to Theorem 4, we have

$$\left\| \frac{1}{p} \mathbf{X}^* \mathbf{X}^{*\top} - \theta \sigma^2 \mathbf{I}_{n \times n} \right\|_2 \lesssim \sigma^2 \sqrt{\frac{n+u}{p}} \quad (2.25)$$

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

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

where in the last inequality, we used $\|\mathbf{A}^*\|_2 = 1$. We define $\Delta_1 = \frac{1}{p\theta\sigma^2} \mathbf{Y} \mathbf{Y}^\top - \mathbf{A}^* \mathbf{A}^{*\top}$. Using the Taylor expansion, we have

$$\begin{aligned} \left(\frac{1}{p\theta\sigma^2} \mathbf{Y} \mathbf{Y}^\top \right)^{-1} &= (\mathbf{A}^* \mathbf{A}^{*\top} + \Delta_1)^{-1} \\ &= \left((\mathbf{A}^* \mathbf{A}^{*\top})^{1/2} \left(I_{r \times r} + (\mathbf{A}^* \mathbf{A}^{*\top})^{-1/2} \Delta_1 (\mathbf{A}^* \mathbf{A}^{*\top})^{-1/2} \right) (\mathbf{A}^* \mathbf{A}^{*\top})^{1/2} \right)^{-1} \\ &= (\mathbf{A}^* \mathbf{A}^{*\top})^{-1} - (\mathbf{A}^* \mathbf{A}^{*\top})^{-1} \Delta_1 (\mathbf{A}^* \mathbf{A}^{*\top})^{-1} + O(\Delta_1^2) \end{aligned}$$

As a result, we have

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

for some constant $C_3 > 0$. Similarly, by Corollary 4.8 from [19], we have

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

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

2.4.2.3 Proof of Lemma 2

Upon choosing $\tau = 1$ in Lemma 4, we have $\kappa(\mathbf{X}^*) < 11/9$ with probability $1 - 2\exp\{-n\}$. Then, by setting $\mathbf{E} = (\mathbf{X} - \mathbf{X}^*) / \|\mathbf{X}^*\|_2$ and $\hat{\mathbf{X}}^* = \mathbf{X}^* / \|\mathbf{X}^*\|_2$, we immediately have

$$\|\mathbf{D} - \mathbf{D}^*\|_F \leq \frac{\kappa(\mathbf{X}^*)^4}{2} \left\| \hat{\mathbf{X}}^* \mathbf{E}^\top - \mathbf{E} \hat{\mathbf{X}}^{*\top} \right\|_F + C_T \|\mathbf{E}\|_F^2 \quad (2.26)$$

after invoking Lemma 5. Now define $\mathcal{T}(\cdot)$ to be the operator that replaces all the diagonal entries of a matrix with zeros. It is easy to see that $\frac{1}{2} \left\| \hat{\mathbf{X}}^* \mathbf{E}^\top - \mathbf{E} \hat{\mathbf{X}}^{*\top} \right\|_F \leq \left\| \mathcal{T}(\hat{\mathbf{X}}^* \mathbf{E}^\top) \right\|_F$. To further investigate this bound, we introduce two matrices. Let matrix $\mathbf{M}_k$ denote an $n \times n$ diagonal matrix of ones and a zero at location (k, k) . Left multiplying $\hat{\mathbf{X}}^*$ by $\mathbf{M}_k$ corresponds to replacing the k th row of $\hat{\mathbf{X}}^*$ with zeros. Let $\tilde{\mathbf{M}}_k$ denote an $p \times p$ diagonal matrix that has ones at entries (i, i) for $i \in \text{supp}(\hat{\mathbf{X}}^*_{(k, \cdot)})$ and zeros elsewhere. Right multiplying $\hat{\mathbf{X}}^*$ by $\tilde{\mathbf{M}}_k$ corresponds to replacing all the columns that are zero at k th row with zeros. Now, we make the following observation:

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

The second equality is due to $\text{supp}(\mathbf{E}) \subseteq \text{supp}(\hat{\mathbf{X}}^*)$, and that the normalization step $\hat{\mathbf{X}}^* = \mathbf{X}^* / \|\mathbf{X}^*\|_2$ does not change the support of $\hat{\mathbf{X}}^*$. We first focus on $\max_{1 \leq k \leq n} \left\| \mathbf{M}_k \hat{\mathbf{X}}^* \tilde{\mathbf{M}}_k \right\|_2$.

Define $\mathbf{G}^k \in \mathbb{R}^{(n-1) \times p}$ as the matrix $\mathbf{M}_k \mathbf{X}^* \tilde{\mathbf{M}}_k$ after removing its k th row. One can see that $\|\mathbf{G}^k\|_2 = \|\mathbf{M}_k \mathbf{X}^* \tilde{\mathbf{M}}_k\|_2$. Without loss of generality, we assume $k = n$, which means that we remove the last row. Recall that right multiplying $\mathbf{X}^*$ by $\tilde{\mathbf{M}}_k$ will replace each column of $\mathbf{X}^*$ whose k th row is zero by an all-zero column vector. Therefore, $\mathbf{G}^k$ has the following property:

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

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

$$\max_{1 \leq k \leq n} \|\mathbf{M}_k \mathbf{X}^* \tilde{\mathbf{M}}_k\|_2 = \max_{1 \leq k \leq n} \|\mathbf{G}^k\|_2 \leq (1 + \tau^{-1/4}) \sqrt{p} \theta \sigma, \quad (2.27)$$

with probability $1 - 2 \exp(\log n - u)$ as long as $p \gtrsim \tau(n+u)/\theta^2$. Combining this result with (2.26), we have

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

with probability $1 - 2 \exp(\log n - u)$. Inequality (a) is due to Lemma 4 and (2.27). Recall that $\theta \leq 1/2$. Therefore, for sufficiently large constant τ , we have $\left(\frac{1+\tau^{-1/4}}{1-\tau^{-1/4}} \right)^5 \sqrt{\theta} \leq 0.8$. Moreover, $C_T \|\mathbf{E}\|_F \leq 0.1$, provided that $\|\mathbf{E}\|_F \leq 1/(10C_T)$. Therefore, we have $\|\mathbf{D} - \mathbf{D}^*\|_F \leq 0.9 \|\mathbf{E}\|_F$. The proof is complete upon choosing $u = n$. $\square$

2.4.2.4 Proof of Lemma 1

The (i, j) -th entry $(\mathbf{D}^\top \mathbf{Y})_{ij}$ of $\mathbf{D}^\top \mathbf{Y}$ can be written as

$$(\mathbf{D}^\top \mathbf{Y})_{ij} = \langle \mathbf{D}_{(\cdot, i)}, \mathbf{Y}_{(\cdot, j)} \rangle = \underbrace{\langle \mathbf{D}_{(\cdot, i)}, \mathbf{D}_{(\cdot, i)}^* \rangle}_{:= \mathcal{A}_{ij}} \mathbf{X}_{ij}^* + \underbrace{\sum_{k \neq i} \langle \mathbf{D}_{(\cdot, i)}, \mathbf{D}_{(\cdot, k)}^* \rangle \mathbf{X}_{kj}^*}_{:= \mathcal{B}_{ij}}. \quad (2.28)$$

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

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

As a result, when $\mathbf{X}_{ij}^* = 0$, we have $\mathcal{A}_{ij} = 0$. Moreover, when $\mathbf{X}_{ij}^* \neq 0$, we have $|\mathcal{A}_{ij}| \geq (1 - \|\mathbf{D}^* - \mathbf{D}\|_{1,2})\Gamma \geq \frac{3\Gamma}{4}$, where the last inequality holds for every $n \geq \frac{16}{\sigma^2\theta}$. 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 $\mathbf{D}$ such that $\|\mathbf{D} - \mathbf{D}^*\|_{1,2} \leq C_1/n$ for some C_1 to be defined later, we have $|\mathcal{B}_{ij}| \leq \Gamma/4$ with high probability. Note that this will lead to $\left|(\mathbf{D}^\top \mathbf{Y})_{ij}\right| \geq |\mathcal{A}_{ij}| - |\mathcal{B}_{ij}| \geq \Gamma/2$, thereby proving the statement. We have

$$\begin{aligned} |\mathcal{B}_{ij}| &= \left| \left\langle \mathbf{D}_{(\cdot,i)}, \sum_{k \neq i} \mathbf{D}_{(\cdot,k)}^* \mathbf{X}_{kj}^* \right\rangle \right| \\ &= \left| \left\langle \mathbf{D}_{(\cdot,i)} - \mathbf{D}_{(\cdot,i)}^*, \sum_{k \neq i} \mathbf{D}_{(\cdot,k)}^* \mathbf{X}_{kj}^* \right\rangle \right| \\ &\leq \|\mathbf{D}_{(\cdot,i)} - \mathbf{D}_{(\cdot,i)}^*\|_2 \left\| \sum_{k \neq i} \mathbf{D}_{(\cdot,k)}^* \mathbf{X}_{kj}^* \right\|_2 \end{aligned} \quad (2.29)$$

$$\begin{aligned} &\leq \|\mathbf{D}_{(\cdot,i)} - \mathbf{D}_{(\cdot,i)}^*\|_2 \left\| \sum_{k=1}^n \mathbf{D}_{(\cdot,k)}^* \mathbf{X}_{kj}^* \right\|_2 \\ &= \|\mathbf{D}_{(\cdot,i)} - \mathbf{D}_{(\cdot,i)}^*\|_2 \|\mathbf{Y}_{(\cdot,j)}\|_2 \\ &= \|\mathbf{D}_{(\cdot,i)} - \mathbf{D}_{(\cdot,i)}^*\|_2 \|\mathbf{X}_{(\cdot,j)}\|_2. \end{aligned} \quad (2.30)$$

Following the argument of Equation (2.11), we have $\|\mathbf{X}_{(\cdot,j)}\|_2 \leq 2\sigma\sqrt{\theta n}$ with probability at least $1 - 2\exp(-Cn)$. With the same probability, we have for any $\mathbf{D}$ that satisfies $\|\mathbf{D} - \mathbf{D}^*\|_{1,2} \leq \frac{\Gamma}{8\sigma\sqrt{\theta n}}$:

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

Upon taking the union bound over all j th column of $\mathbf{X}$, 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.

□

2.4.3 Rank-one Updates for the Preconditioner

We define

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

The above rank-one update algorithm for Cholesky decomposition (Algorithm 7) is already implemented in MATLAB function `cholupdate`.

Algorithm 7 Rank-one update for the preconditioner

```

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

```

2.5 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 16GB unified memory for a serial implementation in MATLAB 2022a. The code is available at: <https://github.com/lianggeyuleo/CompleteDL.git>.

2.5.1 Synthetic Dataset

We validate our theoretical results using synthetic dataset. We consider the generative model $\mathbf{Y} = \mathbf{A}^*\mathbf{X}^*$ where $\mathbf{A}^*$ is a randomly generated orthogonal or full-rank dictionary. Moreover, $\mathbf{X}^*$ is generated from a Gaussian-Bernoulli distribution followed by a truncation step, where the entries $\mathbf{X}_{ij}^*$ with $|\mathbf{X}_{ij}^*| < \Gamma = 0.3$ are replaced by $\mathbf{X}_{ij}^* = \text{sgn}(\mathbf{X}_{ij}^*)\Gamma$. 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 1. We invoke the warm-start method in Algorithm 6 to generate the initial points. For each trial, we stop the algorithm when the consecutive iterates are close to each other ($\|\mathbf{D}^{(t-1)} - \mathbf{D}^{(t)}\|_2 \leq 10^{-7}$) and record the running time and final error ($\|\mathbf{D}^{(T)} - \mathbf{D}^*\|_F$). We select the following algorithms as candidates:

Many algorithms for dictionary learning exhibit poor scalability in this setting. For instance, the convergence time of *KSVD method* [4], which is perhaps the most well-known alternating minimization algorithm for dictionary learning, exceeds 300 seconds even for our smallest instance $n = 50$. Another example is the alternating minimization based on ℓ_1 -regularization [2], which suffers from a similar scalability issue. Instead, in Figure 2.3, we consider two other candidate algorithms that have comparable scalability with ours: *Gradient-based method* [7]: In this method, the dictionary update step (Step 4 in Algorithm 3) is replaced by a gradient step $\mathbf{D}^{(t+1)} = \mathbf{D}^{(t)} - \eta \nabla_{\mathbf{D}^{(t)}} L$ 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. *ℓ_4 -maximization-based method* [139]: The work [139] introduces a projected gradient ascend for maximizing the objective function $f(\mathbf{D}) = \|\mathbf{D}^\top \mathbf{Y}\|_4^4$. This method is shown to have superior performance compared to SPAMS [54] and Subgradient method [10].

As can be seen in Figure 2.3, 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 2.4, we report the performance of Algorithm 4 with warm-start for varying sample sizes. The true dictionary $\mathbf{A}^*$ 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. Similar to Figure 2.2, 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 with the additional error term, which diminishes to zero as $p \rightarrow \infty$.

Finally, we numerically test the stability and sample complexity of our algorithm as $\mathbf{A}^*$ becomes increasingly ill-conditioned. In Figure 2.5 (right), we present the results, illustrating the sample size required to achieve a final error of $\frac{\|\mathbf{A}^{(T)} - \mathbf{A}^*\|_F}{\|\mathbf{A}^*\|_F} \leq 0.1$ as the condition number of $\mathbf{A}^*$ varies from 1 to 1000. Surprisingly, our practical findings indicate that the relationship 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

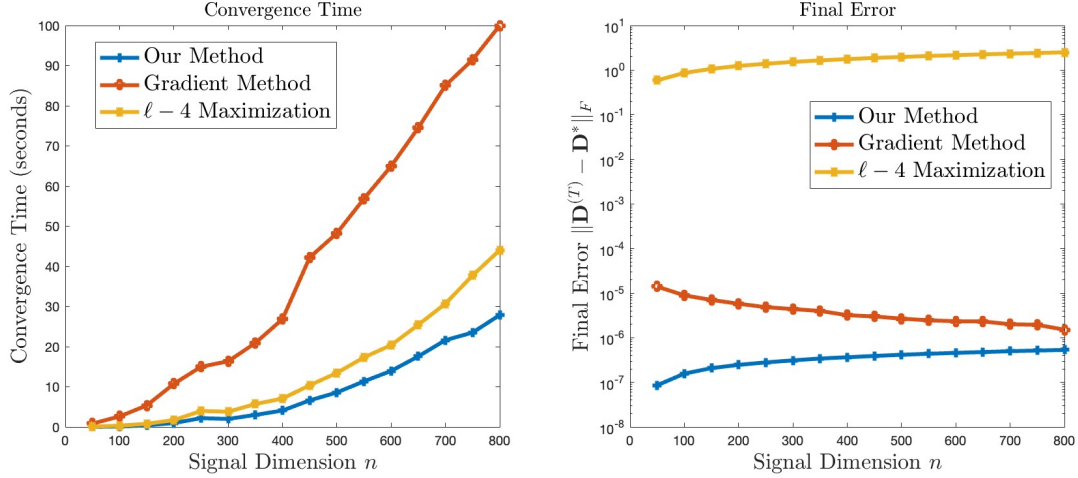


Figure 2.3: We compare three different dictionary learning methods with their running time and final error until convergence. The results above are averaged over 5 independent trials. All methods use the same initial point. The stopping criteria is when consecutive iterates are close to each other ($\|D^{(t-1)} - D^{(t-1)}\|_2 \leq 10^{-7}$).

a noisy setting, where $\mathbf{Y} = \mathbf{A}^* \mathbf{X}^* + \mathcal{E}$, with $\mathcal{E}$ representing a noise matrix whose elements follow a Gaussian distribution with zero mean and variance β^2 . As illustrated in Figure 2.5 (left), our algorithm remains unaffected by the increasing condition number $\kappa(\mathbf{A}^*)$ in the noiseless scenario. However, it encounters difficulties in recovering $\mathbf{A}^*$ 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.

2.5.2 Real Dataset

In this section, we showcase the performance of Algorithm 5 in learning a dictionary for the Landscape Dataset [121] consisting of 7,000 colored images of different landscapes, 20 of which are set aside as 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 [4] 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 $\mathbf{Y}$ of size $300 \times 10,000$ with $n = 300$ and $p = 10,000$. Our goal is to learn a dictionary $\mathbf{A}$ of size 300×300 for this data matrix.

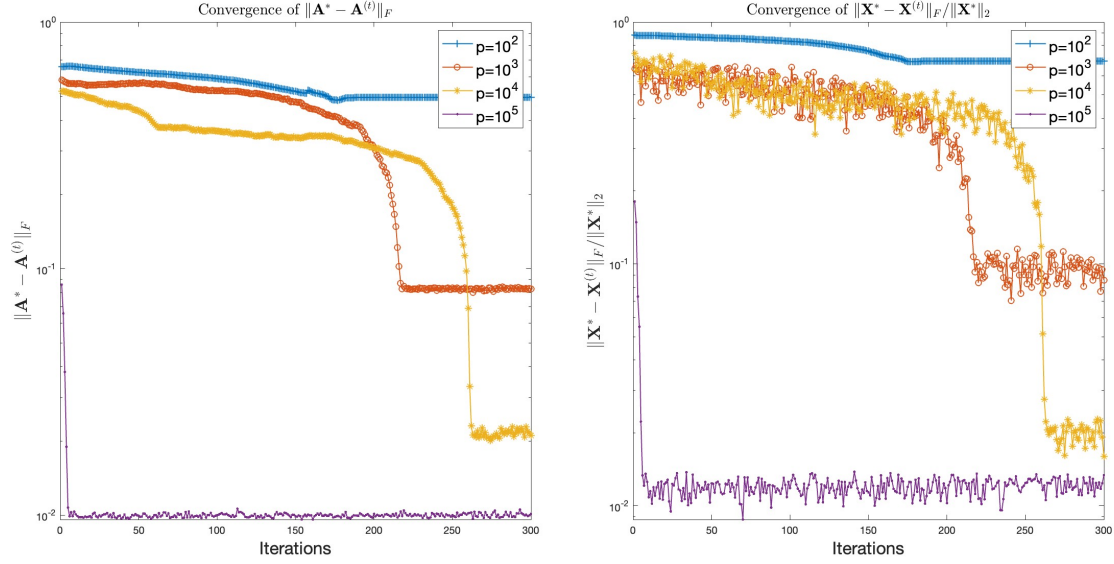


Figure 2.4: Results of Algorithm 4 with $n = 5$, $\theta = 0.3$, $\tilde{p} = 100$, and varying p . The specific number of iterations to reach convergence varies with the sample size since it depends on the distance between the ground truth and initialization.

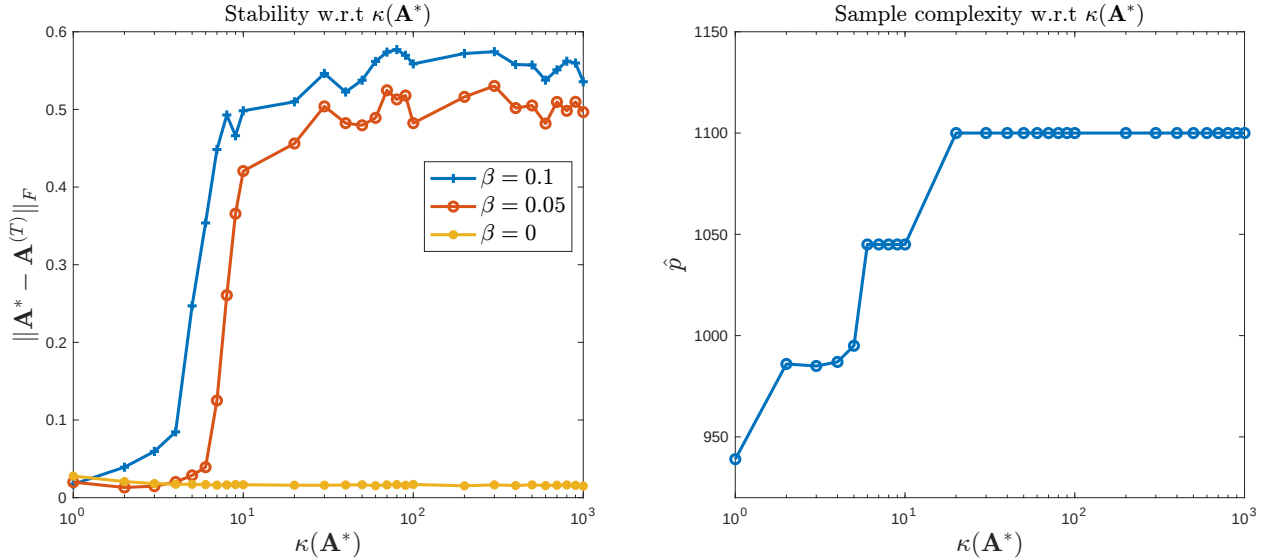


Figure 2.5: (left) Final error of Algorithm 2.2 in relation to condition number $\kappa(\mathbf{A}^*)$ and varied noise levels β for a fixed sample size of $\tilde{p} = p = 10^5$. (right) Required sample size to achieve $\frac{\|\mathbf{A}^{(T)} - \mathbf{A}^*\|_F}{\|\mathbf{A}^*\|_F} \leq 0.1$ as a function of the condition number $\kappa(\mathbf{A}^*)$. In both settings, we fix $n = 5$, $\theta = 0.3$, and $T = 1000$.

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 RGB 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 [81].

For both the image denoising examples in Figure 2.1 and Figure 2.7, we set $p_1 = p_2 = 500$ in Algorithm 5 (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 $\mathbf{Y}$. 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 2.7. 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 [81] in Matlab.

We allow KSVD a total running time of 1100 seconds. A comparison of denoised results using dictionaries learned with Algorithm 5 and KSVD are shown in Figure 2.1. Here, we also see that the dictionary learned by Algorithm 5 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 2.6 that our algorithm achieves a similarity score of 0.9 in 120 seconds, while KSVD can only achieve a similarity score of 0.28 in 2500 seconds.

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 blackout 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 5 for a total of $T = 2,000$, which consumed approximately 2,100 seconds. For the reconstruction process, we maintain a sparsity level of approximately 10%, as we did previously.

Our reconstruction results are depicted in Figure 2.7. We randomly blackout 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

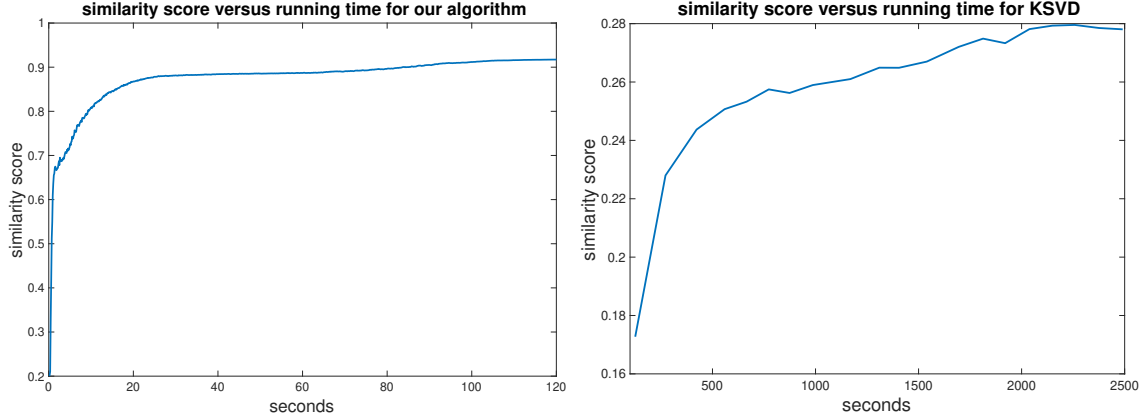


Figure 2.6: **Similarity score between reconstructed image and original image** We display the similarity score between the denoised image and the original one for our algorithm (left) and KSVD (right). As time progresses, the dictionary obtained from our algorithm improves, leading to increasingly better reconstructions. On the other hand, while the dictionary obtained from KSVD improves with time, the reconstruction quality exhibits a slower rate of improvement.

dictionary performs well in reconstructing the images. However, when dealing with larger 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.

2.6 Concluding Remarks

In this chapter, 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 chapter 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.



Figure 2.7: **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.

CHAPTER 3

Dictionary Learning for Heterogeneous Distribution

This chapter is based on the paper *Personalized Dictionary Learning for Heterogeneous Datasets* [74] by Geyu Liang, Naichen Shi, Raed Al Kontar, Salar Fattahi, which is accepted at 37th Conference on Neural Information Processing Systems (NeurIPS 2023).

3.1 Introduction

Recall that, given a set of n signals $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_n] \in \mathbb{R}^{d \times p}$, *dictionary learning* (DL) aims to find a *dictionary* $\mathbf{D} \in \mathbb{R}^{d \times n}$ and a corresponding *code* $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n] \in \mathbb{R}^{r \times p}$ such that: (1) each data sample $\mathbf{y}_i$ can be written as $\mathbf{y}_i = \mathbf{D}\mathbf{x}_i$ for $1 \leq i \leq n$, and (2) the code $\mathbf{X}$ has as few nonzero elements as possible. The columns of the dictionary $\mathbf{D}$, also known as *atoms*, encode the “common features” whose linear combinations form the data samples. Once again we recall the typical approach to solve DL is via the following optimization problem:

$$\min_{\mathbf{X}, \mathbf{D}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 + \lambda \|\mathbf{X}\|_{\ell_q}, \quad (\text{DL})$$

Here $\|\cdot\|_{\ell_q}$ is often modeled as a ℓ_1 -norm [7, 2] or ℓ_0 -(pseudo-)norm [113, 75] and has the role of promoting sparsity in the estimated sparse code. Due to its effective feature extraction and representation, DL has found immense applications in data analytics, with applications ranging from clustering and classification ([96, 123]), to image denoising ([71]), to document detection([56]), to medical imaging ([142]), and to many others.

However, the existing formulations of DL hinge on a critical assumption: the homogeneity of the data. It is assumed that the samples share the same set of features (atoms) collected in a *single* dictionary D . This assumption, however, is challenged in practice as the data is typically collected and processed in heterogeneous edge devices (clients). These clients (for instance, smartphones and wearable devices) operate in different conditions [62] while

sharing some congruity. Accordingly, the collected datasets are naturally endowed with heterogeneous features while potentially sharing common ones. In such a setting, the classical formulation of DL faces a major dilemma: on the one hand, a reasonably-sized dictionary (with a moderate number of atoms n) may overlook the unique features specific to different clients. On the other hand, collecting both shared and unique features in a single enlarged dictionary (with a large number of atoms n) may lead to computational, privacy, and identifiability issues. In addition, both approaches fail to provide information about “what is shared and unique” which may offer standalone intrinsic value and can potentially be exploited for improved clustering, classification and anomaly detection, amongst others.

With the goal of addressing data heterogeneity in dictionary learning, in this chapter, we propose *personalized dictionary learning* (PerDL); a framework that can untangle and recover *global* and *local (unique)* dictionaries from heterogeneous datasets. The global dictionary, which collects atoms that are shared among all clients, represents the common patterns among datasets and serves as a conduit of collaboration in our framework. The local dictionaries, on the other hand, provide the necessary flexibility for our model to accommodate data heterogeneity.

We summarize our contributions below:

- *Identifiability of local and global atoms:* We provide conditions under which the local and global dictionaries can be provably identified and separated by solving a nonconvex optimization problem. At a high level, our identifiability conditions entail that the true dictionaries are column-wise incoherent, and the local atoms do not have a significant alignment along any nonzero vector.
- *Federated meta-algorithm:* We present a fully federated meta-algorithm, called PerMA (Algorithm 8), for solving PerDL. PerMA only requires communicating the estimated dictionaries among the clients, thereby circumventing the need for sharing any raw data. A key property of PerMA is its ability to untangle global and local dictionaries by casting it as a series of shortest path problems over a *directed acyclic graph* (DAG).
- *Theoretical guarantees:* We prove that, under moderate conditions on the generative model and the clients, PerMA enjoys a linear convergence to the ground truth up to a statistical error. Additionally, PerMA borrows strength from strong clients to improve the performance of the weak ones. More concretely, through collaboration, our framework provides weak clients with the extra benefits of *averaged* initial condition, convergence rate, and final statistical error.
- *Practical performance:* We showcase the performance of PerMA on a synthetic dataset,

as well as different realistic learning tasks, such as training with imbalanced datasets and video surveillance. These experiments highlight that our method can effectively extract shared global features while preserving unique local ones, ultimately improving performance through collaboration.

3.1.1 Related Works

Dictionary Learning [113, 75] provide conditions under which DL can be provably solved, provided that the dictionary is a square matrix (also known as *complete* DL). For the more complex case of *overcomplete* DL with $n > d$, [8, 7, 2] show that alternating minimization achieves desirable statistical and convergence guarantees. Inspired by recent results on the benign landscape of matrix factorization [39, 35], [117] show that a smoothed variant of DL is devoid of spurious local solutions. In contrast, distributed or federated variants of DL are far less explored. [52, 44] study DL in the federated setting. However, they do not provide any provable guarantees on their proposed method.

Federated Learning & Personalization Recent years have seen explosive interest in federated learning (FL) following the seminal paper on federated averaging [85]. Literature along this line has primarily focused on predictive modeling using deep neural networks (DNN), be it through enabling faster convergence [55], improving aggregation schemes at a central server [130], promoting fairness across all clients [135] or protecting against potential adversaries [18]. Numerous variants have been proposed to solve related problems [32, 107]. A comprehensive survey of existing methodology can be found in [61]. More recently, the focus has been placed on tackling heterogeneity across client datasets through personalization. The key idea is to allow each client to retain their own tailored model instead of learning one model that fits everyone. Approaches along this line either split weights of a DNN into shared and unique ones and collaborate to learn the shared weights [76, 119], or follow a train-then-personalize approach where a global model is learned and fine-tuned locally, often iteratively [72]. Again such models have mainly focused on predictive models. Whilst this literature abounds, personalization that aims to identify what is shared and unique across datasets is very limited. Very recently, personalized PCA [108] was proposed to address this challenge through identifiably extracting shared and unique principal components using distributed Riemannian gradient descent. However, PCA cannot accommodate sparsity in representation and requires orthogonality constraints that may limit its application. In contrast, our work considers a broader setting via sparse dictionary learning.

Notation. In this chapter, for a matrix $\mathbf{A}$, we use $\mathbf{A}_i$ to indicate that it belongs to client i . We use $\mathcal{P}(n)$ to denote the set of $n \times n$ signed permutation matrices. We define $[n] = \{1, 2, \dots, n\}$.

3.2 PerDL: Personalized Dictionary Learning

In PerDL, we are given N clients, each with p_i samples collected in $\mathbf{Y}_i \in \mathbb{R}^{d \times p_i}$ and generated as a sparse linear combination of n^g global atoms and n_i^l local atoms:

$$\mathbf{Y}_i = \mathbf{D}_i^* \mathbf{X}_i^*, \quad \text{where} \quad \mathbf{D}_i^* = \begin{bmatrix} \mathbf{D}^{g*} & \mathbf{D}_i^{l*} \end{bmatrix}, \quad \text{for} \quad i = 1, \dots, N. \quad (3.1)$$

Here $\mathbf{D}^{g*} \in \mathbb{R}^{d \times n^g}$ denotes a global dictionary that captures the common features shared among all clients, whereas $\mathbf{D}_i^{l*} \in \mathbb{R}^{d \times n_i^l}$ denotes the local dictionary specific to each client. Let $n_i = n^g + n_i^l$ denote the total number of atoms in $\mathbf{D}_i^*$. Without loss of generality, we assume the columns of $\mathbf{D}_i^*$ have unit ℓ_2 -norm. This assumption is without loss of generality since, for any dictionary-code pair $(\mathbf{D}_i, \mathbf{X}_i)$, the columns of $\mathbf{D}_i$ can be normalized to have unit norm by re-weighting the corresponding rows of $\mathbf{X}_i$. The goal in PerDL is to recover $\mathbf{D}^{g*}$ and $\{\mathbf{D}_i^{l*}\}_{i=1}^N$, as well as the sparse codes $\{\mathbf{X}_i^*\}_{i=1}^N$, given the datasets $\{\mathbf{Y}_i\}_{i=1}^N$. Before presenting our approach for solving PerDL, we first consider the following fundamental question: under what conditions is the recovery of the dictionaries $\mathbf{D}^{g*}, \{\mathbf{D}_i^{l*}\}_{i=1}^N$ and sparse codes $\{\mathbf{X}_i^*\}_{i=1}^N$ well-posed?

To answer this question, we first note that it is only possible to recover the dictionaries and sparse codes up to a signed permutation: given any signed permutation matrix $\mathbf{\Pi} \in \mathcal{P}(n_i)$, the dictionary-code pairs $(\mathbf{D}_i, \mathbf{X}_i)$ and $(\mathbf{D}_i \mathbf{\Pi}_i, \mathbf{\Pi}_i^\top \mathbf{X}_i)$ are equivalent. This invariance with respect to signed permutation gives rise to an equivalent class of true solutions with a size that grows exponentially with the dimension. To guarantee the recovery of a solution from this equivalent class, we need the μ -incoherency of the true dictionaries.

Assumption 2 (μ -incoherency). *For each client $1 \leq i \leq N$, the dictionary $\mathbf{D}_i^*$ is μ -incoherent for some constant $\mu > 0$, that is,*

$$\max_{j,k} \left| \left\langle (\mathbf{D}_i^*)_{(:,j)}, (\mathbf{D}_i^*)_{(:,k)} \right\rangle \right| \leq \frac{\mu}{\sqrt{d}}. \quad (3.2)$$

Assumption 2 is standard in dictionary learning ([2, 7, 24]) and was independently introduced by [37, 125] in signal processing and [141, 86] in statistics. Intuitively, it requires the atoms in each dictionary to be approximately orthogonal. To see the necessity of this assumption, consider a scenario where two atoms of $\mathbf{D}_i^*$ are perfectly aligned (i.e., $\mu = \sqrt{d}$).

In this case, using either of these two atoms achieve a similar effect in reconstructing $\mathbf{Y}_i$, contributing to the ill-posedness of the problem.

Our next assumption guarantees the separation of local dictionaries from the global one in PerDL. First, we introduce several signed permutation-invariant distance metrics for dictionaries, which will be useful for later analysis.

Definition 1. For two dictionaries $\mathbf{D}_1, \mathbf{D}_2 \in \mathbb{R}^{d \times r}$, we define their signed permutation-invariant $\ell_{1,2}$ -distance and ℓ_2 -distance as follows:

$$d_{1,2}(\mathbf{D}_1, \mathbf{D}_2) := \min_{\mathbf{\Pi} \in \mathcal{P}(r)} \|\mathbf{D}_1 \mathbf{\Pi} - \mathbf{D}_2\|_{1,2}, \quad (3.3)$$

$$d_2(\mathbf{D}_1, \mathbf{D}_2) := \min_{\mathbf{\Pi} \in \mathcal{P}(r)} \|\mathbf{D}_1 \mathbf{\Pi} - \mathbf{D}_2\|_2. \quad (3.4)$$

Furthermore, suppose $\mathbf{\Pi}^* = \arg \min_{\mathbf{\Pi} \in \mathcal{P}(r)} \|\mathbf{D}_1 \mathbf{\Pi} - \mathbf{D}_2\|_{1,2}$. For any $1 \leq j \leq r$, we define

$$d_{2,(j)}(\mathbf{D}_1, \mathbf{D}_2) := \left\| (\mathbf{D}_1 \mathbf{\Pi}^* - \mathbf{D}_2)_{(:,j)} \right\|_2. \quad (3.5)$$

Assumption 3 (β -identifiability). The local dictionaries $\{\mathbf{D}_i^{l*}\}_{i=1}^N$ are β -identifiable for some constant $0 < \beta < 1$, that is, there exists no vector $\mathbf{v} \in \mathbb{R}^d$ with $\|\mathbf{v}\|_2 = 1$ such that

$$\max_{1 \leq i \leq N} \min_{1 \leq j \leq n^l} d_2 \left((\mathbf{D}_i^{l*})_{(:,j)}, \mathbf{v} \right) \leq \beta. \quad (3.6)$$

Suppose there exists a unit-norm $\mathbf{v}$ satisfying (3.6) for some small $\beta > 0$. This implies that $\mathbf{v}$ is sufficiently close to at least one atom from each local dictionary. Indeed, one may treat this atom as part of the global dictionary, thereby violating the identifiability of local and global dictionaries. On the other hand, the infeasibility of (3.6) for large $\beta > 0$ implies that the local dictionaries are sufficiently dispersed, which in turn facilitates their identification.

With the above assumptions in place, we are ready to present our proposed optimization problem for solving PerDL:

$$\min_{\mathbf{D}^g, \{\mathbf{D}_i\}, \{\mathbf{X}_i\}} \sum_{i=1}^N \|\mathbf{Y}_i - \mathbf{D}_i \mathbf{X}_i\|_F^2 + \lambda \sum_{i=1}^N \|\mathbf{X}_i\|_{\ell_q}, \quad \text{s.t.} \quad (\mathbf{D}_i)_{(1:n^g)} = \mathbf{D}^g \quad \text{for } 1 \leq i \leq N. \quad (\text{PerDL-NCVX})$$

For each client i , PerDL-NCVX aims to recover a dictionary-code pair $(\mathbf{D}_i, \mathbf{X}_i)$ that match $\mathbf{Y}_i$ under the constraint that dictionaries for individual clients share the same global components.

3.3 PerMA: Meta-algorithm of Solving PerDL

In this section, we introduce our meta-algorithm (Algorithm 8) for solving PerDL-NCVX, which we call *Personalized Matching and Averaging* (PerMA). In what follows, we explain

Algorithm 8 PerMA: Federated Matching and Averaging

```

1: Input:  $\{\mathbf{Y}_i\}_{i=1}^N$ .
2: for client  $i = 1, \dots, N$  do
3:   Client: Obtain  $\mathbf{D}_i^{(0)}$  based on  $\mathbf{Y}_i$ . // Initialization step
4:   Client: Send  $\mathbf{D}_i^{(0)}$  to the server.
5: end for
6: Server:  $(\mathbf{D}^{g,(0)}, \{\mathbf{D}_i^{l,(0)}\}_{i=1}^N) = \text{global\_matching}(\{\mathbf{D}_i^{(0)}\}_{i=1}^N)$ 
   // Separating local from global dictionaries
7: Server: Broadcast  $(\mathbf{D}^{g,(0)}, \{\mathbf{D}_i^{l,(0)}\}_{i=1}^N)$ 
8: for  $t = 0, 1, \dots, T$  do
9:   for client  $i = 1, \dots, N$  do
10:    Client:  $(\mathbf{D}_i^{g,(t+1)}, \mathbf{D}_i^{l,(t+1)}) = \text{local\_update}(\mathbf{Y}_i, \mathbf{D}^{g,(t)}, \mathbf{D}_i^{l,(t)})$ 
    // Updating the local and global dictionaries for each client
11:    Client: Send  $\mathbf{D}_i^{g,(t+1)}$  to the server.
12:   end for
13:   Server: Calculate  $\mathbf{D}^{g,(t+1)} = \frac{1}{N} \sum_{i=1}^N \mathbf{D}_i^{g,(t+1)}$ . // Averaging global dictionaries
14:   Server: Broadcast  $\mathbf{D}^{g,(t+1)}$ .
15: end for
16: return  $(\mathbf{D}^{g,(T)}, \{\mathbf{D}_i^{l,(T)}\}_{i=1}^N)$ .

```

the steps of PerMA:

Local initialization (Step 3): PerMA starts with a warm-start step where each client runs their own initialization scheme to obtain $\mathbf{D}_i^{(0)}$. This step is necessary even for the classical DL to put the initial point inside a basin of attraction of the ground truth. Several spectral methods were proposed to provide a theoretically good initialization[7, 2], while in practice, it is reported that random initialization followed by a few iterations of alternating minimization approach will suffice [98, 75].

Global matching scheme (Step 6) Given the clients' initial dictionaries, our global matching scheme separates the global and local parts of each dictionary by solving a series of shortest path problems on an auxiliary graph. Then, it obtains a refined estimate of the global dictionary via simple averaging. A detailed explanation of this step is provided in the next section.

Dictionary update at each client (Step 10) During each communication round, the clients refine their own dictionary based on the available data, the aggregated global dictionary, and the previous estimate of their local dictionary. A detailed explanation of this step is provided in the next section.

Global aggregation (Step 13) At the end of each round, the server updates the clients' estimate of the global dictionary by computing their average.

A distinguishing property of **PerMA** is that it only requires the clients to communicate their dictionaries and not their sparse codes. In fact, after the global matching step on the initial dictionaries, the clients only need to communicate their global dictionaries, keeping their local dictionaries private.

3.3.1 Global Matching and Local Updates

In this section, we provide detailed implementations of `global_matching` and `local_update` subroutines in **PerMA** (Algorithm 8).

Given the initial approximations of the clients' dictionaries $\{\mathbf{D}_i^{(0)}\}_{i=1}^N$, `global_matching` seeks to identify and aggregate the global dictionary by extracting the similarities among the atoms of $\{\mathbf{D}_i^{(0)}\}$. To identify the global components, one approach is to solve the following optimization problem

$$\min_{\mathbf{\Pi}_i} \sum_{i=1}^{N-1} \left\| \left(\mathbf{D}_i^{(0)} \mathbf{\Pi}_i \right)_{(:,1:n^g)} - \left(\mathbf{D}_{i+1}^{(0)} \mathbf{\Pi}_{i+1} \right)_{(:,1:n^g)} \right\|_2 \quad \text{s.t.} \quad \mathbf{\Pi}_i \in \mathcal{P}(n_i) \quad \text{for} \quad 1 \leq i \leq N. \quad (3.7)$$

The above optimization aims to obtain the appropriate signed permutation matrices $\{\mathbf{\Pi}_i\}_{i=1}^N$ that align the first n^g atoms of the permuted dictionaries. In the ideal regime where $\mathbf{D}_i^{(0)} = \mathbf{D}_i^*$, $1 \leq i \leq N$, the optimal solution $\{\mathbf{\Pi}_i^*\}_{i=1}^N$ yields a zero objective value and satisfies $\left(\mathbf{D}_i^{(0)} \mathbf{\Pi}_i^* \right)_{(:,1:n^g)} = \mathbf{D}^{g*}$. However, there are two main challenges with the above optimization. First, it is a nonconvex, combinatorial problem over the discrete sets $\{\mathcal{P}(n_i)\}$. Second, the initial dictionaries may not coincide with their true counterparts. To address the first challenge, we show that the optimal solution to the optimization (3.7) can be efficiently obtained by solving a series of shortest path problems defined over an auxiliary graph. To alleviate the second challenge, we show that our proposed algorithm is robust against possible errors in the initial dictionaries.

Consider a weighted N -layered *directed acyclic graph* (DAG) $\mathcal{G}$ with n_i nodes in layer i representing the n_i atoms in $\mathbf{D}_i$. We connect any node a from layer i to any node b from layer $i + 1$ with a directed edge with weight $w(a, b) = d_2 \left((\mathbf{D}_i)_{(:,a)}, (\mathbf{D}_{i+1})_{(:,b)} \right)$. We add a *source*

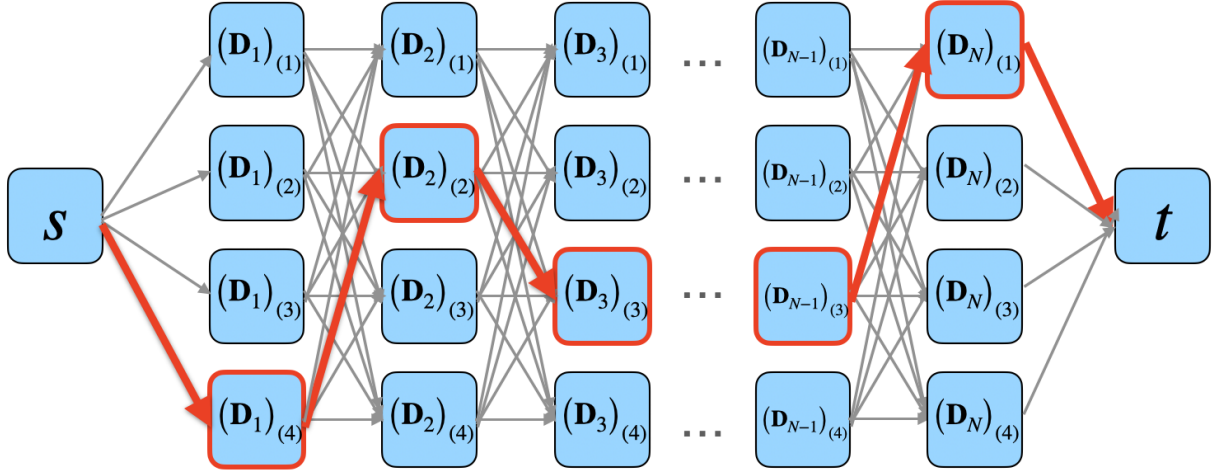


Figure 3.1: A schematic diagram for `global_matching` (Algorithm 9). At each iteration, we find the shortest path from s to t (highlighted with red), estimate one atom of $\mathbf{D}^{g*}$ using all passed vertices and remove the path (including the vertices) from $\mathcal{G}$.

node s and connect it to all nodes in layer 1 with weight 0. Similarly, we include a *terminal* node t and connect all nodes in layer N to t with weight 0. A schematic construction of this graph is presented in Figure 3.1. Given the constructed graph, Algorithm 9 aims to solve (3.7) by running n^g rounds of the shortest path problem: at each round, the algorithm identifies the most aligned atoms in the initial dictionaries by obtaining the shortest path from s to t . Then it removes the used nodes in the path for the next round. The correctness and robustness of the proposed algorithm are established in the next theorem.

Algorithm 9 `global_matching`

- 1: **Input:** $\{\mathbf{D}_i^{(0)}\}_{i=1}^N$ and n^g .
 - 2: Construct the weighted N -layer DAG $\mathcal{G}$ described in Section 3.3.1.
 - 3: Initialize $\{\text{index}_i\}_{i=1}^N$ as empty sets and $\mathbf{D}^{g,(0)}$ as an empty matrix.
 - 4: **for** $j = 1, \dots, n^g$ **do**
 - 5: Find the shortest path $\mathcal{P} = (s, (\mathbf{D}_1^{(0)})_{(:,\alpha_1)}, (\mathbf{D}_1^{(0)})_{(:,\alpha_2)}, \dots, (\mathbf{D}_N^{(0)})_{(:,\alpha_N)}, t)$.
 - 6: Add $\frac{1}{N} \sum_{i=1}^N \text{sign} \left(\left\langle (\mathbf{D}_i^{(0)})_{(:,\alpha_i)}, (\mathbf{D}_1^{(0)})_{(:,\alpha_1)} \right\rangle \right) (\mathbf{D}_i^{(0)})_{(:,\alpha_i)}$ as a new column of $\mathbf{D}^{g,(0)}$.
 - 7: Add α_i to index_i for every $i = 1, \dots, N$.
 - 8: Remove $\mathcal{P}$ from $\mathcal{G}$.
 - 9: **end for**
 - 10: Set $\mathbf{D}_i^{l,(0)} = (\mathbf{D}_i^{(0)})_{(:, [n_i] \setminus \text{index}_i)}$ for every $i = 1, \dots, N$.
 - 11: **return** $\left(\mathbf{D}^{g,(0)}, \{\mathbf{D}_i^{l,(0)}\}_{i=1}^N \right)$.
-

Theorem 7 (Correctness and robustness of `global_matching`). Suppose $\{\mathbf{D}_i^*\}_{i=1}^N$ are μ -incoherent (Assumption 2) and β -identifiable (Assumption 3). Suppose the initial dictionaries $\{\mathbf{D}_i^{(0)}\}_{i=1}^N$ satisfy $d_{1,2}(\mathbf{D}_i^{(0)}, \mathbf{D}_i^*) \leq \epsilon_i$ with $4 \sum_{i=1}^N \epsilon_i \leq \min \left\{ \sqrt{2 - 2\frac{\mu}{\sqrt{d}}}, \beta \right\}$. Then, the output of Algorithm 9 satisfies:

$$d_{1,2}(\mathbf{D}^{g,(0)}, \mathbf{D}^{g*}) \leq \frac{1}{N} \sum_{i=1}^N \epsilon_i, \quad \text{and} \quad d_{1,2}(\mathbf{D}_i^{l,(0)}, \mathbf{D}_i^{l*}) \leq \epsilon_i, \quad \text{for } 1 \leq i \leq N. \quad (3.8)$$

According to the above theorem, `global_matching` can robustly separate the clients' initial dictionaries into global and local parts, provided that the aggregated error in the initial dictionaries is below a threshold. Specific initialization schemes that can satisfy the condition of Theorem 7 include Algorithm 1 from [3] and Algorithm 3 from [7]. We also remark that since the constructed graph is a DAG, the shortest path problem can be solved in time linear in the number of edges, which is $\mathcal{O}(n_1 + n_N + \sum_{i=1}^{N-1} n_i n_{i+1})$, via a simple labeling algorithm (see, e.g., [5, Chapter 4.4]). Since we need to solve the shortest path problem n^g times, this brings the computational complexity of Algorithm 9 to $\mathcal{O}(n^g N n_{\max}^2)$, where $n_{\max} = \max_i n_i$.

Given the initial local and global dictionaries, the clients progressively refine their estimates by applying T rounds of `local_update` (Algorithm 10). At a high level, each client runs a single iteration of a *linearly convergent algorithm* $\mathcal{A}_i$ (see Definition 2), followed by an alignment step that determines the global atoms of the updated dictionary using $\mathbf{D}^{g,(t)}$ as a "reference point". Notably, our implementation of `local_update` is adaptive to different DL algorithms. This flexibility is indeed intentional to provide a versatile meta-algorithm for clients with different DL algorithms.

Algorithm 10 `local_update`

- 1: **Input:** $\mathbf{D}_i^{(t)} = [\mathbf{D}^{g,(t)} \quad \mathbf{D}_i^{l,(t)}], \mathbf{Y}_i$
 - 2: $\mathbf{D}_i^{(t+1)} = \mathcal{A}_i(\mathbf{Y}_i, \mathbf{D}_i^{(t)})$ // One iteration of a linearly-convergent algorithm.
 - 3: Initialize $\mathcal{S}$ as an empty set and $\mathbf{P} \in \mathbb{R}^{n^g \times n^g}$ as an all-zero matrix.
 - 4: **for** $j = 1, \dots, n^g$ **do**
 - 5: Find $k^* = \arg \min_k d_2 \left((\mathbf{D}^{g,(t)})_{(:,j)}, (\mathbf{D}_i^{(t+1)})_{(:,k)} \right)$.
 - 6: Append k^* to $\mathcal{S}$.
 - 7: Set (i, i) -th entry of $\mathbf{P}$ to $\text{sign} \left(\left\langle (\mathbf{D}^{g,(t)})_{(:,j)}, (\mathbf{D}_i^{(t+1)})_{(:,k^*)} \right\rangle \right)$.
 - 8: **end for**
 - 9: **Output:** $\mathbf{D}_i^{g,(t+1)} = (\mathbf{D}_i^{(t+1)})_{(:,\mathcal{S})} \mathbf{P}$ and $\mathbf{D}_i^{l,(t+1)} = (\mathbf{D}_i^{(t+1)})_{(:, [n_i] \setminus \mathcal{S})}$.
-

3.4 Theoretical Guarantees

In this section, we show that our proposed meta-algorithm provably solves PerDL under suitable initialization, identifiability, and algorithmic conditions. To achieve this goal, we first present the definition of a linearly-convergent DL algorithm.

Definition 2. *Given a generative model $\mathbf{Y} = \mathbf{D}^* \mathbf{X}^*$, a DL algorithm $\mathcal{A}$ is called (δ, ρ, ψ) -linearly convergent for some parameters $\delta, \psi > 0$ and $0 < \rho < 1$ if, for any $\mathbf{D} \in \mathbb{R}^{d \times r}$ such that $d_{1,2}(\mathbf{D}, \mathbf{D}^*) \leq \delta$, the output of one iteration $\mathbf{D}^+ = \mathcal{A}(\mathbf{D}, \mathbf{Y})$, satisfies*

$$d_{2,(j)}(\mathbf{D}^+, \mathbf{D}^*) \leq \rho d_{2,(j)}(\mathbf{D}, \mathbf{D}^*) + \psi, \quad \forall 1 \leq j \leq r. \quad (3.9)$$

One notable linearly convergent algorithm is introduced by [7, Algorithm 5]; we will discuss this algorithm in more detail in the following section. Assuming all clients are equipped with linearly convergent algorithms, our next theorem establishes the convergence of PerMA.

Theorem 8 (Convergence of PerMA). *Suppose $\{\mathbf{D}_i^*\}_{i=1}^N$ are μ -incoherent (Assumption 2) and β -identifiable (Assumption 3). Suppose, for every client i , the DL algorithm $\mathcal{A}_i$ used in **local.update** (Algorithm 10) is $(\delta_i, \rho_i, \psi_i)$ -linearly convergent with $4 \sum_{i=1}^N \delta_i \leq \min \left\{ \sqrt{2 - 2\frac{\mu}{\sqrt{d}}}, \beta \right\}$. Assume the initial dictionaries $\{\mathbf{D}_i^{(0)}\}_{i=1}^N$ satisfy:*

$$d_{1,2} \left(\frac{1}{N} \sum_{i=1}^N \mathbf{D}_i^{g,(0)}, \mathbf{D}^{g*} \right) \leq \min_{1 \leq i \leq N} \delta_i, \quad d_{1,2}(\mathbf{D}_i^{l,(0)}, \mathbf{D}_i^{l*}) \leq \delta_i, \quad \text{for } i = 1, \dots, N. \quad (3.10)$$

Then, for every $t \geq 0$, PerMA (Algorithm 8) satisfies

$$d_{1,2}(\mathbf{D}^{g,(t)}, \mathbf{D}^{g*}) \leq \left(\frac{1}{N} \sum_{i=1}^N \rho_i \right) d_{1,2}(\mathbf{D}^{g,(0)}, \mathbf{D}^{g*}) + \frac{1}{N} \sum_{i=1}^N \psi_i, \quad (3.11)$$

$$d_{1,2}(\mathbf{D}_i^{l,(t)}, \mathbf{D}_i^{l*}) \leq \rho_i d_{1,2}(\mathbf{D}_i^{l,(0)}, \mathbf{D}_i^{l*}) + \psi_i, \quad \text{for } 1 \leq i \leq N. \quad (3.12)$$

The above theorem sheds light on a number of key benefits of PerMA:

Relaxed initial condition for weak clients. Our algorithm relaxes the initial condition on the global dictionaries. In particular, it only requires the average of the initial global dictionaries to be close to the true global dictionary. Consequently, it enjoys a provable convergence guarantee even if some of the clients do not provide a high-quality initial dictionary to the server.

Improved convergence rate for slow clients. During the course of the algorithm, the global dictionary error decreases at an average rate of $\frac{1}{N} \sum_{i=1}^N \rho_i$, improving upon the convergence rate of the slow clients.

Improved statistical error for weak clients. A linearly convergent DL algorithm $\mathcal{A}_i$ stops making progress upon reaching a neighborhood around the true dictionary $\mathbf{D}_i^*$ with radius $O(\psi_i)$. This type of guarantee is common among DL algorithms [7, 75] and often corresponds to their statistical error. In light of this, PerMA improves the performance of weak clients (i.e., clients with weak statistical guarantees) by borrowing strength from strong ones.

3.5 Further Discussion on Linearly Convergent Algorithms

In this section, we discuss a linearly convergent DL algorithm that satisfies the conditions of our Theorem 8. In particular, the next theorem is adapted from [7, Theorem 12] and shows that a modified variant of Algorithm 11 introduced in [7, Algorithm 5] is indeed linearly-convergent.

Theorem 9 (Linear convergence of Algorithm 5 in [7]). *Suppose that the data matrix satisfies $\mathbf{Y} = \mathbf{D}^* \mathbf{X}^*$, where $\mathbf{D}^*$ is an μ -incoherent dictionary and the sparse code $\mathbf{X}^*$ satisfies the generative model introduced in Section 1.2 and Section 4.1 of [7]. For any initial dictionary $\|\mathbf{D}^{(0)}\|_2 \leq 1$, Algorithm 5 in [7] is (δ, ρ, ψ) -linearly convergent with $\delta = O(1/\log d)$, $\rho \in (1/2, 1)$, and $\psi = O(d^{-\omega(1)})$.*

Algorithm 5 in [7] is a refinement of Algorithm 11, where the error is further reduced by projecting out the components along the column currently being updated. For brevity, we do not discuss the exact implementation of the algorithm; an interested reader may refer to [7] for more details. Indeed, we have observed in our experiments that the additional projection step does not provide a significant benefit over Algorithm 11.

3.6 Proof of Theorems

For simplicity, we use $(\mathbf{D}_i)_j$ to replace $(\mathbf{D}_i)_{(:,j)}$ in this section.

3.6.1 Proof of Theorem 7

To begin with, we establish a triangular inequality for $d_{1,2}(\cdot, \cdot)$, which will be important in our subsequent arguments:

Lemma 6 (Triangular inequality for $d_{1,2}(\cdot, \cdot)$). *For any dictionary $\mathbf{D}_1, \mathbf{D}_2, \mathbf{D}_3 \in \mathbb{R}^{d \times r}$, we have*

$$d_{1,2}(\mathbf{D}_1, \mathbf{D}_2) \leq d_{1,2}(\mathbf{D}_1, \mathbf{D}_3) + d_{1,2}(\mathbf{D}_3, \mathbf{D}_2) \quad (3.13)$$

Proof. Suppose $\mathbf{\Pi}_{1,3}$ and $\mathbf{\Pi}_{3,2}$ satisfy $d_{1,2}(\mathbf{D}_1, \mathbf{D}_3) = \|\mathbf{D}_1 \mathbf{\Pi}_{1,3} - \mathbf{D}_3\|_{1,2}$ and $d_{1,2}(\mathbf{D}_3, \mathbf{D}_2) = \|\mathbf{D}_3 - \mathbf{D}_2 \mathbf{\Pi}_{3,2}\|_{1,2}$. Then we have

$$\begin{aligned} d_{1,2}(\mathbf{D}_1, \mathbf{D}_3) + d_{1,2}(\mathbf{D}_3, \mathbf{D}_2) &= \|\mathbf{D}_1 \mathbf{\Pi}_{1,3} - \mathbf{D}_3\|_{1,2} + \|\mathbf{D}_3 - \mathbf{D}_2 \mathbf{\Pi}_{3,2}\|_{1,2} \\ &\geq \|\mathbf{D}_1 \mathbf{\Pi}_{1,3} - \mathbf{D}_2 \mathbf{\Pi}_{3,2}\|_{1,2} \\ &\geq d_{1,2}(\mathbf{D}_1, \mathbf{D}_2). \end{aligned} \quad (3.14)$$

□

Given how the directed graph $\mathcal{G}$ is constructed and modified, any directed path from s to t will be of the form $\mathcal{P} = s \rightarrow (\mathbf{D}_1^{(0)})_{\alpha(1)} \rightarrow (\mathbf{D}_2^{(0)})_{\alpha(2)} \rightarrow \cdots \rightarrow (\mathbf{D}_N^{(0)})_{\alpha(N)} \rightarrow t$. Specifically, each layer (or client) contributes exactly one node (or atom), and the path is determined by $\alpha(\cdot) : [N] \rightarrow [r]$.

Recall that $\mathbf{D}_i^* = [\mathbf{D}^{g*} \quad \mathbf{D}_i^{l*}]$ for every $1 \leq i \leq N$. Assume, without loss of generality, that for every client $1 \leq i \leq N$,

$$\mathbf{I}_{n_i \times n_i} = \arg \min_{\mathbf{\Pi} \in \mathcal{P}(n_i)} \left\| \mathbf{D}_i^* \mathbf{\Pi} - \mathbf{D}_i^{(0)} \right\|_{1,2}. \quad (3.15)$$

In other words, the first n^g atoms in the initial dictionaries $\{\mathbf{D}_i^{(0)}\}_{i=1}^N$ are aligned with the global dictionary. Now consider the special path $\mathcal{P}_j^*$ for $1 \leq j \leq n^g$ defined as

$$\mathcal{P}_j^* = s \rightarrow (\mathbf{D}_1^{(0)})_j \rightarrow (\mathbf{D}_2^{(0)})_j \rightarrow \cdots \rightarrow (\mathbf{D}_N^{(0)})_j \rightarrow t. \quad (3.16)$$

To prove that Algorithm 9 correctly selects and aligns global atoms from clients, it suffices to show that $\{\mathcal{P}_j^*\}_{j=1}^{n^g}$ are the top- n^g shortest paths from s to t in $\mathcal{G}$. The length of the path

$\mathcal{P}_j^*$ can be bounded as

$$\begin{aligned}
\mathcal{L}(\mathcal{P}_j^*) &= \sum_{i=1}^{N-1} d_2 \left((\mathbf{D}_i^{(0)})_j, (\mathbf{D}_{i+1}^{(0)})_j \right) \\
&= \sum_{i=1}^{N-1} \min \left\{ \|(\mathbf{D}_i^{(0)})_j - (\mathbf{D}_{i+1}^{(0)})_j\|_2, \|(\mathbf{D}_i^{(0)})_j + (\mathbf{D}_{i+1}^{(0)})_j\|_2 \right\} \\
&\leq \sum_{i=1}^{N-1} \|(\mathbf{D}_i^{(0)})_j - (\mathbf{D}_{i+1}^{(0)})_j\|_2 \\
&\leq \sum_{i=1}^{N-1} \|(\mathbf{D}_i^{(0)})_j - (\mathbf{D}^{g*})_j\|_2 + \|(\mathbf{D}_{i+1}^{(0)})_j - (\mathbf{D}^{g*})_j\|_2 \\
&\leq \sum_{i=1}^{N-1} (\epsilon_i + \epsilon_{i+1}) \\
&\leq 2 \sum_{i=1}^N \epsilon_i.
\end{aligned} \tag{3.17}$$

We move on to prove that all the other paths from s to t will have a distance longer than $2 \sum_{i=1}^N \epsilon_i$. Consider a general directed path $\mathcal{P} = s \rightarrow (\mathbf{D}_1^{(0)})_{\alpha(1)} \rightarrow (\mathbf{D}_2^{(0)})_{\alpha(2)} \rightarrow \cdots \rightarrow (\mathbf{D}_N^{(0)})_{\alpha(N)} \rightarrow t$ that is not in $\{\mathcal{P}_j^*\}_{j=1}^{n^g}$. Based on whether or not $\mathcal{P}$ contains atoms that align with the true global ground atoms, there are two situations:

Case 1: Suppose there exists $1 \leq i \leq N$ such that $\alpha(i) \leq n^g$. Given Model 3.1 and the assumed equality (3.15), we know that for layer i , $\mathcal{P}$ contains a global atom. Since $\mathcal{P}$ is not in $\{\mathcal{P}_j^*\}_{j=1}^{n^g}$, there must exist $k \neq i$ such that $\alpha(k) \neq \alpha(i)$. As a result, we have

$$\begin{aligned}
\mathcal{L}(\mathcal{P}) &\stackrel{(a)}{\geq} d_{1,2} \left((\mathbf{D}_i^{(0)})_{\alpha(i)}, (\mathbf{D}_k^{(0)})_{\alpha(k)} \right) \\
&\stackrel{(b)}{\geq} \min \left\{ \|(\mathbf{D}_i^*)_{\alpha(i)} - (\mathbf{D}_k^*)_{\alpha(k)}\|_2, \|(\mathbf{D}_i^*)_{\alpha(i)} + (\mathbf{D}_k^*)_{\alpha(k)}\|_2 \right\} \\
&\quad - \|(\mathbf{D}_i^*)_{\alpha(i)} - (\mathbf{D}_i^{(0)})_{\alpha(i)}\|_2 - \|(\mathbf{D}_k^*)_{\alpha(k)} - (\mathbf{D}_k^{(0)})_{\alpha(k)}\|_2 \\
&\stackrel{(c)}{\geq} \sqrt{2 - 2 |\langle (\mathbf{D}_k^*)_{\alpha(i)}, (\mathbf{D}_k^*)_{\alpha(k)} \rangle|} - 2 \max_{1 \leq i \leq N} \epsilon_i \\
&\stackrel{(d)}{\geq} \sqrt{2 - 2 \frac{\mu}{\sqrt{d}}} - 2 \max_{1 \leq i \leq N} \epsilon_i \\
&\stackrel{(e)}{\geq} 2 \sum_{i=1}^N \epsilon_i^g
\end{aligned} \tag{3.18}$$

Here (a) and (b) are due to Lemma 6, (c) is due to assumed equality (3.15), (d) is due to the μ -incoherency of $\mathbf{D}_k^*$, and finally (e) is given by the assumption of Theorem 7.

Case 2: Suppose $\alpha(i) > n^g$ for all $1 \leq i \leq N$, which means that the path $\mathcal{P}$ only uses approximations to local atoms. Consider the ground truth of these approximations, $(\mathbf{D}_1^*)_{\alpha(1)}, (\mathbf{D}_2^*)_{\alpha(2)}, \dots, (\mathbf{D}_N^*)_{\alpha(N)}$. There must exist $1 \leq i, j \leq N$ such that $d_{1,2}((\mathbf{D}_i^*)_{\alpha(i)}, (\mathbf{D}_j^*)_{\alpha(j)}) \geq \beta$. Otherwise, it is easy to see that $\{\mathbf{D}_i^*\}_{i=1}^N$ would not be β -identifiable because any $(\mathbf{D}_i^*)_{\alpha(i)}$ will satisfy (3.6). As a result, we have the following:

$$\begin{aligned}
\mathcal{L}(\mathcal{P}) &\geq d_{1,2}\left((\mathbf{D}_i^{(0)})_{\alpha(i)}, (\mathbf{D}_j^{(0)})_{\alpha(j)}\right) \\
&\geq d_{1,2}\left((\mathbf{D}_i^*)_{\alpha(i)}, (\mathbf{D}_j^*)_{\alpha(j)}\right) - \|(\mathbf{D}_i^*)_{\alpha(i)} - (\mathbf{D}_i^{(0)})_{\alpha(i)}\|_2 - \|(\mathbf{D}_j^*)_{\alpha(j)} - (\mathbf{D}_j^{(0)})_{\alpha(j)}\|_2 \\
&\geq \beta - 2 \max_i \epsilon_i \\
&\geq 2 \sum_{i=1}^N \epsilon_i
\end{aligned} \tag{3.19}$$

So we have shown that $\{\mathcal{P}_j^*\}_{j=1}^{n^g}$ are the top- n^g shortest paths from s to t in $\mathcal{G}$. Moreover, it is easy to show that $\text{sign}\left(\left\langle (\mathbf{D}_1^{(0)})_j, (\mathbf{D}_i^{(0)})_j \right\rangle\right) = 1$ for small enough $\{\epsilon_i\}_{i=1}^N$. Therefore, the proposed algorithm correctly recovers the global dictionaries (with the correct identity permutation). Finally, we have $\mathbf{D}^{g,(0)} = \frac{1}{N} \sum_{i=1}^N (\mathbf{D}_i^{(0)})_{1:n^g}$, which leads to:

$$\begin{aligned}
d_{1,2}(\mathbf{D}^{g,(0)}, \mathbf{D}^{g*}) &\leq \max_{1 \leq j \leq n^g} \left\| \frac{1}{N} \sum_{i=1}^N (\mathbf{D}_i^{(0)})_j - (\mathbf{D}^{g*})_j \right\|_2 \\
&\leq \max_{1 \leq j \leq n^g} \frac{1}{N} \sum_{i=1}^N \|(\mathbf{D}_i^{(0)})_j - (\mathbf{D}^{g*})_j\|_2 \\
&\leq \max_{1 \leq j \leq n^g} \frac{1}{N} \sum_{i=1}^N \epsilon_i \\
&= \frac{1}{N} \sum_{i=1}^N \epsilon_i.
\end{aligned} \tag{3.20}$$

This completes the proof of Theorem 7. □

3.6.2 Proof of Theorem 8

Throughout this section, we define:

$$\bar{\rho} := \frac{1}{N} \sum_{i=1}^N \rho_i, \quad \bar{\psi} := \frac{1}{N} \sum_{i=1}^N \psi_i. \tag{3.21}$$

We will prove the convergence of the global dictionary in Theorem 8 by proving the following induction: at each $t \geq 1$, we have

$$d_{1,2}(\mathbf{D}^{g,(t+1)}, \mathbf{D}^{g*}) \leq \bar{\rho} d_{1,2}(\mathbf{D}^{g,(t)}, \mathbf{D}^{g*}) + \bar{\psi}. \quad (3.22)$$

At the beginning of communication round t , each client i performs `local_update` to get $\mathbf{D}_i^{(t+1)}$ given $\begin{bmatrix} \mathbf{D}^{g,(t)} & \mathbf{D}_i^{l,(t)} \end{bmatrix}$. Without loss of generality, we assume

$$\mathbf{I}_{n_i \times n_i} = \arg \min_{\mathbf{\Pi} \in \mathcal{P}(n_i)} \left\| \mathbf{D}_i^* \mathbf{\Pi} - \begin{bmatrix} \mathbf{D}^{g,(t)} & \mathbf{D}_i^{l,(t)} \end{bmatrix} \right\|_{1,2}, \quad (3.23)$$

$$\mathbf{I}_{n_i \times n_i} = \arg \min_{\mathbf{\Pi} \in \mathcal{P}(n_i)} \left\| \mathbf{D}_i^* \mathbf{\Pi} - \mathbf{D}_i^{(t+1)} \right\|_{1,2}. \quad (3.24)$$

Assumed equalities (3.23) and (3.24) imply that the permutation matrix that aligns the input and the output of $\mathcal{A}_i$ is also $\mathbf{I}_{n_i \times n_i}$. Specifically, the linear convergence property of $\mathcal{A}_i$ and Theorem 7 thus suggest:

$$\left\| \left(\mathbf{D}_i^{(t+1)} \right)_j - (\mathbf{D}_i^*)_j \right\|_2 \leq \rho_i \left\| (\mathbf{D}^{g,(t)})_j - (\mathbf{D}_i^*)_j \right\|_2 + \psi_i \quad \forall 1 \leq j \leq n^g, 1 \leq i \leq N. \quad (3.25)$$

However, our algorithm is unaware of this trivial alignment. We will next show the remaining steps in `local_update` correctly recovers the identity permutation. The proof is very similar to the proof of Theorem 7 since we are essentially running Algorithm 9 on a two-layer $\mathcal{G}$. For every $1 \leq i \leq N$, $1 \leq j \leq n^g$, we have

$$\begin{aligned} d_{1,2} \left(\left(\mathbf{D}_i^{(t+1)} \right)_j, (\mathbf{D}^{g,(t)})_j \right) &\leq d_{1,2} \left(\left(\mathbf{D}_i^{(t+1)} \right)_j, (\mathbf{D}_i^*)_j \right) + d_{1,2} \left((\mathbf{D}_i^*)_j, (\mathbf{D}^{g,(t)})_j \right) \\ &\leq 2\delta_i. \end{aligned} \quad (3.26)$$

Meanwhile for $k \neq j$,

$$\begin{aligned} &d_{1,2} \left(\left(\mathbf{D}_i^{(t+1)} \right)_k, (\mathbf{D}^{g,(t)})_j \right) \\ &\geq d_{1,2} \left((\mathbf{D}_i^*)_k, (\mathbf{D}_i^*)_j \right) - d_{1,2} \left(\left(\mathbf{D}_i^{(t+1)} \right)_k, (\mathbf{D}_i^*)_k \right) - d_{1,2} \left((\mathbf{D}_i^*)_j, (\mathbf{D}^{g,(t)})_j \right) \\ &\geq \sqrt{2 - \frac{2\mu}{\sqrt{d}}} - 2\delta_i. \\ &\geq 2\delta_i. \end{aligned} \quad (3.27)$$

As a result, we successfully recover the identity permutation, which implies

$$\left\| \left(\mathbf{D}_i^{g,(t+1)} \right)_j - \left(\mathbf{D}_i^{g*} \right)_j \right\|_2 \leq \rho_i \left\| \left(\mathbf{D}^{g,(t)} \right)_j - \left(\mathbf{D}_i^{g*} \right)_j \right\|_2 + \psi_i \quad \forall 1 \leq j \leq n^g, 1 \leq i \leq N. \quad (3.28)$$

Finally, the aggregation step (Step 13 in Algorithm 8) gives:

$$\begin{aligned} d_{1,2} \left(\mathbf{D}^{g,(t+1)}, \mathbf{D}^{g*} \right) &\leq \left\| \frac{1}{N} \sum_{i=1}^N \mathbf{D}_i^{g,(t+1)} - \mathbf{D}^{g*} \right\|_{1,2} \\ &= \max_{1 \leq j \leq n^g} \left\| \left(\frac{1}{N} \sum_{i=1}^N \mathbf{D}_i^{g,(t+1)} \right)_j - \left(\mathbf{D}^{g*} \right)_j \right\| \\ &\leq \max_{1 \leq j \leq n^g} \frac{1}{N} \sum_{i=1}^N \left\| \left(\mathbf{D}_i^{g,(t+1)} \right)_j - \left(\mathbf{D}_i^{g*} \right)_j \right\|_2 \\ &\leq \max_{1 \leq j \leq n^g} \frac{1}{N} \sum_{i=1}^N \left(\rho_i \left\| \left(\mathbf{D}^{g,(t)} \right)_j - \left(\mathbf{D}_i^{g*} \right)_j \right\|_2 + \psi_i \right) \\ &\leq \frac{1}{N} \sum_{i=1}^N \left(\rho_i d_{1,2} \left(\mathbf{D}^{g,(t)}, \mathbf{D}^{g*} \right) + \psi_i \right) \\ &= \bar{\rho} d_{1,2} \left(\mathbf{D}^{g,(t)}, \mathbf{D}^{g*} \right) + \bar{\psi}. \end{aligned} \quad (3.29)$$

As a result, we prove the induction (3.22) for all $0 \leq t \leq T - 1$. Inequality (3.12) is a by-product of the accurate separation of global and local atoms and can be proved by similar arguments. The proof is hence complete. $\square$

3.7 Experiments Overview

In this section, we showcase the effectiveness of Algorithm 8 using synthetic and real data. All experiments are performed on a MacBook Pro 2021 with the Apple M1 Pro chip and 16GB unified memory for a serial implementation in MATLAB 2022a. Due to limited space, we will only provide the high-level motivation and implication of our experiments. We defer implementation details and comparisons with the existing methods to the next section.

3.7.1 Synthetic Dataset

In this section, we validate our theoretical results on a synthetic dataset. We consider ten clients, each with a dataset generated according to the model 3.1. The details of our construction are presented in the the next section. Specifically, we compare the performances

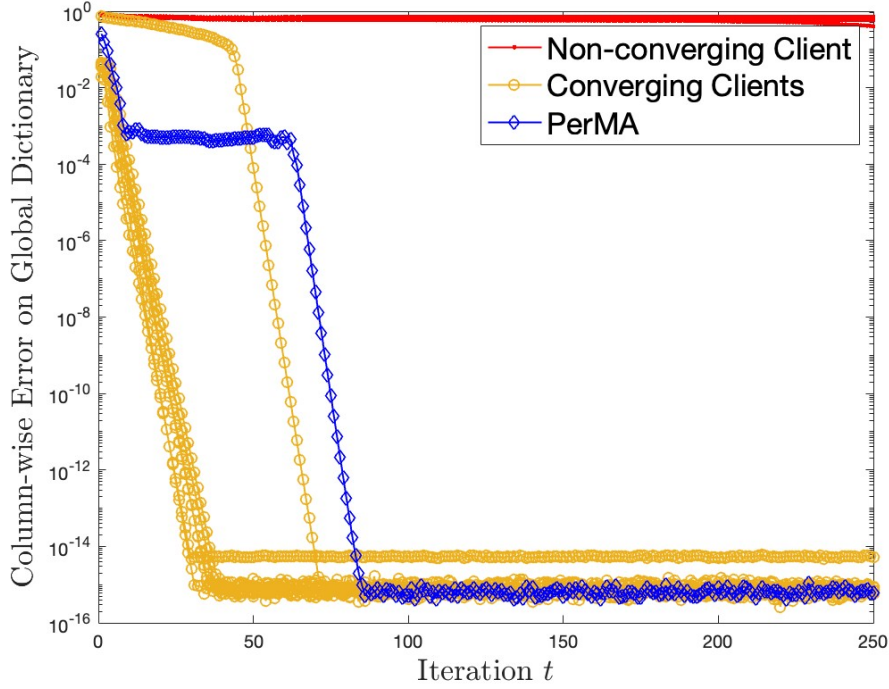


Figure 3.2: PerMA improves the accuracy of the recovered global dictionary for *all* clients, even if some (three out of ten) are weak learners.

of two strategies: (1) *independent strategy*, where each client solves DL without any collaboration, and (2) *collaborative strategy*, where clients collaboratively learn the ground truth dictionaries by solving PerDL via the proposed meta-algorithm PerMA. We initialize both strategies using the same $\{\mathbf{D}_i^{(0)}\}_{i=1}^N$. The initial dictionaries are obtained via a warm-up method proposed in [75, Algorithm 4]. For a fair comparison between independent and collaborative strategies, we use the same DL algorithm ([75, Algorithm 1]) for different clients. Note that in the independent strategy, the clients cannot separate global from local dictionaries. Nonetheless, to evaluate their performance, we collect the atoms that best align with the true global dictionary $\mathbf{D}^{g*}$ and treat them as the estimated global dictionaries. As can be seen in Figure 3.2, three out of ten clients are weak learners and fail to recover the global dictionary with desirable accuracy. On the contrary, in the collaborative strategy, all clients recover the same global dictionary almost exactly.

3.7.2 Training with Imbalanced Data

In this section, we showcase the application of PerDL in training with imbalanced datasets. We consider an image reconstruction task on MNIST dataset. This dataset corresponds

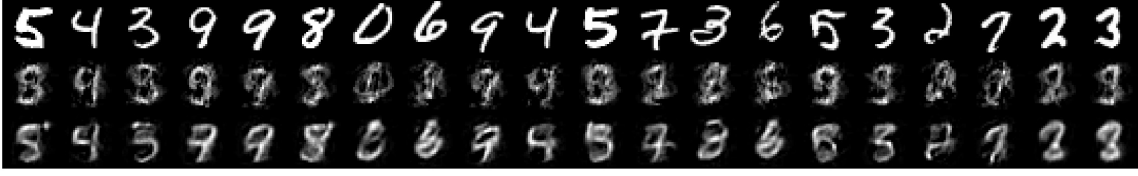


Figure 3.3: PerMA improves training with imbalanced datasets. We consider the image reconstruction task on the imbalanced MNIST dataset using only five atoms from a learned global dictionary. The first row corresponds to the original images. The second row is based on the dictionary learned on a single client with an imbalanced dataset. The third row shows the improved performance of the learned dictionary using our proposed method on the same client.

to a set of handwritten digits (see the first row of Figure 3.3). The goal is to recover a *single* concise global dictionary that can be used to reconstruct the original handwritten digits as accurately as possible. In particular, we study a setting where the clients have *imbalanced label distributions*. Indeed, data imbalance can drastically bias the performance of the trained model in favor of the majority groups, while hurting its performance on the minority groups [69, 120]. Here, we consider a setting where the clients have highly imbalanced datasets, where 90% of their samples have the same label. More specifically, for client i , we assume that 90% of the samples correspond to the handwritten digit “ i ”, with the remaining 10% corresponding to other digits. The second row of Figure 3.3 shows the effect of data imbalance on the performance of the recovered dictionary on a single client, when the clients do not collaborate. The last row of Figure 3.3 shows the improved performance of the recovered dictionary via PerDL on the same client. Our experiment clearly shows the ability of PerDL to effectively address the data imbalance issue by combining the strengths of different clients.

3.7.3 Surveillance Video Dataset

As a proof of concept, we consider a video surveillance task, where the goal is to separate the background from moving objects. Our data is collected from [126] (see the first column of Figure 3.4). As these frames are taken from one surveillance camera, they share the same background corresponding to the global features we aim to extract. The frames also exhibit heterogeneity as moving objects therein are different from the background. This problem can indeed be modeled as an instance of PerDL, where each video frame can be assigned to a “client”, with the global dictionary capturing the background and local dictionaries modeling the moving objects. We solve PerDL by applying PerMA to obtain a global dictionary and

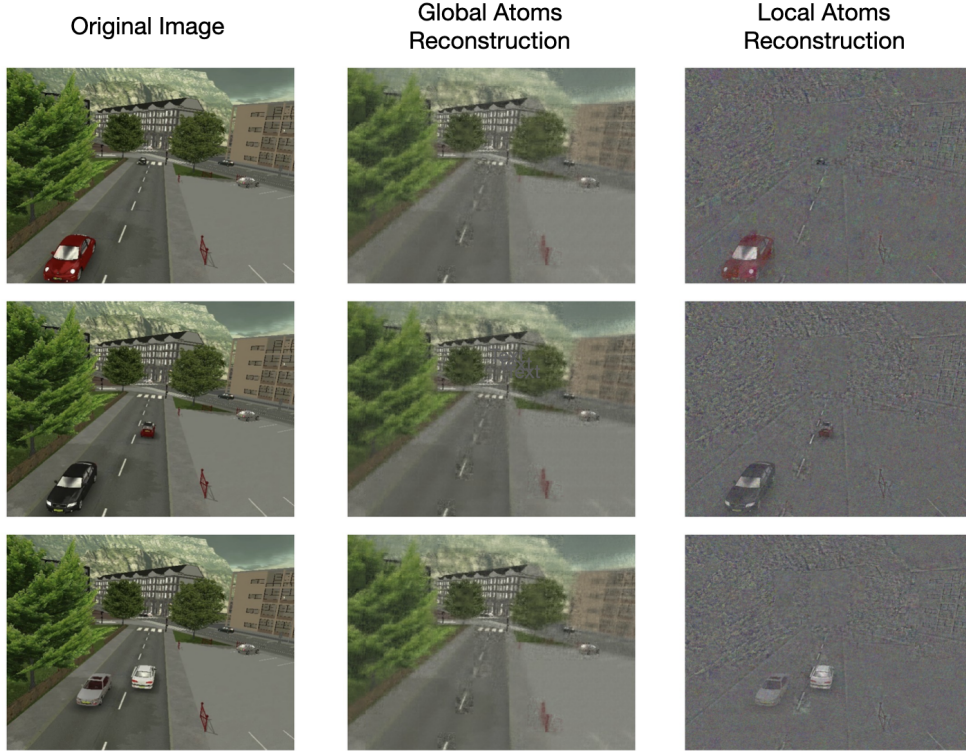


Figure 3.4: PerMA effectively separates the background from moving objects in video frames. Here we reconstruct the surveillance video frames using global and local dictionaries learned by PerMA. We reconstruct the frames using only 50 atoms from the combined dictionaries.

several local dictionaries for this dataset. Figure 3.4 shows the reconstructed background and moving objects via the recovered global and local dictionaries. Our results clearly show the ability of our proposed framework to separate global and local features. We note that moving object detection in video frames has been extensively studied in the literature and typically solved very accurately via different representation learning methods (such as robust PCA and neural network modeling); see [133] for a recent survey. Here, we use this case study as a proof of concept to illustrate the versatility of PerMA in handling heterogeneity, even in settings where the data is not physically distributed among clients.

3.8 Further Details on the Experiments

In this section, we provide further details on the numerical experiments reported in Section 3.7.

3.8.1 Details of Section 3.7.1

In Section 3.7.1, we generate the synthetic datasets according to Model 3.1 with $N = 10$, $d = 6$, $n_i = 6$, and $p_i = 200$ for each $1 \leq i \leq 10$. Each $\mathbf{D}_i^*$ is an orthogonal matrix with the first $n^g = 3$ columns shared with every other client and the last $n_i^l = 3$ columns unique to themselves. Each $\mathbf{X}_i^*$ is first generated from a Gaussian-Bernoulli distribution where each entry is non-zero with a probability 0.2. Then, $\mathbf{X}_i^*$ is further truncated, where all the entries $(\mathbf{X}_i^*)_{(j,k)}$ with $|(\mathbf{X}_i^*)_{(j,k)}| < 0.3$ are replaced by $(\mathbf{X}_i^*)_{(j,k)} = 0.3 \times \text{sign}((\mathbf{X}_i^*)_{(j,k)})$.

We use the orthogonal DL algorithm (Algorithm 3) introduced in the chapter 2 as the local DL algorithm for each client. This algorithm is simple to implement and comes equipped with a strong convergence guarantee (see Theorem 1). Here $\text{HT}_\zeta(\cdot)$ denotes the hard-thresholding operator at level ζ , which is defined as:

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

Specifically, we use $\zeta = 0.15$ for the experiments in Section 3.7.1.

For a fair comparison, we initialize both strategies using the same $\{\mathbf{D}_i^{(0)}\}_{i=1}^N$, which is obtained by iteratively calling Algorithm 3 with a random initial dictionary and shrinking thresholds ζ . For a detailed discussion on such an initialization scheme we refer the reader to [75].

3.8.2 Details of Section 3.7.2

In section 3.7.2, we aim to learn a dictionary with imbalanced data collected from MNIST dataset [67]. Specifically, we consider $N = 10$ clients, each with 500 handwritten images. Each image is comprised of 28×28 pixels. Instead of randomly assigning images, we construct dataset i such that it contains 450 images of digit i and 50 images of other digits. Here client 10 corresponds to digit 0. After vectorizing each image into a 784×1 one-dimension signal, our imbalanced dataset contains 10 matrices $\mathbf{Y}_i \in \mathbb{R}^{784 \times 500}$, $i = 1, \dots, 10$.

We first use Algorithm 3 to learn an orthogonal dictionary for each client, using their own imbalanced dataset. For client i , given the output of Algorithm 3 after T iterations $\mathbf{D}_i^{(T)}$, we reconstruct a new signal $\mathbf{y}$ using the top k atoms according to the following steps: first, we solve a *sparse coding* problem to find the sparse code $\mathbf{x}$ such that $\mathbf{y} \approx \mathbf{D}_i^{(T)} \mathbf{x}$. This can be achieved by the hard thresholding step in Algorithm 3. Second, we find the top k entries in $\mathbf{x}$ that have the largest magnitude: $\mathbf{x}_{(\alpha_1,1)}, \mathbf{x}_{(\alpha_2,1)}, \dots, \mathbf{x}_{(\alpha_k,1)}$. Finally, we calculate the

reconstructed signal $\tilde{\mathbf{y}}$ as

$$\tilde{\mathbf{y}} = \sum_{j=1}^k \mathbf{x}_{(\alpha_h, 1)} \left(\mathbf{D}_i^{(T)} \right)_{(:, \alpha_h)}.$$

The second row of Figure 3.3 is generated by the above procedure with $k = 5$ using the dictionary learned by Client 1. The third row of Figure 3.3 corresponds to the reconstructed images using the output of PerMA.

3.8.3 Details of Section 3.7.3

Our considered dataset in section 3.7.3 contains 62 frames, each of which is a $480 \times 640 \times 3$ RGB image. We consider each frame as one client ($N = 62$). After dividing each frame into 40×40 patches, we obtain each data matrix $\mathbf{Y}_i \in \mathbb{R}^{576 \times 1600}$. Then we apply PerMA to $\{\mathbf{Y}_i\}_{i=1}^{62}$ with $n_i = 576$ for all i and $n^g = 30$. Consider $\mathbf{D}_i^{(T)} = [\mathbf{D}_i^{g, (T)} \quad \mathbf{D}_i^{l, (T)}]$, which is the output of PerMA for client i . We reconstruct each $\mathbf{Y}_i$ using the procedure described in the previous section with $k = 50$. Specifically, we separate the contribution of $\mathbf{D}_i^{g, (T)}$ from $\mathbf{D}_i^{l, (T)}$. Consider the reconstructed matrix $\tilde{\mathbf{Y}}_i$ as

$$\tilde{\mathbf{Y}}_i = [\mathbf{D}_i^{g, (T)} \quad \mathbf{D}_i^{l, (T)}] \begin{bmatrix} \mathbf{X}_i^g \\ \mathbf{X}_i^l \end{bmatrix} = \underbrace{\mathbf{D}_i^{g, (T)} \mathbf{X}_i^g}_{\tilde{\mathbf{Y}}_i^g} + \underbrace{\mathbf{D}_i^{l, (T)} \mathbf{X}_i^l}_{\tilde{\mathbf{Y}}_i^l}$$

The second column and the third column of Figure 3.4 correspond to reconstructed results of $\tilde{\mathbf{Y}}_i^g$ and $\tilde{\mathbf{Y}}_i^l$ respectively. We can see clear separation of the background (which is shared among all frames) from the moving objects (which is unique to each frame).

One notable difference between this experiment and the previous one is in the choice of the DL algorithm $\mathcal{A}_i$. To provide more flexibility, we relax the orthogonality condition for the dictionary. Therefore, we use the alternating minimization algorithm introduced in [7] for each client (see Algorithm 11). The main difference between this algorithm and Algorithm 3 is that the polar decomposition step in Algorithm 3 is replaced by a single iteration of the gradient descent applied to the loss function $\mathcal{L}(\mathbf{D}, \mathbf{X}) = \|\mathbf{DX} - \mathbf{Y}\|_F^2$.

Algorithm 11 Alternating minimization for general dictionary learning ([7])

- 1: **Input:** $\mathbf{Y}_i, \mathbf{D}_i^{(t)}$
 - 2: Set $\mathbf{X}_i^{(t)} = \text{HT}_\zeta \left(\mathbf{D}_i^{(t)\top} \mathbf{Y}_i \right)$
 - 3: Set $\mathbf{D}_i^{(t+1)} = \mathbf{D}_i^{(t)} - 2\eta \left(\mathbf{D}_i^{(t)} \mathbf{X}_i^{(t)} - \mathbf{Y}_i \right) \mathbf{X}_i^{(t)\top}$
 - 4: **return** $\mathbf{D}_i^{(t+1)}$
-

Even with the computational saving brought up by Algorithm 11, the runtime significantly slows down for **PerMA** due to large N , d , and p . Here we report a practical trick to speed up **PerMA**, which is a local refinement procedure (Algorithm 12) added immediately before **local_update** (Step 10 of Algorithm 8). At a high level, **local_dictionary_refinement** first finds the local residual data matrix $\mathbf{Y}_i^l$ by removing the contribution of the global dictionary. Then it iteratively refines the local dictionary with respect to $\mathbf{Y}_i^l$. We observed that **local_dictionary_refinement** significantly improves the local reconstruction quality. We leave its theoretical analysis as a possible direction for future work.

Algorithm 12 `local_dictionary_refinement`

```

1: Input:  $\mathbf{D}_i^{(t)} = \begin{bmatrix} \mathbf{D}^{g,(t)} & \mathbf{D}_i^{l,(t)} \end{bmatrix}, \mathbf{Y}_i$ 
2: Find  $\begin{bmatrix} \mathbf{X}_i^g \\ \mathbf{X}_i^l \end{bmatrix}$  such that  $\mathbf{Y}_i \approx \begin{bmatrix} \mathbf{D}^{g,(t)} & \mathbf{D}_i^{l,(t)} \end{bmatrix} \begin{bmatrix} \mathbf{X}_i^g \\ \mathbf{X}_i^l \end{bmatrix}$ 
// Solving a sparse coding problem
3: Set  $\mathbf{Y}_i^l = \mathbf{Y}_i - \mathbf{D}^{g,(t)} \mathbf{X}_i^g$ 
4: Set  $\mathbf{D}_i^{\text{refine},(0)} = \mathbf{D}_i^{l,(t)}$ .
5: for  $\tau = 0, 1, \dots, T^{\text{refine}} - 1$  do
6:   Set  $\mathbf{D}_i^{\text{refine},(\tau+1)} = \mathcal{A}_i \left( \mathbf{Y}_i^l, \mathbf{D}_i^{\text{refine},(\tau)} \right)$ 
// Improving local dictionary
7: end for
8: return  $\mathbf{D}_i^{\text{refine},(T^{\text{refine}})}$  as refined  $\mathbf{D}_i^{l,(t)}$ 

```

3.8.4 Additional Experiments

In this section, we present the results of some additional experiments to better showcase the efficiency of **PerMA** compared with the existing methods and its potential to adapt to a parameter-free algorithm.

3.8.4.1 Comparison with Existing Methods

To the best of our knowledge, the problem of personalized dictionary learning (PerDL) has not been previously explored and formally defined. While existing methods in federated learning bear some resemblance to PerDL, they lack provable guarantees in recovering the global and local dictionaries. To clarify these distinctions, we present a detailed comparison between our work and the most closely related papers by [52] and [44]. We compare our method with these methods under the same setting as in Section 3.7.1. The results can be seen in Figure 3.5, which shows that **PerMA** consistently outperforms methods proposed by [44] and [52].

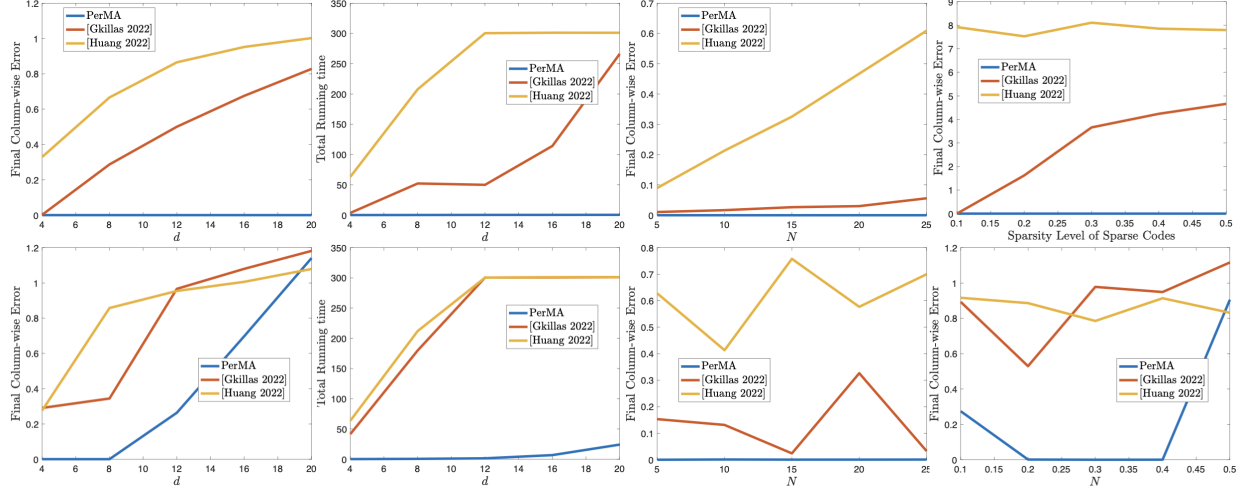


Figure 3.5: Comparisons of different methods on synthetic datasets. In the first row, clients are provided with heterogeneous datasets with similar sizes; in the second row, we consider the special cases in which one of the clients has an insufficient sample size and evaluate the performance of the dictionary learned by that specific client. The first column corresponds to final errors with varying d ; the second column corresponds to total running times with varying d ; the third column corresponds to final errors with varying N ; and the forth column corresponds to final errors with varying sparsity level. All the results are averaged over 3 independent trials.

Next, in the context of Section 3.7.2, we compare the quality of the reconstructed images using dictionaries learned from different methods under three metrics: MSE, PSNR and SSIM. A smaller MSE, a larger PSNR, and an SSIM closer to 1 indicate better image reconstruction quality. In Table 3.1, k denotes the number of atoms used to reconstruct the image. As can be seen in the table, **PerMA** achieves the best result in all sections except for the training time.

Finally, in this paper we use **PerMA** on the surveillance video datasets, with the goal of separating common elements shared by all clients (the background) and unique elements (different cars). Such a task cannot be accomplished by [44] and [52] due to their lack of personalization. On the other hand, a recently proposed method based on personalized PCA (PerPCA) has been shown to be effective in separating common and unique elements [108]. As a result, we run PerPCA on the same dataset to compare its performance with our method. According to Figure 3.6, PerDL outperforms PerPCA by achieving better separation and higher resolution.

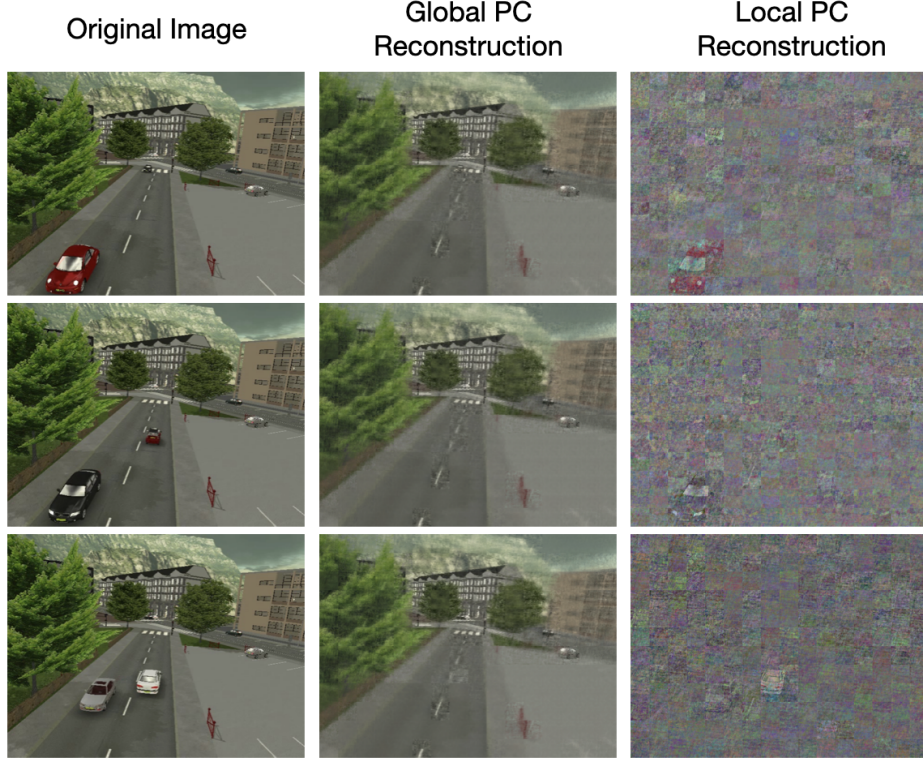


Figure 3.6: Objects and background separation by PerPCA.

3.8.4.2 Auto-tuning of n^g

One can easily see from Section 3.3 that n^g is an important hyper-parameter of our algorithm. A larger n^g means more global atoms are sent between central servers and clients, which leads to a stronger collaboration between them. In the synthetic experiment, we assume to know the value of n^g , while in real-life applications, one needs to fine-tune this parameter. An interesting observation we made is that the proposed PerMA algorithm can be augmented by a simple detection mechanism for identifying the correct choice of n^g . Specifically, during the Global Matching step, where we iteratively remove shortest paths, we can monitor the length of the obtained shortest path. By terminating the removal of paths (i.e., adding global atoms) when the length of the path experiences a significant increase beyond a predefined threshold, we can effectively identify the appropriate value of n^g without requiring prior knowledge. This detection mechanism alleviates the burden of fine-tuning n^g and allows for a more practical and robust implementation of the algorithm.

To validate the efficacy of this approach, we conducted a series of experiments, the results of which are presented in Figure 3.7. We use different $n^g = 4, 6, 8$ with $r = 10$ and monitor the lengths of paths. As evident from the outcomes, a clear and drastic increase in the length

of the $n^g + 1$ -th shortest path is observed, signifying the correct value of n^g .

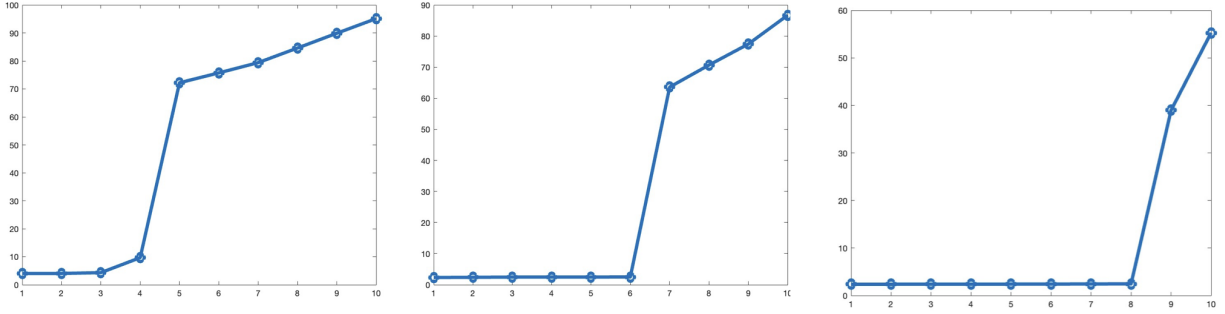


Figure 3.7: Increase of path length for different n^g . Here the x -axis denotes the number of iterations for Algorithm 9 and the x -axis denotes the distance of the $\mathcal{P}$ for each iteration.

Methods	MSE		PSNR		SSIM		Running Time (seconds)
	$k = 10$	$k = 20$	$k = 10$	$k = 20$	$k = 10$	$k = 20$	
PerMA	0.0319	0.0207	15.3795	17.3771	0.6286	0.7074	40.8
Local DL	0.0448	0.0330	14.1936	15.7203	0.5538	0.6162	4.9
Gkillas et al.	0.1069	0.1084	9.8930	9.8360	0.2736	0.2659	3600+
Huang et al.	0.1062	0.1062	9.9203	9.9212	0.2737	0.2628	3600+
PerPCA	0.0837	0.0734	11.0118	11.6426	0.3520	0.3959	59.3

Table 3.1: Comparisons of different methods on MNIST datasets. A smaller MSE, a larger PSNR, and an SSIM closer to 1 indicate better image reconstruction quality. Here k denotes the number of atoms used to reconstruct the image.

3.9 Concluding Remarks

The novel approach for personalized dictionary learning presents a versatile solution with immediate applications across various domains, such as video surveillance and object detection. While these applications offer valuable benefits, they also bring to the forefront ethical and societal concerns, particularly concerning privacy, bias, and potential misuse. In the context of video surveillance, deploying object detection algorithms may inadvertently capture private information, leading to concerns about violating individuals' right to privacy. However, it is important to emphasize that the proposed algorithm is specifically designed to focus on separating unique and common features within the data, without delving into the realm of personal information. Thus, it adheres to ethical principles by ensuring that private

data is not processed or used without explicit consent. Bias is another critical aspect that necessitates careful consideration in the deployment of object detection algorithms. Biases can manifest in various forms, such as underrepresentation or misclassification of certain groups, leading to discriminatory outcomes. Our approach acknowledges the importance of mitigating biases by solely focusing on the distinction between common and unique features, rather than introducing any inherent bias into the learning process. Furthermore, the potential misuse of object detection algorithms in unauthorized surveillance or invasive tracking scenarios raises valid concerns. As responsible researchers, we are cognizant of such risks and stress that our proposed algorithm is meant to be deployed in a controlled and legitimate manner, adhering to appropriate regulations and ethical guidelines.

Even though our meta-algorithm PerMA enjoys strong theoretical guarantees and practical performance, there are still several avenues for improvement. For instance, the theoretical success of PerMA, especially the Global Matching step, relies on an individual initial error of $O(1/N)$. In other words, the initial error should decrease as the number of clients grows. As a future work, we plan to relax such dependency via a more delicate analysis. We also note that imposing an upper bound on the initial error is not unique to our setting, as virtually all existing algorithms for classical (non-personalized) dictionary learning require certain conditions on the initial error. On the other hand, once the assumption on the initial error is satisfied, our meta-algorithm achieves a final error with the same dependency on d (the dimensionality of the data) and n (the number of samples) as the state-of-the-art algorithms for classical dictionary learning ([2], [7]). Remarkably, this implies that personalization is achieved without incurring any additional cost on d or n , making PerMA highly efficient and competitive in its performance.

CHAPTER 4

Sparse Representation Learning and Explainable AI

This chapter is based on the paper *Enhancing Performance of Explainable AI Models with Constrained Concept Refinement* [73] by Geyu Liang, Senne Michielssen, Salar Fattahi.

4.1 Introduction

ML algorithms are often caught in a dilemma between interpretability and performance. Models such as linear regression [50] and decision trees [93] provide straightforward interpretability through parameter weights and rule-based predictions. However, they frequently underperform on complex tasks. On the other hand, high-performing models, such as deep neural networks [66] and large language models [127], are notoriously opaque, given their large parameter spaces and intricate architectures.

Numerous methods have been proposed to extract interpretability from complex models [9, 110, 138, 109, 104, 112, 60, 114]. However, these approaches typically adopt post-hoc strategies, utilizing sensitivity analysis to identify the key parameters that influence predictions. Consequently, these methods lack guarantees of providing explanations for random prediction or ensuring their trustworthiness [1, 99, 58, 42, 111].

An alternative solution is to develop models that are *interpretable by design*. These models intrinsically integrate transparency into their architecture, providing explanations directly tied to their predictions. Concept Bottleneck Models (CBMs, [59]) exemplify this approach by mapping input data to an intermediate representation of human-defined concepts, which is then used for prediction. While concept-based models are promising, they require datasets annotated with concept scores, limiting their applicability. Extensions of CBMs aim to improve the approach by utilizing pretrained encoders like CLIP [136, 88]. A recent line of research [27, 25, 26] has introduced a decision-tree-based architecture with enhanced

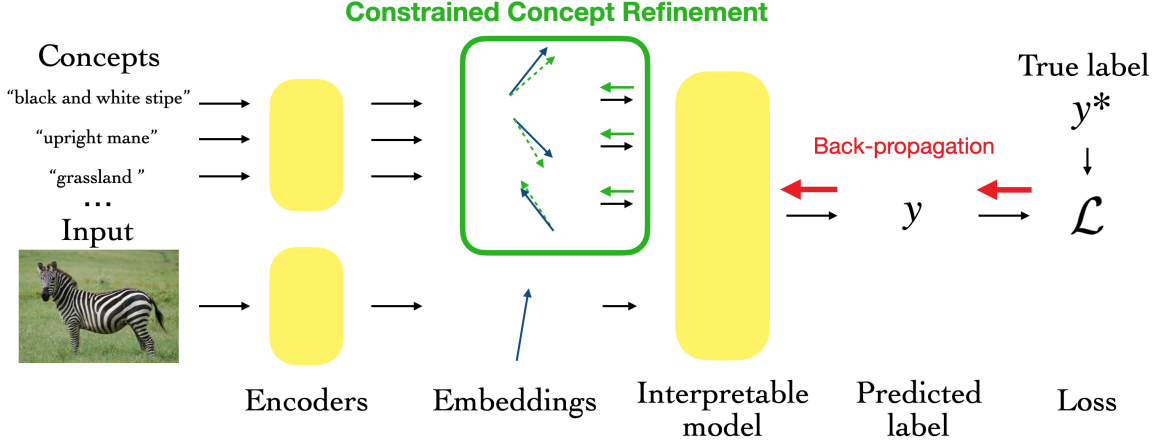


Figure 4.1: The red arrows represent the backpropagation training process for classic explainable AI models. The algorithm introduced in this chapter extends the training process to refine concept embeddings with constraints on their deviation from initial embeddings, represented by green arrows and box.

explainability. The predictions of the model are generated through a sequence of queries, each closely associated with human-specified concepts.

A high level design paradigm of explainable AI is illustrated in Figure 4.1. Within these models, concept embeddings play a pivotal role: they not only influence prediction generation but also serve as the source of interpretability. However, recent studies have raised several questions against its reliability. These challenges are centered around its ambiguity [84, 132, 57, 83], fragility [38], and low accuracy [137, 28]. For embeddings generated with pretrained models, their encoders are reported to face issues such as domain adaptation [105, 45, 36, 131], biases in pretraining data [49], and sensitivity to encoding errors [97].

Efforts to address these challenges include introducing unsupervised modules to enhance concept expressiveness [100], training residual learning modules in parallel with concept encoders [136], learning dual embeddings per concept [137], and adding specialized data encoders to address domain shifts [28]. However, these approaches all involve introducing additional black-box learning modules, which undermines the purpose of explainable AI. Moreover, theoretical guarantees for such approaches, which is of particular interest for safety-critical domains, such as medical imaging [14], autonomous vehicles [15] and industrial design [118], remain largely unexplored.

To address these challenges, this chapter proposes a novel framework termed *Constrained Concept Refinement (CCR)*. Unlike existing methods that introduce additional learning modules, our approach directly optimizes concept embeddings within a constrained parameter space. By restricting embeddings to a small neighborhood around their initialization, our

framework offers a tunable trade-off between performance and interpretability, and in certain settings, it can simultaneously improve both. Specifically, we present the following contributions:

- **Theoretical necessity.** We rigorously demonstrate the necessity of refining concept embeddings by establishing a non-vanishing worst-case lower bound on model performance when such refinements are not applied to the concepts (Theorem 11). This highlights and motivates the idea of concept embedding refinement, which is at the crux of our proposed method.
- **Accuracy and interpretability guarantees.** We show that CCR overcomes the aforementioned challenge by eliminating performance degradation through the appropriate refinement of concepts (Theorem 12). Furthermore, to quantify the performance of CCR, we consider a generative model as a test-bed, where both “interpretability” and “accuracy” admit crisp mathematical definitions. Under this model, we demonstrate that CCR achieves zero training loss while progressively enhancing interpretability (Theorem 13).
- **Application in interpretable image classification.** We demonstrate the practical efficacy of CCR on multiple image classification benchmarks including CIFAR 10/100 [64], ImageNet [31], CUB200 [129] and Places365 [143]. In all benchmarks except CUB-200, CCR outperforms two recently developed explainable methods for visual classification tasks—CLIP-IP-OMP [26] and label-free CBM (lf-CBM) [88]—in terms of prediction accuracy while preserving interpretability, achieving this with a tenfold reduction in runtime.

4.2 Related Literature

In this section, we provide a more comprehensive literature review.

Explainable AI. In the main body of the paper, we provided an overview of related methods in explainable AI. Here, we focus on two approaches that are closely related to and comparable with our proposed method. The first is the CLIP-IP-OMP method introduced by [26], which leverages CLIP embeddings to greedily solve sparse coding problems for each input, subsequently passing the results through a linear layer to generate final predictions. Our method builds upon the same theoretical framework established in [26]. Notably, the baseline model (without concept refinement) considered in Section 4.4 and Section 4.7 serves

as a differentiable surrogate for CLIP-IP-OMP. The second approach is the label-free CBM (lf-CBM) proposed by [88], which employs the inner product between CLIP embeddings to achieve state-of-the-art results in interpretable models. Specifically, lf-CBM was the first CBM model scaled to ImageNet. A detailed comparison of the performance of our method against these two approaches is provided in Section 4.7.

Sparse coding and dictionary learning. Sparse Coding [89] refers to the process of representing a given signal (or vector) as a sparse linear combination of fixed signals, collectively referred to as a *dictionary*. Dictionary Learning [4] extends Sparse Coding to a bi-variable setting, allowing the dictionary itself to be optimized. The objective is to identify a dictionary capable of generating effective sparse codes for a specific signal distribution. [26] establishes a novel connection between sparse coding and explainable AI, treating concept embeddings as the dictionary. Our proposed method directly optimizes concept embeddings, which draws parallels to dictionary learning, although the two approaches address fundamentally different objectives. Among works on dictionary learning, [79] is particularly relevant, as it optimizes dictionaries for downstream tasks. However, our method distinguishes itself through the constrained concept requirement. Interestingly, [79] aligns with our framework as a differentiable learning module, and integrating this approach with our method presents an intriguing avenue for future work. Another promising direction for future research is leveraging the *personalized* dictionary learning framework proposed by [74] to extend our approach to Explainable AI in the context of heterogeneous datasets.

Theoretical guarantees. To the best of our knowledge, the result in Chapter 4 is the first to establish a convergence guarantee in the setting where dictionary atoms are updated for downstream tasks. The most closely related work is [26], which introduced IP-OMP. However, the connection between IP-OMP and column-orthogonal matching pursuit lacks rigor due to the presence of normalization terms (see Theorem 10), making it difficult to prove that IP-OMP minimizes the loss $\mathcal{L}$ or $\mathcal{L}_m$ using the optimality conditions of OMP [124]. Another related approach is task-driven dictionary learning [79], which establishes the differentiability of the objective function but does not provide a convergence guarantee.

Constrained optimization. Our approach enforces interpretability by incorporating a constraint that restricts the deviation of the parameters, thereby formulating our objective as a constrained optimization problem [21]. Although various techniques have been introduced to address the general form of such problems [20, 17, 16], we employ a method analogous to the classical projected gradient descent algorithm [21] and conduct a refined trajectory

analysis to establish its convergence.

4.3 Revisiting IP-OMP

We consider a classical *interpretable-by-design* setting in the literature [27, 26], namely the task of predicting a random variable by sequentially selecting a list of related random variables, termed *queries*, and checking their values. The problem is defined as follows:

Definition 3 (Single Variable Prediction by Query Selection). *Given an input feature $\mathbf{x} \in \mathbb{R}^d$, a set of queries $\{q_i\}_{i=1}^n \subset \mathbb{R}$, and their associated query features (also referred to as concept embeddings) $\{\mathbf{d}_i\}_{i=1}^n \subset \mathbb{R}^d$, the objective is to predict the target random variable $y \in \mathbb{R}$ by selecting at most k queries from $\{q_i\}_{i=1}^n$, leveraging the query features $\{\mathbf{d}_i\}_{i=1}^n$ to guide the selection process. Specifically, the query set $\{q_i\}_{i=1}^n$ consists of random variables that are correlated with y , whose observed values can contribute to predicting y .*

- **Input Feature $\mathbf{x}$:** Denotes the input feature for this learning task, where $\mathbf{x} \in \mathbb{R}^d$.
- **Target Random Variable y :** Represents the random variable that we aim to predict.
- **Query Set $\{q_i\}_{i=1}^n$:** A collection of random variables correlated with y , whose observed values can be examined.
- **Query Features $\{\mathbf{d}_i\}_{i=1}^n$:** For each query in the query set, a feature vector $\mathbf{d}_i \in \mathbb{R}^d$ is available, encoding information about q_i . The feature vector $\mathbf{d}_i$ may be utilized to determine whether to inspect q_i .
- **Budget k :** The maximum number of queries that can be selected.

Definition 3 encompasses a broad spectrum of ML problems with applications in animal identification [65], medical diagnosis [46, 29], visual question-answering [134], image retrieval [23], and so on.

One illustrative example is the game played among friends, where the target random variable refers to the answer to the question “what am I thinking now?”. In this context, the queries $\{q_i\}_{i=1}^n$ are binary random variables representing “yes/no” answers to specific questions like “Is it an animal?”. Here, the query feature is a vector embedding of each question, encapsulating the information relevant to that question. Additionally, the participants are limited to asking at most k questions.

In [40], the authors introduced a classic greedy-style algorithm termed *Information Pursuit* (IP) to address the task outlined in Definition 3. Specifically, IP iteratively selects the

most informative query based on the observed values of previously selected queries. We provide a formal definition of IP for subsequent discussion:

Definition 4 (Information Pursuit). *At every iteration $t = 1, \dots, k$, IP selects $\pi(t)$ -th query according to:*

$$\pi(t) = \operatorname{argmax}_{1 \leq i \leq n} \left\{ I(q_i, y \mid q_{\pi(1)} = r_{\pi(1)}, \right. \\ \left. q_{\pi(2)} = r_{\pi(2)}, \dots, q_{\pi(t-1)} = r_{\pi(t-1)}) \right\}.$$

Here $I(\cdot, \cdot)$ denotes mutual information, $\pi(\cdot) : [k] \rightarrow [n]$ is an injective map representing the selection of queries, and r_i denotes the observed value for q_i . The final output of IP is defined as the maximum likelihood estimator given all observed queries:

$$y_{\text{IP}} = \operatorname{argmax}_{\tilde{y}} \left\{ \mathbb{P}(y = \tilde{y} \mid q_{\pi(1)} = r_{\pi(1)}, \right. \\ \left. q_{\pi(2)} = r_{\pi(2)}, \dots, q_{\pi(k)} = r_{\pi(k)}) \right\}. \quad (4.1)$$

In [25], IP is demonstrated to achieve highly interpretable predictions with competitive accuracies. However, a significant limitation of IP is its substantial computational expense, as mutual information depends not only on the set $\{\mathbf{d}_i\}_{i=1}^n$ but also on previous observations, leading to an exponentially large input space. [26] addresses this complexity by imposing additional assumptions on the underlying generative model. The first quantifies the connection between y and q_i , while the second defines their connections with $\mathbf{d}_i$.

Assumption 4 (Generative model for IP-OMP). *There exists an unknown random vector $\mathbf{z} \in \mathbb{R}^d$ drawn from a standard Normal distribution that satisfies $y = \langle \mathbf{x}, \mathbf{z} \rangle$. Moreover, the queries satisfy $q_i = \langle \mathbf{v}_i, \mathbf{z} \rangle$, where $\mathbf{v}_i$ is a latent feature vector associated with query i .*

Under the above assumption, the target random variable and the queries are correlated through the unknown random variable $\mathbf{z}$. In this setting, a natural way to define the query features is by setting $\mathbf{d}_i = \mathbf{v}_i$. This entails an exact prior knowledge of the latent feature vectors.

Assumption 5 (Prior knowledge of latent feature vectors). *The latent feature vectors $\{\mathbf{v}_i\}_{i=1}^n$ are precisely observed.*

Under Assumption 4 and 5, [26] establish a connection between IP and Orthogonal Matching Pursuit (OMP) [91], a seminal algorithm in the field of sparse coding.

Theorem 10 (IP-OMP [26]). *Under Assumption 4 and 5 and upon setting $\mathbf{d}_i = \mathbf{v}_i$ for $i = 1, \dots, n$, the selection rule $\pi(\cdot)$ defined in Definition 4 admits a closed-form expression given by:*

$$\pi(t) = \operatorname{argmax}_{1 \leq i \leq n} \frac{|\langle \Pi_{\mathbf{D}^{(t-1)}}^\perp \mathbf{d}_i, \Pi_{\mathbf{D}^{(t-1)}}^\perp \mathbf{x} \rangle|}{\|\Pi_{\mathbf{D}^{(t-1)}}^\perp \mathbf{d}_i\|_2 \|\Pi_{\mathbf{D}^{(t-1)}}^\perp \mathbf{x}\|_2},$$

where $\mathbf{D}^{(t-1)} = [\mathbf{d}_{\pi(1)} \quad \mathbf{d}_{\pi(2)} \quad \dots \quad \mathbf{d}_{\pi(t-1)}]$. This selection criterion, referred to as IP-OMP, differs from OMP solely by the inclusion of the normalization term $\|\Pi_{\mathbf{D}^{(t-1)}}^\perp \mathbf{d}_i\|_2 \|\Pi_{\mathbf{D}^{(t-1)}}^\perp \mathbf{x}\|_2$.

The main theoretical contribution of Theorem 10 lies in its ability to transform the computation of mutual information into that of a linear projection, thereby significantly reducing computational complexity. Although the equivalence is derived under simplifying assumptions, IP-OMP demonstrates remarkable generalization capabilities across various benchmark experiments, outperforming models based on the classical IP algorithm [26].

4.3.1 A Major Hurdle: Inaccurate Query Features

As discussed in the previous section, Theorem 10, and as a result, the success of IP-OMP is contingent upon correctly choosing the query features $\{\mathbf{d}_i\}_{i=1}^n$.

For learning algorithms, the effective utilization of $\{\mathbf{d}_i\}_{i=1}^n$ is crucial for the accurate prediction of y . On the other hand, humans derive comprehension from $\{\mathbf{d}_i\}_{i=1}^n$ as these elements are intended to embed concepts that are interpretable by humans. In practical applications, $\{\mathbf{d}_i\}_{i=1}^n$ are either embedded and learned from predefined datasets where concepts are explicitly labeled [59] or generated by pretrained multimodal models, with CLIP being the most popular example [88].

However, these learned embeddings are often misaligned or inaccurate due to the inherent ambiguity and noise in the training of the models used to generate. For example, unlike carefully curated datasets, CLIP relies on large-scale, noisy image-text pairs scraped from the internet. These pairs often include mislabeled data, vague descriptions, or culturally biased associations, resulting in inconsistencies in representation, as recently reported by [33].

This suggests that Assumption 5 may be overly optimistic. In this section, we will illustrate how the violation of Assumption 5—specifically, deviations of the available query features from the latent feature vectors—results in a proportional degradation in the performance of IP-OMP. For the remainder of this chapter, we denote the observed query feature set compactly by the matrix $\mathbf{D} = [\mathbf{d}_1 \quad \mathbf{d}_2 \quad \dots \quad \mathbf{d}_n] \in \mathbb{R}^{d \times n}$. This matrix $\mathbf{D}$ is also referred to as a *feature query matrix*. We also refer to $\mathbf{D}^* = [\mathbf{v}_1 \quad \mathbf{v}_2 \quad \dots \quad \mathbf{v}_n] \in \mathbb{R}^{d \times n}$ as the ground

truth *latent feature matrix*. Assumption 5 can now be interpreted as assuming $\mathbf{D} = \mathbf{D}^*$, stating that we are using the ground truth latent feature vectors as the feature query matrix. This matrix can then be utilized to determine which queries should be observed.

Let $f_{\mathbf{D}}(\mathbf{x}, \{r_{\pi(i)}\}_{i=1}^k)$ be the maximum likelihood estimator of y , defined as Equation (4.1), when the selection rule $\pi(\cdot)$ is obtained by executing IP-OMP using $\mathbf{D}$ as the query feature set in Theorem 10. For brevity, when the context is clear, we refer to this maximum likelihood estimator simply as $f_{\mathbf{D}}$. Moreover, define $\mathcal{L}(y_{\text{pred}})$ as the squared population loss of any estimator y_{pred} :

$$\mathcal{L}(y_{\text{pred}}) = \mathbb{E}_{\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{d \times d})} [(y - y_{\text{pred}})^2].$$

where $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{d \times d})$ is the unknown random vector used in the generative model described in Assumption 4.

To showcase the effect of deviation in the query features, we consider a scenario where the input $\mathbf{x}$ can be written as a linear combination of k latent feature vectors (corresponding to k columns of $\mathbf{D}^*$). This assumption is prevalent in the sparse coding literature [7, 2, 75] and ensures that using $\mathbf{D}^*$ as the query feature set guarantees good prediction performance. However, even in this ideal scenario, we demonstrate that even a small column-wise deviation from $\mathbf{D}^*$ can lead to performance degradation of IP-OMP.

Theorem 11. *Suppose that Assumption 4 holds. Furthermore, suppose that $\mathbf{D}^*$ is column-orthogonal and $\mathbf{x} = \mathbf{D}^* \beta$ for some k -sparse vector β with $\|\beta\|_0 = k$. Additionally, assume that the non-zero entries in β have absolute values bounded below by γ and above by Γ . Then, for any $\epsilon \in \left(0, \frac{1}{\sqrt{1+16\Gamma^2/\gamma^2}}\right)$, there exists another orthonormal matrix $\tilde{\mathbf{D}} \in \mathbb{R}^{d \times n}$ such that $\|\tilde{\mathbf{D}} - \mathbf{D}^*\|_{1,2} \leq \epsilon$ and*

$$\mathcal{L}(f_{\tilde{\mathbf{D}}}) - \mathcal{L}(f_{\mathbf{D}^*}) \geq \frac{81(k-1)\epsilon^2\gamma^2}{200}. \quad (4.2)$$

We present the complete proof of Theorem 11 in Section 4.5.1, which proceeds as follows: (1) we derive the closed form solution for $\mathcal{L}(f_{\mathbf{D}})$ using the column-orthogonality of $\mathbf{D}$ and the distribution of $\mathbf{z}$; (2) we then construct the example that satisfies Equation (4.2) by rotating columns of $\mathbf{D}^*$ alongside the subspace spanned by itself.

When Γ/γ is bounded by a constant, we have $\|\mathbf{x}\|_2^2 = \|\beta\|_2^2 = \Theta(k\gamma^2)$ which simplifies Equation (4.2) to $\mathcal{L}(f_{\tilde{\mathbf{D}}}) - \mathcal{L}(f_{\mathbf{D}^*}) = \Omega(\epsilon^2\|\mathbf{x}\|_2^2)$. This indicates that the perturbation ϵ is fully captured by the resulting gap in the squared loss, scaled by a factor of $\|\mathbf{x}\|_2^2$.

4.4 Constrained Concept Refinement

Building upon our discussion in the preceding section, it is pertinent to consider the following question:

Since an inaccurate $\mathbf{D}$ can adversely affect the performance of IP-OMP, can $\mathbf{D}$ be optimized to mitigate this effect while preserving—or even enhancing—interpretability?

There are two critical challenges to address before answering this question. First, modifying $\mathbf{D}$ may diminish the encoded information that is interpretable by humans, thereby compromising the *interpretable-by-design* principle of information pursuit. Second, treating the query features as variables introduces significant complexity in deriving the optimal decision rule, as the closed-form expression of IP-OMP presented in Theorem 10 is no longer valid.

To overcome these challenges, we propose obtaining a *correction* $\Delta\mathbf{D}$ that minimizes $\mathcal{L}(\tilde{f}_{\mathbf{D}+\Delta\mathbf{D}})$, where $\tilde{f}_{\mathbf{D}}$ serves as a differentiable surrogate of the original estimator produced by IP. To preserve interpretability, we constrain the corrected query feature matrix $\mathbf{D}+\Delta\mathbf{D}$ to remain within a small neighborhood around $\mathbf{D}$, where $\mathbf{D}$ represents the potentially inaccurate initial query feature matrix (e.g., the one obtained from CLIP). This leads to the following constrained optimization problem:

$$\min_{\|\Delta\mathbf{D}\|_{1,2} \leq \rho} \mathcal{L}(\tilde{f}_{\mathbf{D}+\Delta\mathbf{D}}) \quad (4.3)$$

Indeed, the choice of the correction radius ρ is crucial as it controls the trade-off between prediction accuracy and interpretability. Setting $\rho = 0$ recovers the IP-OMP, and therefore, may suffer from the aforementioned performance degradation. Conversely, choosing a large value for ρ can mitigate the performance degradation of IP-OMP, but at the cost of compromising the interpretability.

Our meta-algorithm for Problem (4.3), called *constrained concept refinement* (CCR), is presented in Algorithm 13.

Algorithm 13 Constrained Concept Refinement

- 1: **Input:** Initial query feature matrix $\mathbf{D}$, correction radius ρ , training dataset $\mathcal{D}$.
 - 2: Initialize $\Delta\mathbf{D}^{(0)} = \mathbf{0}_{d \times n}$
 - 3: **while** $t = 1, 2, \dots, T$ **do**
 - 4: **Forward propagation:** calculate $\mathcal{L}(\tilde{f}_{\mathbf{D} + \Delta\mathbf{D}^{(t-1)}})$.
 - 5: **Backward propagation:** update $\Delta\mathbf{D}^{(t)}$ using the gradient $\partial\mathcal{L}/\partial\Delta\mathbf{D}^{(t-1)}$.
 - 6: Perform projection to ensure $\|\Delta\mathbf{D}^{(t)}\|_{1,2} \leq \rho$.
 - 7: **end while**
 - 8: Return $\tilde{f}_{\mathbf{D} + \Delta\mathbf{D}^{(T)}}$
-

Below, we provide a description of its various steps.

Choice of the surrogate estimator. To motivate the choice of the surrogate estimator $\tilde{f}_{\mathbf{D}}$, let us revisit the generative model in Assumption 4, and additionally assume that the ground truth latent feature matrix $\mathbf{D}^*$ is column-orthogonal.

By assuming $\mathbf{D}^*$ to be column-orthogonal, we effectively assume an ideal scenario in which queries are mutually independent. Specifically, it follows that $\langle \mathbf{d}_i^*, \mathbf{z} \rangle$ is independent of $\langle \mathbf{d}_j^*, \mathbf{z} \rangle$ when $\mathbf{d}_i^* \perp \mathbf{d}_j^*$ and $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{d \times d})$. The primary motivation for focusing on a column-orthogonal $\mathbf{D}^*$ is that, under this assumption, the estimator $f_{\mathbf{D}}$ obtained from IP-OMP becomes inherently differentiable with respect to $\mathbf{D}$:

Lemma 7. *Assuming that $\mathbf{D} = \mathbf{D}^*$ and $\mathbf{D}^*$ is column-orthogonal, we have*

$$f_{\mathbf{D}}(\mathbf{x}, \{r_{\pi(i)}\}_{i=1}^k) = \sum_{i \in S} \langle \mathbf{d}_i, \mathbf{x} \rangle r_i,$$

where $S = \operatorname{argmax}_{T \subseteq [n], |T| \leq k} \sum_{i \in T} |\langle \mathbf{d}_i, \mathbf{x} \rangle|$ corresponds to the indices of the top- k largest values of $\{|\langle \mathbf{d}_i, \mathbf{x} \rangle|\}_{i=1}^n$.

The proof of Lemma 7 is deferred to Section 4.5.4.1. Assuming a small initial error, i.e., $\|\mathbf{D}^* - \mathbf{D}\| \leq \rho$ for some small $\rho > 0$, the iterates $\mathbf{D}^{(t)} = \mathbf{D} + \Delta\mathbf{D}^{(t)}$ of Algorithm 13 stay approximately column-orthogonal. This consideration motivates the introduction of the following surrogate estimator $\tilde{f}$:

$$\begin{aligned} \tilde{f}_{\mathbf{D}}(\mathbf{x}, \{r_{\pi(i)}\}_{i=1}^k) &= \sum_{i \in S} \langle \mathbf{d}_i, \mathbf{x} \rangle r_i, \\ \text{where } S &= \operatorname{argmax}_{T \subseteq [n], |T| \leq k} \sum_{i \in T} |\langle \mathbf{d}_i, \mathbf{x} \rangle|. \end{aligned} \tag{4.4}$$

It is important to note that we do not require $\tilde{f}$ to be differentiable everywhere; rather, differentiability is required only almost surely along the trajectory of Algorithm 13, which we will rigorously demonstrate. Alternative differentiable surrogates for f include the task-driven dictionary learning method [79] and the unrolled dictionary learning method [82, 122], which can be used in place of Equation (4.4).

Backward propagation and projection For the backward propagation and projection steps, we propose to adopt gradient descent updates, followed by a projection step based on ℓ_2 -norm.

$$\Delta \mathbf{D}^{(t+1)} = \operatorname{argmin}_{\|\Delta \mathbf{D}\|_{1,2} \leq \rho} \left\| \Delta \mathbf{D} - \left(\Delta \mathbf{D}^{(t)} - \eta \frac{\partial \mathcal{L}(\tilde{f}_{\mathbf{D} + \Delta \mathbf{D}^{(t)}})}{\partial \Delta \mathbf{D}^{(t)}} \right) \right\|_2,$$

We note that the specific instantiation of the proposed meta-algorithm and its steps is inherently task-dependent. For example, while we have only discussed the squared loss, a more common choice in image classification tasks is the cross-entropy loss. Additionally, our choice of the surrogate estimator relies on the assumption that the ground truth latent feature matrix is column-orthogonal. When this assumption is not met, it becomes necessary to enforce it via *concept dispersion*. Both of these modifications are discussed extensively within the context of image classification in Section 4.7.

However, a more fundamental question has remained unanswered: how can “interpretability”—a concept inherently meaningful only to humans—be measured and quantified? One approach is to assess interpretability in an ad-hoc manner by running the candidate method and relying on human judgment to determine whether the predictions are interpretable for different individual samples. This has been the de facto approach adopted in nearly all previous works. An alternative way—which we seek in this work—is to study a simple test-bed for investigation, where “interpretability” admits a crisp mathematical formulation.

4.4.1 Accuracy and Interpretability Guarantees

Consider the following probabilistic generative model.

Assumption 6 (Probabilistic generative model). *Suppose that the target random variable y and queries are generated according to Assumption 4. Moreover, suppose that ground truth latent feature matrix $\mathbf{D}^*$ is column-orthogonal and the input $\mathbf{x}$ satisfies $\mathbf{x} = \mathbf{D}^* \beta$, where: (1) the support of β , denoted as S^* , is drawn uniformly from the set of k -element subsets of $[n]$; and (2) the non-zero elements of β are i.i.d. and satisfy $\mathbf{E}[\beta_i] = 0$, $\operatorname{Var}[\beta_i] = \sigma^2$, and $\gamma \leq |\beta_i| \leq \Gamma$.*

The generative model in Assumption 6 is akin to those explored in the sparse coding literature [7, 2, 98, 75]. Under this model, prediction error can be evaluated using the squared loss, while interpretability can be assessed by the proximity of the query feature matrix $\mathbf{D}^{(T)}$, produced by CCR, to the ground truth latent feature matrix $\mathbf{D}^*$.

Our next theorem shows that CCR effectively resolves the challenge faced by IP-OMP, as outlined in Theorem 11. We note that the result of Theorem 11 also holds for the generative model described in Assumption 6. Let $\mathbf{D}^{(t)} = \mathbf{D} + \Delta\mathbf{D}^{(t)}$ be the corrected query feature matrix generated by CCR at iteration t .

Theorem 12. *Suppose that Assumption 6 holds. Moreover, suppose that the initial query feature matrix $\mathbf{D}$ satisfies $\|\mathbf{D} - \mathbf{D}^*\|_{1,2} = \rho \leq \frac{\gamma}{8\sqrt{k}\Gamma}$ and the step-size η satisfies $0 < \eta < \frac{1}{2\|\mathbf{x}\|_2^2}$. Algorithm 13 with surrogate estimator (4.4) and update rule (4.4) satisfies:*

$$\mathcal{L}(\tilde{f}_{\mathbf{D}^{(t+1)}}) \leq (1 - 2\eta\|\mathbf{x}\|_2^2)^2 \mathcal{L}(\tilde{f}_{\mathbf{D}^{(t)}}).$$

While Theorem 12 addresses the performance limitations of IP-OMP, it does not guarantee improved interpretability. This outcome is unsurprising, since the reliance of the method on only *a single* input $\mathbf{x}$ ensures that only the columns of $\mathbf{D}$ with indices in S^* can be improved, essentially leaving the columns outside S^* unmodified. Moreover, if $\mathbf{D}^*$ is not full-rank, our analysis shows that $\mathbf{D}$ may deviate from $\mathbf{D}^*$ in directions column-orthogonal to the column space of $\mathbf{D}^*$; these deviations cannot be eliminated by Section 4.4. This observation is further supported by numerical experiments (see Figure 4.2).

To tackle these challenges, we assume that $\mathbf{D}^*$ is full-rank. Moreover, it is essential to use a sufficient number of i.i.d. input samples to ensure that each column of $\mathbf{D}^*$ contributes to at least one input sample. This aligns more closely with practical scenarios, such as image classification, where multiple samples are typically used during the training phase. We denote the i.i.d. input samples and their corresponding target random variables, generated according to Assumption 6, as $\{\mathbf{x}^h\}_{h=1}^m$ and $\{y^h\}_{h=1}^m$, respectively. For any estimator $\mathbf{y}_{\text{pred}} = \{y_{\text{pred}}^h\}_{h=1}^m$, the aggregated squared loss is defined as

$$\mathcal{L}_m(\mathbf{y}_{\text{pred}}) = \frac{1}{m} \sum_{h=1}^m \mathbb{E}_{\mathbf{z}^h \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{d \times d})} [(y^h - y_{\text{pred}}^h)^2]. \quad (4.5)$$

Theorem 13. *Suppose that $\{\mathbf{x}^h\}_{h=1}^m$ and $\{y^h\}_{h=1}^m$ are i.i.d. samples generated from Assumption 6 with a fixed full-rank column-orthogonal $\mathbf{D}^*$. Suppose that $m = \Omega\left(\frac{n^6}{\sigma^2 k^5}\right)$. Moreover, suppose that the initial query feature matrix $\mathbf{D}$ satisfies $\|\mathbf{D} - \mathbf{D}^*\|_{1,2} = \rho \leq \frac{\gamma}{8\sqrt{k}\Gamma}$ and the step-size η satisfies $\eta = O\left(\frac{1}{\sigma^2}\right)$. With high probability, Algorithm 13 with surrogate estimator (4.4) and update rule (4.4) applied to the aggregated squared loss (4.5) satisfies:*

- **(Interpretability)** $\|\mathbf{D}^{(t+1)} - \mathbf{D}^*\|_{1,2} \leq \tau \|\mathbf{D}^{(t)} - \mathbf{D}^*\|_{1,2}$, where $\tau = \sqrt{1 - \frac{k(k-1)\sigma^2}{2n^2}\eta}$.
- **(Accuracy)** $\mathcal{L}_m(\tilde{f}_{\mathbf{D}^{(t)}}) \leq \frac{k \sum_{h=1}^m \|\mathbf{x}^h\|_2^2}{m} \|\mathbf{D}^{(t)} - \mathbf{D}^*\|_{1,2}^2$.

According to Theorem 13, by leveraging multiple samples, CCR achieves the best of both worlds: it guarantees convergence to the ground truth latent feature matrix $\mathbf{D}^*$, progressively enhancing interpretability, while simultaneously driving $\mathcal{L}_m$ to zero, thereby achieving perfect accuracy.

Contrary to traditional analyses in dictionary learning [7, 75], the proofs of Theorem 12 and Theorem 13 require a careful alignment of the updates from gradient descent with the direction of the true solution. In the setting of Theorem 13, this is even more challenging because the finite sample size m inevitably causes deviations from the population-level behavior. Moreover, we must further control the deviation introduced by the projection step. The detailed proofs of these theorems are provided in Section 4.5. We also include a detailed discussion on both theorems in Section 4.4.2 and experiments that validate them in Section 4.6.

4.4.2 Further Discussion

In this section, we provide further discussion on Theorem 12 and Theorem 13.

Tolerance, Sample Size, and Convergence Speed. Both theorems establish that the population loss converges to zero, provided that the initial dictionary estimate is within $\frac{1}{\sqrt{k}}$ in the column-wise $\|\cdot\|_{1,2}$ -norm from the ground-truth dictionary $\mathbf{D}^*$. Assuming that a constant fraction of the queries in $\mathbf{D}^*$ is utilized to generate each $\mathbf{x}^h$, i.e., $k = \Omega(n)$, Theorem 13 indicates that each column of $\mathbf{D}^*$ can be accurately recovered with a sample size of $m = O(n)$ and a convergence rate of $\alpha = 1 - \Omega(1)$. Notably, the linear sample size and linear convergence rate is consistent with the best-known results in dictionary learning [7, 75].

Sparsity Level. It is well established in the dictionary learning literature that higher sparsity levels in ground-truth sparse codes (i.e., larger values of k in Assumption 6) present greater challenges for recovery; see [7]. Indeed, [7] asserts that sparsity levels beyond $k = \Omega\left(\frac{n}{\log n}\right)$ rarely succeed in practice, even though several approaches have been proposed to handle cases where $k = \Omega(n)$ [117, 75]. In contrast, our results reveal that the sparsity level k plays a nuanced role when optimizing $\mathbf{D}$ for the downstream task $\mathcal{L}$. On the one hand, a larger k imposes a more stringent initial error bound for both theorems. On the other hand, a larger k in Theorem 13 improves the convergence rate τ and reduces the required sample size m . The former phenomenon is consistent with findings in dictionary learning literature,

where exact support recovery becomes more feasible for sparser generative models. A high-level explanation for the latter is that a denser generative model for β provides each $\mathbf{x}$ with more information, thereby expediting the training process.

4.5 Proof of Theorems

4.5.1 Proof of Theorem 11

Our proof strategy follows two main steps:

- **Step 1:** We prove that $\mathcal{L}(f_{\mathbf{D}^*}) = 0$. To do so, we first derive an explicit form of $f_{\mathbf{D}^*}$, and then show $\mathcal{L}(f_{\mathbf{D}^*}) = 0$ for this explicit form.
- **Step 2:** We explicitly design a column-orthogonal $\tilde{\mathbf{D}}$ such that $\|\tilde{\mathbf{D}} - \mathbf{D}^*\|_{1,2} \leq \epsilon$ and $\mathcal{L}(f_{\tilde{\mathbf{D}}}) \geq \frac{2(k-1)\epsilon^2\gamma^2}{5}$.

To streamline the presentation, we denote the i -th column of $\mathbf{D}^*$ as $\mathbf{d}_i^*$. Note that $\mathbf{d}_i^*$ is identical to $\mathbf{v}_i$ used in the paper.

- **Step 1: Establishing $\mathcal{L}(f_{\mathbf{D}^*}) = 0$.** We first provide an explicit characterization of $f_{\mathbf{D}^*}$.

Lemma 8. *We have*

$$f_{\mathbf{D}^*}(\mathbf{x}, \{r_{\pi(i)}\}_{i=1}^k) = \sum_{i \in S} \langle \mathbf{d}_i^*, \mathbf{x} \rangle r_i,$$

where $S = \{\pi(1), \pi(2), \dots, \pi(k)\}$ is the index set selected by IP-OMP applied to $\mathbf{x}$ and $\mathbf{D}^*$.

Our next goal is to provide an explicit characterization of S , introduced in the above lemma.

Lemma 9. *For any column-orthogonal $\mathbf{D}$, the index set $S = \{\pi(1), \pi(2), \dots, \pi(k)\}$ selected by IP-OMP applied to $\mathbf{x}$ and $\mathbf{D}$ corresponds to the indices of the top- k largest values of $\{|\langle \mathbf{d}_i, \mathbf{x} \rangle|\}_{i=1}^n$:*

$$S = \operatorname{argmax}_{T \subseteq [n], |T| \leq k} \sum_{i \in T} |\langle \mathbf{d}_i, \mathbf{x} \rangle|.$$

Moreover, if $\mathbf{D} = \mathbf{D}^*$, we have $S = S^*$, where S^* is the support of β defined in Theorem 11.

We defer the proofs of Lemma 8 and Lemma 9 to Section 4.5.4.2 and Section 4.5.4.3, respectively. Combining the above two lemmas, we obtain:

$$f_{\mathbf{D}^*}(\mathbf{x}, \{r_{\pi(i)}\}_{i=1}^k) = \sum_{i \in S} \langle \mathbf{d}_i^*, \mathbf{x} \rangle r_i, \text{ where } S = \operatorname{argmax}_{T \subseteq [n], |T| \leq k} \sum_{i \in T} |\langle \mathbf{d}_i^*, \mathbf{x} \rangle|$$

Given this explicit form of $f_{\mathbf{D}^*}$ generated by IP-OMP, we are now ready to investigate the population loss $\mathcal{L}$ of that estimator.

For any column-column-orthogonal $\mathbf{D}$, one can write

$$\begin{aligned} \mathcal{L}(f_{\mathbf{D}}) &= \mathbf{E}_{\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{d \times d})} \left[\left(\sum_{i \in S} \langle \mathbf{d}_i, \mathbf{x} \rangle r_i - \langle \mathbf{x}, \mathbf{z} \rangle \right)^2 \right] \\ &= \mathbf{E}_{\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{d \times d})} \left[\left(\sum_{i \in S} \langle \mathbf{d}_i, \mathbf{x} \rangle \langle \mathbf{d}_i^*, \mathbf{z} \rangle - \langle \mathbf{x}, \mathbf{z} \rangle \right)^2 \right] \\ &= \mathbf{E}_{\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{d \times d})} \left[\left(\sum_{i \in S} \langle \mathbf{d}_i^* \mathbf{d}_i^\top \mathbf{x}, \mathbf{z} \rangle - \langle \mathbf{x}, \mathbf{z} \rangle \right)^2 \right] \\ &= \mathbf{E}_{\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{d \times d})} \left[\left(\langle \mathbf{D}_S^* \mathbf{D}_S^\top \mathbf{x}, \mathbf{z} \rangle - \langle \mathbf{x}, \mathbf{z} \rangle \right)^2 \right] \\ &= \mathbf{E}_{\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{d \times d})} \left[\left(\langle \mathbf{D}_S^* \mathbf{D}_S^\top \mathbf{x} - \mathbf{x}, \mathbf{z} \rangle \right)^2 \right] \end{aligned} \tag{4.6}$$

It is easy to verify that random variable $\langle \mathbf{D}_S^* \mathbf{D}_S^\top \mathbf{x} - \mathbf{x}, \mathbf{z} \rangle$ is distributed as $\mathcal{N}(0, \|\mathbf{D}_S^* \mathbf{D}_S^\top \mathbf{x} - \mathbf{x}\|_2^2)$. As a result, we can conclude that

$$\mathcal{L}(f_{\mathbf{D}}) = \|\mathbf{D}_S^* \mathbf{D}_S^\top \mathbf{x} - \mathbf{x}\|_2^2 \text{ for any column-orthogonal } \mathbf{D}. \tag{4.7}$$

Given Lemma 9, we have $S = S^*$. We can subsequently conclude that:

$$\mathbf{D}_S^* \mathbf{D}_S^{*\top} \mathbf{x} = \mathbf{D}_{S^*}^* \mathbf{D}_{S^*}^{*\top} \mathbf{x} = \mathbf{D}_{S^*}^* \mathbf{D}_{S^*}^{*\top} \mathbf{D}_{S^*}^* \beta_{S^*} = \mathbf{D}_{S^*}^* \beta_{S^*} = \mathbf{x},$$

This leads to $\mathcal{L}(f_{\mathbf{D}^*}) = \|\mathbf{D}_S^* \mathbf{D}_S^{*\top} \mathbf{x} - \mathbf{x}\|_2^2 = 0$, thereby completing the proof of our first step.

• **Step 2: Constructing $\tilde{\mathbf{D}}$.** Without loss of generality, let us assume $S^* = [k]$, where k to be even. The case where k is odd is easily proved by following the argument below and

replacing k with $k - 1$. Consider the following explicit form for $\tilde{\mathbf{D}}$:

$$\begin{cases} \tilde{\mathbf{d}}_i = \cos \theta \mathbf{d}_i^* + \sin \theta \mathbf{d}_{i+k/2}^*, & 1 \leq i \leq k/2, \\ \tilde{\mathbf{d}}_{i+k/2} = -\sin \theta \mathbf{d}_i^* + \cos \theta \mathbf{d}_{i+k/2}^*, & 1 \leq i \leq k/2, \\ \tilde{\mathbf{d}}_i = \mathbf{d}_i^* & i > k, \end{cases}$$

where θ is an angle to be determined later.

Intuitively, $\tilde{\mathbf{D}}$ is constructed by iteratively selecting a pair of columns in $\mathbf{D}_S^*$ and rotating them by an angle $\theta > 0$ within the two-dimensional subspace spanned by each pair. Based on its definition, we can establish the following properties for $\tilde{\mathbf{D}}$:

Lemma 10. *$\tilde{\mathbf{D}}$ has the following properties:*

- $\tilde{\mathbf{D}}$ is column-orthogonal.
- $\|\tilde{\mathbf{D}} - \mathbf{D}^*\|_{1,2} = 2 \sin \frac{\theta}{2}$.

The proof of Lemma 10 is presented in Section 4.5.4.4

Our next step is to calculate $\mathcal{L}(f_{\tilde{\mathbf{D}}})$. Note that $\tilde{\mathbf{D}}$ is column-orthogonal according to Lemma 10. Therefore, according to Equation (4.7), we have $\mathcal{L}(f_{\tilde{\mathbf{D}}}) = \|\mathbf{D}_S^* \tilde{\mathbf{D}}_S^\top \mathbf{x} - \mathbf{x}\|_2^2$. According to Lemma 9, in order to obtain the index set S selected by IP-OMP applied to $\mathbf{x}$ and $\tilde{\mathbf{D}}$, we need to find the indices of the top- k largest values of $|\langle \tilde{\mathbf{d}}_i, \mathbf{x} \rangle|$. For $1 \leq i \leq k/2$, we have:

$$\begin{aligned} |\langle \tilde{\mathbf{d}}_i, \mathbf{x} \rangle| &= \left| \left\langle \cos \theta \mathbf{d}_i^* + \sin \theta \mathbf{d}_{i+k/2}^*, \sum_{j=1}^k \beta_j \mathbf{d}_j^* \right\rangle \right| \\ &= |\cos \theta \beta_i + \sin \theta \beta_{i+k/2}| \end{aligned}$$

Upon choosing θ such that $\tan \theta < \gamma/\Gamma$, we have

$$|\cos \theta \beta_i| \geq \left| \frac{\sin \theta \gamma}{\tan \theta} \right| > |\sin \theta \Gamma| \geq |\sin \theta \beta_{i+k/2}|,$$

which guarantees that $|\langle \tilde{\mathbf{d}}_i, \mathbf{x} \rangle| > 0$. Similarly, for $1 + k/2 \leq i \leq k$, we have:

$$\begin{aligned} |\langle \tilde{\mathbf{d}}_i, \mathbf{x} \rangle| &= \left| \left\langle \cos \theta \mathbf{d}_i^* - \sin \theta \mathbf{d}_{i-k/2}^*, \sum_{j=1}^k \beta_j \mathbf{d}_j^* \right\rangle \right| \\ &= |\cos \theta \beta_i - \sin \theta \beta_{i-k/2}| \\ &> 0. \end{aligned}$$

For $i > k$, we have $|\langle \tilde{\mathbf{d}}_i, \mathbf{x} \rangle| = |\langle \mathbf{d}_i^*, \mathbf{x} \rangle| = 0$. As a result, when $\tan \theta < \gamma/\Gamma$, Lemma 9 guarantees that the index set selected by IP-OMP is $S = [k] = S^*$. Given this fact, we can calculate $\mathcal{L}(f_{\tilde{\mathbf{D}}})$ as:

$$\begin{aligned} \mathcal{L}(f_{\tilde{\mathbf{D}}}) &= \|\mathbf{D}_S^* \tilde{\mathbf{D}}_S^\top \mathbf{x} - \mathbf{x}\|_2^2 \\ &= \|\mathbf{D}_{S^*}^* \tilde{\mathbf{D}}_{S^*}^\top \mathbf{x} - \mathbf{x}\|_2^2 \\ &= \|\mathbf{D}_{S^*}^* \tilde{\mathbf{D}}_{S^*}^\top \sum_{i=1}^k \beta_i \mathbf{d}_i^* - \sum_{i=1}^k \beta_i \mathbf{d}_i^*\|_2^2 \\ &\stackrel{(a)}{=} \left\| \sum_{i=1}^{k/2} (\cos \theta \beta_i + \sin \theta \beta_{i+k/2}) \mathbf{d}_i^* + \sum_{i=1+k/2}^k (\cos \theta \beta_i - \sin \theta \beta_{i-k/2}) \mathbf{d}_i^* - \sum_{i=1}^k \beta_i \mathbf{d}_i^* \right\|_2^2 \\ &= \left\| \sum_{i=1}^{k/2} ((\cos \theta - 1) \beta_i + \sin \theta \beta_{i+k/2}) \mathbf{d}_i^* + \sum_{i=1+k/2}^k ((\cos \theta - 1) \beta_i - \sin \theta \beta_{i-k/2}) \mathbf{d}_i^* \right\|_2^2 \\ &\stackrel{(b)}{=} \sum_{i=1}^{k/2} ((\cos \theta - 1) \beta_i + \sin \theta \beta_{i+k/2})^2 + \sum_{i=1+k/2}^k ((\cos \theta - 1) \beta_i - \sin \theta \beta_{i-k/2})^2, \end{aligned} \tag{4.8}$$

where (a) is based on the construction of $\tilde{\mathbf{D}}$, and (b) is due to the assumption that $\mathbf{D}^*$ is

column-orthogonal. When $\sin \frac{\theta}{2} \leq \frac{1}{\sqrt{1+16\Gamma^2/\gamma^2}}$, we have for $1 \leq i \leq k/2$:

$$\begin{aligned}
\sin \theta &= 2 \sin \frac{\theta}{2} \cos \frac{\theta}{2} \\
&= 2 \sin \frac{\theta}{2} \sqrt{1 - \sin^2 \frac{\theta}{2}} \\
&\geq 2 \sin \frac{\theta}{2} \cdot \frac{4\Gamma}{\gamma} \frac{1}{\sqrt{1 + 16\Gamma^2/\gamma^2}} \\
&\geq \frac{8\Gamma}{\gamma} \sin^2 \frac{\theta}{2} \\
&= \frac{4\Gamma}{\gamma} (1 - \cos \theta),
\end{aligned}$$

which leads to

$$\begin{aligned}
&((\cos \theta - 1)\beta_i + \sin \theta \beta_{i+k/2})^2 \\
&= (\cos \theta - 1)^2 \beta_i^2 + 2(\cos \theta - 1) \sin \theta \beta_i \beta_{i+k/2} + \sin^2 \theta \beta_{i+k/2}^2 \\
&\geq 2(\cos \theta - 1) \sin \theta \beta_i \beta_{i+k/2} + \sin^2 \theta \beta_{i+k/2}^2 \\
&\geq -\left| \frac{\gamma}{2\Gamma} \sin^2 \theta \beta_i \beta_{i+k/2} \right| + \sin^2 \theta \beta_{i+k/2}^2 \\
&\geq \frac{1}{2} \sin^2 \theta \beta_{i+k/2}^2.
\end{aligned}$$

With identical arguments, we have for $1 + k/2 \leq i \leq k$:

$$((\cos \theta - 1)\beta_i + \sin \theta \beta_{i-k/2})^2 \geq \frac{1}{2} \sin^2 \theta \beta_{i-k/2}^2.$$

As a result, Equation (4.8) reduces to:

$$\begin{aligned}
\mathcal{L}(f_{\tilde{\mathbf{D}}}) &\geq \sum_{i=1}^{k/2} \frac{1}{2} \sin^2 \theta \beta_{i+k/2}^2 + \sum_{i=1+k/2}^k \frac{1}{2} \sin^2 \theta \beta_{i-k/2}^2 \\
&\geq \frac{k}{2} \sin^2 \theta \gamma^2
\end{aligned}$$

Finally, we need to perform a change of variable by setting $\epsilon = 2 \sin \frac{\theta}{2}$. Elementary calculation gives $\sin \theta = \frac{\epsilon \sqrt{4-\epsilon^2}}{2}$ and $\tan \theta = \frac{\epsilon \sqrt{4-\epsilon^2}}{2-\epsilon^2}$. When $\epsilon < \frac{1}{2}$, we have $\sin \theta \geq 9\epsilon/10$ and $\tan \theta \leq 6\epsilon/5$. Recall that our proof requires $\tan \theta < \gamma/\Gamma$ and $\sin \frac{\theta}{2} \leq \frac{1}{\sqrt{1+16\Gamma^2/\gamma^2}}$. As a result, when

$\epsilon = 2 \sin \frac{\theta}{2} \leq \min \left(\frac{1}{2}, \frac{5\gamma}{6\Gamma}, \frac{1}{\sqrt{1+16\Gamma^2/\gamma^2}} \right) = \frac{1}{\sqrt{1+16\Gamma^2/\gamma^2}}$, we have

$$\mathcal{L}(f_{\tilde{\mathbf{D}}}) - \mathcal{L}(f_{\mathbf{D}^*}) = \mathcal{L}(f_{\tilde{\mathbf{D}}}) \geq \frac{81k\epsilon^2\gamma^2}{200}.$$

As we mentioned, for an odd value of k , an analogous argument can be made to arrive at a similar bound of

$$\mathcal{L}(f_{\tilde{\mathbf{D}}}) - \mathcal{L}(f_{\mathbf{D}^*}) = \mathcal{L}(f_{\tilde{\mathbf{D}}}) \geq \frac{81(k-1)\epsilon^2\gamma^2}{200},$$

This completes the proof of Theorem 11. $\square$

4.5.2 Proof of Theorem 12

To prove the convergence of $\mathcal{L}(\tilde{f}_{\mathbf{D}^{(t)}})$, we use an inductive approach where at each iteration, we will subsequently prove: (1) $\tilde{f}$ recovers the exact support S^* ; (2) gradient descent will make progress towards one of the optimal minimizers which will be explicitly defined later; (3) $\mathcal{L}(\tilde{f}_{\mathbf{D}^{(t)}})$ will decrease linearly.

Recall that $\mathbf{D}^{(t)} = \mathbf{D} + \Delta\mathbf{D}^{(t)}$. The equations below rewrite Section 4.4 for each column $i \in [n]$ of $\mathbf{D}^{(t)}$.

$$\mathbf{d}_i^{(t+0.5)} = \mathbf{d}_i^{(t)} - \eta \frac{\partial \mathcal{L}(\tilde{f}_{\mathbf{D}^{(t)}})}{\partial \mathbf{d}_i^{(t)}}, \quad (4.9)$$

$$\mathbf{d}_i^{(t+1)} = \operatorname{argmin}_{\mathbf{d}: \|\mathbf{d} - \mathbf{d}_i^{(0)}\|_2 \leq \rho} \|\mathbf{d} - \mathbf{d}_i^{(t+0.5)}\|_2. \quad (4.10)$$

It is easy to verify that the above update rules are equivalent to Section 4.4. We will use induction to prove this theorem. In particular, define

$$A(t) : \quad \mathcal{L}(\tilde{f}_{\mathbf{D}^{(t)}}) \leq \tau^2 \mathcal{L}(\tilde{f}_{\mathbf{D}^{(t-1)}}), \text{ where } \tau = 1 - 2\eta\|\mathbf{x}\|_2^2.$$

To prove this we define an auxiliary feature vector which will play an important role in our arguments:

$$\hat{\mathbf{d}}_i = \mathbf{d}_i^{(0)} + \frac{\beta_i - \mathbf{x}^\top \mathbf{d}_i^{(0)}}{\|\mathbf{x}\|_2^2} \mathbf{x}, \quad \forall i \in [n]. \quad (4.11)$$

Based on this auxiliary feature vector, we define the following event:

$$B(t) : \quad \widehat{\mathbf{d}}_i - \mathbf{d}_i^{(t)} = \frac{\beta_i - \mathbf{x}^\top \mathbf{d}_i^{(t)}}{\|\mathbf{x}\|_2^2} \mathbf{x}, \quad \forall i \in [n].$$

Now, our proof strategy is to show that, for $t = 0, 1, \dots$:

- $B(t)$ implies $B(t+1)$,
- $B(t)$ implies $A(t+1)$.

This two statements combined will establish the correctness of Theorem 12. Indeed, the base case $A(0)$ is trivially satisfied due to Equation (4.11).

• **Establishing $B(t) \implies B(t+1)$.** To begin with, we use the following lemma to show that $\tilde{f}$ is able to find the correct support when $\mathbf{D}^{(t)}$ is close enough to $\mathbf{D}$:

Lemma 11. *For any $\mathbf{D}$ such that $\|\mathbf{D} - \mathbf{D}^*\|_{1,2} \leq 2\rho < \frac{\gamma}{4\sqrt{k}\Gamma}$, we have*

$$S = \operatorname{argmax}_{T \subseteq [n], |T| \leq k} \sum_{i \in T} |\langle \mathbf{d}_i, \mathbf{x} \rangle| = S^*.$$

We defer the proof of Lemma 11 to Section 4.5.4.5. To invoke Lemma 11, we have

$$\|\mathbf{D}^{(t)} - \mathbf{D}^*\|_{1,2} \leq \|\mathbf{D}^{(t)} - \mathbf{D}^{(0)}\|_{1,2} + \|\mathbf{D}^{(0)} - \mathbf{D}^*\|_{1,2} \leq \rho + \rho = 2\rho.$$

The bound on $\|\mathbf{D}^{(t)} - \mathbf{D}^{(0)}\|_{1,2}$ follows from the projection step in Equation (4.10) while the bound on $\|\mathbf{D}^{(0)} - \mathbf{D}^*\|_{1,2}$ follows from the initial error bound. We can then conclude that $S = S^*$ at every iteration t , which means that Equation (4.9) and Equation (4.10) will only change $i \in S^*$, while for other $i \notin S^*$, we have $\mathbf{d}_i^{(t+1)} = \mathbf{d}_i^{(t)}$. This immediately implies that $B(t)$ is trivially satisfied for $\forall i \notin S^*$ and $\forall t$. Now, the goal is to show that $B(t) \implies B(t+1)$ for $\forall i \in S^*$.

Based on Equation (4.7), we have

$$\mathcal{L}(\tilde{f}_{\mathbf{D}^{(t)}}) = \|\mathbf{D}_{S^*}^* \mathbf{D}_{S^*}^{(t)\top} \mathbf{x} - \mathbf{x}\|_2^2,$$

Subsequently, the gradient of $\mathcal{L}$ at $\mathbf{D}_{S^*}^{(t)}$ can be written as:

$$\begin{aligned}
\frac{\partial \mathcal{L}(\tilde{f}_{\mathbf{D}^{(t)}})}{\partial \mathbf{D}_{S^*}^{(t)}} &= 2\mathbf{x} \left(\mathbf{D}_{S^*}^* \mathbf{D}_{S^*}^{(t)\top} \mathbf{x} - \mathbf{x} \right)^\top \mathbf{D}_{S^*}^* \\
&= 2\mathbf{x} \left(\mathbf{D}_{S^*}^* \mathbf{D}_{S^*}^{(t)\top} \mathbf{x} - \mathbf{D}_{S^*}^* \beta_{S^*} \right)^\top \mathbf{D}_{S^*}^* \\
&= 2\mathbf{x} \left(\mathbf{D}_{S^*}^{(t)\top} \mathbf{x} - \beta_{S^*} \right)^\top \mathbf{D}_{S^*}^{*\top} \mathbf{D}_{S^*}^* \\
&= 2\mathbf{x} \left(\mathbf{D}_{S^*}^{(t)\top} \mathbf{x} - \beta_{S^*} \right)^\top.
\end{aligned}$$

For $i \in S^*$, we have

$$\frac{\partial \mathcal{L}(\tilde{f}_{\mathbf{D}^{(t)}})}{\partial \mathbf{d}_i^{(t)}} = 2(\mathbf{d}_i^{(t)\top} \mathbf{x} - \beta_i) \mathbf{x}.$$

This implies that:

$$\begin{aligned}
\widehat{\mathbf{d}}_i - \mathbf{d}_i^{(t+0.5)} &= \widehat{\mathbf{d}}_i - \mathbf{d}_i^{(t)} + 2\eta(\mathbf{d}_i^{(t)\top} \mathbf{x} - \beta_i) \mathbf{x} \\
&\stackrel{A(t)}{=} \frac{\beta_i - \mathbf{x}^\top \mathbf{d}_i^{(t)}}{\|\mathbf{x}\|_2^2} \mathbf{x} + 2\eta(\mathbf{d}_i^{(t)\top} \mathbf{x} - \beta_i) \mathbf{x} \\
&= \frac{\beta_i - \mathbf{x}^\top \left(\mathbf{d}_i^{(t+0.5)} + 2\eta(\mathbf{d}_i^{(t)\top} \mathbf{x} - \beta_i) \mathbf{x} \right)}{\|\mathbf{x}\|_2^2} \mathbf{x} + 2\eta(\mathbf{d}_i^{(t)\top} \mathbf{x} - \beta_i) \mathbf{x} \\
&= \frac{\beta_i - \mathbf{x}^\top \mathbf{d}_i^{(t+0.5)}}{\|\mathbf{x}\|_2^2} \mathbf{x} - 2\eta(\mathbf{d}_i^{(t)\top} \mathbf{x} - \beta_i) \mathbf{x} + 2\eta(\mathbf{d}_i^{(t)\top} \mathbf{x} - \beta_i) \mathbf{x} \\
&= \frac{\beta_i - \mathbf{x}^\top \mathbf{d}_i^{(t+0.5)}}{\|\mathbf{x}\|_2^2} \mathbf{x}.
\end{aligned}$$

In other words, $B(t)$ implies $B(t+0.5)$.

Suppose $\|\mathbf{d}_i^{(t+0.5)} - \mathbf{d}_i^{(0)}\|_2 \leq \rho$, then we have $\mathbf{d}_i^{(t+1)} = \mathbf{d}_i^{(t+0.5)}$, which readily implies $B(t+1)$. On the other hand, if $\|\mathbf{d}_i^{(t+0.5)} - \mathbf{d}_i^{(0)}\|_2 > \rho$, we have:

$$\begin{aligned}
\mathbf{d}_i^{(t+0.5)} - \mathbf{d}_i^{(0)} &= \left(\mathbf{d}_i^{(t+0.5)} - \widehat{\mathbf{d}}_i \right) - \left(\mathbf{d}_i^{(0)} - \widehat{\mathbf{d}}_i \right) \\
&\stackrel{A(t+0.5), A(0)}{=} -\frac{\beta_i - \mathbf{x}^\top \mathbf{d}_i^{(t+0.5)}}{\|\mathbf{x}\|_2^2} \mathbf{x} + \frac{\beta_i - \mathbf{x}^\top \mathbf{d}_i^{(0)}}{\|\mathbf{x}\|_2^2} \mathbf{x} \\
&= \frac{\mathbf{x}^\top \mathbf{d}_i^{(t+0.5)} - \mathbf{x}^\top \mathbf{d}_i^{(0)}}{\|\mathbf{x}\|_2^2} \mathbf{x},
\end{aligned} \tag{4.12}$$

Therefore, when $\|\mathbf{d}_i^{(t+0.5)} - \mathbf{d}_i^{(0)}\|_2 > \rho$, we have:

$$\begin{aligned}
\mathbf{d}_i^{(t+1)} &= \operatorname{argmin}_{\mathbf{d}: \|\mathbf{d} - \mathbf{d}_i^{(0)}\|_2 \leq \rho} \|\mathbf{d} - \mathbf{d}_i^{(t+0.5)}\|_2 \\
&= \mathbf{d}_i^{(0)} + \rho \frac{\mathbf{d}_i^{(t+0.5)} - \mathbf{d}_i^{(0)}}{\|\mathbf{d}_i^{(t+0.5)} - \mathbf{d}_i^{(0)}\|_2} \\
&= \mathbf{d}_i^{(0)} + \rho \operatorname{sign} \left(\mathbf{x}^\top \mathbf{d}_i^{(t+0.5)} - \mathbf{x}^\top \mathbf{d}_i^{(0)} \right) \frac{\mathbf{x}}{\|\mathbf{x}\|_2}.
\end{aligned} \tag{4.13}$$

This in turn implies

$$\begin{aligned}
&\widehat{\mathbf{d}}_i - \mathbf{d}_i^{(t+1)} \\
&= \widehat{\mathbf{d}}_i - \mathbf{d}_i^{(0)} - \rho \operatorname{sign} \left(\mathbf{x}^\top \mathbf{d}_i^{(t+0.5)} - \mathbf{x}^\top \mathbf{d}_i^{(0)} \right) \frac{\mathbf{x}}{\|\mathbf{x}\|_2} \\
&\stackrel{A(0)}{=} \frac{\beta_i - \mathbf{x}^\top \mathbf{d}_i^{(0)}}{\|\mathbf{x}\|_2^2} \mathbf{x} - \rho \operatorname{sign} \left(\mathbf{x}^\top \mathbf{d}_i^{(t+0.5)} - \mathbf{x}^\top \mathbf{d}_i^{(0)} \right) \frac{\mathbf{x}}{\|\mathbf{x}\|_2} \\
&= \frac{\beta_i - \mathbf{x}^\top \left(\mathbf{d}_i^{(t+1)} - \rho \operatorname{sign} \left(\mathbf{x}^\top \mathbf{d}_i^{(t+0.5)} - \mathbf{x}^\top \mathbf{d}_i^{(0)} \right) \frac{\mathbf{x}}{\|\mathbf{x}\|_2} \right)}{\|\mathbf{x}\|_2^2} \mathbf{x} - \rho \operatorname{sign} \left(\mathbf{x}^\top \mathbf{d}_i^{(t+0.5)} - \mathbf{x}^\top \mathbf{d}_i^{(0)} \right) \frac{\mathbf{x}}{\|\mathbf{x}\|_2} \\
&= \frac{\beta_i - \mathbf{x}^\top \mathbf{d}_i^{(t+1)}}{\|\mathbf{x}\|_2^2} \mathbf{x},
\end{aligned}$$

establishing $B(t+1)$, as desired.

• **Establishing $B(t) \implies A(t+1)$.** First, we establish $\|\widehat{\mathbf{d}}_i - \mathbf{d}_i^{(t+1)}\|_2 \leq \tau \|\widehat{\mathbf{d}}_i - \mathbf{d}_i^{(t)}\|_2$. We do so in two steps. First, we prove that $\|\widehat{\mathbf{d}}_i - \mathbf{d}_i^{(t+0.5)}\|_2 \leq \tau \|\widehat{\mathbf{d}}_i - \mathbf{d}_i^{(t)}\|_2$. Then, we show that $\|\widehat{\mathbf{d}}_i - \mathbf{d}_i^{(t+1)}\|_2 \leq \|\widehat{\mathbf{d}}_i - \mathbf{d}_i^{(t+0.5)}\|_2$.

To establish $\|\widehat{\mathbf{d}}_i - \mathbf{d}_i^{(t+1)}\|_2 \leq \tau \|\widehat{\mathbf{d}}_i - \mathbf{d}_i^{(t)}\|_2$, one can write:

$$\begin{aligned}
\|\widehat{\mathbf{d}}_i - \mathbf{d}_i^{(t+0.5)}\|_2 &= \|\widehat{\mathbf{d}}_i - \mathbf{d}_i^{(t)} + 2\eta(\mathbf{d}_i^{(t)\top} \mathbf{x} - \beta_i) \mathbf{x}\|_2 \\
&\stackrel{A(t)}{=} \left\| \widehat{\mathbf{d}}_i - \mathbf{d}_i^{(t)} - 2\eta \|\mathbf{x}\|_2^2 \left(\widehat{\mathbf{d}}_i - \mathbf{d}_i^{(t)} \right) \right\|_2 \\
&\leq |1 - 2\eta \|\mathbf{x}\|_2^2| \|\widehat{\mathbf{d}}_i - \mathbf{d}_i^{(t)}\|_2 \\
&= \tau \|\widehat{\mathbf{d}}_i - \mathbf{d}_i^{(t)}\|_2.
\end{aligned}$$

Next, we show that $\|\widehat{\mathbf{d}}_i - \mathbf{d}_i^{(t+1)}\|_2 \leq \|\widehat{\mathbf{d}}_i - \mathbf{d}_i^{(t+0.5)}\|_2$. To do so, we need the following elementary lemma:

Lemma 12. *Given a fixed vector $\mathbf{v}$ and $\mathbf{a} = a\mathbf{v}$, $\mathbf{b} = b\mathbf{v}$, $\mathbf{c} = c\mathbf{v}$, where $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{c}$ are vectors and a , b , c are scalars, if $|c| \geq |b| \geq a$ and $bc > 0$, we have $\|\mathbf{c} - \mathbf{a}\|_2 \geq \|\mathbf{b} - \mathbf{a}\|_2$.*

We will prove Lemma 12 in Section 4.5.4.6. Now we invoke Lemma 12 with:

$$\begin{aligned} \mathbf{v} &= \frac{\mathbf{x}}{\|\mathbf{x}\|_2}, \\ a &= \frac{\beta_i - \mathbf{x}^\top \mathbf{d}_i^{(0)}}{\|\mathbf{x}\|_2}, \\ b &= \rho \text{sign} \left(\mathbf{x}^\top \mathbf{d}_i^{(t+0.5)} - \mathbf{x}^\top \mathbf{d}_i^{(0)} \right), \\ c &= \frac{\mathbf{x}^\top \mathbf{d}_i^{(t+0.5)} - \mathbf{x}^\top \mathbf{d}_i^{(0)}}{\|\mathbf{x}\|_2}. \end{aligned}$$

Here $bc > 0$ is given by their definition. Moreover $|c| \geq |b|$ is established as:

$$\begin{aligned} |c| &= \left| \frac{\mathbf{x}^\top \mathbf{d}_i^{(t+0.5)} - \mathbf{x}^\top \mathbf{d}_i^{(0)}}{\|\mathbf{x}\|_2} \right| \\ &= \left\| \frac{\mathbf{x}^\top \mathbf{d}_i^{(t+0.5)} - \mathbf{x}^\top \mathbf{d}_i^{(0)}}{\|\mathbf{x}\|_2^2} \mathbf{x} \right\|_2 \\ &\stackrel{(4.12)}{=} \|\mathbf{d}_i^{(t+0.5)} - \mathbf{d}_i^{(0)}\|_2 \\ &\geq \rho \\ &= |b|. \end{aligned}$$

Finally, $|b| > a$ is given by:

$$\begin{aligned} a &= \frac{\beta_i - \mathbf{x}^\top \mathbf{d}_i^{(0)}}{\|\mathbf{x}\|_2} \\ &= \frac{\mathbf{x}^\top \mathbf{d}_i^* - \mathbf{x}^\top \mathbf{d}_i^{(0)}}{\|\mathbf{x}\|_2} \\ &\leq \frac{\|\mathbf{d}_i^* - \mathbf{d}_i^{(0)}\|_2 \|\mathbf{x}\|_2}{\|\mathbf{x}\|_2} \\ &\leq \|\mathbf{D}^* - \mathbf{D}^{(0)}\|_{1,2} \\ &= \rho \\ &= |b|. \end{aligned} \tag{4.14}$$

Now, upon substituting these parameters in Lemma 12, we have $\|\mathbf{c} - \mathbf{a}\|_2 \geq \|\mathbf{b} - \mathbf{a}\|_2$,

which can be rewritten as:

$$\begin{aligned} & \left\| \frac{\mathbf{x}^\top \mathbf{d}_i^{(t+0.5)} - \mathbf{x}^\top \mathbf{d}_i^{(0)}}{\|\mathbf{x}\|_2^2} \mathbf{x} - \frac{\beta_i - \mathbf{x}^\top \mathbf{d}_i^{(0)}}{\|\mathbf{x}\|_2^2} \mathbf{x} \right\|_2 \\ & \geq \left\| \rho \text{sign} \left(\mathbf{x}^\top \mathbf{d}_i^{(t+0.5)} - \mathbf{x}^\top \mathbf{d}_i^{(0)} \right) \frac{\mathbf{x}}{\|\mathbf{x}\|_2} - \frac{\beta_i - \mathbf{x}^\top \mathbf{d}_i^{(0)}}{\|\mathbf{x}\|_2^2} \mathbf{x} \right\|_2. \end{aligned}$$

The above inequality leads to:

$$\begin{aligned} \left\| \left(\mathbf{d}_i^{(t+0.5)} - \mathbf{d}_i^{(0)} \right) - \left(\widehat{\mathbf{d}}_i - \mathbf{d}_i^{(0)} \right) \right\|_2 & \stackrel{(a)}{\geq} \left\| \left(\mathbf{d}_i^{(t+1)} - \mathbf{d}_i^{(0)} \right) - \left(\widehat{\mathbf{d}}_i - \mathbf{d}_i^{(0)} \right) \right\|_2 \\ \left\| \mathbf{d}_i^{(t+0.5)} - \widehat{\mathbf{d}}_i \right\|_2 & \geq \left\| \mathbf{d}_i^{(t+1)} - \widehat{\mathbf{d}}_i \right\|_2 \end{aligned}$$

Here inequality (a) is obtained by Equation (4.12), Equation (4.11), and Equation (4.13). We can now conclude that:

$$\left\| \mathbf{d}_i^{(t+1)} - \widehat{\mathbf{d}}_i \right\|_2 \leq \left\| \mathbf{d}_i^{(t+0.5)} - \widehat{\mathbf{d}}_i \right\|_2 \leq \tau \left\| \mathbf{d}_i^{(t)} - \widehat{\mathbf{d}}_i \right\|_2. \quad (4.15)$$

Finally, we are ready to establish $A(t+1)$. We rewrite $\mathcal{L}(\tilde{f}_{\mathbf{D}(t)})$ as:

$$\begin{aligned} \mathcal{L}(\tilde{f}_{\mathbf{D}(t)}) &= \|\mathbf{D}_{S^*}^* \mathbf{D}_{S^*}^{(t)\top} \mathbf{x} - \mathbf{x}\|_2^2 \\ &= \|\mathbf{D}_{S^*}^* \mathbf{D}_{S^*}^{(t)\top} \mathbf{x} - \mathbf{D}_{S^*}^* \beta_{S^*}\|_2^2 \\ &= \|\mathbf{D}_{S^*}^{(t)\top} \mathbf{x} - \beta_{S^*}\|_2^2 \\ &= \sum_{i \in S^*} \left(\mathbf{x}^\top \mathbf{d}_i^{(t)} - \beta_i \right)^2 \\ &= \sum_{i \in S^*} \left\| \frac{\beta_i - \mathbf{x}^\top \mathbf{d}_i^{(t)}}{\|\mathbf{x}\|_2^2} \mathbf{x} \right\|_2^2 \|\mathbf{x}\|_2^2 \\ &\stackrel{B(t)}{=} \sum_{i \in S^*} \left\| \mathbf{d}_i^{(t)} - \widehat{\mathbf{d}}_i \right\|_2^2 \|\mathbf{x}\|_2^2 \end{aligned}$$

On the other hand,

$$\mathcal{L}(\tilde{f}_{\mathbf{D}(t+1)}) = \sum_{i \in S^*} \left\| \mathbf{d}_i^{(t+1)} - \widehat{\mathbf{d}}_i \right\|_2^2 \|\mathbf{x}\|_2^2$$

can be established by identical arguments. Finally, we have

$$\mathcal{L}(\tilde{f}_{\mathbf{D}^{(t+1)}}) = \sum_{i \in S^*} \left\| \mathbf{d}_i^{(t+1)} - \hat{\mathbf{d}}_i \right\|_2^2 \|\mathbf{x}\|_2^2 \stackrel{(4.15)}{\leq} \tau^2 \sum_{i \in S^*} \left\| \mathbf{d}_i^{(t)} - \hat{\mathbf{d}}_i \right\|_2^2 \|\mathbf{x}\|_2^2 = \mathcal{L}(\tilde{f}_{\mathbf{D}^{(t)}}),$$

which is exactly $B(t+1)$. As a result, we have proved Theorem 12.

4.5.3 Proof of Theorem 13

To prove the statements of this theorem, we first establish the convergence of $\|\mathbf{d}_i^{(t)} - \mathbf{d}_i^*\|_2$ for every $i \in [n]$. Then, we provide the desired upper bound on $\mathcal{L}_m(\tilde{f}_{\mathbf{D}^{(t)}})$ in terms of $\|\mathbf{D}^{(t)} - \mathbf{D}^*\|_{1,2}^2$.

We will follow the same index convention defined in Equation (4.9) and Equation (4.10), with $\mathcal{L}$ replaced by $\mathcal{L}_m$. For each $h \in [m]$, we use β^h and S^{h*} to denote the corresponding variables in Assumption 6. We define the set Q_i as the index set of h such that i is in the support of β^h :

$$Q_i := \{h \in [m] \mid i \in S^{h*}\}.$$

Given Lemma 11, we have:

$$\frac{\partial \mathcal{L}_m(\tilde{f}_{\mathbf{D}^{(t)}})}{\partial \mathbf{d}_i^{(t)}} = \frac{1}{m} \sum_{h \in Q_i} 2(\mathbf{d}_i^{(t)\top} \mathbf{x}^h - \beta_i^h) \mathbf{x}^h, \quad \forall i \in [n].$$

To ensure that $\frac{\partial \mathcal{L}_m(\tilde{f}_{\mathbf{D}^{(t)}})}{\partial \mathbf{d}_i^{(t)}}$ is indeed aligned with $\mathbf{d}_i^{(t)} - \mathbf{d}_i^*$, we first need the following lemma:

Lemma 13. *Suppose that $m = \Omega\left(\frac{n^6}{\sigma^2 k^5}\right)$. With probability at least $1 - 2 \exp\{\log n - n\} - 2 \exp\{\log n - \frac{km}{8n}\}$, for all $i \in [n]$,*

$$\begin{aligned} \frac{1}{m} \sigma_d \left(\sum_{h \in Q_i} \mathbf{x}^h \mathbf{x}^{h\top} \right) &\geq \frac{k(k-1)\sigma^2}{4n^2}, \\ \frac{1}{m} \sigma_1 \left(\sum_{h \in Q_i} \mathbf{x}^h \mathbf{x}^{h\top} \right) &\leq \frac{4k\sigma^2}{n}. \end{aligned}$$

We defer the proof of Lemma 13 to Section 4.5.4.7. Now, we can bound $\|\mathbf{d}_i^{(t+0.5)} - \mathbf{d}_i^*\|_2^2$

as:

$$\begin{aligned}
\|\mathbf{d}_i^{(t+0.5)} - \mathbf{d}_i^*\|_2^2 &= \left\| \mathbf{d}_i^{(t)} - \frac{\eta}{m} \sum_{h \in Q_i} 2(\mathbf{d}_i^{(t)\top} \mathbf{x}^h - \beta_i^h) \mathbf{x}^h - \mathbf{d}_i^* \right\|_2^2 \\
&= \|\mathbf{d}_i^{(t)} - \mathbf{d}_i^*\|_2^2 - 4\eta \left\langle \mathbf{d}_i^{(t)} - \mathbf{d}_i^*, \frac{1}{m} \sum_{h \in Q_i} (\mathbf{d}_i^{(t)\top} \mathbf{x}^h - \beta_i^h) \mathbf{x}^h \right\rangle + 4\eta^2 \left\| \frac{1}{m} \sum_{h \in Q_i} (\mathbf{d}_i^{(t)\top} \mathbf{x}^h - \beta_i^h) \mathbf{x}^h \right\|_2^2.
\end{aligned} \tag{4.16}$$

For the second term on the right hand side, we have:

$$\begin{aligned}
&\left\langle \mathbf{d}_i^{(t)} - \mathbf{d}_i^*, \frac{1}{m} \sum_{h \in Q_i} (\mathbf{d}_i^{(t)\top} \mathbf{x}^h - \beta_i^h) \mathbf{x}^h \right\rangle \\
&= \left\langle \mathbf{d}_i^{(t)} - \mathbf{d}_i^*, \frac{1}{m} \sum_{h \in Q_i} (\mathbf{d}_i^{(t)\top} \mathbf{x}^h - \mathbf{d}_i^{*\top} \mathbf{x}^h) \mathbf{x}^h \right\rangle \\
&= \left\langle \mathbf{d}_i^{(t)} - \mathbf{d}_i^*, \frac{1}{m} \sum_{h \in Q_i} \mathbf{x}^h \mathbf{x}^{h\top} (\mathbf{d}_i^{(t)} - \mathbf{d}_i^*) \right\rangle \\
&\geq \frac{1}{m} \sigma_d \left(\sum_{h \in Q_i} \mathbf{x}^h \mathbf{x}^{h\top} \right) \|\mathbf{d}_i^{(t)} - \mathbf{d}_i^*\|_2^2 \\
&\stackrel{\text{Lemma 13}}{\geq} \frac{k(k-1)\sigma^2}{4n^2} \|\mathbf{d}_i^{(t)} - \mathbf{d}_i^*\|_2^2.
\end{aligned}$$

For the third term on the right hand side of Equation (4.16), we have:

$$\begin{aligned}
&\left\| \frac{1}{m} \sum_{h \in Q_i} (\mathbf{d}_i^{(t)\top} \mathbf{x}^h - \beta_i^h) \mathbf{x}^h \right\|_2^2 \\
&= \left\| \frac{1}{m} \sum_{h \in Q_i} \mathbf{x}^h \mathbf{x}^{h\top} (\mathbf{d}_i^{(t)} - \mathbf{d}_i^*) \right\|_2^2 \\
&\leq \frac{1}{m^2} \sigma_1^2 \left(\sum_{h \in Q_i} \mathbf{x}^h \mathbf{x}^{h\top} \right) \|\mathbf{d}_i^{(t)} - \mathbf{d}_i^*\|_2^2 \\
&\stackrel{\text{Lemma 13}}{\leq} \frac{16k^2\sigma^4}{n^2} \|\mathbf{d}_i^{(t)} - \mathbf{d}_i^*\|_2^2.
\end{aligned}$$

When $\eta < \frac{k-1}{128k\sigma^2}$, the above two inequalities can be combined with Equation (4.16) to arrive

at:

$$\begin{aligned}
\|\mathbf{d}_i^{(t+0.5)} - \mathbf{d}_i^*\|_2^2 &\leq \left(1 - \frac{k(k-1)\sigma^2}{n^2}\eta + \frac{64k^2\sigma^4}{n^2}\eta^2\right) \|\mathbf{d}_i^{(t)} - \mathbf{d}_i^*\|_2^2 \\
&\leq \left(1 - \frac{k(k-1)\sigma^2}{2n^2}\eta\right) \|\mathbf{d}_i^{(t)} - \mathbf{d}_i^*\|_2^2 \\
&\leq \tau^2 \|\mathbf{d}_i^{(t)} - \mathbf{d}_i^*\|_2^2.
\end{aligned} \tag{4.17}$$

Next, we aim to show that $\|\mathbf{d}_i^{(t+1)} - \mathbf{d}_i^*\|_2^2 \leq \|\mathbf{d}_i^{(t+0.5)} - \mathbf{d}_i^*\|_2^2$. Recall that:

$$\mathbf{d}_i^{(t+1)} = \operatorname{argmin}_{\mathbf{d} \in D} \|\mathbf{d} - \mathbf{d}_i^{(t+0.5)}\|_2^2 \quad \text{where} \quad D = \{\mathbf{d} : \|\mathbf{d} - \mathbf{d}_i^{(0)}\|_2 \leq \rho\}.$$

It is obvious that D is convex and $\|\mathbf{d} - \mathbf{d}_i^{(t+0.5)}\|_2^2$ is strongly convex with respect to $\mathbf{d}$. As a result, first order stationary condition requires that:

$$\left\langle \nabla_{\mathbf{d}=\mathbf{d}_i^{(t+1)}} \|\mathbf{d} - \mathbf{d}_i^{(t+0.5)}\|_2^2, \tilde{\mathbf{d}} - \mathbf{d}_i^{(t+1)} \right\rangle = 2 \left\langle \mathbf{d}_i^{(t+1)} - \mathbf{d}_i^{(t+0.5)}, \tilde{\mathbf{d}} - \mathbf{d}_i^{(t+1)} \right\rangle \geq 0,$$

for any $\tilde{\mathbf{d}} \in D$. Given the initial error bound $\|\mathbf{D}^{(0)} - \mathbf{D}^*\|_{1,2} \leq \rho$, we have $\mathbf{d}^* \in D$, which results in:

$$\left\langle \mathbf{d}_i^{(t+1)} - \mathbf{d}_i^{(t+0.5)}, \mathbf{d}^* - \mathbf{d}_i^{(t+1)} \right\rangle \geq 0.$$

We subsequently have:

$$\begin{aligned}
\|\mathbf{d}_i^* - \mathbf{d}_i^{(t+0.5)}\|_2^2 &= \left\| \left(\mathbf{d}_i^{(t+1)} - \mathbf{d}_i^{(t+0.5)} \right) + \left(\mathbf{d}_i^* - \mathbf{d}_i^{(t+1)} \right) \right\|_2^2 \\
&= \|\mathbf{d}_i^{(t+1)} - \mathbf{d}_i^{(t+0.5)}\|_2^2 + 2 \left\langle \mathbf{d}_i^{(t+1)} - \mathbf{d}_i^{(t+0.5)}, \mathbf{d}_i^* - \mathbf{d}_i^{(t+1)} \right\rangle + \|\mathbf{d}_i^* - \mathbf{d}_i^{(t+1)}\|_2^2 \\
&\geq \|\mathbf{d}_i^* - \mathbf{d}_i^{(t+1)}\|_2^2,
\end{aligned} \tag{4.18}$$

where the last inequality follows from the fact that both $\|\mathbf{d}_i^{(t+1)} - \mathbf{d}_i^{(t+0.5)}\|_2^2$ and $\left\langle \mathbf{d}_i^{(t+1)} - \mathbf{d}_i^{(t+0.5)}, \mathbf{d}_i^* - \mathbf{d}_i^{(t+1)} \right\rangle$ are non-negative. Combining Equation (4.17) and Equation (4.18), we have

$$\|\mathbf{d}_i^{(t+1)} - \mathbf{d}_i^*\|_2^2 \leq \|\mathbf{d}_i^{(t+0.5)} - \mathbf{d}_i^*\|_2^2 \leq \tau^2 \|\mathbf{d}_i^{(t)} - \mathbf{d}_i^*\|_2^2.$$

Finally, note that

$$\|\mathbf{D}^{(t+1)} - \mathbf{D}^*\|_{1,2} = \max_{i \in [n]} \{\|\mathbf{d}_i^{(t+1)} - \mathbf{d}_i^*\|_2\} \leq \tau \max_{i \in [n]} \{\|\mathbf{d}_i^{(t)} - \mathbf{d}_i^*\|_2\} \leq \tau \|\mathbf{D}^{(t)} - \mathbf{D}^*\|_{1,2},$$

Next, we show that $\mathcal{L}_m(\mathbf{D}^{(t)})$ is upper bounded by $\|\mathbf{D}^{(t)} - \mathbf{D}^*\|_{1,2}$:

$$\begin{aligned}
\mathcal{L}_m(\tilde{f}_{\mathbf{D}^{(t)}}) &= \frac{1}{m} \sum_{h=1}^m \|\mathbf{D}_{S^{h*}}^* \mathbf{D}_{S^{h*}}^{(t)\top} \mathbf{x}^h - \mathbf{x}^h\|_2^2 \\
&= \frac{1}{m} \sum_{h=1}^m \|\mathbf{D}_{S^{h*}}^{(t)\top} \mathbf{x}^h - \beta_{S^{h*}}^h\|_2^2 \\
&= \frac{1}{m} \sum_{h=1}^m \sum_{i \in S^{h*}} \left(\mathbf{x}^h{}^\top \mathbf{d}_i^{(t)} - \mathbf{x}^h{}^\top \mathbf{d}_i^* \right)^2 \\
&\leq \frac{1}{m} \sum_{h=1}^m \sum_{i \in S^{h*}} \|\mathbf{x}^h\|_2^2 \|\mathbf{d}_i^{(t)} - \mathbf{d}_i^*\|_2^2 \\
&\leq \frac{1}{m} \sum_{h=1}^m k \|\mathbf{x}^h\|_2^2 \|\mathbf{D}^{(t)} - \mathbf{D}^*\|_{1,2}^2 \\
&= \frac{k \sum_{h=1}^m \|\mathbf{x}^h\|_2^2}{m} \|\mathbf{D}^{(t)} - \mathbf{D}^*\|_{1,2}^2.
\end{aligned}$$

To complete the proof of Theorem 13, we need to examine the success probability which the above linear convergence occurs with. The only probability statement we use is Lemma 13 and we only invoke it once. So the total success probability is at least:

$$1 - 2 \exp\{\log n - n\} - 2 \exp\left\{\log n - \frac{km}{8n}\right\},$$

which is $1 - n^{-\omega(1)}$ when $m = \Omega\left(\frac{n^6}{\sigma^2 k^5}\right)$. This completes the proof of Theorem 13. $\square$

4.5.4 Proof of Lemmas

4.5.4.1 Proof of Lemma 7

Lemma 7 directly follows from Lemma 8 and Lemma 9.

4.5.4.2 Proof of Lemma 8

We can decompose y as follows:

$$y = \langle \mathbf{x}, \mathbf{z} \rangle = \langle \Pi_{\mathbf{D}_S^*} \mathbf{x}, \mathbf{z} \rangle + \langle \Pi_{\mathbf{D}_S^*}^\perp \mathbf{x}, \mathbf{z} \rangle$$

Note that $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{d \times d})$. This implies that $\Pi_{\mathbf{D}_S^*}^\perp \mathbf{z}$ is independent of $\Pi_{\mathbf{D}_S^*} \mathbf{z}$ [6], which in turn entails that $\langle \Pi_{\mathbf{D}_S^*}^\perp \mathbf{x}, \mathbf{z} \rangle$ is independent of $\langle \Pi_{\mathbf{D}_S^*} \mathbf{x}, \mathbf{z} \rangle$ and the events $\{\langle \mathbf{d}_i^*, \mathbf{z} \rangle = r_i \quad \forall i \in S\}$.

Therefore, we have:

$$\begin{aligned} \text{MLE}(\langle \mathbf{x}, \mathbf{z} \rangle \mid \langle \mathbf{d}_i^*, \mathbf{z} \rangle = r_i \quad \forall i \in S) &= \text{MLE}(\langle \Pi_{\mathbf{D}_S^*} \mathbf{x}, \mathbf{z} \rangle \mid \langle \mathbf{d}_i^*, \mathbf{z} \rangle = r_i \quad \forall i \in S) \\ &\quad + \text{MLE}(\langle \Pi_{\mathbf{D}_S^*}^\perp \mathbf{x}, \mathbf{z} \rangle). \end{aligned}$$

It turns out that the first term on the right hand side is deterministic:

$$\begin{aligned} \langle \Pi_{\mathbf{D}_S^*} \mathbf{x}, \mathbf{z} \rangle &= \langle \mathbf{D}_S^* \mathbf{D}_S^{*\top} \mathbf{x}, \mathbf{z} \rangle \\ &= \langle \sum_{i \in S} \langle \mathbf{d}_i^*, \mathbf{x} \rangle \mathbf{d}_i^*, \mathbf{z} \rangle \\ &= \sum_{i \in S} \langle \mathbf{d}_i^*, \mathbf{x} \rangle \langle \mathbf{d}_i^*, \mathbf{z} \rangle \\ &= \sum_{i \in S} \langle \mathbf{d}_i^*, \mathbf{x} \rangle r_i. \end{aligned}$$

For the last equality we used $q_i = r_i$. For the second term, we have:

$$\begin{aligned} \text{MLE}(\langle \Pi_{\mathbf{D}_S^*}^\perp \mathbf{x}, \mathbf{z} \rangle) &= \text{MLE}(\langle \Pi_{\mathbf{D}_S^*}^\perp \mathbf{x}, \Pi_{\mathbf{D}_S^*}^\perp \mathbf{z} \rangle) \\ &\stackrel{(a)}{=} 0. \end{aligned}$$

The equality (a) is due to the observation that $\Pi_{\mathbf{D}_S^*}^\perp \mathbf{z}$ is indeed distributed as $\mathcal{N}(\mathbf{0}, \mathbf{I}_{(d-k) \times (d-k)})$ after a simple dimensionality reduction [6]. The resulting random variable $\langle \Pi_{\mathbf{D}_S^*}^\perp \mathbf{x}, \Pi_{\mathbf{D}_S^*}^\perp \mathbf{z} \rangle$ is distributed as $\mathcal{N}(0, \|\Pi_{\mathbf{D}_S^*}^\perp \mathbf{x}\|_2^2)$, leading to the equality (a). To sum up, we have

$$\text{MLE}(\langle \mathbf{x}, \mathbf{z} \rangle \mid \langle \mathbf{d}_i^*, \mathbf{z} \rangle = r_i \quad \forall i \in S) = \sum_{i \in S} \langle \mathbf{d}_i^*, \mathbf{x} \rangle r_i.$$

□

4.5.4.3 Proof of Lemma 9

Let us define

$$S_k = \text{argmax}_{T \subseteq [n], |T| \leq k} \sum_{i \in T} |\langle \mathbf{d}_i, \mathbf{x} \rangle|.$$

We will use induction to show that S_k corresponds to the indices selected by IP-OMP after k iterations. At the first iteration, according to [26], IP-OMP simply selects the index $i = \arg \max_{j \in [n]} |\langle \mathbf{d}_j, \mathbf{x} \rangle| / \|\mathbf{d}_j\|_2 \|\mathbf{x}\|_2 = \arg \max_{j \in [n]} |\langle \mathbf{d}_j, \mathbf{x} \rangle|$, where the second equality is due

to $\|\mathbf{d}_j\|_2 = 1$ for all $j \in [n]$. As a result we have $S_1 = \{i\}$ and the base case is established.

Suppose that the statement holds for $S_{k-1} = \arg \max_{T \subseteq [n], |T| \leq k-1} \sum_{i \in T} |\langle \mathbf{d}_i, \mathbf{x} \rangle|$. Let $\mathbf{D}_{S_{k-1}}$ be the submatrix of $\mathbf{D}$ which consists of columns indexed by S_{k-1} . Then IP-OMP will select

$$i = \arg \max_{j \in [n], j \notin S_{k-1}} \frac{|\langle \Pi_{\mathbf{D}_{S_{k-1}}}^\perp \mathbf{d}_j, \Pi_{\mathbf{D}_{S_{k-1}}}^\perp \mathbf{x} \rangle|}{\|\Pi_{\mathbf{D}_{S_{k-1}}}^\perp \mathbf{d}_j\|_2 \|\Pi_{\mathbf{D}_{S_{k-1}}}^\perp \mathbf{x}\|_2}.$$

As $\mathbf{D}$ is column-orthogonal, we have $\Pi_{\mathbf{D}_{S_{k-1}}}^\perp \mathbf{d}_j = \mathbf{d}_j$ and $\Pi_{\mathbf{D}_{S_{k-1}}} \mathbf{d}_j = \mathbf{0} \forall j \notin S_{k-1}$. Therefore, for $\forall j \notin S_{k-1}$, we have

$$\begin{aligned} \langle \mathbf{d}_j, \mathbf{x} \rangle &= \langle \Pi_{\mathbf{D}_{S_{k-1}}}^\perp \mathbf{d}_j, \Pi_{\mathbf{D}_{S_{k-1}}}^\perp \mathbf{x} \rangle + \langle \Pi_{\mathbf{D}_{S_{k-1}}} \mathbf{d}_j, \Pi_{\mathbf{D}_{S_{k-1}}} \mathbf{x} \rangle \\ &= \langle \Pi_{\mathbf{D}_{S_{k-1}}}^\perp \mathbf{d}_j, \Pi_{\mathbf{D}_{S_{k-1}}}^\perp \mathbf{x} \rangle. \end{aligned}$$

As a result, we can rewrite the selection rule of IP-OMP as

$$\begin{aligned} i &= \arg \max_{j \in [n], j \notin S_{k-1}} \frac{|\langle \Pi_{\mathbf{D}_{S_{k-1}}}^\perp \mathbf{d}_j, \Pi_{\mathbf{D}_{S_{k-1}}}^\perp \mathbf{x} \rangle|}{\|\Pi_{\mathbf{D}_{S_{k-1}}}^\perp \mathbf{d}_j\|_2 \|\Pi_{\mathbf{D}_{S_{k-1}}}^\perp \mathbf{x}\|_2} \\ &= \arg \max_{j \in [n], j \notin S_{k-1}} \frac{|\langle \mathbf{d}_j, \mathbf{x} \rangle|}{\|\mathbf{d}_j\|_2 \|\mathbf{x}\|_2} \\ &= \arg \max_{j \in [n], j \notin S_{k-1}} |\langle \mathbf{d}_j, \mathbf{x} \rangle|. \end{aligned}$$

The last equality is due to $\|\mathbf{d}_j\|_2 = 1$ and the fact that $\|\mathbf{x}\|_2$ is the same for different j . This implies that, at iteration k , IP-OMP will select the index $j \notin S_{k-1}$ with the largest $|\langle \mathbf{d}_j, \mathbf{x} \rangle|$, which completes the proof of our induction.

Finally, when $\mathbf{D} = \mathbf{D}^*$, one can verify that $\langle \mathbf{d}_i, \mathbf{x} \rangle = \langle \mathbf{d}_i^*, \mathbf{D}^* \beta \rangle = 0$ for $i \notin S^*$. This implies that $S_k = S^*$. $\square$

4.5.4.4 Proof of Lemma 10

To prove $\tilde{\mathbf{D}}$ is column-orthogonal, we first observe that, for $i \leq k$, we have $\|\tilde{\mathbf{d}}_i\|_2^2 = \cos^2 \theta + \sin^2 \theta = 1$. Moreover, for any pair $i < j \leq k/2$, we have

$$\langle \tilde{\mathbf{d}}_i, \tilde{\mathbf{d}}_j \rangle = \langle \cos \theta \mathbf{d}_i^* + \sin \theta \mathbf{d}_{i+k/2}^*, \cos \theta \mathbf{d}_j^* + \sin \theta \mathbf{d}_{j+k/2}^* \rangle = 0,$$

which follows since $\langle \mathbf{d}_i^*, \mathbf{d}_j^* \rangle = \langle \mathbf{d}_{i+k/2}^*, \mathbf{d}_j^* \rangle = \langle \mathbf{d}_i^*, \mathbf{d}_{j+k/2}^* \rangle = \langle \mathbf{d}_{i+k/2}^*, \mathbf{d}_{j+k/2}^* \rangle = 0$. A similar argument can be made for any $i < j \leq k$ and $j \neq i + k/2$. For any $i \leq k$ and $j = i + k/2$,

we have

$$\langle \tilde{\mathbf{d}}_i, \tilde{\mathbf{d}}_j \rangle = \langle \cos \theta \mathbf{d}_i^* + \sin \theta \mathbf{d}_{i+k/2}^*, -\sin \theta \mathbf{d}_i^* + \cos \theta \mathbf{d}_{i+k/2}^* \rangle = -\sin \theta \cos \theta + \sin \theta \cos \theta = 0.$$

Finally, for any $j > i > k$ we trivially have $\langle \tilde{\mathbf{d}}_i, \tilde{\mathbf{d}}_j \rangle = 0$. This completes the proof of column-orthogonality of $\tilde{\mathbf{D}}$.

To prove the second statement, note that $\|\tilde{\mathbf{d}}_i - \mathbf{d}_i^*\|_2 = 0$ for every $i > k$. Moreover, for every $i \leq k$, we have

$$\|\tilde{\mathbf{d}}_i - \mathbf{d}_i^*\|_2 = \|(\cos \theta - 1)\mathbf{d}_i^* + \sin \theta \mathbf{d}_{i+k}^*\|_2 = \sqrt{(\cos \theta - 1)^2 + \sin^2 \theta} = \sqrt{2 - 2 \cos \theta} = 2 \sin \frac{\theta}{2}.$$

This completes the proof. $\square$

4.5.4.5 Proof of Lemma 11

The proof strategy we adopt here is similar with the proof of Lemma 3.1 in [75]. However, we consider a different generative model in this paper, so we present the full proof for the purpose of completeness. We will show that for $i \in S^*$, $|\langle \mathbf{d}_i, \mathbf{x} \rangle| > \gamma/2$ and for $i \notin S^*$, $|\langle \mathbf{d}_i, \mathbf{x} \rangle| < \gamma/2$, which will immediately result in $S = \operatorname{argmax}_{T \subseteq [n], |T| \leq k} \sum_{i \in T} |\langle \mathbf{d}_i, \mathbf{x} \rangle| = S^*$, which proves Lemma 11.

We decompose $\langle \mathbf{d}_i, \mathbf{x} \rangle$ as:

$$\begin{aligned} \langle \mathbf{d}_i, \mathbf{x} \rangle &= \langle \mathbf{d}_i, \sum_{j \in S^*} \mathbf{d}_j^* \beta_j \rangle \\ &= \underbrace{\langle \mathbf{d}_i, \mathbf{d}_i^* \rangle \beta_i}_{:= \mathcal{A}_i} + \underbrace{\sum_{j \neq i, j \in S^*} \langle \mathbf{d}_i, \mathbf{d}_j^* \rangle \beta_j}_{:= \mathcal{B}_i} \end{aligned} \quad (4.19)$$

For $\mathcal{A}_i$, we have:

$$\mathcal{A}_i = \langle \mathbf{d}_i, \mathbf{d}_i^* \rangle \beta_i = \langle \mathbf{d}_i^*, \mathbf{d}_i^* \rangle \beta_i + \langle \mathbf{d}_i - \mathbf{d}_i^*, \mathbf{d}_i^* \rangle \beta_i = \beta_i + \langle \mathbf{d}_i - \mathbf{d}_i^*, \mathbf{d}_i^* \rangle \beta_i.$$

When $i \in S^*$, we have

$$\begin{aligned} |\beta_i + \langle \mathbf{d}_i - \mathbf{d}_i^*, \mathbf{d}_i^* \rangle \beta_i| &\geq (1 - \|\mathbf{d}_i - \mathbf{d}_i^*\|_2) |\beta_i| \\ &\geq (1 - \|\mathbf{D} - \mathbf{D}^*\|_{1,2}) |\beta_i| \\ &\geq (1 - 2\rho) |\beta_i|. \end{aligned}$$

Given that $2\rho < \frac{\gamma}{4\sqrt{k}\Gamma} \leq \frac{1}{4}$, we can conclude

$$|\mathcal{A}_i| \begin{cases} \geq \frac{3\gamma}{4} & \text{if } i \in S^* \\ = 0 & \text{if } i \notin S^* \end{cases}, \quad (4.20)$$

given that $\beta_i = 0$ when $i \notin S^*$. For $\mathcal{B}_i$, we have:

$$\begin{aligned} |\mathcal{B}_i| &= \left| \sum_{j \neq i, j \in S^*} \langle \mathbf{d}_i, \mathbf{d}_j^* \rangle \beta_j \right| \\ &= \left| \sum_{j \neq i, j \in S^*} \langle \mathbf{d}_i^* + (\mathbf{d}_i - \mathbf{d}_i^*), \mathbf{d}_j^* \rangle \beta_j \right| \\ &= \left| \sum_{j \neq i, j \in S^*} \langle \mathbf{d}_i - \mathbf{d}_i^*, \mathbf{d}_j^* \rangle \beta_j \right| \\ &\leq \sum_{j \neq i, j \in S^*} |\langle \mathbf{d}_i - \mathbf{d}_i^*, \mathbf{d}_j^* \rangle| |\beta_j| \\ &\leq \Gamma \|\mathbf{d}_i - \mathbf{d}_i^*\|_1 \\ &\leq \sqrt{k}\Gamma \|\mathbf{d}_i - \mathbf{d}_i^*\|_2 \\ &\leq 2\sqrt{k}\Gamma\rho \\ &\leq \gamma/4 \end{aligned} \quad (4.21)$$

Combining Equation (4.20) and Equation (4.21), we have that for all i :

$$|\langle \mathbf{d}_i, \mathbf{x} \rangle| = |\mathcal{A}_i + \mathcal{B}_i| \begin{cases} > \frac{3\gamma}{4} - \frac{\gamma}{4} = \frac{\gamma}{2} & \text{if } i \in S^* \\ < \frac{\gamma}{4} & \text{if } i \notin S^* \end{cases}, \quad (4.22)$$

which proves the exact support recovery.

4.5.4.6 Proof of Lemma 12

It is easy to see that $\|\mathbf{c} - \mathbf{a}\|_2 \geq \|\mathbf{b} - \mathbf{a}\|_2$ is equivalent to $|c - a| \geq |b - a|$.

If $c > 0$, by $bc > 0$ we must have $b > 0$. Then $|c| \geq |b| \geq a$ becomes $c \geq b \geq a$, which gives $|c - a| \geq |b - a|$.

If $c < 0$, by $bc > 0$ we must have $b < 0$. Then $|c| \geq |b| \geq a$ becomes $a \geq b \geq c$, which gives $|c - a| \geq |b - a|$ as well.

4.5.4.7 Proof of Lemma 13

The proof of Lemma 13 consists of two steps:

1. We first use classic concentration inequalities from covariance estimation literature to bound σ_d and σ_1 given $|Q_i|$.
2. Then we will show that when m is large enough, with high probability, we have $|Q_i|$ bounded above and below.

Combining these two steps and taking the union bound will complete the proof of Lemma 13.

For the first step, we first notice that:

$$\begin{aligned}\sigma_d \left(\sum_{h \in Q_i} \mathbf{x}^h \mathbf{x}^{h\top} \right) &= \sigma_d \left(\sum_{h \in Q_i} \mathbf{D}^* \beta^h \beta^{h\top} \mathbf{D}^{*\top} \right) \\ &= \sigma_d \left(\sum_{h \in Q_i} \beta^h \beta^{h\top} \right)\end{aligned}$$

The last equation is due to the fact that $\mathbf{D}^*$ is a full-rank orthogonal matrix. Consider the vector β^h for $h \in Q_i$. According to our generative model and the definition of Q_i , we have $\beta_i^h \neq 0$, and its support outside i is selected uniformly over all $(k-1)$ -element subsets of $[n] \setminus \{i\}$. Each nonzero entry of β^h has a zero mean, variance of σ^2 , and an absolute value bounded between γ and Γ . Therefore, its covariance matrix defined as $\Sigma := \mathbf{E} \left[\beta^h \beta^{h\top} \right]$ is diagonal, with its (i, i) -th entry corresponding to σ^2 , and its (j, j) -th entry (with $j \neq i$) corresponding to $\frac{(k-1)\sigma^2}{n}$. We claim that when $|Q_i|$ is sufficiently large, $\sum_{h \in Q_i} \mathbf{x}^h \mathbf{x}^{h\top}$ will concentrate around $|Q_i| \Sigma$. Specifically, we will use the following well-established result:

Theorem 14 ([128]). *Let $\mathbf{w}$ be a zero-mean sub-Gaussian random vector in $\mathbb{R}^d$ with covariance matrix Σ , such that*

$$\|\langle \mathbf{w}, \mathbf{q} \rangle\|_{\psi_2} \leq K \left(\mathbf{E}[\langle \mathbf{w}, \mathbf{q} \rangle^2] \right)^{1/2} \quad \text{for any } \mathbf{q} \in \mathbb{R}^d,$$

for some $K \geq 1$. Here $\|\cdot\|_{\psi_2}$ denotes the sub-Gaussian norm of a random variable. Let $\mathbf{W} \in \mathbb{R}^{d \times \hat{m}}$ be a matrix whose columns have identical and independent distribution as $\mathbf{w}$. Then, for any $u \geq 0$ and with probability at least $1 - 2 \exp(-u)$, we have

$$\left\| \frac{1}{\hat{m}} \mathbf{W} \mathbf{W}^\top - \Sigma \right\|_2 \leq CK^2 \left(\sqrt{\frac{d+u}{\hat{m}}} + \frac{d+u}{\hat{m}} \right) \|\Sigma\|_2$$

for some universal constant C .

To invoke Theorem 14, we note that β^h is indeed zero-mean and sub-Gaussian. Therefore, upon setting $\mathbf{w} = \beta^h$, we notice that for any unit-norm $\mathbf{q}$:

$$\|\langle \beta^h, \mathbf{q} \rangle\|_{\psi_2}^2 \leq \sum_{j=1}^d \mathbf{q}_j^2 \|\beta_j^h\|_{\psi_2}^2 \leq C_0 \sigma^2,$$

for some universal constant C_0 . We also have

$$\mathbf{E} \left[\langle \beta^h, \mathbf{q} \rangle^2 \right] = \mathbf{q}^\top \Sigma \mathbf{q} \geq \frac{(k-1)\sigma^2}{n}.$$

By setting $K^2 = \frac{C_0 n}{k-1}$, we can invoke Theorem 14 with $u = n$ and conclude that, with probability at least $1 - 2 \exp\{-n\}$:

$$\sigma_d \left(\sum_{h \in Q_i} \beta^h \beta^{h\top} \right) \geq |Q_i| \sigma_d(\Sigma) - \frac{Cn\sigma}{k-1} \sqrt{2n|Q_i|} = \frac{|Q_i|(k-1)\sigma^2}{n} - \frac{Cn\sigma}{k-1} \sqrt{2n|Q_i|}, \quad (4.23)$$

$$\sigma_1 \left(\sum_{h \in Q_i} \beta^h \beta^{h\top} \right) \leq |Q_i| \sigma_1(\Sigma) + \frac{Cn\sigma}{k-1} \sqrt{2n|Q_i|} = |Q_i| \sigma^2 + \frac{Cn\sigma}{k-1} \sqrt{2n|Q_i|}, \quad (4.24)$$

for some universal constant C , which concludes the first step. Here we recall that $d = n$ in the full-rank and orthogonal setting.

For the second step, consider one specific i from $[n]$. The random variable $|Q_i|$ follows a binomial distribution with m trials and $\frac{k}{n}$ success rate. We next recall the Chernoff bound:

Theorem 15 ([87]). *Let $X_1, X_2, \dots, X_N$ be independent Bernoulli random variables with $\mathbf{P}(X_i = 1) = p_i$, and let*

$$X = \sum_{i=1}^N X_i \quad \text{and} \quad \mu = \mathbb{E}[X] = \sum_{i=1}^N p_i.$$

Then for any $0 < \delta < 1$, we have

$$\mathbf{P}(X \leq (1 - \delta)\mu) \leq \exp\left\{-\frac{\delta^2 \mu}{2}\right\},$$

and

$$\mathbf{P}(X \geq (1 + \delta)\mu) \leq \exp\left\{-\frac{\delta^2 \mu}{2}\right\}.$$

We can then invoke Theorem 15 with $\delta = \frac{1}{2}$ to obtain

$$\frac{km}{2n} \leq |Q_i| \leq \frac{2km}{n}, \quad (4.25)$$

with probability at least $1 - 2 \exp \left\{ -\frac{km}{8n} \right\}$. Finally, by combining Equation (4.23) and Equation (4.25), we have

$$\begin{aligned} \frac{1}{m} \sigma_d \left(\sum_{h \in Q_i} \beta^h \beta^{h^\top} \right) &\geq \frac{|Q_i|(k-1)\sigma^2}{nm} - \frac{Cn\sigma}{(k-1)m} \sqrt{2n|Q_i|} \\ &\geq \frac{k(k-1)\sigma^2}{2n^2} - \frac{Cn\sigma}{(k-1)} \sqrt{\frac{4k}{m}}. \end{aligned} \quad (4.26)$$

When $m \geq \frac{64C^2n^6}{\sigma^2k(k-1)^4}$, we have

$$\frac{Cn\sigma}{(k-1)} \sqrt{\frac{4k}{m}} \leq \frac{k(k-1)\sigma^2}{4n^2},$$

which reduces Equation (4.26) to:

$$\frac{1}{m} \sigma_d \left(\sum_{h \in Q_i} \beta^h \beta^{h^\top} \right) \geq \frac{k(k-1)\sigma^2}{2n^2} - \frac{k(k-1)\sigma^2}{4n^2} = \frac{k(k-1)\sigma^2}{4n^2}.$$

With a similar argument, we can combine Equation (4.24) and Equation (4.25) to get:

$$\frac{1}{m} \sigma_1 \left(\sum_{h \in Q_i} \beta^h \beta^{h^\top} \right) \leq |Q_i|\sigma^2 + \frac{Cn\sigma}{k-1} \sqrt{2n|Q_i|} \leq \frac{4k\sigma^2}{n}.$$

Finally, we take the union bound for all $i \in [n]$, leading to the overall probability of $1 - 2n \exp\{-n\} - 2n \exp\left\{-\frac{km}{8n}\right\}$. This completes the proof. $\square$

4.6 Experiments on Generative Model

All experiments reported in this section were performed in Python 3.9 on a MacBook Pro (14-inch, 2021) equipped with an Apple M1 Pro chip.

We generate samples according to the described generative model in Assumption 6, we construct the input data $\mathbf{x}$ or $\{\mathbf{x}^h\}_{h=1}^m$ by sampling each non-zero entry from a uniform distribution over the interval $[\gamma, \Gamma]$. The matrix $\mathbf{D}^*$ is chosen to be a randomly generated orthonormal matrix, and we construct $\mathbf{D}_{\text{init}}$ by setting $\mathbf{D}_{\text{init}} = \mathbf{D}^* + \mathbf{E}$, where each column of $\mathbf{E}$ is uniformly drawn from the set $\{\mathbf{e} \mid \|\mathbf{e}\|_2 \leq \rho\}$. For the results shown in Figure 4.2, we set $d = 10$, $k = 5$, $\rho = 0.2$, $\gamma = 0.5$, and $\Gamma = 1$. The first column of Figure 4.2 illustrates the scenario corresponding to Theorem 12, in which only a single input feature $\mathbf{x}$ is available. Here, we choose $n = 8$ and $\eta = 10^{-2}$. The plot showing $\mathcal{L}(\tilde{f}_{\mathbf{D}^{(t)}})$ confirms

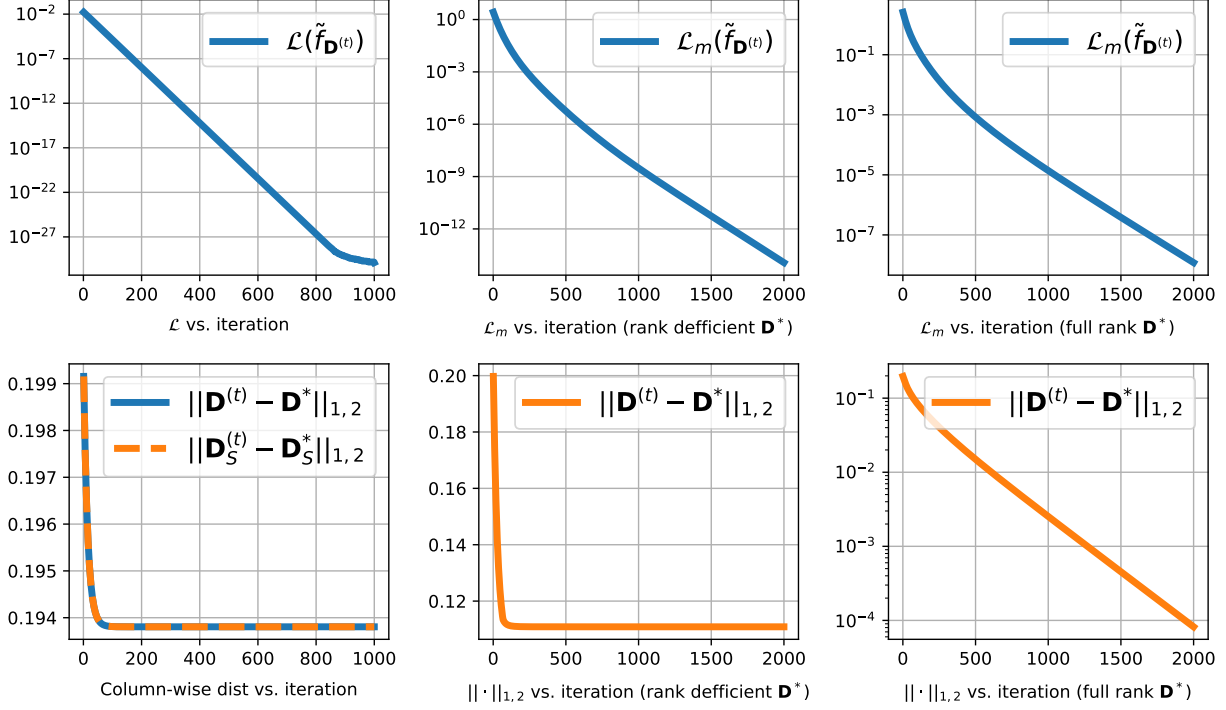


Figure 4.2: Results on synthetic dataset.

the linear convergence of the loss function to zero, in agreement with Theorem 12. However, as previously noted, this setting does not ensure exact recovery for each queried feature. Indeed, as shown in the right panel of the first row, $\|\mathbf{D}^{(t)} - \mathbf{D}^*\|_{1,2}$ does not converge to zero. Even if we only consider columns that are activated at each iteration, $\|\mathbf{D}_S^{(t)} - \mathbf{D}_S^*\|_{1,2}$ also remains nonzero, indicating that optimizing $\mathcal{L}$ with a single $\mathbf{x}$ is insufficient to recover the ground-truth dictionary $\mathbf{D}^*$.

In the second column of Figure 4.2, we apply projected gradient descent to minimize $\mathcal{L}_m$, as defined in Equation (4.5), under the assumption that $\mathbf{D}^*$ is rank-deficient. Specifically, we set $n = 8 < d$ and $\eta = 10^{-1}$. Evidently, $\|\mathbf{D}^{(t)} - \mathbf{D}^*\|_{1,2}$ does not converge to zero because the component of $\mathbf{D}^{(t)} - \mathbf{D}^*$ orthogonal to the column space of $\mathbf{D}^*$ remains unaffected throughout the iterations. In contrast, the third column of Figure 4.2 corresponds to the setting $n = d$, which guarantees the full-rankness of $\mathbf{D}^*$ is full rank. In this case, $\|\mathbf{D}_S^{(t)} - \mathbf{D}_S^*\|_{1,2}$ converges to zero, thus supporting the conclusion of Theorem 13.

4.7 Application: Interpretable Image Classification

In this section, we showcase the performance of our proposed CCR framework from Algorithm 13 on interpretable image classification task. The Python implementation of algorithm can be found here: github.com/lianggeyuleo/CCR.git.

The image classification setting differs slightly from the setting provided in Definition 3. Specifically, we are given a dataset $\{\mathbf{a}^i, \mathbf{y}^i\}_{i=1}^m$ consisting of m image ($\mathbf{a}$)-label ($\mathbf{y}$) pairs, where the goal is to predict the correct label for a given image. Here, $\mathbf{y}$ represents a one-hot label vector. Additionally, this setting provides access to a concept set $\{\mathbf{c}_i\}_{i=1}^n$, which consists of n key concepts that can aid in classifying an image. These concepts are typically textual; for example, in the case of animal images, the concept set might include descriptors like “stocky body” or “black stripes”.

Algorithm 14 CCR for Interpretable Image Classification

- 1: **Input:** Dataset of image-label pairs $\{\mathbf{a}^i, \mathbf{y}^i\}_{i=1}^m$, concept set $\{\mathbf{c}_i\}_{i=1}^n$.
 - 2: Use CLIP to embed the images $\{\mathbf{a}^i\}_{i=1}^m$ into input features $\{\mathbf{x}^i\}_{i=1}^m$, and the concepts $\{\mathbf{c}_i\}_{i=1}^n$ into query features $\{\mathbf{d}_i\}_{i=1}^n$.
 - 3: Initialize $\mathbf{D}^{(0)}$ via concept dispersion on $\{\mathbf{d}_i\}_{i=1}^n$.
 - 4: Initialize $\mathbf{s}_i^{(0)} = \text{HT}_\lambda \left(\mathbf{D}^{(0)\top} \mathbf{x}^i \right)$ for $i = 1, \dots, m$.
 - 5: Initialize $\mathbf{L}^{(0)}$ with random values.
 - 6: **while** $t = 1, 2, \dots, T$ **do**
 - 7: Calculate $\mathcal{L}_m = \sum_i \text{CE} \left(\mathbf{L} \mathbf{s}_i^{(t-1)}, \mathbf{y}^i \right)$.
 - 8: Update $\mathbf{D}^{(t)} = \mathbf{D}^{(t-1)} - \eta_{\mathbf{D}} \partial \mathcal{L} / \partial \mathbf{D}^{(t-1)}$.
 - 9: Normalize and project $\mathbf{D}^{(t)}$.
 - 10: Update $\mathbf{L}^{(t)} = \mathbf{L}^{(t-1)} - \eta_{\mathbf{L}} \partial \mathcal{L} / \partial \mathbf{L}^{(t-1)}$.
 - 11: Update $\mathbf{s}_i^{(t)} = \text{HT}_\lambda \left(\mathbf{D}^{(t)\top} \mathbf{x}^i \right)$ for $i = 1, \dots, m$.
 - 12: **end while**
 - 13: **Return** $\mathbf{D}^{(T)}$, $\mathbf{L}^{(T)}$, and $\{\mathbf{s}_i^{(T)}\}_{i=1}^m$.
-

We formally introduce our algorithm in Algorithm 14. Prior to comparing our approach with other interpretable AI methods, we provide a detailed explanation of the key components in Algorithm 14.

CLIP embedding. We employ the multi-modal model CLIP [94]—a recently introduced large Vision-Language Model—to embed both images and concepts into the same latent space.

Concept dispersion. After obtaining the CLIP embeddings, the query features tend to cluster too closely, making the query feature matrix $\mathbf{D}^{(0)}$ far from column-orthogonal. To address this, we introduce a *concept dispersion* procedure (see Algorithm 15 in the following

section). This heuristic approach enhances the mutual orthogonality of query features by increasing their relative angles, thereby improving the orthogonality of $\mathbf{D}^{(0)}$. Notably, this is achieved while preserving interpretability by maintaining the relative positions of features in the embedded space. Further details on the dispersion step are provided in Section 4.7.1.

Hard-thresholding. In Equation (4.4), $\tilde{f}$ is computed in two steps: (1) constructing the sparse code $\mathbf{s}$ by keeping only the entries with the top- k absolute values of $\mathbf{D}^\top \mathbf{x}$; and (2) setting $\tilde{f}$ as the inner product of $\mathbf{s}$ with $\mathbf{r} = [r_1, \dots, r_i]^\top \in \mathbb{R}^n$. Here, we replace the top- k selection with the entry-wise hard-thresholding operator:

$$\text{HT}_\lambda(x) = \begin{cases} x, & \text{if } |x| \geq \lambda, \\ 0, & \text{if } |x| < \lambda. \end{cases}$$

This approach allows for parallelization and reduced computational cost, although the exact sparsity level k can no longer be directly specified; instead, k is determined implicitly by λ . Nonetheless, adjusting λ provides similar control over sparsity.

Linear Layer. To compute $\tilde{f}_{\mathbf{D}}$ as defined in Equation (4.4), access to $\mathbf{r}$ is required, which is unavailable in the image classification setting. To address this, we train a linear classifier that takes the sparse code $\mathbf{s}$ as input and outputs a weight vector over the possible labels. The label with the highest weight is selected as the prediction. The loss function is defined as the cross-entropy between the linear layer’s output and the true label $\mathbf{y}$.

Embedding normalization and projection. After each iteration, we perform a projection step similar to Section 4.4 for each updated concept embedding. Additionally, we normalize each embedding to ensure that it remains on the unit ball (see Algorithm 16 in Section 4.7.1).

4.7.1 More Details on Algorithm 14

Here we provide a detailed description for the **concept dispersion** step and the **embedding normalization and projection** step in Algorithm 14. In Algorithm 15, we present the pseudo-code for concept dispersion. The algorithm first calculates the mean concept embedding for the given $\{\mathbf{d}_i\}_{i=1}^n$ and calculates the angle between each $\mathbf{d}$ and the mean. Then, Algorithm 15 increases these angles by a constant factor r .

Algorithm 16 presents the pseudo-code for the projection and normalization step in Algorithm 16.

Algorithm 15 Concept dispersion

- 1: **Input:** $\{\mathbf{d}_i\}_{i=1}^n$, dispersion factor r .
 - 2: Calculate the mean concept embedding $\bar{\mathbf{d}} = \sum_{i=1}^n \mathbf{d}_i / \|\sum_{i=1}^n \mathbf{d}_i\|_2$.
 - 3: **for** each $\mathbf{d}_i$ **do**
 - 4: Decompose $\mathbf{d}_i$ as $\mathbf{d}_i = \cos \alpha_i \bar{\mathbf{d}} + \sin \alpha_i \mathbf{e}_i$, where $\alpha_i \in [0, \pi/2]$, $\|\mathbf{e}_i\|_2 = 1$, and $\mathbf{e}_i \perp \bar{\mathbf{d}}$.
 - 5: Calculate the dispersed query feature $\mathbf{d}_i^{\text{new}} = \cos(r\alpha_i) \bar{\mathbf{d}} + \sin(r\alpha_i) \mathbf{e}_i$.
 - 6: Add $\mathbf{d}_i^{\text{new}}$ as the i th column of $\mathbf{D}^{\text{init}}$.
 - 7: **end for**
 - 8: **Return** $\mathbf{D}^{\text{init}}$.
-

Algorithm 16 Embedding normalization and projection

- 1: **Input:** current query feature matrix $\mathbf{D}$, initial query feature matrix $\mathbf{D}^{\text{init}}$, and radius ρ .
 - 2: **for** each query feature $\mathbf{d}_i$ in $\mathbf{D}$ **do**
 - 3: Normalize $\mathbf{d}_i$ as $\mathbf{d}_i = \mathbf{d}_i / \|\mathbf{d}_i\|_2$.
 - 4: **if** $\|\mathbf{d}_i - \mathbf{d}_i^{\text{init}}\|_2 \geq \rho$ **then**
 - 5: $\mathbf{d}_i = \arg \min_{\mathbf{d}: \|\mathbf{d} - \mathbf{d}_i^{\text{init}}\|_2 \leq \rho, \|\mathbf{d}\|_2 = 1} \|\mathbf{d} - \mathbf{d}_i\|_2$.
 - 6: **end if**
 - 7: **end for**
 - 8: **Return** $\mathbf{D}$.
-

4.7.2 Performance

We compare the performance of Algorithm 14 against two recently proposed explainable AI methods, CLIP-IP-OMP [26] and label-free CBM (lf-CBM) [88], as well as the CCR baseline without the concept refinement step, in the context of explainable image classification.

For the baseline version of CCR, we set $\eta_{\mathbf{D}} = 0$ at Step 8 of Algorithm 14, ensuring that the comparison isolates the effect of concept refinement. The two methods we compare against represent state-of-the-art explainable AI approaches, particularly in terms of scalability. Lf-CBM was the first CBM-type model to be scaled to datasets as large as ImageNet, while CLIP-IP-OMP further improves computational efficiency while maintaining competitive accuracy. For a detailed introduction to both methods, we refer to Section 4.2.

The evaluation is conducted across five image classification benchmarks: CIFAR-10, CIFAR-100 [64], ImageNet [31], CUB-200 [129], and Places365 [143].

To ensure a fair comparison, all methods use the same concept set, which is generated by GPT-3 [22]. A detailed description of the generation process can be found in [88]. For the CIFAR-10/100 and CUB-200 datasets, we tune CLIP-IP-OMP to match the average sparsity level of $\mathbf{s}$, also referred to as the *explanation length* or k , used in CCR. For ImageNet and

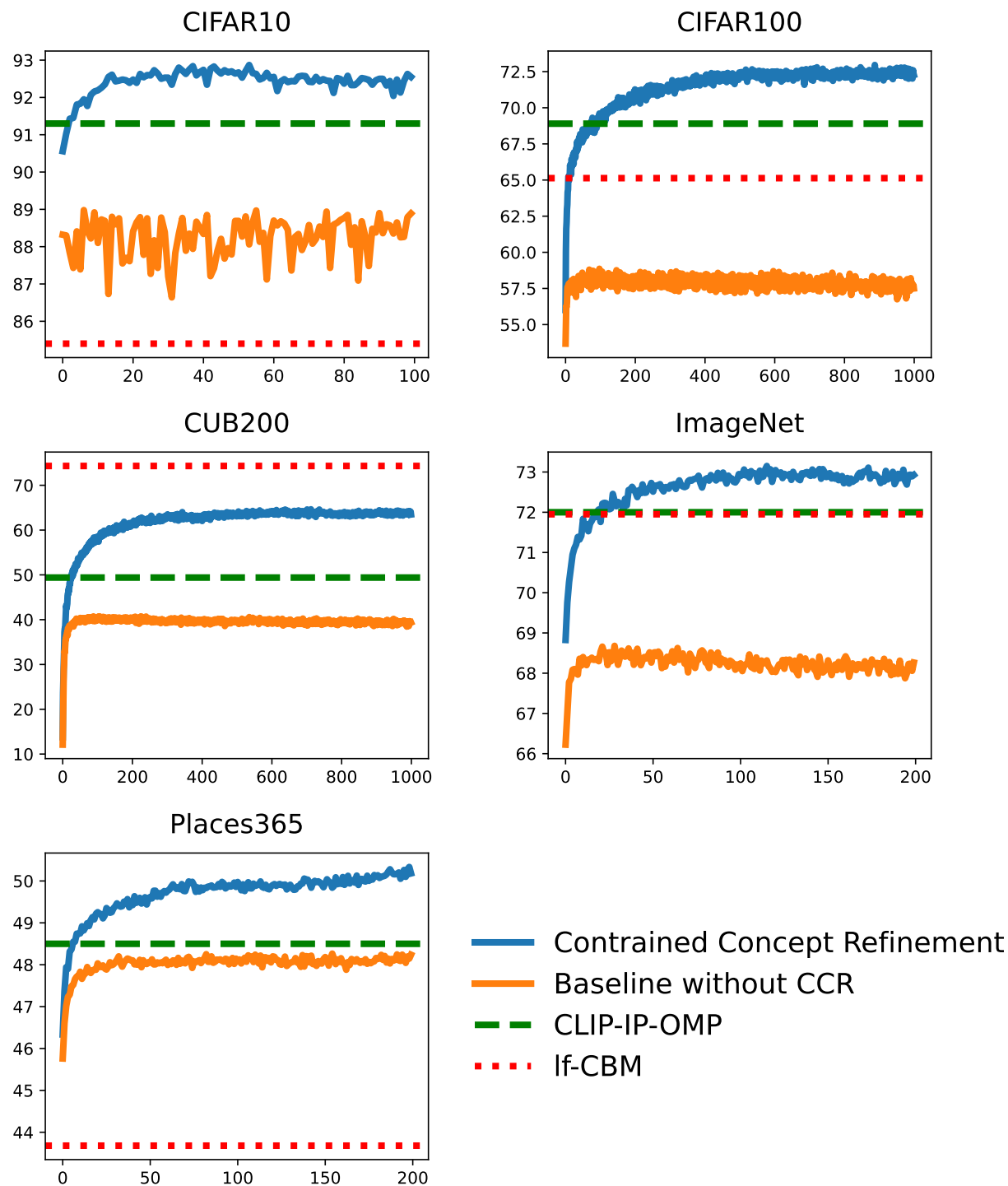


Figure 4.3: Prediction accuracy of CCR and its baseline across iterations, with the final test accuracy of CLIP-IP-OMP and lf-CBM indicated for reference.

Places365, we report the best accuracy achieved by CLIP-IP-OMP across all explanation lengths. Since Lf-CBM does not allow tuning of its explanation length, we directly report its accuracy. The constraint parameter ρ for CCR is fixed at 0.1 for all experiments.

As shown in Figure 4.3, CCR consistently outperforms its baseline, CLIP-IP-OMP, and lf-CBM across all benchmarks except CUB-200. This exception is expected, as lf-CBM was built on a ResNet-18 backbone specifically trained on CUB-200, whereas both CLIP-IP-OMP and CCR rely on CLIP as their encoder, which was trained on more diverse and general datasets.

Additionally, we report the average explanation length (AEL, k), average sparsity ratio (ASR, k/n), and average concept embedding deviation (ACED, $\frac{1}{n} \sum_{i=1}^n \|\mathbf{d}_i - \mathbf{d}_i^{(0)}\|_2$) of CCR in the following table:

	AEL	ASR	ACED
CIFAR10	11.09	8.66%	0.056
CIFAR100	19.67	2.39%	0.061
CUB200	27.52	13.2%	0.062
ImageNet	48.97	1.08%	0.097
Places365	44.43	2.01%	0.041

Table 4.1: Average explanation length (AEL), average sparsity ratio(ASR) and average concept embedding deviation (ACED) for CCR.

Thanks to the parallelization enabled by hard thresholding and the computational efficiency of backpropagation, CCR significantly reduces computational costs compared to CLIP-IP-OMP and lf-CBM, particularly when applied to large-scale datasets such as ImageNet (≈ 1.2 million images) and Places365 (≈ 1.8 million images). As reported in [26], training lf-CBM on ImageNet requires ≈ 50 hours on an NVIDIA RTX A5000 GPU. Under the same experimental conditions, CLIP-IP-OMP, with an average explanation length of ≈ 50 , incurs a computational cost of ≈ 40 hours. In our computational environment, using an NVIDIA Tesla V100 GPU, CLIP-IP-OMP remains comparably expensive, requiring ≈ 33 hours for $k = 50$. In contrast, CCR (with $k \approx 49$, as shown in Table 4.1) processes ImageNet in only ≈ 2 hours (corresponding to 200 iterations) while achieving even higher accuracy, demonstrating a substantial improvement in computational efficiency. To make CLIP-IP-OMP more computationally feasible, one can reduce k ; however, for $k = 10$, while the processing time drops to ≈ 6 hours, it comes at the cost of a significant accuracy decline to $\approx 63\%$.

4.7.3 Interpretability

To illustrate the interpretability of Algorithm 14, we follow the methodology outlined in [26] and present the most significant coefficients and weights from the algorithm on different samples from the Places365 dataset. As shown in Figure 4.4, we highlight the top 10 concepts in $\mathbf{s}$ with the highest values (left) along with their corresponding weights in the linear layer $\mathbf{L}$ for the predicted label (right). The selected concepts are semantically relevant to the image, and the linear layer effectively assigns substantial weight to key concepts such as “designated fairways and greens”, enabling accurate prediction while appropriately disregarding concepts that, although relevant, may be misleading, such as “a flag” or “a country”. For additional case studies on other datasets, we refer the reader to Section 4.7.4.

4.7.4 More Experiments on Interpretability

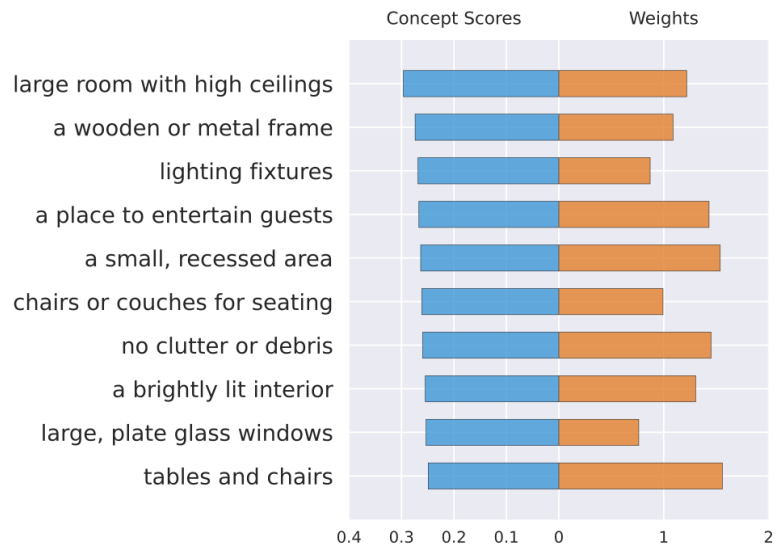
In this section, we expand on the discussion of CCR’s interpretability from Section 4.7.3 by conducting a comprehensive case study across all five datasets used to evaluate CCR. For each dataset, we select three representative image-label pairs and present the ten highest-ranking concept scores on the left, alongside their corresponding weights associated with the predicted label on the right. Based on our case study, we derive several key observations:

- **Semantic correlation.** In most cases, concepts with higher concept scores exhibit a strong semantic correlation with the input images. This finding substantiates the reliability of our algorithm as an interpretable AI model. A notable example is illustrated in Figure 4.12 and Figure 4.13, where the algorithm effectively identifies the primary distinguishing features between two visually similar bird species—their coloration—leading to an accurate classification.
- **Weight distribution.** In instances where misleading elements are present in an image, certain concepts unrelated to the ground truth label may receive high concept scores. Nevertheless, the linear layer appropriately assigns a small weight to such concepts. These weights encapsulate the algorithm’s interpretation of the label, thereby rendering it comprehensible to human users. Examples of such cases include “a beak” in Figure 4.6, “a plow” in Figure 4.8, “a seagull”/“a bird” in Figure 4.10, “a highlighter” in Figure 4.14, “iridescent” in Figure 4.15, and “abandoned buildings”/“a sunset” in Figure 4.17.
- **Debugging capability for incorrect predictions.** Inevitably, our framework makes incorrect predictions for some samples. A key application of explainable AI is to help



True label: dining room

Predicted label: dining room



True label: golf course

Predicted label: golf course

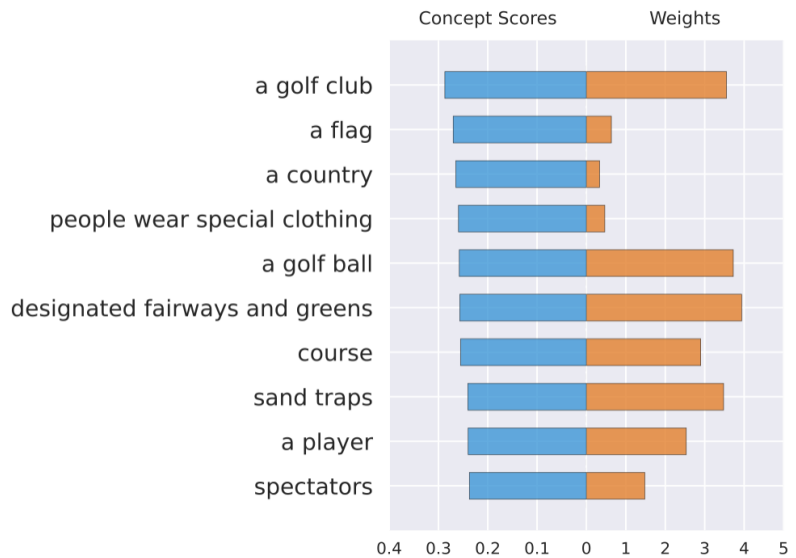


Figure 4.4: The first example illustrates a *simple case* where CCR successfully learns the correct concepts. The second example represents a *misleading case*, where the image contains concepts like “a flag” that, while relevant and visually apparent, could potentially mislead classification. However, CCR effectively extracts both useful and misleading information, assigning appropriate weights to ensure the correct prediction.

human experts understand the reasoning behind these errors. Our findings indicate that the proposed algorithm effectively captures interpretable misconceptions introduced by encoders. For instance, in Figure 4.18, concepts such as “a display case” and “exhibits” suggest that the primary misleading factor causing the model to predict “natural history museum” instead of the correct label “shopping mall” is the manner in which the store displays its goods. This insight is valuable for human intervention and model correction.

- **Reliance on the richness of the concept set.** The effectiveness of the algorithm is inherently influenced by the quality of the concept set, a phenomenon that aligns with the fundamental motivation behind explainable AI. As demonstrated in Figure 4.19, the concept set lacks distinctive features that differentiate between “bus interior” and “train interior”. Consequently, the model is unable to distinguish between these two labels.

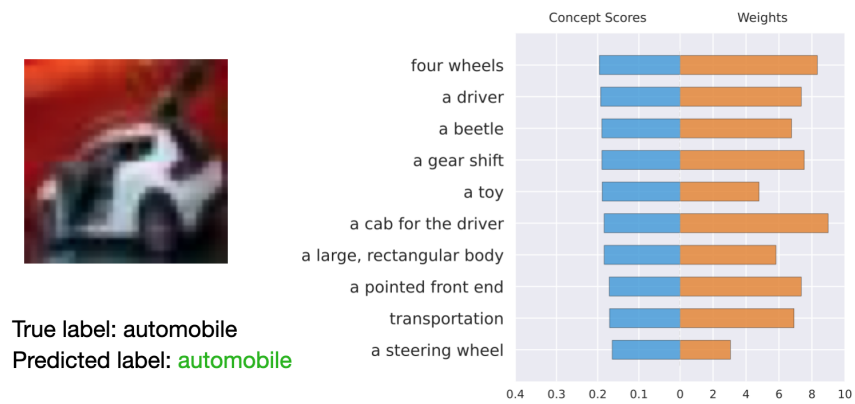


Figure 4.5: CIFAR-10 (a)

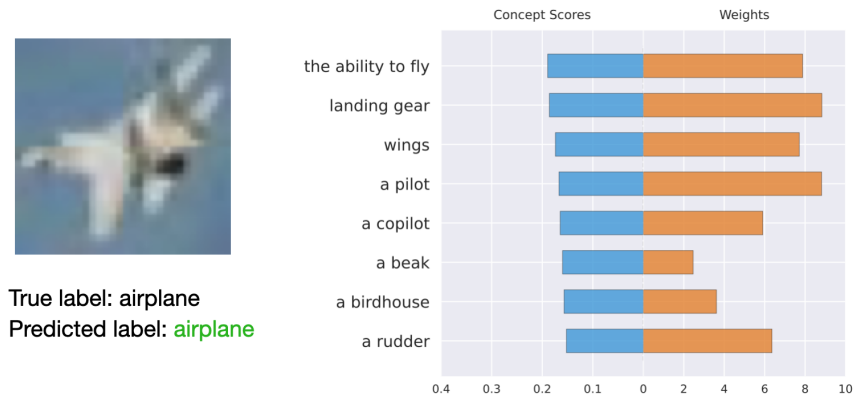


Figure 4.6: CIFAR-10 (b)

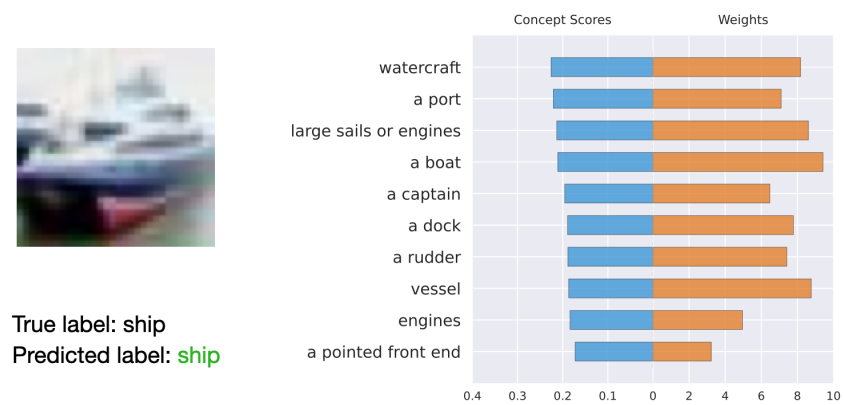


Figure 4.7: CIFAR-10 (c)



True label: pickup truck
 Predicted label: pickup truck

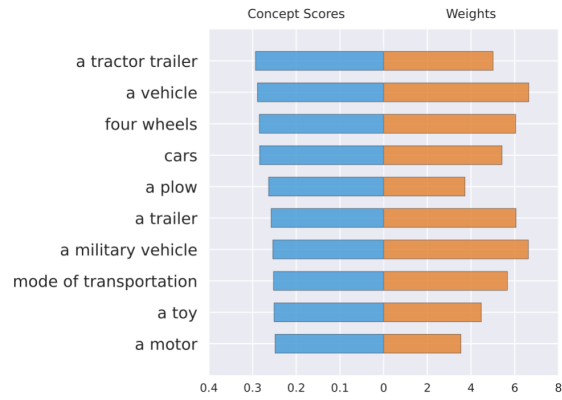
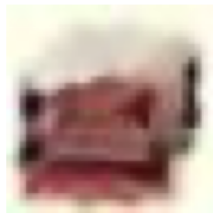


Figure 4.8: CIFAR-100 (a)



True label: couch
 Predicted label: couch

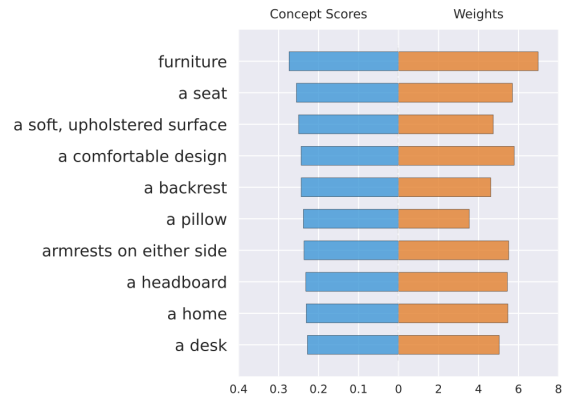
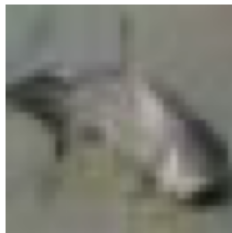


Figure 4.9: CIFAR-100 (b)



True label: shark
 Predicted label: shark

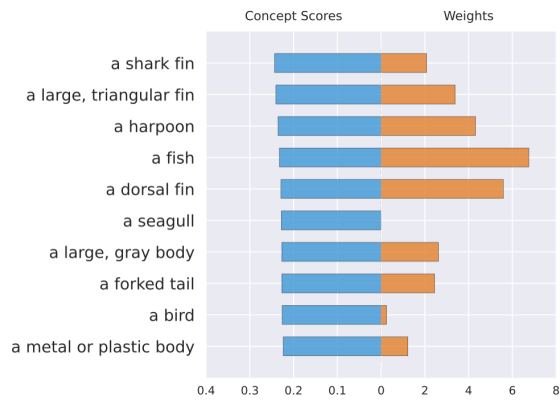


Figure 4.10: CIFAR-100 (c)

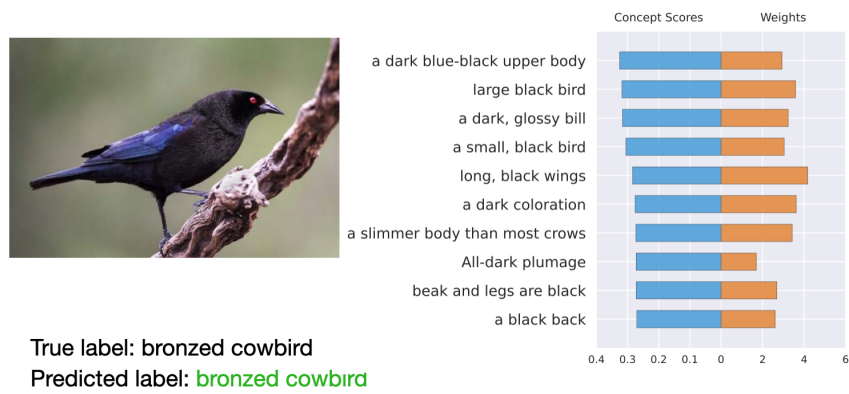


Figure 4.11: CUB-200 (a)

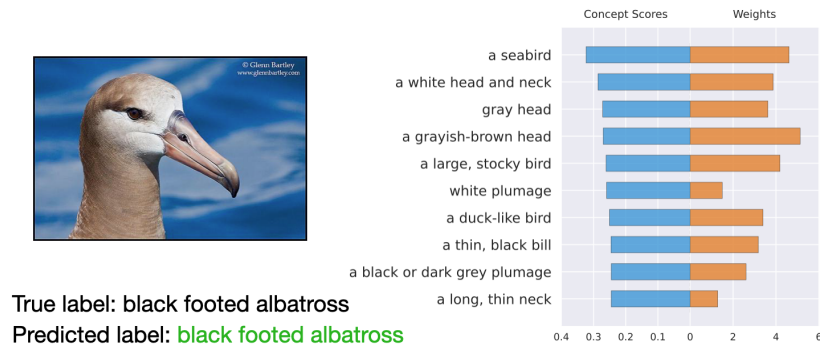


Figure 4.12: CUB-200 (b)

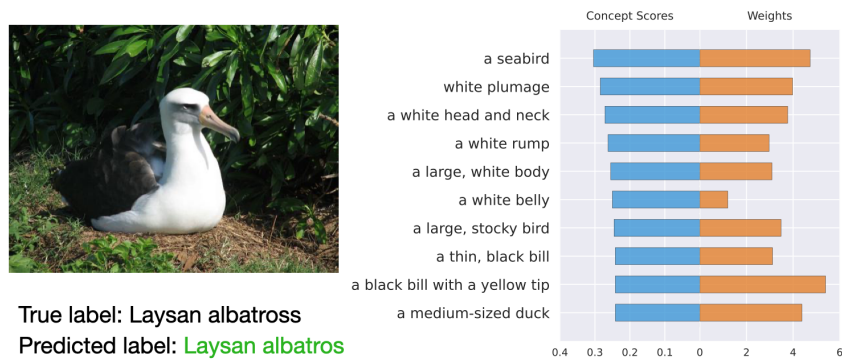
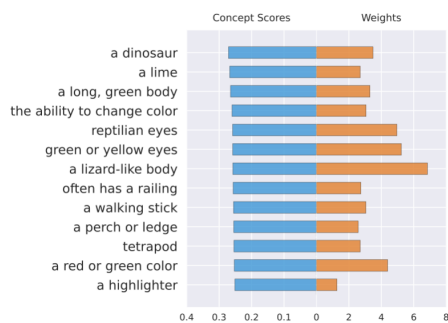


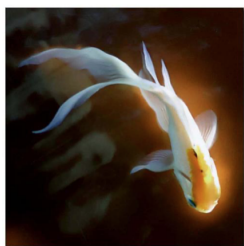
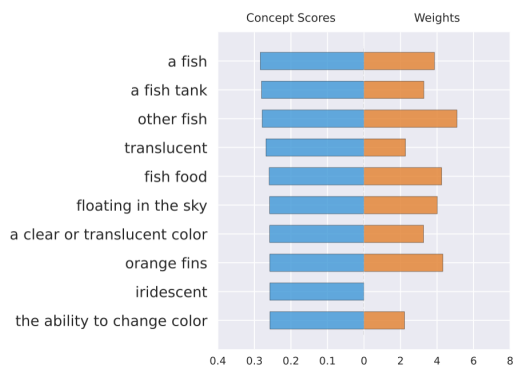
Figure 4.13: CUB-200 (c)



True label: African chameleon

Predicted label: African chameleon

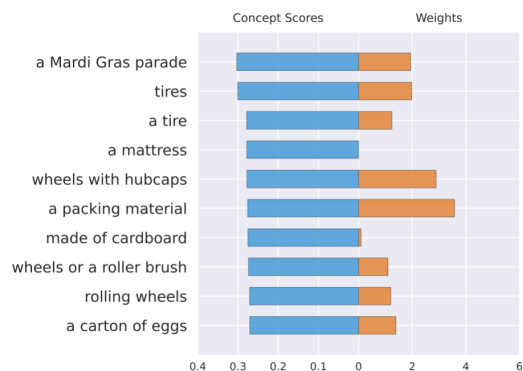
Figure 4.14: ImageNet (a)



True label: goldfish

Predicted label: goldfish

Figure 4.15: ImageNet (b)



True label: toilet tissue

Predicted label: car wheel

Figure 4.16: ImageNet (c)



True label: campus
Predicted label: **campus**

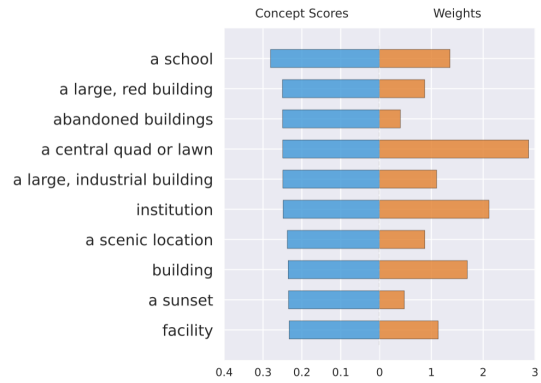


Figure 4.17: Places365 (a)



True label: shopping mall
Predicted label: **natural**
history museam

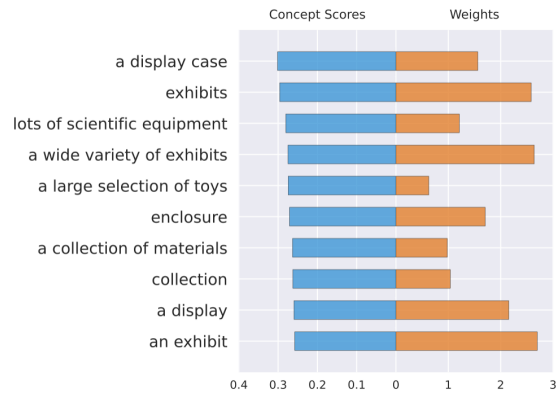


Figure 4.18: Places365 (b)



True label: bus interior
Predicted label: **train interior**

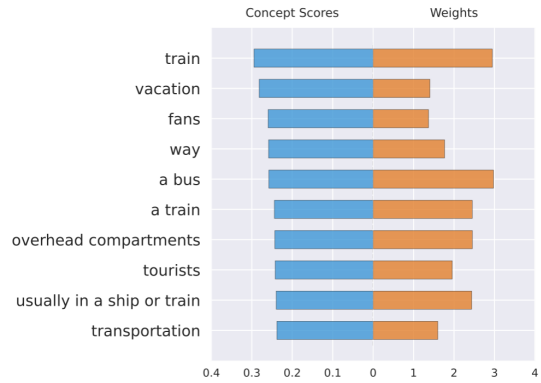


Figure 4.19: Places365 (c)

4.8 Concluding Remarks

This chapter addresses the long-standing trade-off between accuracy and interpretability in machine learning by proposing a novel framework that optimizes concept embeddings while maintaining interpretability constraints. Through theoretical analysis and empirical evaluations, we demonstrate that our approach achieves zero loss in a generative setting while progressively enhancing interpretability. Furthermore, our framework outperforms existing explainable methods in image classification tasks, achieving improved accuracy with significantly lower computational cost. These findings highlight the potential of principled concept optimization in advancing interpretable machine learning without compromising performance. Future work may explore extensions to broader model architectures and application domains.

BIBLIOGRAPHY

- [1] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *Advances in neural information processing systems*, 31, 2018.
- [2] Alekh Agarwal, Animashree Anandkumar, Prateek Jain, and Praneeth Netrapalli. Learning sparsely used overcomplete dictionaries via alternating minimization. *SIAM Journal on Optimization*, 26(4):2775–2799, 2016.
- [3] Alekh Agarwal, Animashree Anandkumar, and Praneeth Netrapalli. Exact recovery of sparsely used overcomplete dictionaries. *stat*, 1050:8–39, 2013.
- [4] Michal Aharon, Michael Elad, and Alfred Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311–4322, 2006.
- [5] Ravindra K Ahuja, Thomas L Magnanti, and James B Orlin. Network flows. 1988.
- [6] Theodore Wilbur Anderson, Theodore Wilbur Anderson, Theodore Wilbur Anderson, and Theodore Wilbur Anderson. *An introduction to multivariate statistical analysis*, volume 2. Wiley New York, 1958.
- [7] Sanjeev Arora, Rong Ge, Tengyu Ma, and Ankur Moitra. Simple, efficient, and neural algorithms for sparse coding. In *Conference on learning theory*, pages 113–149. PMLR, 2015.
- [8] Sanjeev Arora, Rong Ge, and Ankur Moitra. New algorithms for learning incoherent and overcomplete dictionaries. In *Conference on Learning Theory*, pages 779–806. PMLR, 2014.
- [9] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. How to explain individual classification decisions. *The Journal of Machine Learning Research*, 11:1803–1831, 2010.
- [10] Yu Bai, Qijia Jiang, and Ju Sun. Subgradient descent learns orthogonal dictionaries. *arXiv preprint arXiv:1810.10702*, 2018.
- [11] Chenglong Bao, Jian-Feng Cai, and Hui Ji. Fast sparsity-based orthogonal dictionary learning for image restoration. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3384–3391, 2013.

- [12] Chenglong Bao, Hui Ji, Yuhui Quan, and Zuowei Shen. L0 norm based dictionary learning by proximal methods with global convergence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3858–3865, 2014.
- [13] Boaz Barak, Jonathan A Kelner, and David 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*, pages 143–151, 2015.
- [14] Ana Barragán-Montero, Umair Javaid, Gilmer Valdés, Dan Nguyen, Paul Desbordes, Benoit Macq, Siri Willems, Liesbeth Vandewinckele, Mats Holmström, Fredrik Löfman, et al. Artificial intelligence and machine learning for medical imaging: A technology review. *Physica Medica*, 83:242–256, 2021.
- [15] Saddek Bensalem, Chih-Hong Cheng, Wei Huang, Xiaowei Huang, Changshun Wu, and Xingyu Zhao. What, indeed, is an achievable provable guarantee for learning-enabled safety-critical systems. In *International Conference on Bridging the Gap between AI and Reality*, pages 55–76. Springer, 2023.
- [16] Albert S Berahas, Jiahao Shi, Zihong Yi, and Baoyu Zhou. Accelerating stochastic sequential quadratic programming for equality constrained optimization using predictive variance reduction. *Computational Optimization and Applications*, 86(1):79–116, 2023.
- [17] Dimitri P Bertsekas. *Constrained optimization and Lagrange multiplier methods*. Academic press, 2014.
- [18] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. Analyzing federated learning through an adversarial lens. In *International Conference on Machine Learning*, pages 634–643. PMLR, 2019.
- [19] Rajendra Bhatia. Matrix factorizations and their perturbations. *Linear Algebra and its applications*, 197:245–276, 1994.
- [20] MJ Box. A new method of constrained optimization and a comparison with other methods. *The Computer Journal*, 8(1):42–52, 1965.
- [21] Stephen P Boyd and Lieven Vandenbergh. *Convex optimization*. Cambridge university press, 2004.
- [22] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020.
- [23] Maxime Bucher, Stéphane Herbin, and Frédéric Jurie. Semantic bottleneck for computer vision tasks. In *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part II 14*, pages 695–712. Springer, 2019.

- [24] Niladri Chatterji and Peter L Bartlett. Alternating minimization for dictionary learning with random initialization. *Advances in Neural Information Processing Systems*, 30, 2017.
- [25] Aditya Chattopadhyay, Kwan Ho Ryan Chan, Benjamin D Haeffele, Donald Geman, and René Vidal. Variational information pursuit for interpretable predictions. *arXiv preprint arXiv:2302.02876*, 2023.
- [26] Aditya Chattopadhyay, Ryan Pilgrim, and Rene Vidal. Information maximization perspective of orthogonal matching pursuit with applications to explainable ai. *Advances in Neural Information Processing Systems*, 36, 2024.
- [27] Aditya Chattopadhyay, Stewart Slocum, Benjamin D Haeffele, Rene Vidal, and Donald Geman. Interpretable by design: Learning predictors by composing interpretable queries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(6):7430–7443, 2022.
- [28] Townim F Chowdhury, Vu Minh Hieu Phan, Kewen Liao, Minh-Son To, Yutong Xie, Anton van den Hengel, Johan W Verjans, and Zhibin Liao. Adacbm: An adaptive concept bottleneck model for explainable and accurate diagnosis. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 35–45. Springer, 2024.
- [29] James R Clough, Ilkay Oksuz, Esther Puyol-Antón, Bram Ruijsink, Andrew P King, and Julia A Schnabel. Global and local interpretability for cardiac mri classification. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 656–664. Springer, 2019.
- [30] Jeremy E Cohen and Nicolas Gillis. Identifiability of complete dictionary learning. *SIAM Journal on Mathematics of Data Science*, 1(3):518–536, 2019.
- [31] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [32] Ali Dorri, Salil S Kanhere, and Raja Jurdak. Multi-agent systems: A survey. *Ieee Access*, 6:28573–28593, 2018.
- [33] Shiladitya Dutta, Hongbo Wei, Lars van der Laan, and Ahmed M Alaa. Estimating uncertainty in multimodal foundation models using public internet data. *arXiv preprint arXiv:2310.09926*, 2023.
- [34] Kjersti Engan, Sven Ole Aase, and J Hakon Husoy. Method of optimal directions for frame design. In *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258)*, volume 5, pages 2443–2446. IEEE, 1999.

- [35] Salar Fattahi and Somayeh Sojoudi. Exact guarantees on the absence of spurious local minima for non-negative rank-1 robust principal component analysis. *Journal of machine learning research*, 2020.
- [36] Ruoyu Feng, Tao Yu, Xin Jin, Xiaoyuan Yu, Lei Xiao, and Zhibo Chen. Rethinking domain adaptation and generalization in the era of clip. In *2024 IEEE International Conference on Image Processing (ICIP)*, pages 2585–2591. IEEE, 2024.
- [37] Jean-Jacques Fuchs. Recovery of exact sparse representations in the presence of bounded noise. *IEEE Transactions on Information Theory*, 51(10):3601–3608, 2005.
- [38] Jack Furby, Daniel Cunningham, Dave Braines, and Alun Preece. Can we constrain concept bottleneck models to learn semantically meaningful input features? *arXiv preprint arXiv:2402.00912*, 2024.
- [39] Rong Ge, Chi Jin, and Yi Zheng. No spurious local minima in nonconvex low rank problems: A unified geometric analysis. In *International Conference on Machine Learning*, pages 1233–1242. PMLR, 2017.
- [40] Donald Geman and Bruno Jedynak. An active testing model for tracking roads in satellite images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(1):1–14, 1996.
- [41] Quan Geng and John Wright. On the local correctness of ℓ_1 -minimization for dictionary learning. In *2014 IEEE International Symposium on Information Theory*, pages 3180–3184. IEEE, 2014.
- [42] Amirata Ghorbani, Abubakar Abid, and James Zou. Interpretation of neural networks is fragile. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3681–3688, 2019.
- [43] Dar Gilboa, Sam Buchanan, and John Wright. Efficient dictionary learning with gradient descent. In *International Conference on Machine Learning*, pages 2252–2259. PMLR, 2019.
- [44] Alexandros Gkillas, Dimitris Ampeliotis, and Kostas Berberidis. Federated dictionary learning from non-iid data. In *2022 IEEE 14th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP)*, pages 1–5. IEEE, 2022.
- [45] Muhammad Waleed Gondal, Jochen Gast, Inigo Alonso Ruiz, Richard Droste, Tommaso Macri, Suren Kumar, and Luitpold Staudigl. Domain aligned clip for few-shot classification. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5721–5730, 2024.
- [46] Mara Graziani, Vincent Andrearczyk, and Henning Müller. Regression concept vectors for bidirectional explanations in histopathology. In *Understanding and Interpreting Machine Learning in Medical Image Computing Applications: First International Workshops, MLCN 2018, DLF 2018, and iMIMIC 2018, Held in Conjunction with*

- MICCAI 2018, Granada, Spain, September 16-20, 2018, Proceedings 1*, pages 124–132. Springer, 2018.
- [47] Rémi Gribonval, Rodolphe Jenatton, and Francis Bach. Sparse and spurious: dictionary learning with noise and outliers. *IEEE Transactions on Information Theory*, 61(11):6298–6319, 2015.
 - [48] Rémi Gribonval and Karin Schnass. Dictionary identification—sparse matrix-factorization via ℓ_1 -minimization. *IEEE Transactions on Information Theory*, 56(7):3523–3539, 2010.
 - [49] Kimia Hamidieh, Haoran Zhang, Walter Gerych, Thomas Hartvigsen, and Marzyeh Ghassemi. Identifying implicit social biases in vision-language models. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, volume 7, pages 547–561, 2024.
 - [50] Trevor Hastie. The elements of statistical learning: data mining, inference, and prediction, 2009.
 - [51] Jingzhou Hu and Kejun Huang. Global identifiability of ℓ_1 -based dictionary learning via matrix volume optimization. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.
 - [52] Keke Huang, Xinyi Liu, Fanbiao Li, Chunhua Yang, Okyay Kaynak, and Tingwen Huang. A federated dictionary learning method for process monitoring with industrial applications. *IEEE Transactions on Artificial Intelligence*, 2022.
 - [53] Rodolphe Jenatton, Rémi Gribonval, and Francis Bach. Local stability and robustness of sparse dictionary learning in the presence of noise. *arXiv preprint arXiv:1210.0685*, 2012.
 - [54] Rodolphe Jenatton, Julien Mairal, Guillaume Obozinski, and Francis R Bach. Proximal methods for sparse hierarchical dictionary learning. In *ICML*, volume 1, page 2. Citeseer, 2010.
 - [55] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pages 5132–5143. PMLR, 2020.
 - [56] Shiva Kasiviswanathan, Huahua Wang, Arindam Banerjee, and Prem Melville. Online ℓ_1 -dictionary learning with application to novel document detection. *Advances in neural information processing systems*, 25, 2012.
 - [57] Eunji Kim, Dahuin Jung, Sangha Park, Siwon Kim, and Sungroh Yoon. Probabilistic concept bottleneck models. *arXiv preprint arXiv:2306.01574*, 2023.

- [58] Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T Schütt, Sven Dähne, Dumitru Erhan, and Been Kim. The (un) reliability of saliency methods. *Explainable AI: Interpreting, explaining and visualizing deep learning*, pages 267–280, 2019.
- [59] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International conference on machine learning*, pages 5338–5348. PMLR, 2020.
- [60] Stefan Kolek, Duc Anh Nguyen, Ron Levie, Joan Bruna, and Gitta Kutyniok. A rate-distortion framework for explaining black-box model decisions. In *International Workshop on Extending Explainable AI Beyond Deep Models and Classifiers*, pages 91–115. Springer, 2020.
- [61] Raed Kontar, Naichen Shi, Xubo Yue, Seokhyun Chung, Eunshin Byon, Mosharaf Chowdhury, Jionghua Jin, Wissam Kontar, Neda Masoud, Maher Nouiehed, et al. The internet of federated things (ioft). *IEEE Access*, 9:156071–156113, 2021.
- [62] Raed Kontar, Shiyu Zhou, Chaitanya Sankavaram, Xinyu Du, and Yilu Zhang. Nonparametric-condition-based remaining useful life prediction incorporating external factors. *IEEE Transactions on Reliability*, 67(1):41–52, 2017.
- [63] Oswin Krause and Christian 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*, pages 129–136, 2015.
- [64] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.
- [65] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *2009 IEEE conference on computer vision and pattern recognition*, pages 951–958. IEEE, 2009.
- [66] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.
- [67] Yann LeCun, Corinna Cortes, Chris Burges, et al. Mnist handwritten digit database, 2010.
- [68] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Ng. Efficient sparse coding algorithms. *Advances in neural information processing systems*, 19, 2006.
- [69] Joffrey L Leevy, Taghi M Khoshgoftaar, Richard A Bauder, and Naeem Seliya. A survey on addressing high-class imbalance in big data. *Journal of Big Data*, 5(1):1–30, 2018.
- [70] Ren-Cang Li. New perturbation bounds for the unitary polar factor. *SIAM Journal on Matrix Analysis and Applications*, 16(1):327–332, 1995.

- [71] Shutao Li, Leyuan Fang, and Haitao Yin. An efficient dictionary learning algorithm and its application to 3-d medical image denoising. *IEEE transactions on biomedical engineering*, 59(2):417–427, 2011.
- [72] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. *International Conference on Machine Learning*, 2021.
- [73] Geyu Liang, Senne Michielssen, and Salar Fattahi. Enhancing performance of explainable ai models with constrained concept refinement. *arXiv preprint arXiv:2502.06775*, 2025.
- [74] Geyu Liang, Naichen Shi, Raed Al Kontar, and Salar Fattahi. Personalized dictionary learning for heterogeneous datasets. *Advances in Neural Information Processing Systems*, 36, 2024.
- [75] Geyu Liang, Gavin Zhang, Salar Fattahi, and Richard Y Zhang. Simple alternating minimization provably solves complete dictionary learning. *arXiv preprint arXiv:2210.12816*, 2022.
- [76] Paul Pu Liang, Terrance Liu, Liu Ziyin, Nicholas B Allen, Randy P Auerbach, David Brent, Ruslan Salakhutdinov, and Louis-Philippe Morency. Think locally, act globally: Federated learning with local and global representations. *arXiv preprint arXiv:2001.01523*, 2020.
- [77] Meng Liu, Liqiang Nie, Xiang Wang, Qi Tian, and Baoquan Chen. Online data organizer: micro-video categorization by structure-guided multimodal dictionary learning. *IEEE Transactions on Image Processing*, 28(3):1235–1247, 2018.
- [78] Cewu Lu, Jiaping Shi, and Jiaya Jia. Online robust dictionary learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 415–422, 2013.
- [79] Julien Mairal, Francis Bach, and Jean Ponce. Task-driven dictionary learning. *IEEE transactions on pattern analysis and machine intelligence*, 34(4):791–804, 2011.
- [80] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th annual international conference on machine learning*, pages 689–696, 2009.
- [81] Julien Mairal, Michael Elad, and Guillermo Sapiro. Sparse representation for color image restoration. *IEEE Transactions on image processing*, 17(1):53–69, 2007.
- [82] Benoît Malézieux, Thomas Moreau, and Matthieu Kowalski. Understanding approximate and unrolled dictionary learning for pattern recovery. In *International Conference on Learning Representations*, 2021.

- [83] Emanuele Marconato, Andrea Passerini, and Stefano Teso. Interpretability is in the mind of the beholder: A causal framework for human-interpretable representation learning. *Entropy*, 25(12):1574, 2023.
- [84] Andrei Margeloiu, Matthew Ashman, Umang Bhatt, Yanzhi Chen, Mateja Jamnik, and Adrian Weller. Do concept bottleneck models learn as intended? *arXiv preprint arXiv:2105.04289*, 2021.
- [85] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agueray Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.
- [86] Nicolai Meinshausen and Peter Bühlmann. High-dimensional graphs and variable selection with the lasso. *Annals of Statistics*, 2006.
- [87] Rajeev Motwani and Prabhakar Raghavan. Randomized algorithms. *ACM Computing Surveys (CSUR)*, 28(1):33–37, 1996.
- [88] Tuomas Oikarinen, Subhro Das, Lam M Nguyen, and Tsui-Wei Weng. Label-free concept bottleneck models. *arXiv preprint arXiv:2304.06129*, 2023.
- [89] Bruno A Olshausen and David J Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325, 1997.
- [90] Marie-Christine Pali and Karin Schnass. Dictionary learning—from local towards global and adaptive. *Information and Inference: A Journal of the IMA*, 12(3):iaad008, 2023.
- [91] Yagyensh Chandra Pati, Ramin Rezaifar, and Perinkulam Sambamurthy Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of 27th Asilomar conference on signals, systems and computers*, pages 40–44. IEEE, 1993.
- [92] Qing Qu, Ju Sun, and John Wright. Finding a sparse vector in a subspace: Linear sparsity using alternating directions. *Advances in Neural Information Processing Systems*, 27, 2014.
- [93] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1:81–106, 1986.
- [94] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.
- [95] Sirisha Rambhatla, Xingguo Li, and Jarvis Haupt. Noodl: Provable online dictionary learning and sparse coding. *arXiv preprint arXiv:1902.11261*, 2019.

- [96] Ignacio Ramirez, Pablo Sprechmann, and Guillermo Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3501–3508. IEEE, 2010.
- [97] Ayush Ranjan, Daniel Wen, and Karthik Bhat. Unveiling glitches: A deep dive into image encoding bugs within clip. *arXiv preprint arXiv:2407.00592*, 2024.
- [98] Saiprasad Ravishankar, Anna Ma, and Deanna Needell. Analysis of fast structured dictionary learning. *Information and Inference: A Journal of the IMA*, 9(4):785–811, 2020.
- [99] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5):206–215, 2019.
- [100] Yoshihide Sawada and Keigo Nakamura. Concept bottleneck model with additional unsupervised concepts. *IEEE Access*, 10:41758–41765, 2022.
- [101] Karin Schnass. Local identification of overcomplete dictionaries. *J. Mach. Learn. Res.*, 16(Jun):1211–1242, 2015.
- [102] Karin Schnass. Convergence radius and sample complexity of itkm algorithms for dictionary learning. *Applied and Computational Harmonic Analysis*, 45(1):22–58, 2018.
- [103] Karin Schnass and Flavio Teixeira. Compressed dictionary learning. *Journal of Fourier Analysis and Applications*, 26(2):33, 2020.
- [104] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.
- [105] Jie-Jing Shao, Jiang-Xin Shi, Xiao-Wen Yang, Lan-Zhe Guo, and Yu-Feng Li. Investigating the limitation of clip models: The worst-performing categories. *arXiv preprint arXiv:2310.03324*, 2023.
- [106] Yifei Shen, Ye Xue, Jun Zhang, Khaled Letaief, and Vincent Lau. Complete dictionary learning via lp-norm maximization. In *Conference on Uncertainty in Artificial Intelligence*, pages 280–289. PMLR, 2020.
- [107] Jiahao Shi and James C Spall. Difference between cyclic and distributed approach in stochastic optimization for multi-agent system. *arXiv preprint arXiv:2409.05155*, 2024.
- [108] Naichen Shi and Raed Al Kontar. Personalized pca: Decoupling shared and unique features. *arXiv preprint arXiv:2207.08041*, 2022.

- [109] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pages 3145–3153. PMIR, 2017.
- [110] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.
- [111] Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 180–186, 2020.
- [112] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.
- [113] Daniel A Spielman, Huan Wang, and John Wright. Exact recovery of sparsely-used dictionaries. In *Conference on Learning Theory*, pages 37–1. JMLR Workshop and Conference Proceedings, 2012.
- [114] Akshayvarun Subramanya, Vipin Pillai, and Hamed Pirsiavash. Fooling network interpretation in image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2020–2029, 2019.
- [115] Ju Sun, Qing Qu, and John Wright. Complete dictionary recovery over the sphere. In *2015 International Conference on Sampling Theory and Applications (SampTA)*, pages 407–410. IEEE, 2015.
- [116] Ju Sun, Qing Qu, and John Wright. Complete dictionary recovery over the sphere i: Overview and the geometric picture. *IEEE Transactions on Information Theory*, 63(2):853–884, 2016.
- [117] Ju Sun, Qing Qu, and John Wright. Complete dictionary recovery over the sphere i: Overview and the geometric picture. *IEEE Transactions on Information Theory*, 63(2):853–884, 2016.
- [118] Wenbo Sun, Jiacheng Liu, Jingwen Hu, Judy Jin, Kevin Siasoco, Rongrong Zhou, and Robert McCoy. Adaptive restraint design for a diverse population through machine learning. *Frontiers in Public Health*, 11:1202970, 2023.
- [119] Canh T Dinh, Nguyen Tran, and Josh Nguyen. Personalized federated learning with moreau envelopes. *Advances in Neural Information Processing Systems*, 33:21394–21405, 2020.
- [120] Fadi Thabtah, Suhel Hammoud, Firuz Kamalov, and Amanda Gonsalves. Data imbalance in classification: Experimental evaluation. *Information Sciences*, 513:429–441, 2020.

- [121] TheBlackMamba31. Landscape image colorization. <https://www.kaggle.com/datasets/theblackmamba31/landscape-image-colorization>, accessed 2023.
- [122] Bahareh Tolooshams and Demba Ba. Stable and interpretable unrolled dictionary learning. *arXiv preprint arXiv:2106.00058*, 2021.
- [123] Ivana Tošić and Pascal Frossard. Dictionary learning. *IEEE Signal Processing Magazine*, 28(2):27–38, 2011.
- [124] Joel A Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Transactions on Information theory*, 50(10):2231–2242, 2004.
- [125] Joel A Tropp. Just relax: Convex programming methods for identifying sparse signals in noise. *IEEE transactions on information theory*, 52(3):1030–1051, 2006.
- [126] Antoine Vacavant, Thierry Chateau, Alexis Wilhelm, and Laurent Lequievre. A benchmark dataset for outdoor foreground/background extraction. In *Computer Vision-ACCV 2012 Workshops: ACCV 2012 International Workshops, Daejeon, Korea, November 5-6, 2012, Revised Selected Papers, Part I 11*, pages 291–300. Springer, 2013.
- [127] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.
- [128] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.
- [129] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.
- [130] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. *arXiv preprint arXiv:2002.06440*, 2020.
- [131] Yingfan Wang and Guoliang Kang. Attention head purification: A new perspective to harness clip for domain generalization. *arXiv preprint arXiv:2412.07226*, 2024.
- [132] David S Watson. Conceptual challenges for interpretable machine learning. *Synthese*, 200(2):65, 2022.
- [133] Mehran Yazdi and Thierry Bouwmans. New trends on moving object detection in video images captured by a moving camera: A survey. *Computer Science Review*, 28:157–177, 2018.
- [134] Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Josh Tenenbaum. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. *Advances in neural information processing systems*, 31, 2018.

- [135] Xubo Yue, Maher Nouiehed, and Raed Al Kontar. Gifair-fl: A framework for group and individual fairness in federated learning. *INFORMS Journal on Data Science*, 2022.
- [136] Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc concept bottleneck models. *arXiv preprint arXiv:2205.15480*, 2022.
- [137] Mateo Espinosa Zarlenga, Pietro Barbiero, Gabriele Ciravegna, Giuseppe Marra, Francesco Giannini, Michelangelo Diligenti, Frederic Precioso, Stefano Melacci, Adrian Weller, Pietro Lio, et al. Concept embedding models. In *NeurIPS 2022-36th Conference on Neural Information Processing Systems*, 2022.
- [138] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part I 13*, pages 818–833. Springer, 2014.
- [139] Yuexiang Zhai, Zitong Yang, Zhenyu Liao, John Wright, and Yi Ma. Complete dictionary learning via l4-norm maximization over the orthogonal group. *Journal of Machine Learning Research*, 21(165):1–68, 2020.
- [140] Cun-Hui Zhang. Nearly unbiased variable selection under minimax concave penalty. *The Annals of statistics*, 38(2):894–942, 2010.
- [141] Peng Zhao and Bin Yu. On model selection consistency of lasso. *The Journal of Machine Learning Research*, 7:2541–2563, 2006.
- [142] Rongchang Zhao, Hong Li, and Xiyao Liu. A survey of dictionary learning in medical image analysis and its application for glaucoma diagnosis. *Archives of Computational Methods in Engineering*, 28(2):463–471, 2021.
- [143] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017.