Joint Estimation of Dictionary and Image from Compressive Samples

Mohammad Aghagolzadeh, Hayder Radha, *Fellow, IEEE*

***Abstract*—Effective Compressed Sensing (CS) of images relies on the prior knowledge of a well-suited dictionary for sparse representation of the target image. In the absence of such knowledge, which is a typical scenario in real-world applications, the following question arises: *Would it be advantageous to take an off-the-shelf overcomplete dictionary and fine-tune the dictionary with respect to (the observed part of) the image?* A primary obstacle in this approach is overfitting; i.e. the loss in model generalization to the whole image. In this paper, we establish that *local* dictionary optimization using the compressive samples reduces the image recovery error—relative to the off-the-shelf recovery—with an overwhelming probability that depends on the sampling matrix. We present JEDI, an iterative algorithm for dictionary fine-tuning from compressive samples and analyze its performance for CS image recovery. Our algorithmic analysis is supplemented with numerical simulations under different random sampling patterns and off-the-shelf dictionary initializations.**

***Index Terms*—Dictionary learning; adaptive image recovery; blind compressed sensing.**

I. Introduction

The theory of Compressed Sensing (CS) [1], [2] establishes that the combinatorial problem of recovering the sparsest vector $x \in \mathbb{R}^n$ from a set of linear measurements $\Phi x = y \in \mathbb{R}^m$ with $m \ll n$, i.e.

$$x^* = \arg\min_{\hat{x}} \|\hat{x}\|_0 \;\; s.t. \;\; \Phi\hat{x} = \Phi x = y \tag{1}$$

can be solved in a polynomial time, given that the measurement matrix Φ, also known as sampling or projection matrix, satisfies certain isometry conditions [3].

While CS can be directly used for recovering sparse signals in the standard basis, it has also been extended [4] to work with dense signals, e.g. natural images, that can be represented as a sparse combination of some *dictionary*'s elements (columns of the dictionary matrix, a.k.a. *atoms*). If the dictionary matrix is denoted by $D \in \mathbb{R}^{n\times p}$ with $p \geq n$, the CS recovery problem can be written as finding the sparsest representation $\hat{x} = D\alpha$, subject to the measurements:

$$\alpha^* = \arg\min_{\alpha} \|\alpha\|_0 \;\; s.t. \;\; \Phi D\alpha = \Phi x = y \tag{2}$$

and $x^* = D\alpha^*$. As an alternative to hand-crafted dictionaries, such as wavelets [5], Dictionary Learning (DL) [6] is a data-driven approach to building the dictionary matrix D.

Training dictionaries over large-scale databases of natural images, e.g. similar to [23], results in universal dictionaries that work well for most images. However, due to the intensive training process, universally trained dictionaries are usually used off-the-shelf. In a related line of work, several efforts have been made into fine-tuning the available off-the-shelf dictionaries using a set of incomplete or noisy image measurements. This technique has been used in image denoising [7], inpainting [8] and most recently, CS [14]. In this paper, we focus on the last application; i.e. CS recovery by starting from an the off-the-shelf dictionary and updating the dictionary using the CS measurements during recovery.

A relevant theoretical framework is Blind Compressed Sensing (BCS) [9] which studies the conditions under which exact sparse recovery is achievable with no prior knowledge of the sparsifying dictionary. To explain more, assume that $X \in \mathbb{R}^{n\times N}$ is formed from a set of N n-dimensional signal vectors and there exists a dictionary $\tilde{D} \in \mathbb{R}^{n\times p}$ $(p \ll N)$ and a sparse matrix $A \in \mathbb{R}^{p\times N}$ such that $X = \tilde{D}A$. If $\tilde{D}$ is known and fixed, estimating X from $Y = \Phi X$ boils down into a typical CS problem. However, when $\tilde{D}$ is unknown, BCS corresponds to estimating both $\tilde{D}$ and A from $Y = \Phi\tilde{D}A$. While BCS represents an intriguing problem, recovering X from $Y = \Phi X$ can be shown to be impossible, if N is small or other certain conditions are not satisfied. Requirements for uniqueness of BCS solutions have been studied in [9]–[11]. However, such theoretical approaches either rely on specific signal and dictionary constraints [9], [10] or can be impractical to implement for large n [11].

For the sole purpose of paper organization, we classify the dictionary fine-tuning algorithms, such as [7], [8], [14], as a subcategory of BCS which we refer to as 'practical BCS'. In spite of the relevance between theoretical and practical BCS, there remains a large gap between them. This gap is described below, followed by a description of our contributions towards reducing it.

A. Prior art

Existing efforts in the area of CS recovery with variable or unknown dictionaries can be divided into two schemes:

1) Theoretical BCS: [9]–[11]. These works target exact recovery guarantees for truly sparse signals but do not provide practical algorithms for the general case. BCS theory was initially conceived in [9] and was later extended in [10] and [11]. Specifically, [9] and [10] focus on the constrained versions of the BCS problem tailored for specific applications while [11] studies BCS in its general form. These theoretical studies rely on strict sparsity assumptions over the coefficients and spark assumptions over the dictionary which are not always true for real-world images. The generic BCS recovery

M. Aghagolzadeh and H. Radha are with the Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI, 48823 USA e-mail: {aghagol1, radha@msu.edu}.

that is studied in [11] involves a time and memory consuming procedure that is not practical for imaging applications.

2) Practical BCS: [12]–[16]. In contrast to theoretical BCS studies, these works explore the practical aspects of dictionary learning from corrupt or compressive measurements without providing strong performance guarantees or analysis. These approaches iteratively update the dictionary and the sparse coefficients to minimize the sparse representation error. For example, [12], [13] adopt the K-SVD strategy of [7] while [14], [16] use gradient descent and [15] employs a recursive Bayesian approach.

It is important to note that the unification of BCS theory [9]–[11] with [12]–[16] is mainly for paper organization.

B. Our contributions

Our aim in this paper is to study the *theoretical aspects of practical BCS* by proposing an efficient dictionary fine-tuning algorithm with provable insights into why and how it reduces the recovery error. In particular, we intend to extend the BCS literature in the following directions:

1) Mathematical justification for practical BCS: Empirical BCS studies [12]–[16] do not provide any justification for the uniqueness, convergence or the accuracy of their algorithms. Therefore, it is crucial to develop a theoretical understanding of these algorithms. In this work, we adopt a gradient descent methodology, similar to [14], [16]. Rather than focusing on deriving new BCS algorithms, our focus is on optimizing and analysis of efficient and well-performing BCS algorithms. Our proposed algorithm for analysis is termed JEDI which stands for Joint Estimation of Dictionary and Image.

2) Empirical evaluations over natural images: In [10], [14], empirical evaluations over natural images are limited to the image inpainting problem which can be viewed as a particular case of CS—in which the compressive measurements are in the standard basis. Meanwhile, in [12]–[14] the generic CS problem is only tested on artificially generated strictly sparse vectors. Overall, the variety of test images and algorithm initializations in [10], [14], [16] are very limited. Finally and most importantly, most of these studies fail to compare the BCS recovery with CS recovery based on off-the-shelf dictionaries. One of our goals in this paper is to improve upon non-adaptive sparse image recovery using pre-trained dictionaries based on well-established DL algorithms such as [7] or [23].

Similar to most efforts in the area of compressive imaging, including the BCS framework, we employ a block compressed sensing or *block-CS* scheme for measurement and recovery of images [18], [19]. Unlike *dense-CS*, where the image is recovered as a single vector using a dense sampling matrix, block-CS attempts to break down the high dimensional image into small non-overlapping blocks and solve a CS over each block. Some advantages of block-CS are: (a) a block-diagonal sampling matrix which significantly reduces the amount of required memory for its storage or communication, (b) high-dimensional CS is computationally challenging[1] while block-CS can take advantage of parallel or distributed processing, (c) sparse modeling or learning dictionaries for high-dimensional global image features is challenging and not well studied.

C. Paper organization

In Section II, we formally overview DL and BCS and propose our algorithm, JEDI, for practical BCS. We present our analytical contributions regarding the performance analysis of JEDI in Section III. Simulation results are presented in Section IV. We conclude this paper in Section V.

D. Notation

Throughout the paper, upper-case letters, except for N and T, are used for matrices and lower-case letters are used for vectors and scalars. I_n denotes the identity matrix of size n. We reserve the following notation: N is the total number of blocks in the image, n is the size of each block (e.g. an 8×8 block has size 64), m is the number of compressive measurements per block ($m \leq n$), p is the number of atoms in the dictionary, t is the iteration index, T is the total number of iterations, $D \in \mathbb{R}^{n\times p}$ denotes the dictionary, $x_j \in \mathbb{R}^n$ denotes the vectorized image block (column-major) number j, $\alpha_j \in \mathbb{R}^p$ is the representation of x_j (i.e. $x_j \approx D\alpha_j$), $\Phi_j \in \mathbb{R}^{m\times n}$ denotes the measurement matrix for block number j and $y_j \in \mathbb{R}^m$ denotes the vector of compressive measurements (i.e. $y_j = \Phi_j x_j$). For simplicity, we drop the block index subscripts in α_j, x_j, Φ_j and y_j when a single block is under consideration.

The vector ℓ_q norm is defined as $\|x\|_q = (\sum_i |x_i|^q)^{\frac{1}{q}}$. Particularly, $\|x\|_0$ denotes the number of non-zero entries of x. The matrix operators $A \otimes B$ and $A \odot B$ respectively denote the Kronecker and the Hadamard (element-wise) products. The operator $\mathrm{vec}(A)$ reshapes the matrix A to its column-major vectorized format. The matrix inner product is defined as $\langle A, B\rangle = \mathrm{Tr}(A^T B)$ with $\mathrm{Tr}(A)$ denoting the matrix trace, i.e. the sum of diagonal entries of A. The squared Frobenius norm of A is defined as $\|A\|_F^2 = \mathrm{Tr}(A^T A) = \sum_{ij} |A_{ij}|^2$. Let $[n] = \{1, 2, \ldots, n\}$ for any $n \in \mathbb{N}$.

II. Problem Statement

A. Overview of the dictionary learning problem

The goal of DL is to obtain a matrix $D \in \mathbb{R}^{n\times p}$—consisting of p atoms in the n-dimensional ambient signal space—such that, for a collection of N training signals $\{x_1, x_2, \ldots, x_N\}$, each $x_j \in \mathbb{R}^n$ can be expressed as $x_j = D\alpha_j$ with $\|\alpha_j\|_0 \leq k$ for some $k \ll p$.

When working with real-world data, such as natural images, the equality $x_j = D\alpha_j$ may not be satisfied for any reasonable p (e.g. a multiple of n) and $k \ll p$. In those cases, the ℓ_2 cost function $\|x_j - D\alpha_j\|_2^2$ is minimized over the training signals in DL. Also, similar to the methodology of CS [3], the ℓ_0-norm constraint $\|\alpha_j\|_0 \leq k$ is typically replaced with an ℓ_1-norm constraint $\|\alpha_j\|_1 \leq \tau$ to make the optimization of α_j convex and tractable. The resulting DL problem can be expressed as:

$$D^* = \arg\min_{D\in\mathcal{D}} \psi(D) \qquad \text{(P1)}$$

[1] A typical consumer image has an order of 10^6 pixels which would make it impractical to be recovered as a single vector using existing sparse recovery methods with quadratic or cubic time complexities.

where $\mathcal{D}$ denotes the set of admissible dictionaries and $\psi(D)$ represents the following *empirical risk function*:

$$\psi(D) = \frac{1}{N}\sum_{j=1}^{N} \min_{\alpha_j} \|x_j - D\alpha_j\|_2^2 \;\; s.t. \;\|\alpha_j\|_1 \leq \tau \qquad (3)$$

An equivalent and arguably more useful definition of $\psi(D)$ employs the unconstrained Lasso [20] formulation:

$$\psi(D) = \frac{1}{N}\sum_{j=1}^{N} \min_{\alpha_j} \|x_j - D\alpha_j\|_2^2 + \lambda\|\alpha_j\|_1 \qquad (4)$$

In this formulation, λ controls the level of sparsity of the signal representation. The optimal value of λ depends on the data and the particular application in mind.

(P1) is usually viewed as a "layered" optimization problem where the *inner-layer* consists of N instances of Lasso to construct the empirical risk function $\psi(D)$ and the *outer-layer* consists of minimizing $\psi(D)$ over D. The typical strategy used in most works [7], [8], [23]–[29] is to alternate between the inner and outer layers until convergence to a local minimum. Formally, the iterative procedure is:

$$D^{(t+1)} = \arg\min_{D\in\mathcal{D}} \psi^{(t)}(D) \qquad (5)$$

where

$$\psi^{(t)}(D) = \frac{1}{N}\sum_{j=1}^{N} \|x_j - D\alpha_j^{(t)}\|_2^2 \qquad (6)$$

$$\alpha_j^{(t)} = \arg\min_{\alpha} \|x_j - D^{(t)}\alpha\|_2^2 + \lambda\|\alpha\|_1 \qquad (7)$$

This procedure starts from an initial dictionary $D = D^{(0)}$. For example, $D^{(0)}$ can be the overcomplete cosine frame [7]. Notably, this iterative procedure is suitable for parallel or distributed computing where each node would be responsible for a subset of the training dataset; see e.g. [30]. Meanwhile, in a streaming input data model, the outer-layer problem can be handled efficiently using online learning [23]. However, to simplify the analysis, we resort to batch learning[2] along with an efficient gradient descent for the outer-layer problem.

Finally, it must be noted that (P1) is in fact trivial without additional constraints over the set $\mathcal{D}$. It is not difficult to see that scaling D in (3) by some constant $c \in (1,\infty)$ is equivalent to scaling the ℓ_1 constraint parameter τ by c. Particularly, the ℓ_1 constraint in (3) would become ineffective as $c \to \infty$. There are two typical bounding methods that have been discussed in more detail in e.g. [28], [29]: *a*) bounding the ℓ_2 vector norm of each of D's atoms (columns) or *b*) bounding the Frobenius matrix norm of D. For example, in the second approach, the constraint set takes the following form:

$$\mathcal{D} = \{D | D \in \mathbb{R}^{n\times p}, \|D\|_F \leq \sqrt{p}\} \qquad (8)$$

The choice of $\mathcal{D}$ is discussed later in this section.

B. Dictionary fine-tuning for blind compressed sensing

The problem of CS is to recover a sparse signal $x \in \mathbb{R}^n$, or a signal that can be approximated by a sparse vector, from a set of compressive measurements $y = \Phi x$. $\Phi \in \mathbb{R}^{m\times n}$ represents a matrix of random i.i.d. entries from a certain distribution, such as a Gaussian distribution. When $m < n$, the linear system of equations is under-determined and the solution set $\{x | x \in \mathbb{R}^n, y = \Phi x\}$ is infinite. However, under sparsity constriants over x, the solution to $y = \Phi x$ can be unique when $m < n$ [32]. Unfortunately, the problem of searching for the sparsest x subject to $y = \Phi x$ is NP-hard and impractical to solve for high-dimensional x. Meanwhile, CS theory indicates that this problem can be provably solved in a polynomial time, by replacing the sparsity constraint with a bounded ℓ_1 norm constraint as in the Lasso regression [20], given that Φ satisfies a Restricted Isometry Property (RIP) [3]. It has been shown that most random designs of Φ would yield RIP with high probabilities [33].

CS can also be used to recover a dense x if x has a sparse representation of the form $x = D\alpha$. Measurements can be written as $y = (\Phi D)\alpha$. Dictionary-based CS recovery of x is $\hat{x} = D\hat{\alpha}$ where

$$\hat{\alpha} = \arg\min_{\alpha} \|\alpha\|_1 \;\; s.t. \;\; \Phi D\alpha = y \qquad \text{(CS)}$$

Real-world signals, such as natural images, may not have exact sparse representations under any dictionary of a bounded size. In such cases, one finds the sparsest representation of x given an upper bound on the error $\|y - \Phi D\alpha\|_2$. This problem, which is usually referred to as Noisy CS, corresponds to the following optimization problem:

$$\hat{\alpha} = \arg\min_{\alpha} \|\alpha\|_1 \;\; s.t. \;\; \|y - \Phi D\alpha\|_2 \leq \sigma \qquad \text{(Noisy CS)}$$

It can be challenging to estimate the value of σ—before or during recovery—due to its dependence on the dictionary and the unknown signal. When σ is unknown, CS with cross-validation [34], [35] may be used: a small portion of CS measurements are dedicated solely for the purpose of selecting a σ that results in the most accurate recovery. We use a similar strategy in our algorithm to avoid overfitting.

BCS corresponds to the problem of recovering a collection of N signals $\{x_1, \ldots, x_N\}$ from their compressive measurements, $y_j = \Phi_j x_j, \forall j \in [N]$, using the same but unknown dictionary for all signals. In this paper, each x_j represents an image block and Φ_j represents the measurement matrix for that block. The goal of BCS theory is to study the requirements (e.g. bounds for N, k, p, m) under which the optimal dictionary and the recovered data are unique. At the same time, it is important to have access to practical BCS algorithms for real-world applications. Unfortunately, existing BCS theory falls short in both aspects of applicability and practicality for real-word data. Specifically, the original BCS theory [9] has been limited to complete dictionaries ($p = n$) with structural constraints (e.g. sparsely representable or block-diagonal dictionaries) and [10] focuses on a block-sparse signal model. Although [11] addresses the unconstrained BCS with overcomplete dictionaries, it involves a computationally complex recovery algorithm.

[2] By batch learning, we refer to the processing of all N blocks at once while online learning refers to the one-by-one or 'mini-batch' processing of blocks in a streaming fashion.

A more practical approach to BCS is finding a locally optimal dictionary, in the vicinity of a universal dictionary. While this approach is different from the theoretical BCS framework that targets the globally optimal dictionary, one would hope that the recovery could benefit from dictionary "adaptation". Recall that typical DL frameworks also target local dictionary optimization. However, unlike typical DL that uses large datasets of training images, our task involves CS measurements of a single image. To distinguish our framework from ordinary DL frameworks, we refer to the process of local dictionary optimization over a single image as dictionary *fine-tuning*. Dictionary fine-tuning is more prone to overfitting due to the small size of the training data. Below, we describe the derivation of a dictionary fine-tuning algorithm for BCS.

To start with, we extend the iterative DL procedure to the scenario of having access only to the CS measurements:

$$D^{(t+1)} = \arg\min_{D \in \mathcal{D}} \hat{\psi}^{(t)}(D) \tag{9}$$

where

$$\hat{\psi}^{(t)}(D) = \frac{1}{N}\sum_{j=1}^{N} \|y_j - \Phi_j D \alpha_j^{(t)}\|_2^2 \tag{10}$$

$$\alpha_j^{(t)} = \arg\min_{\alpha} \|\alpha\|_1 \;\; s.t. \;\; \|y_j - \Phi_j D^{(t)}\alpha\|_2 \leq \sigma_j^{(t)} \tag{11}$$

Note that the computation of $\alpha_j^{(t)}$ is based on the Noisy CS framework which relies on an estimation of the noise variance $\sigma_j^{(t)}$. Our complete algorithm, termed JEDI for Joint Estimation of Dictionary and Image, is described below.

C. JEDI

The outer-layer constrained quadratic optimization step defined in (9) can be computationally challenging for real-time applications. Specifically, it has a running time complexity of $O(Nn^2p^2 + n^3p^3)$ and is memory intensive. The strategy employed in this paper, similar to what has been proposed in [6], [24], [28], is to perform a gradient update:

$$D^{(t+1)} = D^{(t)} - \mu^{(t)} \nabla_D \hat{\psi}^{(t)}(D^{(t)}) \tag{12}$$

In an 'exact' line search framework, the optimal step size can be computed as:

$$\mu^{(t)} = \arg\min_{\mu} \hat{\psi}^{(t)}\left(D^{(t)} - \mu \nabla_D \hat{\psi}^{(t)}(D^{(t)})\right)$$
$$= \frac{\|\nabla_D \hat{\psi}^{(t)}(D^{(t)})\|_F^2}{\sum_{j=1}^{N} \|\Phi_j \nabla_D \hat{\psi}^{(t)}(D^{(t)})\alpha_j^{(t)}\|_2^2}$$

A low complexity alternative to exact line search is backtracking [31]. Meanwhile, it is known that a properly selected fixed step size enjoys the same convergence rate [31]. The gradient update in (12) with a fixed step size is significantly faster than performing a second-order Newton update which involves computing the inverse of the Hessian matrix $H_t \in \mathbb{R}^{np \times np}$ or its approximation as in BFGS.

JEDI is formalized in Alg. 1. The initial estimate for the noise variance is obtained as follows:

$$\sigma_j^{(0)} = \gamma^{-1}\|y_j - \Phi_j D^{(0)}\tilde{\alpha}_j\|_2 \tag{13}$$

$$\tilde{\alpha}_j = \arg\min_{\alpha} \|y_j - \Phi_j D^{(0)}\alpha\|_2^2 + \lambda_0\|\alpha\|_1 \tag{14}$$

for some $\gamma \in (0, 1)$ that is discussed later in the following section. Selection of λ_0 is discussed in Section IV.

The time complexity of computing $\nabla_D \hat{\psi}^{(t)}(D^{(t)})$:

$$\nabla_D \hat{\psi}^{(t)}(D^{(t)}) = -\frac{1}{N}\sum_{j=1}^{N} \Phi_j^T (y_j - \Phi_j D^{(t)}\alpha_j^{(t)})\alpha_j^{(t)T} \tag{15}$$

is $O(Nnmk)$. Also, the time complexity of Noisy CS based on the Least Angle Regression algorithm of [20] is $O(pnk + pk^2)$ [23]. Therefore, the total complexity of Alg. 1 is $O(TNnk(m+p) + TNpk^2)$. The dominant term $O(TNnpk)$ is linear in each of the parameters.

Finally, it is important to note that, in JEDI, no constraints are imposed over the dictionary. Specifically, the sparse coding step in (11) is different from (7) in that scaling D by $c \in (1, \infty)$ in (11) merely scales the coefficient vector α_j (by c^{-1}), not affecting its sparsity. This scaling is tolerable up to the machine precision. Our numerical tests show that $\|D^{(T)}\|_F$ is often less than an order of magnitude greater than $\|D^{(0)}\|_F$ under JEDI. Relaxing the dictionary norm constraint helps in simplifying the mathematical analysis.

D. Remarks on the measurement matrix

In the proposed block-CS framework, entries of each Φ_j are random i.i.d. instances of a zero-mean Gaussian with variance $1/m$. This choice of variance results in $E\{\Phi_j^T\Phi_j\} = I_n$ which alleviates the need for normalization of the error function. Other distributions of Φ_j are also valid under our framework but generally result in inferior performances. In particular, a typical choice for Φ_j is the one with a single non-zero entry in each row; i.e. each measurement corresponds a pixel value and a random subset of pixels are measured from each block. We also evaluate JEDI under such measurement scenarios (which is referred to as image inpainting).

We conclude by mentioning that block-wise measurements could be arranged in a single equation:

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \underline{\Phi}\,(I_N \otimes D) \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{bmatrix} \tag{16}$$

where

$$\underline{\Phi} = \begin{bmatrix} \Phi_1 & & & \\ & \Phi_2 & & \\ & & \ddots & \\ & & & \Phi_N \end{bmatrix} \tag{17}$$

represents the block-diagonal measurement matrix.

III. Mathematical Analysis of JEDI

A. Preliminaries

Define the Oracle Error (OE) at the end of each iteration:

$$h_t := \frac{1}{nN}\sum_{j=1}^{N} \|x_j - D^{(t)}\alpha_j^{(t-1)}\|_2^2 \tag{18}$$

Also, define $h_0 := \frac{1}{nN}\sum_{j=1}^{N}\|x_j - D^{(0)}\alpha_j^{(0)}\|_2^2$. We have employed the term 'oracle' to highlight the fact that x_j is not visible to the algorithm and h_t can only be computed by an oracle who has full access to the underlying image.

Recall that our ultimate goal is to prove that our proposed solution for BCS outperforms CS. The performance of most recovery methods is quantified using the recovery Peak-Signal-to-Noise-Ratio (PSNR), typically expressed in dBs. Assuming that the pixel values are normalized within the range $[0, 1]$,

$$\text{PSNR(CS)} = -10\log_{10}(h_0)$$
$$\text{PSNR(BCS)} = -10\log_{10}(h_T)$$

Therefore, a decrease in OE is equivalent to an increase the recovery PSNR. Consequently, we base our analysis around OE and the extent to which JEDI reduces OE.

Clearly, OE cannot be directly minimized (or be evaluated) in a BCS framework. Instead, JEDI aims at minimizing the Projected Error (PE):

$$g_t := \frac{1}{nN}\sum_{j=1}^{N}\|y_j - \Phi_j D^{(t)}\alpha_j^{(t-1)}\|_2^2 = \frac{1}{nN}\sum_{j=1}^{N}{\sigma_j^{(t)}}^2 \tag{19}$$

which is an approximation of h_t. It is important to note that a reduction in g_t (i.e. $g_t < g_{t-1}$) does not necessarily result in a reduction in h_t. This issue is the subject of our analysis. In particular, we intend to answer the following question.

Problem 1. ***Let h_t denote the OE (oracle error) at iteration t of JEDI. What is the decay rate of h_t given that JEDI has only access to the projected error g_t?***

Throughout this section, we show that the penalty in the decay rate of h_t, due to the use of $m \ll n$ measurements per signal, is of order $O(N^{-\frac{1}{2}}m^{-\frac{1}{2}})$ which becomes tolerable for a reasonably large N, even when m is small. To obtain this result, we intend the following plan of attack:

- We show that h_t is bounded within an 'envelope' around g_t which shrinks at the rate $O(N^{-\frac{1}{2}}m^{-\frac{1}{2}})$.
- We show that the decay rate of g_t, under a plain gradient descent algorithm, is at least of order $O(N^{-\frac{1}{2}})$.
- We put together the above results to compute the overall decay rate of h_t, compared to the case of having complete measurements.

Define the oracle and projected *error functions*, for fixed x_j's and α_j's and variable D, respectively denoted by $h_t(D)$ and $g_t(D)$, as:

$$h_t(D) := \frac{1}{nN}\sum_{j=1}^{N}\|x_j - D\alpha_j^{(t-1)}\|_2^2 \tag{20}$$

$$g_t(D) := \frac{1}{nN}\sum_{j=1}^{N}\|\Phi_j(x_j - D\alpha_j^{(t-1)})\|_2^2 \tag{21}$$

Clearly, $g_t = g_t(D^{(t)})$ and $h_t = h_t(D^{(t)})$. A crucial link between $h_t(D)$ and $g_t(D)$ is stated in the following asymptotic result which is enhanced (with non-asymptotic bounds) and proved later in this section.

Lemma 1. ***Asymptotically almost surely (i.e. almost surely as $N \to \infty$) for $\forall D \in \mathcal{D}, \forall \epsilon \in (0, 1)$:***

$$(1-\epsilon)h_t(D) \le g_t(D) \le (1+\epsilon)h_t(D) \tag{22}$$

An immediate consequence of Lemma 1 would be the following inequality, which is simply the application of $D = D^{(t)}$:

$$(1-\epsilon)h_t \le g_t \le (1+\epsilon)h_t \qquad \text{(a.a.s.)} \tag{23}$$

Algorithm 1 Joint Estimation of Dictionary and Image (JEDI)

```
Require: $y_j$, $\Phi_j$, $\sigma_j^{(0)}$ ($\forall j \in [N]$), $D^{(0)}$, $T$, $\gamma$
  $t \leftarrow 0$
  while $t < T$ do
      Randomly partition the set of measurements into two sets of equal size: $\forall j \in [N]$: $[{y_j^{(1)}}^T {y_j^{(2)}}^T] = x_j^T[{\Phi_j^{(1)}}^T {\Phi_j^{(2)}}^T]$
      ________ Noisy CS (using measurement set 1) ________
      $\forall j$: ${\alpha_j^{(1)}}^{(t)} = \arg\min_\alpha \|\alpha\|_1 \; s.t. \; \|y_j^{(1)} - \Phi_j^{(1)} D^{(t)}\alpha\|_2 \le \gamma\sigma_j^{(t)}$
      ________ Dictionary Update (using measurement set 2) ________
      $\nabla_D\hat{\psi}^{(t)}(D^{(t)}) = -\sum_{j=1}^N {\Phi_j^{(2)}}^T(y_j^{(2)} - {\Phi_j^{(2)}}^T D^{(t)}{\alpha_j^{(1)}}^{(t)}){{\alpha_j^{(1)}}^{(t)}}^T$
      $D^{(t+1)} = D^{(t)} - \mu^{(t)}\nabla_D\hat{\psi}^{(t)}(D^{(t)})$
      ____________________________________________
      $\forall j$: $\alpha_j^{(t)} = \arg\min_\alpha \|\alpha\|_1 \; s.t. \; \|y_j - \Phi_j D^{(t)}\alpha\|_2 \le \gamma\sigma_j^{(t)}$
      $\forall j$: $\sigma_j^{(t+1)} = \|y_j - \Phi_j D^{(t+1)}\alpha_j^{(t)}\|_2$
      $t \leftarrow t + 1$
  end while
  return $D^{(T)}$ and $\{\alpha_1^{(T-1)}, \ldots, \alpha_N^{(T-1)}\}$
```

Unfortunately, applying BCS over a single image can result in a rather limited N. For a concrete example, assume that the image under recovery is 512×512 and the block size is 8×8. The total number of non-overlapping blocks is thus $N = 4096$ which may not be sufficient for the asymptotic statements of Lemma 1 and (23). Therefore, it is important to study the non-asymptotic counterpart of (23):

$$(1-\epsilon_N)h_t \leq g_t \leq (1+\epsilon_N)h_t \qquad \text{(w.h.p.)} \tag{24}$$

where 'with high probability' is meant as being satisfied with a fixed probability that is close to one. $\epsilon_N \in (0,1)$ is a descending function of N. Hereafter, for simplicity, 'w.h.p.' is implicit in the expressions that contain ϵ_N.

B. Convergence rates of g_t and h_t

According to Alg. 1, the PE sequence $g_0, g_1, \ldots, g_T$ is strictly decreasing due to $\gamma < 1$ and the gradient descent step which results in $g_t(D^{(t)}) \leq g_t(D^{(t-1)})$. In particular,

$$\begin{aligned} \frac{g_t - g_T}{g_{t-1} - g_T} \leq \frac{g_t}{g_{t-1}} &= \frac{g_t(D^{(t)})}{g_{t-1}(D^{(t-1)})} \\ &= \frac{g_t(D^{(t)})}{g_t(D^{(t-1)})} \cdot \frac{g_t(D^{(t-1)})}{g_{t-1}(D^{(t-1)})} \\ &\leq \frac{g_t(D^{(t)})}{g_t(D^{(t-1)})} \cdot \gamma \end{aligned} \tag{25}$$

However, same cannot be stated about the OE sequence $h_0, h_1, \ldots, h_T$ without further analysis. Our main goal in this subsection is to study the decrease or convergence rate of h_t which depends on the convergence rate of g_t as it is established by the following proposition.

Proposition 2. ***Suppose*** $\exists r \in (0,1)\colon g_t \leq r g_{t-1}$***. Then,***

$$h_t \leq r\left(\frac{1+\epsilon_N}{1-\epsilon_N}\right) h_{t-1} \tag{26}$$

Proof:

$$(1-\epsilon_N)h_t \leq g_t \leq r g_{t-1} \leq r(1+\epsilon_N)h_{t-1}$$

■

The following can be stated about the convergence rate of h_t:

$$\frac{h_t - h_T}{h_{t-1} - h_T} \leq \frac{h_t}{h_{t-1}} \leq \left(\frac{1+\epsilon_N}{1-\epsilon_N}\right) \frac{g_t}{g_{t-1}} \tag{27}$$

Proposition 2 asserts that convergence of the OE sequence is slower than the PE sequence by a factor that vanishes as ϵ_N approaches zero. In particular, $h_t < h_{t-1}$ when

$$\frac{g_t}{g_{t-1}} < \frac{1-\epsilon_N}{1+\epsilon_N}. \tag{28}$$

Letting

$$\gamma < \frac{1-\epsilon_N}{1+\epsilon_N} \tag{29}$$

would guarantee (28). However, it is crucial that g_t's decay rate at the inner layer (controlled by γ) is balanced with respect to its decay at the outer layer. For instance, an overly small γ results in the quick convergence of the sequence g_t which does not allow the dictionary to adapt itself to the data using the designated gradient descent step. On the other hand, (29) may not be satisfied if $\epsilon_N \approx 1$ due to insufficient training signals or an extremely small number of measurements per block.

As mentioned before and stated in (25), another factor that contributes to the reduction in g_t is the gradient descent step. Studying the decay rate of g_t under the gradient step is, unfortunately, a perplexing task since it depends on the data as well as the initialization point. More details follow.

C. Decay rate of g_t under the gradient descent step

The change in $g_t(D)$ under a single iteration of gradient descent depends on what is known as the *condition number* of the Hessian matrix of $g_t(D)$ at $D = D^{(t-1)}$, defined as:

$$\kappa_t = \frac{\lambda_{\max}(H_t)}{\lambda_{\min}(H_t)} \tag{30}$$

where H_t denotes the Hessian matrix that, as shown in detail in Appendix A, evaluates to:

$$H_t = \frac{2}{nN} \sum_{j=1}^{N} \alpha_j^{(t-1)} {\alpha_j^{(t-1)}}^T \otimes {\Phi_j}^T \Phi_j \tag{31}$$

Having κ_t, the convergence rate of $g_t(D)$ can be bounded [31] as:

$$\frac{g_t(D^{(t)}) - g_t^*}{g_t(D^{(t-1)}) - g_t^*} \leq 1 - \kappa_t^{-1} \tag{32}$$

where $g_t^* = \min_D g_t(D)$.

For the fastest decay, we must have $\kappa_t \approx 1$. However, since κ_t is data dependent, it may not be possible to bound it without imposing assumptions over the data. Instead, we measure its deviation *relative* to the oracle's condition number. To explain more, let $\bar{H}_t$ denote the (oracle) Hessian matrix of $h_t(D)$ at $D = D^{(t-1)}$. Specifically,

$$\bar{H}_t = \frac{2}{nN} \sum_{j=1}^{N} \alpha_j^{(t-1)} {\alpha_j^{(t-1)}}^T \otimes I_n, \tag{33}$$

and let $\bar{\kappa}_t$ denote its condition number. Using tools from the areas of random matrix theory and concentration of measure, mainly utilizing [36], we attemp to bound $|\kappa_t - \bar{\kappa}_t|$. We start by reviewing a crucial argument about the deviation of the extreme eigenvalues of H_t from those of $\bar{H}_t$.

Lemma 3. ***(Matrix Chernoff*** [36]***; Theorem 5.1.1). Consider a finite sequence*** $\{Q_k\} \subset \mathbb{R}^{d \times d}$ ***of random independent positive semidefinite Hermitian matrices that satisfy*** $\lambda_{\max}(Q_k) \leq \xi$ ***(almost surely). Define*** $S := \sum_k Q_k$***,*** $\mu_{\max} := \lambda_{\max}(\mathrm{E}\{S\})$ ***and*** $\mu_{\min} := \lambda_{\min}(\mathrm{E}\{S\})$***. Then, for*** $\forall\theta > 0$***:***

$$\mathrm{E}\{\lambda_{\max}(S)\} \leq \frac{e^\theta - 1}{\theta}\mu_{\max} + \frac{1}{\theta}\xi \log d \tag{34}$$

and

$$\mathrm{E}\{\lambda_{\min}(S)\} \geq \frac{1 - e^{-\theta}}{\theta}\mu_{\min} - \frac{1}{\theta}\xi \log d \tag{35}$$

Furthermore, for $\delta \geq 0$***:***

$$\Pr\{\lambda_{\max}(S) \geq (1+\delta)\mu_{\max}\} \leq d\left(\frac{e^\delta}{(1+\delta)^{(1+\delta)}}\right)^{\mu_{\max}/\xi} \tag{36}$$

and for $\delta \in [0,1)$***:***

$$\Pr\{\lambda_{\min}(S) \le (1-\delta)\mu_{\min}\} \le d\left(\frac{e^{-\delta}}{(1-\delta)^{(1-\delta)}}\right)^{\mu_{\min}/\xi} \quad (37)$$

We can customize the above result for our task which is summarized in the following lemma.

Lemma 4. ***For*** $\forall\delta \in [0,1)$***:***

$$\Pr\left\{\lambda_{\max}(H_t) \ge (1+\delta)\lambda_{\max}(\bar{H}_t)\right\} \le np\left(\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right)^{\lambda_{\max}(\bar{H}_t)/\xi_N} \quad (38)$$

and

$$\Pr\left\{\lambda_{\min}(H_t) \le (1-\delta)\lambda_{\min}(\bar{H}_t)\right\} \le np\left(\frac{e^{-\delta}}{(1-\delta)^{(1-\delta)}}\right)^{\lambda_{\min}(\bar{H}_t)/\xi_N} \quad (39)$$

with $\xi_N \le \frac{2}{N}(1+\sqrt{\frac{c}{m}})$***.***

Proof: Note that $E\{H_t\} = \bar{H}_t$. Furthermore,

$$\begin{aligned}\xi_N &= \frac{2}{nN}\lambda_{\max}(\alpha_j^{(t-1)}\alpha_j^{(t-1)^T} \otimes \Phi_j^T\Phi_j)\\ &= \frac{2}{nN}\lambda_{\max}(\alpha_j^{(t-1)}\alpha_j^{(t-1)^T})\lambda_{\max}(\Phi_j^T\Phi_j)\end{aligned}$$

It has been proved [33] that, for a Gaussian Φ_j,

$$\Pr\left\{\lambda_{\max}(\Phi_j^T\Phi_j) \ge (1+\zeta)\right\} \le 2e^{-m(\zeta^2/4-\zeta^3/6)} \quad (40)$$

Therefore, with a probability of at least $1-p_0$, we have

$$\lambda_{\max}(\Phi_j^T\Phi_j) \le 1+\sqrt{\frac{12}{m}\log(\frac{2}{p_0})} \quad (41)$$

Furthermore,

$$\frac{1}{n}\lambda_{\max}(\alpha_j^{(t-1)}\alpha_j^{(t-1)^T}) = \frac{1}{n}\|\alpha_j^{(t-1)}\|_2^2 \le 1 \quad (42)$$

Hence, $\xi_N \le \frac{2}{N}(1+\sqrt{\frac{12}{m}\log(\frac{2}{p_0})})$ with probability $1-p_0$. Proof is completed using (36) and (37).

■

Define

$$p_1 := np\left(\frac{e^{\delta_N}}{(1+\delta_N)^{(1+\delta_N)}}\right)^{\lambda_{\max}(\bar{H}_t)/\xi_N} \quad (43)$$

$$p_2 := np\left(\frac{e^{-\delta_N}}{(1-\delta_N)^{(1-\delta_N)}}\right)^{\lambda_{\min}(\bar{H}_t)/\xi_N} \quad (44)$$

Taking natural logarithm of the above equalities and some trivial manipulation, we arrive at

$$\delta_N = \sqrt{\frac{2\log(np)}{N}}\max\left(\sqrt{\frac{\log(\frac{1}{p_1})}{\lambda_{\max}(\bar{H}_t)}}, \sqrt{\frac{\log(\frac{1}{p_2})}{\lambda_{\min}(\bar{H}_t)}}\right) \quad (45)$$

Therefore, $\delta_N = O(N^{-\frac{1}{2}})$ for fixed p_1 and p_2. The following proposition is based on a simple union bound argument where it is assumed that the perturbations in $\lambda_{\max}(H_t)$ and $\lambda_{\min}(H_t)$ are independent. It must be noted that the independency assumption corresponds to the worst-case scenario for the perturbation of the ratio $\lambda_{\max}(H_t)/\lambda_{\min}(H_t)$.

Proposition 5. ***For*** $\forall\delta_N \in [0,1)$***:***

$$\Pr\left\{\kappa_t \ge \frac{1+\delta_N}{1-\delta_N}\bar{\kappa}_t\right\} \le p_1+p_2-p_1p_2 \quad (46)$$

I.e., with a probability of $(1-p_1)(1-p_2)$***, for*** δ_N ***provided in (45):***

$$\kappa_t \le \left(\frac{1+\delta_N}{1-\delta_N}\right)\bar{\kappa}_t \quad (47)$$

In conclusion, having $\bar{\kappa}_t \approx 1$ guarantees that $\kappa_t \approx 1$ with an overwhelming probability for large N. Likewise, having $\bar{\kappa}_t \gg 1$ could result in $\kappa_t \gg 1$ and thus a slow convergence of the gradient descent. In essence, instead of imposing assumptions directly over the data, we evaluated the convergence *relative* to the oracle convergence. More on this is provided in the last part of this section.

D. Non-asymptotic bounds for ϵ_N

Our analysis in this subsection is based on an extension of the well-known Johnson-Lindenstrauss (a.k.a. the stable embedding) lemma [38] for block-diagonal random measurement matrices [39].

Theorem 6. [39] ***Let*** m_j ***denote the number of measurements from the block*** x_j***. Define the sub-Gaussian norm of the r.v.*** ϕ ***as*** $\|\phi\|_{\psi_2} := \sup_{p\ge1} p^{-1/2}(E\{|\phi|^p\})^{1/p}$***. Let*** ϕ ***denote a sub-Gaussian random variable with zero mean and unit variance and let*** $\{\Phi_j\}_{j=1}^N$ ***denote the set of random matrices drawn independently, where each*** $\Phi_j \in \mathbb{R}^{m_j\times n}$ ***is populated with i.i.d. realizations of the renormalized random variable*** $\frac{\phi}{\sqrt{m_j}}$***. Furthermore, define:***

$$\Gamma_2 := \frac{\left(\sum_{j=1}^N \|x_j\|_2^2\right)^2}{\sum_{j=1}^N \frac{\|x_j\|_2^4}{m_j}}, \quad \Gamma_\infty := \frac{\sum_{j=1}^N \|x_j\|_2^2}{\max_j \frac{\|x_j\|_2^2}{m_j}} \quad (48)$$

Then,

$$\Pr\left\{\left|\sum_{j=1}^N \|\Phi_j x_j\|_2^2 - \sum_{j=1}^N \|x_j\|_2^2\right| > \epsilon\sum_{j=1}^N \|x_j\|_2^2\right\} \le 2\exp\left[-c_1\min\left(\frac{c_2^2\epsilon^2}{\|\phi\|_{\psi_2}^4}\Gamma_2, \frac{c_2\epsilon}{\|\phi\|_{\psi_2}^2}\Gamma_\infty\right)\right] \quad (49)$$

where c_1 ***and*** c_2 ***are positive constants.***

The above theorem can be customized for the class of Gaussian measurements. Specifically, if $\phi \sim N(0,1)$, $\|\phi\|_{\psi_2} = \sqrt{2/\pi}$ [40]. Using the proof of Theorem 6 in [39], one can show that $c_2 = \frac{1}{4}$ for a standard normal ϕ. Hence, the right hand side of the inequality in (49) becomes:

$$2\exp\left[-c\epsilon\min\left(\frac{\pi}{8}\epsilon\Gamma_2, \Gamma_\infty\right)\right] \quad (50)$$

Note that $\Gamma_\infty \le \Gamma_2$ (which can be shown by multiplying both the numerator and the denominator of Γ_∞ with $\sum_{j=1}^N \|x_j\|_2^2$). Therefore, whether $\frac{\pi}{8}\epsilon\Gamma_2$ is smaller or Γ_∞ in

(50) is not absolute and depends on the data. Also, note that both Γ_2 and Γ_∞ grow linearly with m and N as long as $\|x_j\|_2 > 0, \forall j \in [N]$.

The above theorem measures the concentration of the signal energy. However, we are interested to see how well Gaussian measurements preserve the energy of the error $x_j - D\alpha_j$. Below, we present an adaptation of Theorem 6 for the purposes of this work.

Theorem 7. $\forall D \in \mathcal{D}, \epsilon \in [0,1]$:

$$\Pr\left\{\frac{|g_t(D) - h_t(D)|}{h_t(D)} > \epsilon\right\} \leq 2\exp\left[-c\epsilon \min\left(\frac{\pi}{8}\epsilon\bar{\Gamma}_2, \bar{\Gamma}_\infty\right)\right] \tag{51}$$

where

$$\bar{\Gamma}_2 := m\frac{\left(\sum_{j=1}^N \|x_j - D\alpha_j^{(t-1)}\|_2^2\right)^2}{\sum_{j=1}^N \|x_j - D\alpha_j^{(t-1)}\|_2^4} \tag{52}$$

$$\bar{\Gamma}_\infty := m\frac{\sum_{j=1}^N \|x_j - D\alpha_j^{(t-1)}\|_2^2}{\max_j \|x_j - D\alpha_j^{(t-1)}\|_2^2} \tag{53}$$

If the error energy is evenly distributed among the blocks, then $\bar{\Gamma}_\infty = \bar{\Gamma}_2 = mN$. While such scenario is unlikely, we can always write $\bar{\Gamma}_\infty \geq mN\eta_t$ and $\bar{\Gamma}_2 \geq mN\eta_t^2$ where

$$\eta_t := \frac{\inf\{\|x_j - \Phi_j D\alpha_j^{(t-1)}\|_2^2\}_{j=1}^N}{\sup\{\|x_j - \Phi_j D\alpha_j^{(t-1)}\|_2^2\}_{j=1}^N} \tag{54}$$

is strictly positive. Note that $\|x_j - \Phi_j D\alpha_j^{(t-1)}\|_2 > 0, \forall j$ (and thus $\eta_t > 0$) due to the algorithm initialization which balances the error and the sparsity using a Lasso optimization. Over the course of iterations, the error energies may become unbalanced which would have an adverse effect on the size of ϵ_N. However, as some of the error energies start to vanish, the algorithm starts to converge and avoid overfitting[3].

In conclusion, under a fixed probability of having

$$(1 - \epsilon_N)h_t \leq g_t \leq (1 + \epsilon_N)h_t, \tag{55}$$

one of the following occurs (depending on the data):

- $\epsilon_N = O(N^{-1}m^{-1})$
- $\epsilon_N = O(N^{-\frac{1}{2}}m^{-\frac{1}{2}})$

In both scenarios, ϵ_N decays significantly by training over a large number of blocks even when the number of measurements per block is not enough to guarantee a stable CS.

E. Putting it all together

We conclude the analysis by putting together the results of the previous subsections. To reduce the clutter and make the analysis simpler, we make the following approximations involving ϵ_N (and δ_N) for sufficiently large N [4]:

$$(1 + \epsilon_N)(1 - \epsilon_N)^{-1} \approx 1 + 2\epsilon_N \tag{56}$$

$$(1 - \epsilon_N)(1 + \epsilon_N)^{-1} \approx 1 - 2\epsilon_N \tag{57}$$

Recall the following relationship between decay rates of h_t and g_t:

$$\frac{h_t - h_T}{h_{t-1} - h_T} \leq \frac{h_t}{h_{t-1}} \tag{58}$$

$$\leq \left(\frac{1 + \epsilon_N}{1 - \epsilon_N}\right)\frac{g_t}{g_{t-1}} \tag{59}$$

$$\leq \left(\frac{1 + \epsilon_N}{1 - \epsilon_N}\right)\gamma\frac{g_t(D^{(t)})}{g_t(D^{(t-1)})} \tag{60}$$

Rearrange (32) into:

$$\frac{g_t(D^{(t)})}{g_t(D^{(t-1)})} \leq 1 - \kappa_t^{-1}\left(\frac{g_t(D^{(t-1)}) - g_t^*}{g_t(D^{(t-1)})}\right) \tag{61}$$

The ratio $(g_t(D^{(t-1)}) - g_t^*)/g_t(D^{(t-1)})$ corresponds to the amount of 'room for improvement'; it quantifies the amount by which $g_t(D)$ can be decreased (under any optimization method). The presence of this quantity in our analysis is inevitable since we are interested in the ratio of $g_t(D^{(t)})$ and $g_t(D^{(t-1)})$, rather than the ratio of their distances from g_t^*. The ratio $g_t(D^{(t)})/g_t(D^{(t-1)})$ is guaranteed to be small given that the Hessian matrix is well-conditioned ($\kappa_t = \kappa(H_t) \approx 1$) and the current function value is away from the optimal value.

Recall that, with a (constant) high probability,

$$\begin{aligned}
&\kappa_t^{-1}\left(1 - \frac{g_t^*}{g_t(D^{(t-1)})}\right) \\
&\geq \left(\frac{1 - \delta_N}{1 + \delta_N}\right)\bar{\kappa}_t^{-1}\left(1 - \left(\frac{1 + \epsilon_N}{1 - \epsilon_N}\right)\frac{h_t^*}{h_t(D^{(t-1)})}\right) \\
&\approx (1 - 2\delta_N)\,\bar{\kappa}_t^{-1}\left(1 - (1 + 2\epsilon_N)\,\frac{h_t^*}{h_t(D^{(t-1)})}\right) \\
&\approx \left(1 - \frac{h_t^*}{h_t(D^{(t-1)})}\right)\bar{\kappa}_t^{-1} \\
&\qquad - 2\epsilon_N\frac{h_t^*}{h_t(D^{(t-1)})}\bar{\kappa}_t^{-1} - 2\delta_N\left(1 - \frac{h_t^*}{h_t(D^{(t-1)})}\right)\bar{\kappa}_t^{-1} \\
&= \left(1 - \frac{h_t^*}{h_t(D^{(t-1)})}\right)\bar{\kappa}_t^{-1} - 2\xi_N
\end{aligned}$$

where

$$\xi_N := \bar{\kappa}_t^{-1}\left[\epsilon_N\frac{h_t^*}{h_t(D^{(t-1)})} + \delta_N\left(1 - \frac{h_t^*}{h_t(D^{(t-1)})}\right)\right]$$

Note that ξ_N, having a convex combination of ϵ_N and δ_N as one of its factors, is of order $O(N^{-\frac{1}{2}})$. Next, we combine the above result with (60) and (61).

$$\begin{aligned}
\frac{h_t}{h_{t-1}} &\leq \left(\frac{1 + \epsilon_N}{1 - \epsilon_N}\right)\gamma\frac{g_t(D^{(t)})}{g_t(D^{(t-1)})} \\
&\leq (1 + 2\epsilon_N)\gamma\left[1 - \left(1 - \frac{h_t^*}{h_t(D^{(t-1)})}\right)\bar{\kappa}_t^{-1} + 2\xi_N\right] \\
&\approx \gamma\left[1 - \left(1 - \frac{h_t^*}{h_t(D^{(t-1)})}\right)\bar{\kappa}_t^{-1}\right] \\
&\qquad + 2\gamma\epsilon_N\left[1 - \left(1 - \frac{h_t^*}{h_t(D^{(t-1)})}\right)\bar{\kappa}_t^{-1}\right] + 2\gamma\xi_N
\end{aligned} \tag{62}$$

It is not difficult to see that the first term of (62), i.e.

$$\gamma\left[1 - \left(1 - \frac{h_t^*}{h_t(D^{(t-1)})}\right)\bar{\kappa}_t^{-1}\right]$$

[3] Note that large values of ϵ_N result in overfitting as the learned parameters cannot be 'generalized' to the unseen part of the data. The cross-validation strategy that is explained in the following section would also prevent overfitting in case ϵ_N becomes significant.

[4] Recall that $\epsilon_N = O(N^{-1})$ or $\epsilon_N = O(N^{-\frac{1}{2}})$ and $\delta_N = O(N^{-\frac{1}{2}})$.

corresponds to JEDI's convergence rate if the measurements where complete. Meanwhile, the second term, i.e.

$$2\gamma\epsilon_N \left[1 - \left(1 - \frac{h_t^*}{h_t(D^{(t-1)})}\right)\bar{\kappa}_t^{-1}\right] + 2\gamma\xi_N$$

can be regarded as the penalty associated with using incomplete measurements which is of order $O(N^{-\frac{1}{2}})$.

In conclusion, we have demonstrated that the fluctuation in the oracle error, which is inversely related to the sparse recovery PSNR, is in the order $O(N^{-\frac{1}{2}})$. This means that, for a reasonably large N, one can achieve a significant boost in the recovery PSNR using only a few iterations of the JEDI algorithm.

IV. Simulations and Results

A. Simulation settings

The set of 12 test images, which are down-scaled due to the limited space, are shown in Fig. 1. Each test image has a resolution of 512×512. In all simulations, we have used a fixed block size of 8×8. Input parameters to Alg. 1 are $\gamma = 0.95$, $\lambda_0 = 0.001, 0.005, 0.01, 0.05$ and $T = 40$. Regarding the choice of $D^{(0)}$ and the measurement matrix, there are four different sets of simulations that are described below.

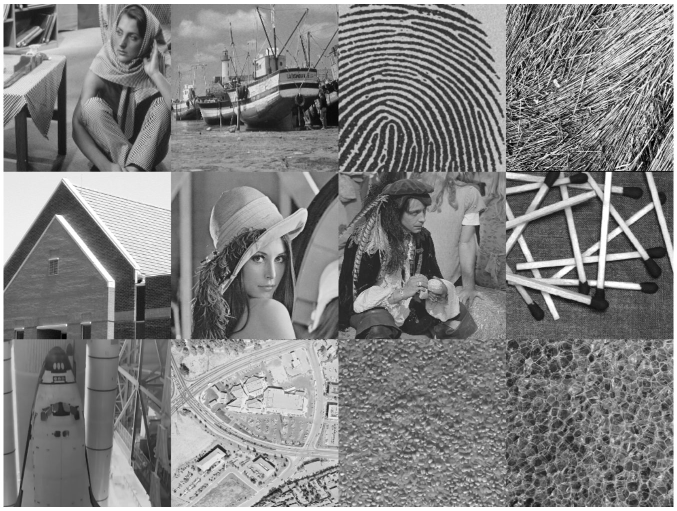

Fig. 1. Name of the test images from left to right and top to bottom: 'Barbara', 'boat', 'fingerprint', 'grass', 'house', 'Lena', 'man', 'matches', 'shuttle', 'aerial', 'barley' and 'bubbles'.

SET1: In this setting, the initial dictionary is computed by cross-validation. Specifically, for each test image in Fig. 1, 10^6 training patches were randomly selected from the other images and fed to the online DL algorithm in [23] for training the initial dictionary with $p = 256$; this results in a distinct initial dictionary for each test image[5]. Gaussian measurements were utilized for this simulation.

SET2: In this setting, for the initial dictionary, we have used a redundant dictionary that was trained over a large set of training images using the K-SVD method [7]. Testing with the K-SVD dictionary helps in benchmarking the performance of the proposed method. Similar to the previous case, Gaussian measurements were utilized in this simulation.

SET3: In addition to testing JEDI with overcomplete dictionaries, in this setting we test the algorithm with an orthogonal Discrete Cosine Transform (DCT). The advantage of using an orthogonal matrix as the initial dictionary is the lower cost of storing the dictionary and a faster running time. Gaussian measurements were utilized as before.

SET4: Finally, we test JEDI for image inpainting under the extreme conditions of having 80%, 65% and 50% of the pixels missing. Sampling a random subset of pixels corresponds to the most practical CS scheme for images. However, Φ_j would not satisfy the mentioned sub-Gaussian concentration inequalities, resulting in weaker recovery guarantees.

B. Using a validation measurement to prevent overfitting

In all of our simulations, a single measurement per block was reserved for the sole purpose of cross-validation that is explained below. Note that the total number of measurements per block (i.e. the sampling rate) does not change.

Denote the validation measurement by $v_j = \phi_j^T x_j \in \mathbb{R}$ where $\phi_j \sim N_n(0, I_n)$. The validation error is computed as:

$$h_t^v = \frac{1}{nN}\sum_{j=1}^{N} \|v_j - \phi_j D^{(t)}\alpha_j^{(t-1)}\|_2^2 \tag{63}$$

which approximates h_t. However, unlike g_t, h_t^v is not directly utilized by the algorithm for training and, therefore, can be used for detection of overfitting. Specifically, the algorithm can be stopped after observing consecutive increases in the value of h_t^v. Also, the validation measurement is utilized for selecting the best λ_0 (in both CS and BCS). In our application, we found that a single validation measurement is sufficient.

C. Results and discussion

The recovery PSNR results for every setting, averaged over 10 trials for each case, are presented in Tables I through IV. Since the block size is $n = 64$, sampling ratios 20%, 35% and 50% respectively correspond to $m = 12$, $m = 22$ and $m = 32$ measurements per block.

As can be seen in Tables I through IV, improvements in the recovery PSNRs of BCS are highly dependent on the image and the choice of dictionary initialization. In most cases, the PSNR gain increases with m. This behavior could be due to the fact that ϵ_N and δ_N become smaller with more measurements and JEDI gets closer to a typical DL algorithm. Meanwhile, it is crucial that the gain stays positive for all sampling ratios.

Unfortunately, there are no open-source BCS software packages with adjustable measurement and initialization parameters. Our inpainting results (**SET4**) can be compared to the BCS results reported in [10] which uses the block-sparse signal model. In [10], the recovery PSNR for Barbara's image with $n = 64$, $p = 256$ and $m = 32$ was reported at 27.93 dB which is 0.25 dB higher than JEDI's recovery. However, in addition to employing the block-sparse model which works well for a subset of natural images, the proposed algorithm in [10] uses

[5] The parameters for the online DL algorithm were set at their default values, as suggested in [22]. Specifically, we have set the maximum number of DL iterations at 1000, the mini-batch size at 400 patches and $\lambda_1 = 0.15$

TABLE I

Recovery PSNRs for **SET1**. PSNRs are in dB. Percentage values are sampling ratios (m/n).

	Universal dictionary			Adaptive dictionary			PSNR gain		
	20%	35%	50%	20%	35%	50%	20%	35%	50%
Barbara	23.31	25.39	27.37	23.89	27.56	31.07	0.58	2.17	3.70
boat	25.75	29.16	31.77	26.15	29.45	32.31	0.40	0.29	0.53
fingerprint	21.73	25.35	27.50	28.19	33.93	38.72	6.46	8.58	11.22
grass	14.01	16.55	18.91	14.38	17.15	19.67	0.37	0.60	0.75
house	26.99	31.13	34.56	29.05	34.10	36.95	2.06	2.97	2.40
Lena	28.67	32.64	35.49	29.15	32.78	35.61	0.49	0.14	0.12
man	26.66	29.78	32.14	26.80	29.86	32.21	0.14	0.08	0.07
matches	25.21	28.80	31.25	26.10	29.10	31.44	0.89	0.30	0.19
shuttle	31.50	38.04	43.23	33.85	39.77	44.06	2.34	1.74	0.84
aerial	22.84	26.17	29.03	22.82	26.17	29.09	-0.02	0.00	0.06
barley	25.37	31.02	36.12	26.90	32.89	37.86	1.53	1.87	1.74
bubbles	22.63	25.74	28.43	22.68	25.75	28.44	0.05	0.01	0.01

TABLE II

Recovery PSNRs for **SET2**. PSNRs are in dB. Percentage values are sampling ratios (m/n).

	Universal dictionary			Adaptive dictionary			PSNR gain		
	20%	35%	50%	20%	35%	50%	20%	35%	50%
Barbara	23.34	25.36	27.45	23.77	27.22	30.72	0.44	1.86	3.27
boat	25.89	29.45	32.20	26.10	29.63	32.45	0.21	0.18	0.25
fingerprint	22.25	26.83	30.38	28.49	34.81	39.96	6.24	7.98	9.58
grass	13.80	16.21	18.75	14.19	16.92	19.57	0.40	0.70	0.82
house	27.03	31.58	35.11	29.15	33.98	36.98	2.11	2.40	1.87
Lena	28.60	32.52	35.41	29.15	32.67	35.39	0.54	0.15	-0.02
man	26.61	29.88	32.51	26.77	29.96	32.60	0.16	0.08	0.09
matches	25.26	28.56	31.19	26.22	29.10	31.51	0.96	0.54	0.33
shuttle	31.89	38.57	43.45	33.63	40.02	44.25	1.74	1.45	0.80
aerial	22.76	26.18	29.31	22.87	26.27	29.37	0.11	0.09	0.06
barley	25.15	30.44	35.28	26.56	32.35	37.34	1.41	1.91	2.05
bubbles	22.41	25.59	28.53	22.64	25.60	28.57	0.24	0.02	0.04

TABLE III

Recovery PSNRs for **SET3**. PSNRs are in dB. Percentage values are sampling ratios (m/n).

	Universal dictionary			Adaptive dictionary			PSNR gain		
	20%	35%	50%	20%	35%	50%	20%	35%	50%
Barbara	22.66	25.27	28.15	24.22	27.97	31.43	1.57	2.70	3.27
boat	23.70	26.49	29.40	26.07	29.34	32.30	2.37	2.85	2.90
fingerprint	19.53	23.33	26.84	26.63	31.02	35.09	7.10	7.69	8.25
grass	12.43	14.28	16.54	14.17	16.65	19.12	1.74	2.37	2.58
house	24.55	27.81	31.46	28.53	33.15	36.13	3.99	5.34	4.67
Lena	25.91	29.07	32.24	28.96	32.30	35.06	3.06	3.23	2.82
man	24.41	26.96	29.54	26.64	29.26	31.68	2.23	2.30	2.14
matches	22.83	26.04	28.76	26.15	29.06	31.34	3.32	3.02	2.58
shuttle	28.32	33.01	37.90	33.50	39.07	42.84	5.18	6.06	4.94
aerial	20.60	22.89	25.68	22.79	25.76	28.58	2.19	2.87	2.90
barley	22.36	26.11	30.40	27.44	32.81	36.49	5.08	6.71	6.09
bubbles	20.57	22.76	25.38	22.85	25.59	28.09	2.28	2.83	2.71

TABLE IV

Recovery PSNRs for **SET4**. PSNRs are in dB. Percentage values are sampling ratios (m/n).

	Universal dictionary			Adaptive dictionary			PSNR gain		
	20%	35%	50%	20%	35%	50%	20%	35%	50%
Barbara	23.08	25.02	26.74	23.50	25.61	27.68	0.42	0.59	0.94
boat	24.85	27.53	29.56	25.75	28.94	31.58	0.90	1.41	2.02
fingerprint	21.07	24.61	26.92	21.93	25.96	29.05	0.86	1.34	2.13
grass	13.93	16.32	18.62	13.97	16.41	18.78	0.04	0.09	0.16
house	25.80	28.67	30.76	26.94	30.74	33.86	1.14	2.07	3.10
Lena	26.89	29.86	31.86	28.25	31.95	34.65	1.36	2.09	2.79
man	25.48	27.92	29.77	26.53	29.48	31.90	1.05	1.56	2.12
matches	23.80	26.63	28.42	24.92	28.35	30.88	1.12	1.72	2.47
shuttle	28.73	32.29	34.81	30.93	35.92	39.68	2.19	3.62	4.86
aerial	22.10	24.60	26.80	22.59	25.64	28.44	0.49	1.04	1.65
barley	23.32	26.73	30.06	24.78	29.30	33.12	1.46	2.57	3.06
bubbles	21.74	24.26	26.32	22.42	25.40	28.08	0.68	1.14	1.76

overlapping block-CS recovery which significantly increases N. This performance boost comes at the cost of a higher computational complexity. Meanwhile, we have employed a non-overlapping framework to stay consistent with the general block-CS recovery where Φ_j may be dense and overlapping block recovery may not be feasible.

V. Conclusion

The analysis and simulation results presented in this paper show that BCS of natural images is both practical and reliable, without the need for additional constraints over the dictionary or the sparse coefficients. We proposed and studied the convergence rate of a practical iterative BCS algorithm, termed JEDI for Joint Estimation of Dictionary and Image. In this framework, the incompleteness of block-wise measurements is rectified by combining a large number of blocks, each with a different random measurement matrix.

Appendix A
Computing the Hessian matrix H_t

First, we write $g_t(D)$ in the standard quadratic format:

$$\begin{aligned} g_t(D) &= \frac{1}{nN}\sum_{j=1}^{N} \|y_j - \Phi_j D\alpha_j^{(t-1)}\|_2^2 \\ &= \frac{1}{nN}\sum_{j=1}^{N} y_j^T y_j + \frac{1}{nN}\sum_{j=1}^{N} \alpha_j^{(t-1)^T} D^T \Phi_j^T \Phi_j D \alpha_j^{(t-1)} \\ &- \frac{2}{nN}\sum_{j=1}^{N} y_j^T \Phi_j D \alpha_j^{(t-1)} \end{aligned}$$

We can further write:

$$\begin{aligned} &\alpha_j^{(t-1)^T} D^T \Phi_j^T \Phi_j D \alpha_j^{(t-1)} \\ &\quad = \mathrm{Tr}(\alpha_j^{(t-1)^T} D^T \Phi_j^T \Phi_j D \alpha_j^{(t-1)}) \\ &\quad = \mathrm{Tr}(D^T \Phi_j^T \Phi_j D \alpha_j^{(t-1)} \alpha_j^{(t-1)^T}) \\ &\quad = \langle D, \Phi_j^T \Phi_j D \alpha_j^{(t-1)} \alpha_j^{(t-1)^T} \rangle \\ &\quad = \mathrm{vec}(D)^T \mathrm{vec}(\Phi_j^T \Phi_j D \alpha_j^{(t-1)} \alpha_j^{(t-1)^T}) \\ &\quad = \mathrm{vec}(D)^T (\alpha_j^{(t-1)} \alpha_j^{(t-1)^T} \otimes \Phi_j^T \Phi_j) \,\mathrm{vec}(D) \end{aligned}$$

and

$$\begin{aligned} y_j^T \Phi_j D \alpha_j^{(t-1)} &= \mathrm{Tr}(y_j^T \Phi_j D \alpha_j^{(t-1)}) \\ &= \mathrm{Tr}(\alpha_j^{(t-1)} y_j^T \Phi_j D) \\ &= \langle \Phi_j^T y_j \alpha_j^{(t-1)^T}, D \rangle \\ &= \mathrm{vec}(\Phi_j^T y_j \alpha_j^{(t-1)^T})^T \,\mathrm{vec}(D) \end{aligned}$$

Letting $\boldsymbol{d} = \mathrm{vec}(D)$, the standard quadratic form of $g_t(D)$ can be written as:

$$g(\boldsymbol{d}) = \frac{1}{2}\boldsymbol{d}^T H_t \boldsymbol{d} + f_t^T \boldsymbol{d} + c \tag{64}$$

with

$$\begin{aligned} H_t &= \frac{2}{nN}\sum_{j=1}^{N} \alpha_j^{(t-1)} \alpha_j^{(t-1)^T} \otimes \Phi_j^T \Phi_j \\ f_t &= -\frac{2}{nN}\,\mathrm{vec}\left(\sum_{j=1}^{N} \Phi_j^T y_j \alpha_j^{(t-1)^T}\right) \\ c &= \frac{1}{nN}\sum_{j=1}^{N} y_j^T y_j \end{aligned}$$

References

[1] E. J. Candes, J. Romberg and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, 52(2), pp. 489-509, February 2006.

[2] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, 52(4), pp. 1289-1306, April 2006.

[3] E. J. Candes, "The restricted isometry property and its implications for compressed sensing," *Compte Rendus de lacademie des Sciences*, Series I, vol. 346, no. 9-10, pp. 589-592, May 2008.

[4] Emmanuel J. Candes, Yonina C. Eldar, Deanna Needell and Paige Randall, "Compressed sensing with coherent and redundant dictionaries," *Applied and Computational Harmonic Analysis*, vol. 31(1), pp. 59-73, 2011.

[5] D. L. Donoho and I. M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, no. 3, pp. 425-455, September 1994.

[6] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1?," *Vis. Res.*, vol. 37, pp. 311-325, 1997.

[7] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," IEEE *Transactions on Image Processing*, vol.15, no.12, pp.3736-3745, Dec 2006.

[8] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Transactions on Image Processing*, 17(1):53-69, January 2008.

[9] S. Gleichman and Y. Eldar, "Blind compressed sensing," *IEEE Transactions on Information Theory*, vol. 57, no. 10, pp. 6958-6975, 2011.

[10] J. Silva, M. Chen, Y. C. Eldar, G. Sapiro and L. Carin, "Blind compressed sensing over a structured union of subspaces," arXiv:1103.2469v1, 2011.

[11] M. Aghagolzadeh and H. Radha, "New Guarantees for Blind Compressed Sensing," *in the 53rd Annual Allerton Conference on Communication, Control and Computing*, 2015.

[12] F. Pourkamali Anaraki and S.M. Hughes, "Compressive K-SVD," *IEEE International Conference on Acoustics, Speech and Signal Processing* (ICASSP), pp.5469-5473, 2013.

[13] F. Pourkamali-Anaraki, Stephen Becker and Shannon M. Hughes, "Efficient Dictionary Learning via Very Sparse Random Projections," *arXiv:1504.01169*, 2015.

[14] C. Studer and R. Baraniuk, "Dictionary learning from sparsely corrupted or compressed signals," in *IEEE International Conference on Acoustics, Speech and Signal Processing* (ICASSP), pp. 3341-3344, 2012.

[15] Z. Xing, M. Zhou, A. Castrodad, G. Sapiro and L. Carin, "Dictionary learning for noisy and incomplete hyperspectral images," *SIAM Journal on Imaging Sciences*, vol. 5, no. 1, pp. 33-56, 2012.

[16] M. Aghagolzadeh and H. Radha, "Adaptive Dictionaries for Compressive Imaging," *IEEE Global Conference on Signal and Information Processing* (GlobalSIP), December 2013.

[17] B. Recht, "A simpler approach to matrix completion," arXiv:0910.0651, 2009.

[18] L. Gan, "Block compressed sensing of natural images," in *IEEE 15th International Conference on Digital Signal Processing*, pp. 403-406, 2007.

[19] S. Mun and J. E. Fowler, "Block compressed sensing of images using directional transforms," in *Proceedings of the International Conference on Image Processing*, pp. 3021-3024, 2009.

[20] B. Efron, T. Hastie, I. Johnstone and R. Tibshirani, "Least angle regression," *Annals of Statistics*, 32, 407-499, 2004.

[21] R. Gribonval and K. Schnass, "Dictionary identification-sparse matrix-factorisation via l1-minimisation," *IEEE Transactions on Information Theory*, 56(7):3523-3539, 2010.

[22] J. Mairal, F. Bach, J. Ponce, G. Sapiro and A. Zisserman, “Non-local sparse models for image restoration,” *In Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2009.

[23] J. Mairal, F. Bach, J. Ponce and G. Sapiro, “Online learning for matrix factorization and sparse coding,” *Journal of Machine Learning Research*, vol. 11, pp. 19-60, 2010.

[24] M. Aharon and M. Elad, “Sparse and redundant modeling of image content using an image signature dictionary,” *SIAM Journal on Imaging Sciences*, 1(3):228-247, July 2008.

[25] K. Engan, S. O. Aase and J. H. Hakon-Husoy, “Method of optimal directions for frame design,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, pp. 2443-2446, 1999.

[26] K. Engan, K. Skretting and J. H. Husoy, “A family of iterative LS-based dictionary learning algorithms, ILS-DLA, for sparse signal representation,” *Digital Signal Processing*, vol. 17, pp. 32-49, Jan. 2007.

[27] K. Skretting and K. Engan, “Recursive least squares dictionary learning algorithm,” *IEEE Transactions on Signal Processing*, vol. 58, no. 4, pp. 2121-2130, 2010.

[28] K. Kreutz-Delgado, J. F. Murray, B. D. Rao, K. Engan, T. Lee and T. J. Sejnowski, “Dictionary learning algorithms for sparse representation,” *Neural Computation*, vol. 15, no. 2, pp. 349-396, 2003.

[29] M. Yaghoobi, T. Blumensath and M.E. Davies, “Regularized dictionary learning for sparse approximation,” In *Proceedings of EUSIPCO*, 2008.

[30] H. Raja and W. U. Bajwa, “Cloud K-SVD: a collaborative dictionary learning algorithm for big, distributed data,” CoRR, abs/1412.7839, 2014.

[31] J. Nocedal and S.J. Wright, “Numerical Optimization,” *Springer*, New York, 1999.

[32] A. M. Bruckstein, D. L. Donoho and M. Elad, “From sparse solutions of systems of equations to sparse modeling of signals and images,” *SIAM Review*, vol. 51, no. 1, pp. 34-81, 2009.

[33] R. Baraniuk, M. Davenport, R. DeVore and M. Wakin, “A simple proof of the restricted isometry property for random matrices,” *Constructive Approximation*, vol. 28, no. 3, pp. 253-263, 2008.

[34] P. Boufounos, M. Duarte and R. Baraniuk, “Sparse signal reconstruction from noisy compressive measurements using cross validation,” *in Proc. of the IEEE Workshop on Statistical Signal Processing*, 2007.

[35] R. Ward, “Compressive sensing with cross validation,” *IEEE Transactions on Information Theory*, 55(12):5773?5782, 2009.

[36] Joel A. Tropp, “User-Friendly Tools for Random Matrices: An Introduction,” NIPS, 2012.

[37] Ryan J. Tibshirani, “The lasso problem and uniqueness,” *Electron. J. Stat.*, 7, pp. 1456-1490, 2013.

[38] S. Dasgupta and A. Gupta, “An elementary proof of a theorem of Johnson and Lindenstrauss,” *Random Structures and Algorithms*, vol. 22, pp. 6065, 2003.

[39] J.Y. Park, H.L. Yap, C.J. Rozell and M.B.Wakin, “Concentration of measure for block diagonal matrices with applications to compressive signal processing,” *IEEE Transactions on Signal Processing*, 59(12):5859-5875, 2011.

[40] R. Vershynin, “Introduction to the non-asymptotic analysis of random matrices,” *Arxiv preprint*, arXiv:1011.3027, 2010.

[41] R. DeVore, “Deterministic constructions of compressed sensing matrices,” *Journal of Complexity*, vol. 23, pp. 918-925, 2007.