Journal of Machine Learning Research ? (2016) 1-50 Submitted ?/?; Published ?/?

Convolutional Neural Networks Analyzed via Convolutional Sparse Coding

Vardan Papyan* VARDANP@CAMPUS.TECHNION.AC.IL
Department of Computer Science
Technion - Israel Institute of Technology
Technion City, Haifa 32000, Israel

Yaniv Romano* YROMANO@TX.TECHNION.AC.IL
Department of Electrical Engineering
Technion - Israel Institute of Technology
Technion City, Haifa 32000, Israel

Michael Elad ELAD@CS.TECHNION.AC.IL
Department of Computer Science
Technion - Israel Institute of Technology
Technion City, Haifa 32000, Israel

Editor: ?

Abstract

Convolutional neural networks (CNN) have led to many state-of-the-art results spanning through various fields. However, a clear and profound theoretical understanding of the forward pass, the core algorithm of CNN, is still lacking. In parallel, within the wide field of sparse approximation, Convolutional Sparse Coding (CSC) has gained increasing attention in recent years. A theoretical study of this model was recently conducted, establishing it as a reliable and stable alternative to the commonly practiced patch-based processing. Herein, we propose a novel multi-layer model, ML-CSC, in which signals are assumed to emerge from a cascade of CSC layers. This is shown to be tightly connected to CNN, so much so that the forward pass of the CNN is in fact the thresholding pursuit serving the ML-CSC model. This connection brings a fresh view to CNN, as we are able to attribute to this architecture theoretical claims such as uniqueness of the representations throughout the network, and their stable estimation, all guaranteed under simple local sparsity conditions. Lastly, identifying the weaknesses in the above pursuit scheme, we propose an alternative to the forward pass, which is connected to deconvolutional, recurrent and residual networks, and has better theoretical guarantees.

Keywords: Deep Learning, Convolutional Neural Networks, Forward Pass, Sparse Representation, Convolutional Sparse Coding, Thresholding Algorithm, Basis Pursuit

∗. The authors contributed equally to this work.

1. Introduction

Deep learning (LeCun et al., 2015), and in particular CNN (LeCun et al., 1990, 1998; Krizhevsky et al., 2012), has gained a copious amount of attention in recent years as it has led to many state-of-the-art results spanning through many fields – including speech recognition (Bengio et al., 2003; Hinton et al., 2012; Mikolov et al., 2013), computer vision (Farabet et al., 2013; Simonyan and Zisserman, 2014; He et al., 2015), signal and image processing (Gatys et al., 2015; Ulyanov et al., 2016; Johnson et al., 2016; Dong et al., 2016), to name a few. In the context of CNN, the forward pass is a multi-layer scheme that provides an end-to-end mapping, from an input signal to some desired output. Each layer of this algorithm consists of three steps. The first convolves the input with a set of learned filters, resulting in a set of feature (or kernel) maps. These then undergo a point wise non-linear function, in a second step, often resulting in a sparse outcome (Glorot et al., 2011). A third (and optional) down-sampling step, termed *pooling*, is then applied on the result in order to reduce its dimensions. The output of this layer is then fed into another one, thus forming the multi-layer structure, often termed *forward pass*.

Despite its marvelous empirical success, a clear and profound theoretical understanding of this scheme is still lacking. A few preliminary theoretical results were recently suggested. In (Mallat, 2012; Bruna and Mallat, 2013) the Scattering Transform was proposed, suggesting to replace the learned filters in the CNN with predefined Wavelet functions. Interestingly, the features obtained from this network were shown to be invariant to various transformations such as translations and rotations. Other works have studied the properties of deep and fully connected networks under the assumption of independent identically distributed random weights (Giryes et al., 2015; Saxe et al., 2013; Arora et al., 2014; Dauphin et al., 2014; Choromanska et al., 2015). In particular, in (Giryes et al., 2015) deep neural networks were proven to preserve the metric structure of the input data as it propagates through the layers of the network. This, in turn, was shown to allow a stable recovery of the data from the features obtained from the network.

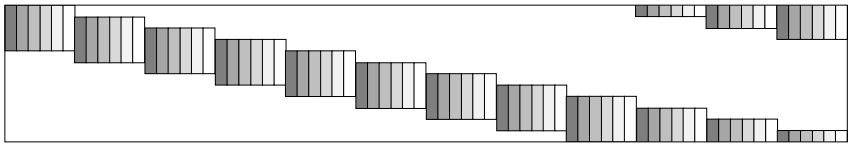
Another prominent paradigm in data processing is the sparse representation concept, being one of the most popular choices for a prior in the signal and image processing communities, and leading to exceptional results in various applications (Elad and Aharon, 2006; Dong et al., 2011; Zhang and Li, 2010; Jiang et al., 2011; Mairal et al., 2014). In this framework, one assumes that a signal can be represented as a linear combination of a few columns (called atoms) from a matrix termed a dictionary. Put differently, the signal is equal to a multiplication of a dictionary by a sparse vector. The task of retrieving the sparsest representation of a signal over a dictionary is called sparse coding or pursuit. Over the years, various algorithms were proposed to tackle this problem, among of which we mention the thresholding algorithm (Elad, 2010) and its iterative variant (Daubechies et al., 2004). When handling natural signals, this model has been commonly used for modeling local patches extracted from the global data mainly due to the computational difficulties related to the task of learning the dictionary (Elad and Aharon, 2006; Dong et al., 2011; Mairal et al., 2014; Romano and Elad, 2015; Sulam and Elad, 2015). However, in recent years an alternative to this patch-based processing has emerged in the form of the Convolutional Sparse Coding (CSC) model (Bristow et al., 2013; Kong and Fowlkes, 2014; Wohlberg, 2014; Gu et al., 2015; Heide et al., 2015; Papyan et al., 2016a,b). This circumvents the afore-

mentioned limitations by imposing a special structure – a union of banded and Circulant matrices – on the dictionary involved. The traditional sparse model has been extensively studied over the past two decades (Elad, 2010; Foucart and Rauhut, 2013). More recently, the convolutional extension was extensively analyzed in (Papyan et al., 2016a,b), shedding light on its theoretical aspects and prospects of success.

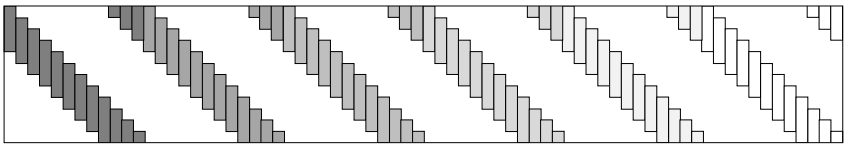
In this work, by leveraging the recent study of CSC, we aim to provide a new perspective on CNN, leading to a clear and profound theoretical understanding of this scheme, along with new insights. Embarking from the classic CSC, our approach builds upon the observation that similar to the original signal, the representation vector itself also admits a convolutional sparse representation. As such, it can be modeled as a superposition of atoms, taken from a different convolutional dictionary. This rationale can be extended to several layers, leading to the definition of our proposed ML-CSC model. Building on the recent analysis of the CSC, we provide a theoretical study of this novel model and its associated pursuits, namely the layered thresholding algorithm and the layered basis pursuit (BP).

Our analysis reveals the relation between the CNN and the ML-CSC model, showing that *the forward pass of the CNN is in fact identical to our proposed pursuit – the layered thresholding algorithm*. This connection is of significant importance since it gives a clear mathematical meaning, objective and model to the CNN architecture, which in turn can be accompanied by guarantees for the success of the forward pass, studied via the layered thresholding algorithm. Specifically, we show that the forward pass is guaranteed to recover an estimate of the underlying representations of an input signal, assuming these are sparse in a local sense. Moreover, considering a setting where a norm-bounded noise is added to the signal, we show that such a mild corruption in the input results in a bounded perturbation in the output – indicating the stability of the CNN in recovering the underlying representations. Lastly, we exploit the answers to the above questions in order to propose an alternative to the commonly used forward pass algorithm, which is tightly connected to both deconvolutional (Zeiler et al., 2010; Pu et al., 2016) and recurrent networks (Bengio et al., 1994), and also related to residual networks (He et al., 2015). The proposed alternative scheme is accompanied by a thorough theoretical study. Although this and the analysis presented throughout this work focus on CNN, we will show that they also hold for fully connected networks.

This paper is organized as follows. In Section 2 we review the basics of both the CNN and the Sparse-Land model. We then define the proposed ML-CSC model in Section 3, together with its corresponding deep sparse coding problem. In Section 4, we aim to solve this using the layered thresholding algorithm, which is shown to be equivalent to the forward pass of the CNN. Next, having established the relevance of our model to CNN, we proceed to its analysis in Section 5. Standing on these theoretical grounds, we then propose in Section 6 a provably improved pursuit, termed the layered BP, accompanied by its theoretical analysis. We revisit the assumptions of our model in Section 7. First, in Section 7.1 we link the double sparsity model to ours by assuming the dictionaries throughout the layers are sparse. Then, in Section 7.2 we consider an idea typically employed in CNN, termed spatial-stride, showing its benefits from a simple theoretical perspective. Combining our insights from Section 7.1 and 7.2, we move to an experimental phase by constructing a family of signals satisfying the assumptions of our model, which are then used in order to verify our theoretical results.



(a) A convolutional matrix.



(b) A concatenation of banded and Circulant matrices.

Figure 1: The two facets of the convolutional structure.

Finally, in Section 9 we conclude the contributions of this paper and present several future directions.

2. Background

This section is divided into two parts: The first is dedicated to providing a simple mathematical formulation of the CNN and the forward pass, while the second reviews the Sparse-Land model and its various extensions. Readers familiar with these two topics can skip directly to Section 3, which moves to serve the main contribution of this work.

2.1 Deep Learning - Convolutional Neural Networks

The fundamental algorithm of deep learning is the *forward pass*, employed both in the training and the inference stages. The first step of this algorithm convolves an input (one dimensional) signal $\mathbf{X} \in \mathbb{R}^N$ with a set of m_1 learned filters of length n_0, creating m_1 feature (or kernel) maps. Equally, this convolution can be written as a matrix-vector multiplication, $\mathbf{W}_1^T\mathbf{X} \in \mathbb{R}^{Nm_1}$, where $\mathbf{W}_1 \in \mathbb{R}^{N\times Nm_1}$ is a matrix containing in its columns the m_1 filters with all of their shifts. This structure, also known as a convolutional matrix, is depicted in Figure 1a. A pointwise nonlinear function is then applied on the sum of the obtained feature maps $\mathbf{W}_1^T\mathbf{X}$ and a bias term denoted by $\mathbf{b}_1 \in \mathbb{R}^{Nm_1}$. Many possible functions were proposed over the years, the most popular one being the Rectifier Linear Unit (ReLU) (Glorot et al., 2011; Krizhevsky et al., 2012), formally defined as $\text{ReLU}(z) = \max(z, 0)$. By cascading the basic block of convolutions followed by a nonlinear function, $\mathbf{Z}_1 = \text{ReLU}(\mathbf{W}_1^T\mathbf{X} + \mathbf{b}_1)$, a multi-layer structure of depth K is constructed. Formally, for two layers this is given by

$$f(\mathbf{X}, \{\mathbf{W}_i\}_{i=1}^2, \{\mathbf{b}_i\}_{i=1}^2) = \mathbf{Z}_2 = \text{ReLU}\Big(\mathbf{W}_2^T \ \text{ReLU}\left(\mathbf{W}_1^T\mathbf{X} + \mathbf{b}_1 \right) + \mathbf{b}_2 \Big), \qquad (1)$$

where $\mathbf{W}_2 \in \mathbb{R}^{Nm_1\times Nm_2}$ is a convolutional matrix (up to a small modification discussed below) constructed from m_2 filters of length n_1m_1 and $\mathbf{b}_2 \in \mathbb{R}^{Nm_2}$ is its corresponding bias. Although the two layers considered here can be readily extended to a much deeper configuration, we defer this to a later stage.

By changing the order of the columns in the convolutional matrix, one can observe that it can be equally viewed as a concatenation of banded and Circulant[1] matrices, as depicted in Figure 1b. Using this observation, the above description for one dimensional signals can be extended to images, with the exception that now every Circulant matrix is replaced by a block Circulant with Circulant blocks one.

An illustration of the forward pass algorithm is presented in Figure 2a and 2b. In Figure 2a one can observe that $\mathbf{W}_2$ is not a regular convolutional matrix but a stride one, since it shifts local filters by skipping m_1 entries at a time. The reason for this becomes apparent once we look at Figure 2b; the convolutions of the second layer are computed by shifting the filters of $\mathbf{W}_2$ that are of size $\sqrt{n_1} \times \sqrt{n_1} \times m_1$ across N places, skipping m_1 indices at a time from the $\sqrt{N} \times \sqrt{N} \times m_1$-sized array. A matrix obeying this structure is called a *stride convolutional matrix*.

Thus far, we have presented the basic structure of CNN. However, oftentimes an additional non-linear function, termed *pooling*, is employed on the resulting feature map obtained from the ReLU operator. In essence, this step summarizes each w_i-dimensional spatial neighborhood from the i-th kernel map $\mathbf{Z}_i$ by replacing it with a single value. If the neighborhoods are non-overlapping, for example, this results in the down-sampling of the feature map by a factor of w_i. The most widely used variant of the above is the max pooling (Krizhevsky et al., 2012; Simonyan and Zisserman, 2014), which picks the maximal value of each neighborhood. In (Springenberg et al., 2014) it was shown that this operator can be replaced by a convolutional layer with increased stride without loss in performance in several image classification tasks. Moreover, the current state-of-the-art in image recognition is obtained by the residual network (He et al., 2015), which does not employ any pooling steps (except for a single layer). As such, we defer the analysis of this operator to a follow-up work.

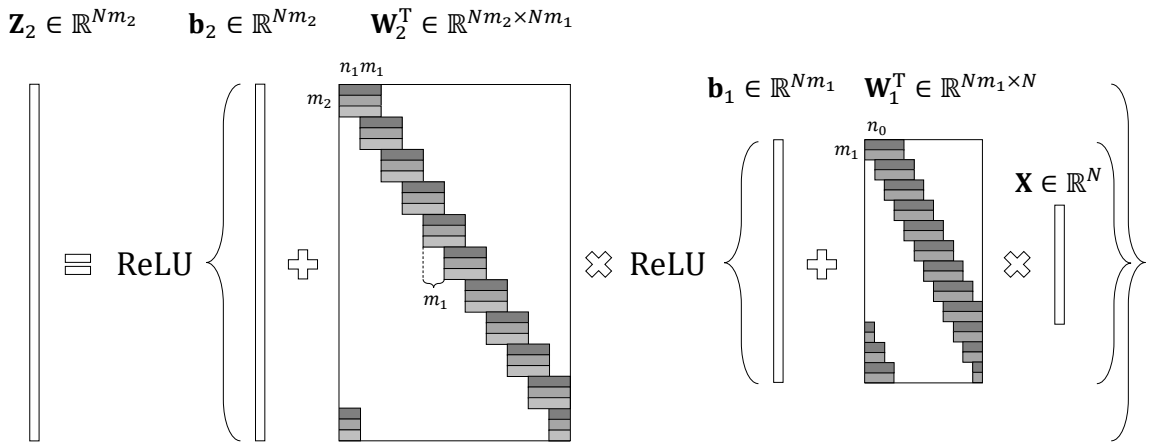
In the context of classification, for example, the output of the last layer is fed into a simple classifier that attempts to predict the label of the input signal $\mathbf{X}$, denoted by $h(\mathbf{X})$. Given a set of signals $\{\mathbf{X}_j\}_j$, the task of learning the parameters of the CNN – including the filters $\{\mathbf{W}_i\}_{i=1}^K$, the biases $\{\mathbf{b}_i\}_{i=1}^K$ and the parameters of the classifier $\mathbf{U}$ – can be formulated as the following minimization problem

$$\min_{\{\mathbf{W}_i\}_{i=1}^K, \{\mathbf{b}_i\}_{i=1}^K, \mathbf{U}} \sum_j \ell\Big(h(\mathbf{X}_j), \mathbf{U}, f\left(\mathbf{X}_j, \{\mathbf{W}_i\}_{i=1}^K, \{\mathbf{b}_i\}_{i=1}^K\right)\Big). \tag{2}$$

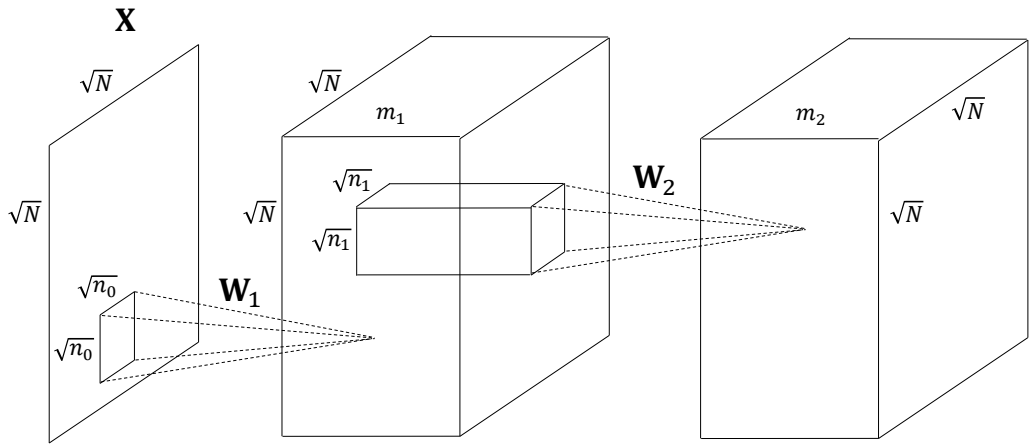
This optimization task seeks for the set of parameters that minimize the mean of the loss function ℓ, representing the price incurred when classifying the signal $\mathbf{X}$ incorrectly. The input for ℓ is the true label $h(\mathbf{X})$ and the one estimated by employing the classifier defined by $\mathbf{U}$ on the final layer of the CNN given by $f\left(\mathbf{X}, \{\mathbf{W}_i\}_{i=1}^K, \{\mathbf{b}_i\}_{i=1}^K\right)$. Similarly one can tackle various other problems, e.g. regression or prediction.

In the remainder of this work we shall focus on the feature extraction process and assume that the parameters of the CNN model are pre-trained and fixed. These, for example, could have been obtained by minimizing the above objective via the backpropagation algorithm and the stochastic gradient descent, as in the VGG network (Simonyan and Zisserman, 2014).

1. We shall assume throughout this paper that boundaries are treated by a periodic continuation, which gives rise to the cyclic structure.



(a) An illustration of Equation (1) for a one dimensional signal $\mathbf{X}$.



(b) The evolution of an image $\mathbf{X}$ throughout the CNN. Notice that the number of channels in $\mathbf{X}$ is equal to one and as such $m_0 = 1$.

Figure 2: The forward pass algorithm for a one dimensional signal (a) and an image (b).

2.2 Sparse-Land

This section presents an overview of the Sparse-Land model and its many extensions. We start with the traditional sparse representation and the core problem it aims to solve, and then proceed to its nonnegative variant. Next, we continue to the dictionary learning task both in the unsupervised and supervised cases. Finally, we describe the recent CSC model, which will lead us in the next section to the proposal of the ML-CSC model. This, in turn, will naturally connect the realm of sparsity to that of the CNN.

2.2.1 Sparse Representation

In the sparse representation model one assumes a signal $\mathbf{X} \in \mathbb{R}^N$ can be described as a multiplication of a matrix $\mathbf{D} \in \mathbb{R}^{N \times M}$, also called a dictionary, by a sparse vector $\mathbf{\Gamma} \in \mathbb{R}^M$. Equally, the signal $\mathbf{X}$ can be seen as a linear combination of a few columns from the dictionary $\mathbf{D}$, coined atoms.

For a fixed dictionary, given a signal $\mathbf{X}$, the task of recovering its sparsest representation $\mathbf{\Gamma}$ is called sparse coding, or simply pursuit, and it attempts to solve the following problem (Donoho and Elad, 2003; Tropp, 2004; Elad, 2010):

$$(\mathrm{P}_0): \ \min_{\mathbf{\Gamma}} \|\mathbf{\Gamma}\|_0 \ \text{ s.t. } \ \mathbf{D}\mathbf{\Gamma} = \mathbf{X}, \tag{3}$$

where we have denoted by $\|\mathbf{\Gamma}\|_0$ the number of non-zeros in $\mathbf{\Gamma}$. The above has a convex relaxation in the form of the Basis-Pursuit (BP) problem (Chen et al., 2001; Donoho and Elad, 2003; Tropp, 2006), formally defined as

$$(\mathrm{P}_1): \ \min_{\mathbf{\Gamma}} \|\mathbf{\Gamma}\|_1 \ \text{ s.t. } \ \mathbf{D}\mathbf{\Gamma} = \mathbf{X}. \tag{4}$$

Many questions arise from the above two defined problems. For instance, given a signal $\mathbf{X}$, is its sparsest representation unique? Assuming that such a unique solution exists, can it be recovered using practical algorithms such as the Orthogonal Matching Pursuit (OMP) (Chen et al., 1989; Pati et al., 1993) and the BP (Chen et al., 2001; Daubechies et al., 2004)? The answers to these questions were shown to be positive under the assumption that the number of non-zeros in the underlying representation is not too high and in particular less than $\frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D})}\right)$ (Donoho and Elad, 2003; Tropp, 2004; Donoho et al., 2006). The quantity $\mu(\mathbf{D})$ is the mutual coherence of the dictionary $\mathbf{D}$, being the maximal inner product of two atoms extracted from it[2]. Formally, we can write

$$\mu(\mathbf{D}) = \max_{i \neq j} |\mathbf{d}_i^T \mathbf{d}_j|.$$

Tighter conditions, relying on sharper characterizations of the dictionary, were also suggested in the literature (Candes et al., 2006; Schnass and Vandergheynst, 2007; Candes et al., 2006; Candes and Tao, 2007). However, at this point, we shall not dwell on these.

One of the simplest approaches for tackling the P_0 and P_1 problems is via the hard and soft thresholding algorithms, respectively. These operate by computing the inner products between the signal $\mathbf{X}$ and all the atoms in $\mathbf{D}$ and then choosing the atoms corresponding to the highest responses. This can be described as solving, for some scalar β, the following problems:

$$\min_{\mathbf{\Gamma}} \frac{1}{2}\|\mathbf{\Gamma} - \mathbf{D}^T\mathbf{X}\|_2^2 + \beta\|\mathbf{\Gamma}\|_0$$

for the P_0, or

$$\min_{\mathbf{\Gamma}} \frac{1}{2}\|\mathbf{\Gamma} - \mathbf{D}^T\mathbf{X}\|_2^2 + \beta\|\mathbf{\Gamma}\|_1, \tag{5}$$

for the P_1. The above are simple projection problems that admit a closed-form solution in the form[3] of $\mathcal{H}_\beta(\mathbf{D}^T\mathbf{X})$ or $\mathcal{S}_\beta(\mathbf{D}^T\mathbf{X})$, where we have defined the hard thresholding operator

2. Hereafter, we assume that the atoms are normalized to a unit ℓ_2 norm.
3. The curious reader may identify the relation between the notations used here and the ones in the previous subsection, which starts to reveal the relation between CNN and sparsity-inspired models. This connection will be made stringer and clearer as we proceed to CSC.

$\mathcal{H}_\beta(\cdot)$ by

$$\mathcal{H}_\beta(z) = \begin{cases} z, & z < -\beta \\ 0, & -\beta \leq z \leq \beta \\ z, & \beta < z, \end{cases}$$

and the soft thresholding operator $\mathcal{S}_\beta(\cdot)$ by

$$\mathcal{S}_\beta(z) = \begin{cases} z+\beta, & z < -\beta \\ 0, & -\beta \leq z \leq \beta \\ z-\beta, & \beta < z. \end{cases}$$

Both of the above, depicted in Figure 3, nullify small entries and thus promote a sparse solution. However, while the hard thresholding operator does not modify large coefficients (in absolute value), the soft thresholding does, by contracting these to zero. This inherent limitation of the soft version will appear later on in our theoretical analysis.

As for the theoretical guarantees for the success of the simple thresholding algorithms; these depend on the properties of $\mathbf{D}$ and on the ratio between the minimal and maximal coefficients in absolute value in $\mathbf{\Gamma}$, and thus are weaker when compared to those found for OMP and BP (Donoho and Elad, 2003; Tropp, 2004; Donoho et al., 2006). Still, under some conditions, both algorithms are guaranteed to find the true support of $\mathbf{\Gamma}$ along with an approximation of its true coefficients. Moreover, a better estimation of these can be obtained by projecting the input signal onto the atoms corresponding to the found support (indices of the non-zero entries) by solving a Least-Squares problem. This step, termed debiasing (Elad, 2010), results in a more accurate identification of the non-zero values.

2.2.2 Nonnegative Sparse Coding

The nonnegative sparse representation model assumes a signal can be decomposed into a multiplication of a dictionary and a *nonnegative* sparse vector. A natural question arising from this is whether such a modification to the original Sparse-Land model affects its expressiveness. To address this, we hereby provide a simple reduction from the original sparse representation to the nonnegative one.

Consider a signal $\mathbf{X} = \mathbf{D}\mathbf{\Gamma}$, where the signs of the entries in $\mathbf{\Gamma}$ are unrestricted. Notice that this can be equally written as

$$\mathbf{X} = \mathbf{D}\mathbf{\Gamma}_P + (-\mathbf{D})(-\mathbf{\Gamma}_N),$$

where we have split the vector $\mathbf{\Gamma}$ to its positive coefficients, $\mathbf{\Gamma}_P$, and its negative ones, $\mathbf{\Gamma}_N$. Since the coefficients in $\mathbf{\Gamma}_P$ and $-\mathbf{\Gamma}_N$ are all positive, one can thus assume the signal $\mathbf{X}$ admits a non-negative sparse representation over the dictionary $[\mathbf{D}, -\mathbf{D}]$ with the vector $[\mathbf{\Gamma}_P, -\mathbf{\Gamma}_N]^T$. Thus, restricting the coefficients in the sparsity inspired model to be nonnegative does not change its expressiveness.

Similar to the original model, in the nonnegative case, one could solve the associated pursuit problem by employing a soft thresholding algorithm. However, in this case a constraint must be added to the optimization problem in Equation (5), forcing the outcome to be positive, i.e.,

$$\min_{\mathbf{\Gamma}} \frac{1}{2}\|\mathbf{\Gamma} - \mathbf{D}^T\mathbf{X}\|_2^2 + \beta\|\mathbf{\Gamma}\|_1 \quad \text{s.t.} \quad \mathbf{\Gamma} \geq \mathbf{0}.$$

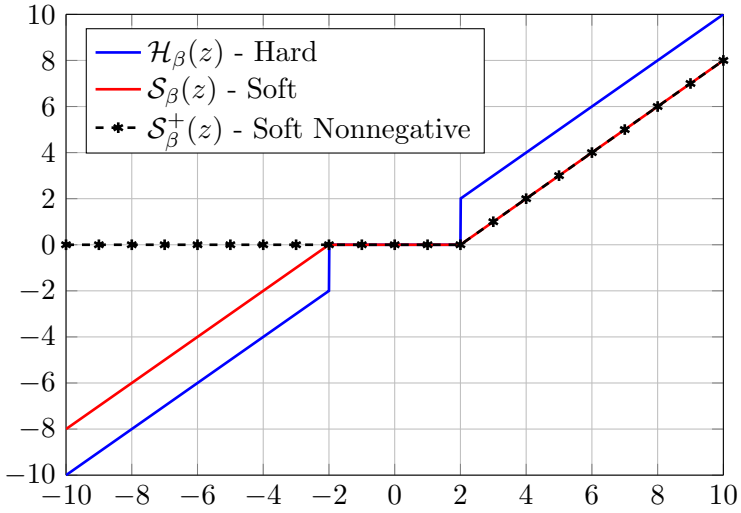


Figure 3: The thresholding operators for a constant $\beta = 2$.

Since the above is a simple projection problem (onto the ℓ_1 ball constrained to positive entries), it admits a closed-form solution $\mathcal{S}_\beta^+(\mathbf{D}^T\mathbf{X})$, where we have defined the soft nonnegative thresholding operator $\mathcal{S}_\beta^+(\cdot)$ as

$$\mathcal{S}_\beta^+(z) = \begin{cases} 0, & z \leq \beta \\ z - \beta, & \beta < z. \end{cases}$$

Remarkably, the above function satisfies

$$\mathcal{S}_\beta^+(z) = \max(z - \beta, 0) = \text{ReLU}(z - \beta).$$

In other words, the ReLU and the soft nonnegative thresholding operator are equal, a fact that will prove to be important later in our work. We should note that a similar conclusion was reached in (Fawzi et al., 2015). To summarize this discussion, we depict in Figure 3 the hard, soft, and nonnegative soft thresholding operators.

2.2.3 Unsupervised and Task Driven Dictionary Learning

At first, the dictionaries employed in conjunction with the sparsity inspired model were analytically defined matrices, such as the Wavelet and the Fourier (Daubechies et al., 1992; Mallat and Zhang, 1993; Elad and Bruckstein, 2002; Mallat, 2008). Although the sparse coding problem under these can be done very efficiently, over the years many have shifted to a data driven approach – adapting the dictionary $\mathbf{D}$ to a set of training signals at hand via some learning procedure. This was empirically shown to lead to sparser representations and better overall performance, at the cost of complicating the involved pursuit, since the dictionary was usually chosen to be redundant (having more columns than rows).

The task of learning a dictionary for representing a set of signals $\{\mathbf{X}_j\}_j$ can be formulated as follows

$$\min_{\mathbf{D},\{\mathbf{\Gamma}^j\}_j} \sum_j \|\mathbf{X}_j - \mathbf{D}\mathbf{\Gamma}^j\|_2^2 + \xi\|\mathbf{\Gamma}^j\|_0.$$

The above formulation is an unsupervised learning procedure, and it was later extended to a supervised setting. In this context, given a set of signals $\{\mathbf{X}_j\}_j$, one attempts to predict their corresponding labels $\{h(\mathbf{X}_j)\}_j$. A common approach for tackling this is first solving a pursuit problem for each signal $\mathbf{X}_j$ over a dictionary $\mathbf{D}$, resulting in

$$\mathbf{\Gamma}^\star(\mathbf{X}_j, \mathbf{D}) = \arg\min_{\mathbf{\Gamma}} \|\mathbf{\Gamma}\|_0 \quad \text{s.t.} \quad \mathbf{D}\mathbf{\Gamma} = \mathbf{X}_j,$$

and then feeding these sparse representations into a simple classifier, defined by the parameters $\mathbf{U}$. The task of learning jointly the dictionary $\mathbf{D}$ and the classifier $\mathbf{U}$ was addressed in (Mairal et al., 2012), where the following optimization problem was proposed

$$\min_{\mathbf{D},\mathbf{U}} \sum_j \ell\Big(h(\mathbf{X}_j), \mathbf{U}, \mathbf{\Gamma}^\star(\mathbf{X}_j, \mathbf{D})\Big).$$

The loss function ℓ in the above objective penalizes the estimated label if it is different from the true $h(\mathbf{X}_j)$, similar to what we have seen in Section 2.1. The above formulation contains in it the unsupervised option as a special case, in which $\mathbf{U}$ is of no importance, and the loss function is the representation error $\sum_j \|\mathbf{X}_j - \mathbf{D}\mathbf{\Gamma}_j^\star\|_2^2$.

Double sparsity – first proposed in (Rubinstein et al., 2010) and later employed in (Sulam et al., 2016) – attempts to benefit from both the computational efficiency of analytically defined matrices, and the adaptability of data driven dictionaries. In this model, one assumes the dictionary $\mathbf{D}$ can be factorized into a multiplication of two matrices, $\mathbf{D}_1$ and $\mathbf{D}_2$, where $\mathbf{D}_1$ is an analytic dictionary with fast implementation, and $\mathbf{D}_2$ is a trained sparse one. As a result, the signal $\mathbf{X}$ can be represented as

$$\mathbf{X} = \mathbf{D}\mathbf{\Gamma}_2 = \mathbf{D}_1\mathbf{D}_2\mathbf{\Gamma}_2,$$

where $\mathbf{\Gamma}_2$ is sparse.

We propose a different interpretation for the above, which is unrelated to practical aspects. Since both the matrix $\mathbf{D}_2$ and the vector $\mathbf{\Gamma}_2$ are sparse, one would expect their multiplication $\mathbf{\Gamma}_1 = \mathbf{D}_2\mathbf{\Gamma}_2$ to be sparse as well. As such, the double sparsity model implicitly assumes that the signal $\mathbf{X}$ can be decomposed into a multiplication of a dictionary $\mathbf{D}_1$ and sparse vector $\mathbf{\Gamma}_1$, which in turn can also be decomposed similarly via $\mathbf{\Gamma}_1 = \mathbf{D}_2\mathbf{\Gamma}_2$.

2.2.4 Convolutional Sparse Coding Model

Due to the computational constraints entailed when deploying trained dictionaries, this approach seems valid only for treatment of low-dimensional signals. Indeed, the sparse representation model is traditionally used for modeling local patches extracted from a global signal. An alternative, which was recently proposed, is the CSC model that attempts to represent the whole signal $\mathbf{X} \in \mathbb{R}^N$ as a multiplication of a global convolutional dictionary $\mathbf{D} \in \mathbb{R}^{N \times Nm}$ and a sparse vector $\mathbf{\Gamma} \in \mathbb{R}^{Nm}$. Interestingly, the former is constructed by

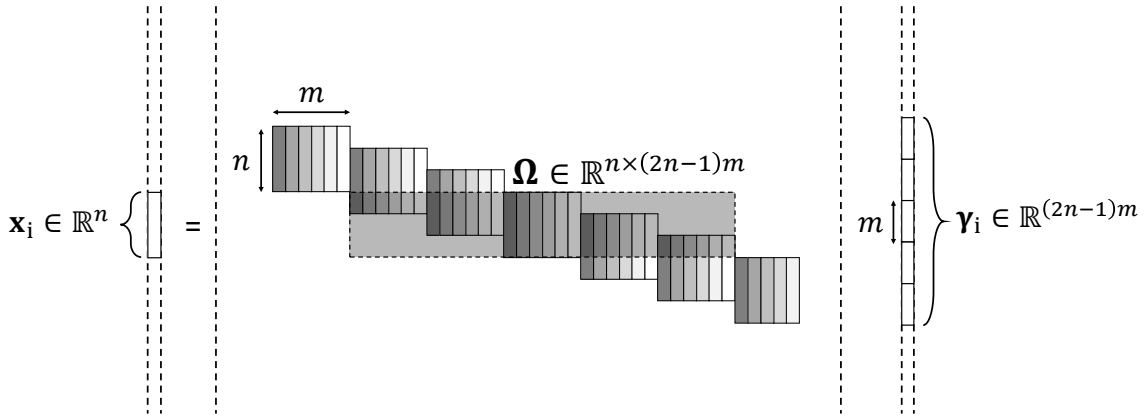


Figure 4: The i-th patch $\mathbf{x}_i$ of the global system $\mathbf{X} = \mathbf{D\Gamma}$, given by $\mathbf{x}_i = \mathbf{\Omega}\boldsymbol{\gamma}_i$.

shifting a local matrix of size $n \times m$ in all possible positions, resulting in the same structure as the one shown in Figure 1a.

In the convolutional model, the classical theoretical guarantees (we are referring to results reported in (Chen et al., 2001; Donoho and Elad, 2003; Tropp, 2006)) for the P_0 problem, defined in Equation (3), are very pessimistic. In particular, the condition for the uniqueness of the underlying solution and the requirement for the success of the sparse coding algorithms depend on the global number of non-zeros being less than $\frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D})}\right)$. Following the Welch bound (Welch, 1974), this expression was shown in (Papyan et al., 2016a) to be impractical, allowing the global number of non-zeros in $\mathbf{\Gamma}$ to be extremely low.

In order to provide a better theoretical understanding of this model, which exploits the inherent structure of the convolutional dictionary, a recent work (Papyan et al., 2016a) suggested to measure the sparsity of $\mathbf{\Gamma}$ in a localized manner. More concretely, consider the i-th n-dimensional patch of the global system $\mathbf{X} = \mathbf{D\Gamma}$, given by $\mathbf{x}_i = \mathbf{\Omega}\boldsymbol{\gamma}_i$. The stripe-dictionary $\mathbf{\Omega}$, which is of size $n \times (2n-1)m$, is obtained by extracting the i-th patch from the global dictionary $\mathbf{D}$ and discarding all the zero columns from it. The stripe vector $\boldsymbol{\gamma}_i$ is the corresponding sparse representation of length $(2n-1)m$, containing all coefficients of atoms contributing to $\mathbf{x}_i$. This relation is illustrated in Figure 4. Notably, the choice of a convolutional dictionary results in signals such that every patch of length n extracted from them can be sparsely represented using a single shift-invariant local dictionary $\mathbf{\Omega}$ – a common assumption usually employed in signal and image processing.

Following the above construction, the $\ell_{0,\infty}$ norm of the global sparse vector $\mathbf{\Gamma}$ is defined to be the maximal number of non-zeros in a stripe of length $(2n-1)m$ extracted from it. Formally,

$$\|\mathbf{\Gamma}\|_{0,\infty}^{\mathbf{s}} = \max_i \|\boldsymbol{\gamma}_i\|_0,$$

where the letter $\mathbf{s}$ emphasizes that the $\ell_{0,\infty}$ norm is computed by sweeping over all stripes. Given a signal $\mathbf{X}$, finding its sparest representation $\mathbf{\Gamma}$ in the $\ell_{0,\infty}$ sense is equal to the following optimization problem:

$$(\mathrm{P}_{0,\infty}): \quad \min_{\mathbf{\Gamma}} \quad \|\mathbf{\Gamma}\|_{0,\infty}^{\mathbf{s}} \quad \text{s.t.} \quad \mathbf{D\Gamma} = \mathbf{X}. \tag{6}$$

Intuitively, this seeks for a global vector $\mathbf{\Gamma}$ that can represent sparsely every patch in the signal $\mathbf{X}$ using the dictionary $\mathbf{\Omega}$. The advantage of the above problem over the traditional P_0 becomes apparent as we move to consider its theoretical aspects. Assuming that the **number of non-zeros per stripe** (and not globally) in $\mathbf{\Gamma}$ is less than $\frac{1}{2}\left(1+\frac{1}{\mu(\mathbf{D})}\right)$, in (Papyan et al., 2016a) it was proven that the solution for the $\mathrm{P}_{0,\infty}$ problem is unique. Furthermore, classical pursuit methods, originally tackling the P_0 problem, are guaranteed to find this representation.

When modeling natural signals, due to measurement noise as well as model deviations, one can not impose a perfect reconstruction such as $\mathbf{X}=\mathbf{D}\mathbf{\Gamma}$ on the signal $\mathbf{X}$. Instead, one assumes $\mathbf{Y}=\mathbf{X}+\mathbf{E}=\mathbf{D}\mathbf{\Gamma}+\mathbf{E}$, where $\mathbf{E}$ is, for example, an ℓ_2-bounded error vector. To address this, the work reported in (Papyan et al., 2016b) considered the extension of the $\mathrm{P}_{0,\infty}$ problem to the $\mathrm{P}_{0,\infty}^{\boldsymbol{\mathcal{E}}}$ one, formally defined as

$$(\mathrm{P}_{0,\infty}^{\boldsymbol{\mathcal{E}}}): \quad \min_{\mathbf{\Gamma}} \|\mathbf{\Gamma}\|_{0,\infty}^{\mathbf{s}} \ \text{ s.t. } \ \|\mathbf{Y}-\mathbf{D}\mathbf{\Gamma}\|_2^2 \leq \boldsymbol{\mathcal{E}}^2.$$

Similar to the $\mathrm{P}_{0,\infty}$ problem, this was also analyzed theoretically, shedding light on the theoretical aspects of the convolutional model in the presence of noise. In particular, a stability claim for the $\mathrm{P}_{0,\infty}^{\boldsymbol{\mathcal{E}}}$ problem and guarantees for the success of both the OMP and the BP were provided. Similar to the noiseless case, these assumed that the number of non-zeros per stripe is low.

3. From Atoms to Molecules: Multi-Layer Convolutional Sparse Model

Convolutional sparsity assumes an inherent structure for natural signals. Similarly, the representations themselves could also be assumed to have such a structure. In what follows, we propose a novel layered model that relies on this rationale.

The convolutional sparse model assumes a global signal $\mathbf{X}\in\mathbb{R}^N$ can be decomposed into a multiplication of a convolutional dictionary $\mathbf{D}_1\in\mathbb{R}^{N\times Nm_1}$, composed of m_1 local filters of length n_0, and a sparse vector $\mathbf{\Gamma}_1\in\mathbb{R}^{Nm_1}$. Herein, we extend this by proposing a similar factorization of the vector $\mathbf{\Gamma}_1$, which can be perceived as an N-dimensional global signal with m_1 channels. In particular, we assume $\mathbf{\Gamma}_1=\mathbf{D}_2\mathbf{\Gamma}_2$, where $\mathbf{D}_2\in\mathbb{R}^{Nm_1\times Nm_2}$ is a stride convolutional dictionary (skipping m_1 entries at a time) and $\mathbf{\Gamma}_2\in\mathbb{R}^{Nm_2}$ is a sparse representation. We denote the number of unique filters constructing $\mathbf{D}_2$ by m_2 and their corresponding length by n_1m_1. Due to the multi-layer nature of this model and the imposed convolutional structure, we name this the ML-CSC model.

Intuitively, $\mathbf{X}=\mathbf{D}_1\mathbf{\Gamma}_1$ assumes that the signal $\mathbf{X}$ is a superposition of **atoms** taken from $\mathbf{D}_1$. While equation $\mathbf{X}=\mathbf{D}_1\mathbf{D}_2\mathbf{\Gamma}_2$ views the signal as a superposition of more complex entities taken from the dictionary $\mathbf{D}_1\mathbf{D}_2$, which we coin **molecules**.

While this proposal can be interpreted as a straightforward fusion between the double sparsity model (Rubinstein et al., 2010) and the convolutional one, it is in fact substantially different. The double sparsity model assumes that $\mathbf{D}_2$ is sparse, and forces **only** the deepest representation $\mathbf{\Gamma}_2$ to be sparse as well. Here, on the other hand, we replace this constraint by forcing $\mathbf{D}_2$ to have a stride convolution structure, putting emphasis on the sparsity of both the representations $\mathbf{\Gamma}_1$ and $\mathbf{\Gamma}_2$. In Section 7.1 we will revisit the double sparsity work and its ties to ours by showing the benefits of injecting the assumption on the sparsity of $\mathbf{D}_2$ into our proposed model.

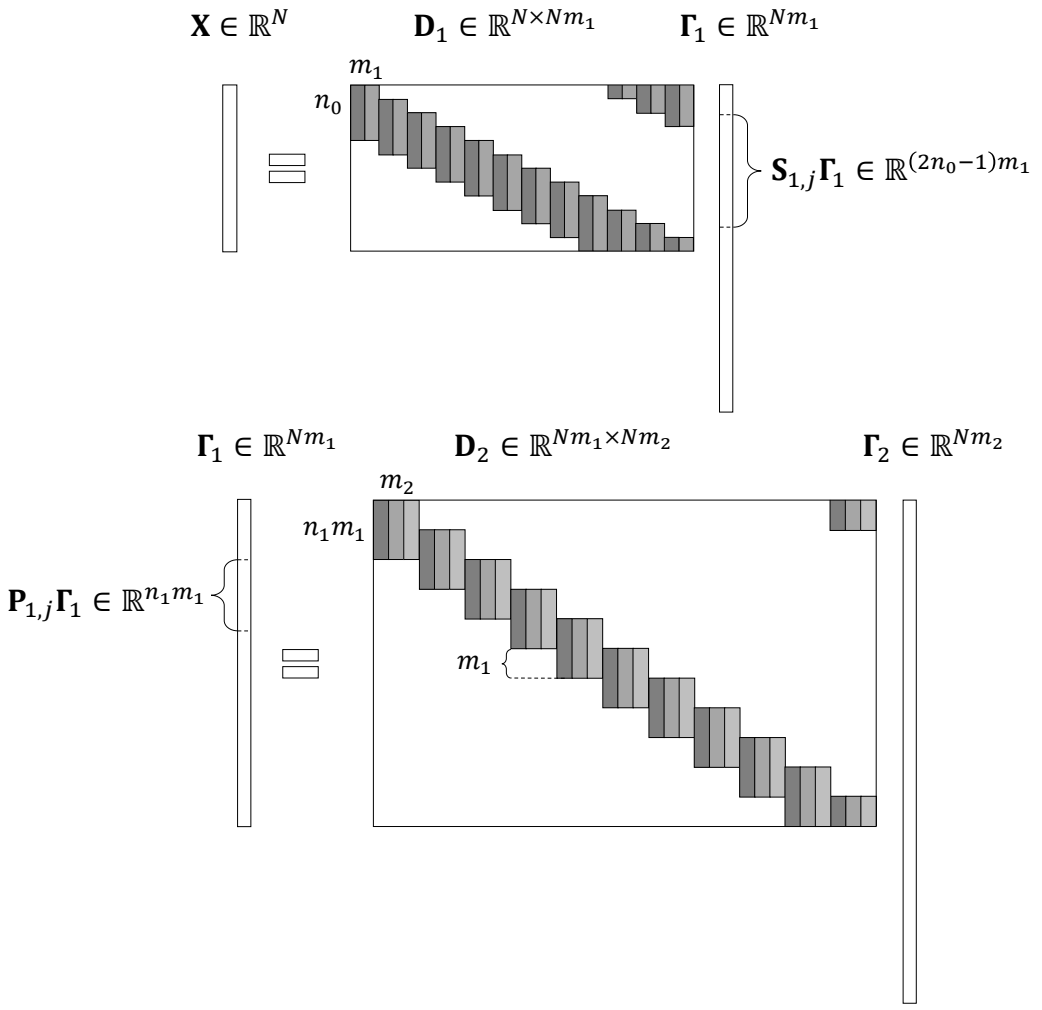


Figure 5: An instance $\mathbf{X} = \mathbf{D}_1\boldsymbol{\Gamma}_1 = \mathbf{D}_1\mathbf{D}_2\boldsymbol{\Gamma}_2$ of the ML-CSC model. Notice that $\boldsymbol{\Gamma}_1$ is built of both stripes $\mathbf{S}_{1,j}\boldsymbol{\Gamma}_1$ and patches $\mathbf{P}_{1,j}\boldsymbol{\Gamma}_1$.

Under the above construction the sparse vector $\boldsymbol{\Gamma}_1$ has two roles. In the context of the system of equations $\mathbf{X} = \mathbf{D}_1\boldsymbol{\Gamma}_1$, it is the convolutional sparse representation of the signal $\mathbf{X}$ over the dictionary $\mathbf{D}_1$. As such, the vector $\boldsymbol{\Gamma}_1$ is composed from $(2n_0 - 1)m_1$-dimensional stripes, $\mathbf{S}_{1,j}\boldsymbol{\Gamma}_1$, where $\mathbf{S}_{i,j}$ is the operator that extracts the j-th stripe from $\boldsymbol{\Gamma}_i$. From another point of view, $\boldsymbol{\Gamma}_1$ is in itself a signal that admits a sparse representation $\boldsymbol{\Gamma}_1 = \mathbf{D}_2\boldsymbol{\Gamma}_2$. Denoting by $\mathbf{P}_{i,j}$ the operator that extracts the j-th patch from $\boldsymbol{\Gamma}_i$, the signal $\boldsymbol{\Gamma}_1$ is composed of patches $\mathbf{P}_{1,j}\boldsymbol{\Gamma}_1$ of length $n_1 m_1$. The above model is depicted in Figure 5, presenting both roles of $\boldsymbol{\Gamma}_1$ and their corresponding constituents – stripes and patches. Clearly, the above construction can be extended to more than two layers, leading to the following definition:

Definition 1 *For a global signal* $\mathbf{X}$*, a set of convolutional dictionaries* $\{\mathbf{D}_i\}_{i=1}^K$*, and a vector* $\boldsymbol{\lambda}$*, define the deep coding problem* $DCP_{\boldsymbol{\lambda}}$ *as:*

$$(DCP_{\boldsymbol{\lambda}}): \quad \text{find} \quad \{\boldsymbol{\Gamma}_i\}_{i=1}^K \qquad \text{s.t.} \qquad \begin{aligned} \mathbf{X} &= \mathbf{D}_1\boldsymbol{\Gamma}_1, & \|\boldsymbol{\Gamma}_1\|_{0,\infty}^{\mathbf{s}} &\leq \lambda_1 \\ \boldsymbol{\Gamma}_1 &= \mathbf{D}_2\boldsymbol{\Gamma}_2, & \|\boldsymbol{\Gamma}_2\|_{0,\infty}^{\mathbf{s}} &\leq \lambda_2 \\ &\vdots & &\vdots \\ \boldsymbol{\Gamma}_{K-1} &= \mathbf{D}_K\boldsymbol{\Gamma}_K, & \|\boldsymbol{\Gamma}_K\|_{0,\infty}^{\mathbf{s}} &\leq \lambda_K, \end{aligned}$$

where the scalar λ_i *is the* i*-th entry of* $\boldsymbol{\lambda}$*.*

Denoting $\boldsymbol{\Gamma}_0$ to be the signal $\mathbf{X}$, the $\text{DCP}_{\boldsymbol{\lambda}}$ can be rewritten compactly as

$$(\text{DCP}_{\boldsymbol{\lambda}}): \quad \text{find} \quad \{\boldsymbol{\Gamma}_i\}_{i=1}^K \quad \text{s.t.} \quad \boldsymbol{\Gamma}_{i-1} = \mathbf{D}_i\boldsymbol{\Gamma}_i, \quad \|\boldsymbol{\Gamma}_i\|_{0,\infty}^{\mathbf{s}} \leq \lambda_i, \quad \forall 1 \leq i \leq K.$$

Intuitively, given a signal $\mathbf{X}$, this problem seeks for a set of representations, $\{\boldsymbol{\Gamma}_i\}_{i=1}^K$, such that each one is locally sparse. As we shall see next, the above can be easily solved using simple algorithms that also enjoy from theoretical justifications. Next, we extend the $\text{DCP}_{\boldsymbol{\lambda}}$ problem to a noisy regime.

Definition 2 *For a global signal* $\mathbf{Y}$*, a set of convolutional dictionaries* $\{\mathbf{D}_i\}_{i=1}^K$*, and vectors* $\boldsymbol{\lambda}$ *and* $\boldsymbol{\mathcal{E}}$*, define the deep coding problem* $DCP_{\boldsymbol{\lambda}}^{\boldsymbol{\mathcal{E}}}$ *as:*

$$(DCP_{\boldsymbol{\lambda}}^{\boldsymbol{\mathcal{E}}}): \quad \text{find} \quad \{\boldsymbol{\Gamma}_i\}_{i=1}^K \qquad \text{s.t.} \qquad \begin{aligned} \|\mathbf{Y} - \mathbf{D}_1\boldsymbol{\Gamma}_1\|_2 &\leq \mathcal{E}_0, & \|\boldsymbol{\Gamma}_1\|_{0,\infty}^{\mathbf{s}} &\leq \lambda_1 \\ \|\boldsymbol{\Gamma}_1 - \mathbf{D}_2\boldsymbol{\Gamma}_2\|_2 &\leq \mathcal{E}_1, & \|\boldsymbol{\Gamma}_2\|_{0,\infty}^{\mathbf{s}} &\leq \lambda_2 \\ &\vdots & &\vdots \\ \|\boldsymbol{\Gamma}_{K-1} - \mathbf{D}_K\boldsymbol{\Gamma}_K\|_2 &\leq \mathcal{E}_{K-1}, & \|\boldsymbol{\Gamma}_K\|_{0,\infty}^{\mathbf{s}} &\leq \lambda_K, \end{aligned}$$

where the scalars λ_i *and* $\mathcal{E}_i$ *are the* i*-th entry of* $\boldsymbol{\lambda}$ *and* $\boldsymbol{\mathcal{E}}$*, respectively.*

We now move to the task of learning the model parameters. Denote by $\text{DCP}_{\boldsymbol{\lambda}}^{\star}(\mathbf{X}, \{\mathbf{D}_i\}_{i=1}^K)$ the representation $\boldsymbol{\Gamma}_K$ obtained by solving the DCP problem (Definition 1, i.e., noiseless) for the signal $\mathbf{X}$ and the set of dictionaries $\{\mathbf{D}_i\}_{i=1}^K$. Relying on this, we now extend the dictionary learning problem, as presented in Section 2.2.3, to the multi-layer convolutional sparse representation setting.

Definition 3 *For a set of global signals* $\{\mathbf{X}_j\}_j$*, their corresponding labels* $\{h(\mathbf{X}_j)\}_j$*, a loss function* ℓ*, and a vector* $\boldsymbol{\lambda}$*, define the deep learning problem* $DLP_{\boldsymbol{\lambda}}$ *as:*

$$(DLP_{\boldsymbol{\lambda}}): \quad \min_{\{\mathbf{D}_i\}_{i=1}^K, \mathbf{U}} \sum_j \ell\Big(h(\mathbf{X}_j), \mathbf{U}, DCP_{\boldsymbol{\lambda}}^{\star}(\mathbf{X}_j, \{\mathbf{D}_i\}_{i=1}^K)\Big).$$

A clarification for the chosen name, deep learning problem, will be provided shortly. The solution for the above results in an end-to-end mapping, from a set of input signals to their corresponding labels. Similarly, we can define the $\text{DLP}_{\boldsymbol{\lambda}}^{\boldsymbol{\mathcal{E}}}$ problem. However, this is omitted for the sake of brevity. We conclude this section by summarizing, for the convenience of the reader, all notations used throughout this work in Table 1.

$\mathbf{X} = \mathbf{\Gamma}_0$	:	a global signal of length N.
$\mathbf{E}$, $\mathbf{Y} = \hat{\mathbf{\Gamma}}_0$	:	a global error vector and its corresponding noisy signal, where generally $\mathbf{Y} = \mathbf{X} + \mathbf{E}$.
K	:	the number of layers.
m_i	:	the number of local filters in $\mathbf{D}_i$, and also the number of channels in $\mathbf{\Gamma}_i$. Notice that $m_0 = 1$.
n_0	:	the size of a local patch in $\mathbf{X} = \mathbf{\Gamma}_0$.
$n_i,\ i \geq 1$	:	the size of a local patch (not including channels) in $\mathbf{\Gamma}_i$.
$n_i m_i$	:	the size of a local patch (including channels) in $\mathbf{\Gamma}_i$.
$\mathbf{D}_1$	:	a (full) convolutional dictionary of size $N \times Nm_1$ with filters of length n_0.
$\mathbf{D}_i,\ i \geq 2$	:	a convolutional dictionary of size $Nm_{i-1} \times Nm_i$ with filters of length $n_{i-1}m_{i-1}$ and a stride equal to m_{i-1}.
$\mathbf{\Gamma}_i$	:	a sparse vector of length Nm_i that is the representation of $\mathbf{\Gamma}_{i-1}$ over the dictionary $\mathbf{D}_i$, i.e. $\mathbf{\Gamma}_{i-1} = \mathbf{D}_i\mathbf{\Gamma}_i$.
$\mathbf{S}_{i,j}$	:	an operator that extracts the j-th stripe of length $(2n_{i-1} - 1)m_i$ from $\mathbf{\Gamma}_i$.
$\|\mathbf{\Gamma}_i\|_{0,\infty}^{\mathbf{S}}$	:	the maximal number of non-zeros in a stripe from $\mathbf{\Gamma}_i$.
$\mathbf{P}_{i,j}$	:	an operator that extracts the j-th $n_i m_i$-dimensional patch from $\mathbf{\Gamma}_i$.
$\|\mathbf{\Gamma}_i\|_{0,\infty}^{\mathbf{P}}$	:	the maximal number of non-zeros in a patch from $\mathbf{\Gamma}_i$ (Definition 6).
$\mathbf{R}_{i,j}$	:	an operator that extracts the filter of length $n_{i-1}m_{i-1}$ from the j-th atom in $\mathbf{D}_i$.
$\|\mathbf{V}\|_{2,\infty}^{\mathbf{P}}$	:	the maximal ℓ_2 norm of a patch extracted from a vector $\mathbf{V}$ (Definition 6).

Table 1: Summary of notations used throughout the paper.

4. Layered Thresholding: The Crux of the Matter

Consider the ML-CSC model defined by the set of dictionaries $\{\mathbf{D}_i\}_{i=1}^K$. Assume we are given a signal

$$\begin{aligned}\mathbf{X} &= \mathbf{D}_1\mathbf{\Gamma}_1 \\ \mathbf{\Gamma}_1 &= \mathbf{D}_2\mathbf{\Gamma}_2 \\ &\vdots \\ \mathbf{\Gamma}_{K-1} &= \mathbf{D}_K\mathbf{\Gamma}_K,\end{aligned}$$

and our goal is to find its underlying representations, $\{\mathbf{\Gamma}_i\}_{i=1}^K$. Tackling this problem by recovering all the vectors at once might be computationally and conceptually challenging; therefore, we propose the *layered thresholding algorithm* that gradually computes the sparse vectors one at a time across the different layers. Denoting by $\mathcal{P}_\beta(\cdot)$ a sparsifying operator that is equal to $\mathcal{H}_\beta(\cdot)$ in the hard thresholding case and $\mathcal{S}_\beta(\cdot)$ in the soft one; we commence by computing $\hat{\mathbf{\Gamma}}_1 = \mathcal{P}_{\beta_1}(\mathbf{D}_1^T\mathbf{X})$, which is an approximation of $\mathbf{\Gamma}_1$. Next, by applying another thresholding algorithm, however this time on $\hat{\mathbf{\Gamma}}_1$, an approximation of $\mathbf{\Gamma}_2$ is obtained, $\hat{\mathbf{\Gamma}}_2 = \mathcal{P}_{\beta_2}(\mathbf{D}_2^T\hat{\mathbf{\Gamma}}_1)$. This process, which is iterated until the last representation $\hat{\mathbf{\Gamma}}_K$ is acquired, is summarized in Algorithm 1.

One might ponder as to why does the application of the thresholding algorithm on the signal $\mathbf{X}$ not result in the true representation $\mathbf{\Gamma}_1$, but instead an approximation of it. As

Algorithm 1 The layered thresholding algorithm.

Input:

$\mathbf{X}$ – a signal.
$\{\mathbf{D}_i\}_{i=1}^K$ – convolutional dictionaries.
$\mathcal{P} \in \{\mathcal{H}, \mathcal{S}, \mathcal{S}^+\}$ – a thresholding operator.
$\{\beta_i\}_{i=1}^K$ – thresholds.

Output:

A set of representations $\{\hat{\mathbf{\Gamma}}_i\}_{i=1}^K$.

Process:

1: $\hat{\mathbf{\Gamma}}_0 \leftarrow \mathbf{X}$
2: **for** $i = 1 : K$ **do**
3: $\quad \hat{\mathbf{\Gamma}}_i \leftarrow \mathcal{P}_{\beta_i}(\mathbf{D}_i^T \hat{\mathbf{\Gamma}}_{i-1})$
4: **end for**

previously described in Section 2.2.1, assuming some conditions are met, the result of the thresholding algorithm, $\hat{\mathbf{\Gamma}}_1$, is guaranteed to have the correct support. In order to obtain the vector $\mathbf{\Gamma}_1$ itself, one should project the signal $\mathbf{X}$ onto this obtained support, by solving a Least-Squares problem. For reasons that will become clear shortly, we choose not to employ this step in the layered thresholding algorithm. Despite this algorithm failing in recovering the exact representations in the noiseless setting, as we shall see in Section 5, the estimated sparse vectors and the true ones are close – indicating the stability of this simple algorithm.

Thus far, we have assumed a noiseless setting. However, the same layered thresholding algorithm could be employed for the recovery of the representations of a noisy signal $\mathbf{Y} = \mathbf{X} + \mathbf{E}$, with the exception that the threshold constants, $\{\beta_i\}_{i=1}^K$, would be different and proportional to the noise level.

Assuming two layers for simplicity, Algorithm 1 can be summarized in the following equation

$$\hat{\mathbf{\Gamma}}_2 = \mathcal{P}_{\beta_2}\Big(\mathbf{D}_2^T \mathcal{P}_{\beta_1}\left(\mathbf{D}_1^T \mathbf{X} \right) \Big).$$

Comparing the above with Equation (1), given by

$$f(\mathbf{X}, \{\mathbf{W}_i\}_{i=1}^2, \{\mathbf{b}_i\}_{i=1}^2) = \text{ReLU}\Big(\mathbf{W}_2^T \ \text{ReLU}\left(\mathbf{W}_1^T \mathbf{X} + \mathbf{b}_1 \right) + \mathbf{b}_2 \Big),$$

one can notice a striking similarity between the two. Moreover, by replacing $\mathcal{P}_\beta(\cdot)$ with the soft nonnegative thresholding, $\mathcal{S}_\beta^+(\cdot)$, we obtain that *the aforementioned pursuit and the forward pass of the CNN are equal*! Notice that we are relying here on the discussion of Section 2.2.2, where we have shown that the ReLU and the soft nonnegative thresholding are equal[4].

4. A slight difference does exist between the soft nonnegative layered thresholding algorithm and the forward pass of the CNN. While in the former a constant threshold β is employed for all entries, the latter uses a bias vector, $\mathbf{b}$, that might not be constant in all of its entries. This is of little significance, however, since a similar approach of an entry-based constant could be used in the layered thresholding algorithm as well.

Recall the optimization problem of the training stage of the CNN as shown in Equation (2), given by

$$\min_{\{\mathbf{W}_i\}_{i=1}^K,\{\mathbf{b}_i\}_{i=1}^K,\mathbf{U}} \sum_j \ell\Big(h(\mathbf{X}_j),\mathbf{U},f\left(\mathbf{X}_j,\{\mathbf{W}_i\}_{i=1}^K,\{\mathbf{b}_i\}_{i=1}^K\right)\Big),$$

and its parallel in the ML-CSC model, the DLP$_{\boldsymbol{\lambda}}$ problem, defined by

$$\min_{\{\mathbf{D}_i\}_{i=1}^K,\mathbf{U}} \sum_j \ell\Big(h(\mathbf{X}_j),\mathbf{U},\mathrm{DCP}_{\boldsymbol{\lambda}}^{\star}(\mathbf{X}_j,\{\mathbf{D}_i\}_{i=1}^K)\Big).$$

Notice the remarkable similarity between both objectives, the only difference being in the feature vector on which the classification is done; in the CNN this is the output of the forward pass algorithm, given by $f\left(\mathbf{X}_j,\{\mathbf{W}_i\}_{i=1}^K,\{\mathbf{b}_i\}_{i=1}^K\right)$, while in the sparsity case this is the result of the DCP$_{\boldsymbol{\lambda}}$ problem. In light of the discussion above, the solution for the DCP$_{\boldsymbol{\lambda}}$ problem can be approximated using the layered thresholding algorithm, which is in turn equal to the forward pass of the CNN. We can therefore conclude that the problems solved by the training stage of the CNN and the DLP$_{\boldsymbol{\lambda}}$ are tightly connected, and in fact are equal once the solution for the DLP$_{\boldsymbol{\lambda}}$ is approximated via the layered thresholding algorithm (hence the name DLP$_{\boldsymbol{\lambda}}$).

5. Theoretical Study

Thus far, we have defined the ML-CSC model and its corresponding pursuits – the DCP$_{\boldsymbol{\lambda}}$ and DCP$_{\boldsymbol{\lambda}}^{\boldsymbol{\mathcal{E}}}$ problems. We have proposed a method to tackle them, coined the layered thresholding algorithm, which was shown to be equivalent to the forward pass of the CNN. Relying on this, we conclude that the proposed ML-CSC is the global Bayesian model implicitly imposed on the signal $\mathbf{X}$ when deploying the forward pass algorithm. Put differently, the ML-CSC answers the question of who are the signals belonging to the model behind the CNN. Having established the importance of our model, we now proceed to its theoretical analysis.

We should emphasize that the following study does not assume any specific form on the network's parameters, apart from a broad coherence property (as will be shown hereafter). This is in contrast to the work of (Bruna and Mallat, 2013) that assumes that the filters are Wavelets, or the analysis in (Giryes et al., 2015) that considers random weights.

5.1 Uniqueness of the DCP$_{\boldsymbol{\lambda}}$ Problem

Consider a signal $\mathbf{X}$ admitting a multi-layer convolutional sparse representation defined by the sets $\{\mathbf{D}_i\}_{i=1}^K$ and $\{\lambda_i\}_{i=1}^K$. Can another set of sparse vectors represent the signal $\mathbf{X}$? In other words, can we guarantee that, under some conditions, the set $\{\boldsymbol{\Gamma}_i\}_{i=1}^K$ is a unique solution to the DCP$_{\boldsymbol{\lambda}}$ problem? In the following theorem we provide an answer to this question.

Theorem 4 *(Uniqueness via the mutual coherence): Consider a signal* $\mathbf{X}$ *satisfying the* $DCP_{\boldsymbol{\lambda}}$ *model,*

$$\begin{aligned}\mathbf{X} &= \mathbf{D}_1\boldsymbol{\Gamma}_1\\ \boldsymbol{\Gamma}_1 &= \mathbf{D}_2\boldsymbol{\Gamma}_2\\ &\vdots\\ \boldsymbol{\Gamma}_{K-1} &= \mathbf{D}_K\boldsymbol{\Gamma}_K,\end{aligned}$$

where $\{\mathbf{D}_i\}_{i=1}^K$ *is a set of convolutional dictionaries and* $\{\mu(\mathbf{D}_i)\}_{i=1}^K$ *are their corresponding mutual coherences. If*

$$\forall\ 1 \le i \le K \qquad \|\boldsymbol{\Gamma}_i\|_{0,\infty}^{\mathbf{s}} < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D}_i)}\right),$$

then the set $\{\boldsymbol{\Gamma}_i\}_{i=1}^K$ *is the unique solution to the* $DCP_{\boldsymbol{\lambda}}$ *problem, assuming that the thresholds* $\{\lambda_i\}_{i=1}^K$ *are chosen to satisfy*

$$\forall\ 1 \le i \le K \qquad \|\boldsymbol{\Gamma}_i\|_{0,\infty}^{\mathbf{s}} \le \lambda_i < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D}_i)}\right).$$

The proof for the above theorem is given in Appendix A. In what follows, we present its importance in the context of CNN. Assume a signal $\mathbf{X}$ is fed into a network, resulting in a set of activation values across the different layers. These, in the realm of sparsity, correspond to the set of sparse representations $\{\boldsymbol{\Gamma}_i\}_{i=1}^K$, which according to the above theorem are in fact unique representations of the signal $\mathbf{X}$.

One might ponder at this point whether there exists an algorithm for obtaining the unique solution guaranteed in this subsection for the $\text{DCP}_{\boldsymbol{\lambda}}$ problem. As previously mentioned, the layered thresholding algorithm is incapable of finding the exact representations, $\{\boldsymbol{\Gamma}_i\}_{i=1}^K$, due to the lack of a Least-Squares step after each layer. One should not despair, however, as we shall see in a following section an alternative algorithm, which manages to overcome this hurdle.

5.2 Global Stability of the $\text{DCP}_{\boldsymbol{\lambda}}^{\boldsymbol{\mathcal{E}}}$ Problem

Consider an instance signal $\mathbf{X}$ belonging to the ML-CSC model, defined by the sets $\{\mathbf{D}_i\}_{i=1}^K$ and $\{\lambda_i\}_{i=1}^K$. Assume $\mathbf{X}$ is contaminated by a noise vector $\mathbf{E}$, generating the perturbed signal $\mathbf{Y} = \mathbf{X} + \mathbf{E}$. Suppose we solve the $\text{DCP}_{\boldsymbol{\lambda}}^{\boldsymbol{\mathcal{E}}}$ problem and obtain a set of solutions $\{\hat{\boldsymbol{\Gamma}}_i\}_{i=1}^K$. How close is every solution in this set, $\hat{\boldsymbol{\Gamma}}_i$, to its corresponding true representation, $\boldsymbol{\Gamma}_i$? In what follows, we provide a theorem addressing this question of stability, the proof of which is deferred to Appendix B.

Theorem 5 *(Stability of the solution to the* $DCP_{\boldsymbol{\lambda}}^{\boldsymbol{\mathcal{E}}}$ *problem): Suppose a signal* $\mathbf{X}$ *that has a decomposition*

$$\begin{aligned}\mathbf{X} &= \mathbf{D}_1\boldsymbol{\Gamma}_1\\ \boldsymbol{\Gamma}_1 &= \mathbf{D}_2\boldsymbol{\Gamma}_2\\ &\vdots\\ \boldsymbol{\Gamma}_{K-1} &= \mathbf{D}_K\boldsymbol{\Gamma}_K\end{aligned}$$

is contaminated with noise $\mathbf{E}$*, where* $\|\mathbf{E}\|_2 \leq \mathcal{E}_0$*, resulting in* $\mathbf{Y} = \mathbf{X}+\mathbf{E}$*. For all* $1 \leq i \leq K$*, if*

1. $\|\mathbf{\Gamma}_i\|_{0,\infty}^{\mathbf{S}} \leq \lambda_i < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D}_i)}\right)$*; and*
2. $\mathcal{E}_i^2 = \frac{4\mathcal{E}_{i-1}^2}{1-(2\|\mathbf{\Gamma}_i\|_{0,\infty}^{\mathbf{S}}-1)\mu(\mathbf{D}_i)}$,

then

$$\|\mathbf{\Gamma}_i - \hat{\mathbf{\Gamma}}_i\|_2^2 \leq \mathcal{E}_i^2,$$

where the set $\{\hat{\mathbf{\Gamma}}_i\}_{i=1}^K$ *is the solution for the DCP*$_{\boldsymbol{\lambda}}^{\boldsymbol{\mathcal{E}}}$ *problem.*

Intuitively, the above claims that as long as all the feature vectors $\{\mathbf{\Gamma}_i\}_{i=1}^K$ are $\ell_{0,\infty}$-sparse, then the representations obtained by solving the DCP$_{\boldsymbol{\lambda}}^{\boldsymbol{\mathcal{E}}}$ problem must be close to the true ones. Interestingly, the obtained bound increases as a function of the depth of the layer. This can be clearly seen from the recursive definition of $\mathcal{E}_i$, leading to the following bound

$$\|\mathbf{\Gamma}_i - \hat{\mathbf{\Gamma}}_i\|_2^2 \leq \mathcal{E}_0^2 \prod_{j=1}^{i} \frac{4}{1-(2\|\mathbf{\Gamma}_j\|_{0,\infty}^{\mathbf{S}}-1)\mu(\mathbf{D}_j)}.$$

Is this necessarily the true behavior of a deep network? Perhaps the answer to this resides in the choice we made above of considering the noise as adversary. A similar, yet somewhat more involved, analysis with a random noise assumption should be done, with the hope to see a better controlled noise propagation in this system. We leave this for our future work.

Another important remark is that the above bounds the *absolute error* between the estimated and the true representation. In practice, however, the *relative error* is of more importance. This is measured in terms of the signal to noise ratio (SNR), which we shall define in Section 8.

Having established the stability of the DCP$_{\boldsymbol{\lambda}}^{\boldsymbol{\mathcal{E}}}$ problem, we now turn to the stability of the algorithms attempting to solve it, the chief one being the forward pass of CNN.

5.3 Stability of the Layered Hard Thresholding

Consider a signal $\mathbf{X}$ that admits a multi-layer convolutional sparse representation, which is defined by the sets $\{\mathbf{D}_i\}_{i=1}^K$ and $\{\lambda_i\}_{i=1}^K$. Assume we run the layered hard thresholding algorithm on $\mathbf{X}$, obtaining the sparse vectors $\{\hat{\mathbf{\Gamma}}_i\}_{i=1}^K$. Under certain conditions, can we guarantee that the estimate $\hat{\mathbf{\Gamma}}_i$ recovers the true support of $\mathbf{\Gamma}_i$? or that the norm of the difference between the two is bounded? Assume $\mathbf{X}$ is contaminated with a noise vector $\mathbf{E}$, resulting in the measurement $\mathbf{Y} = \mathbf{X} + \mathbf{E}$. Assume further that this signal is then fed to the layered thresholding algorithm, resulting in another set of representations. How do the answers to the above questions change? To tackle these, we commence by presenting a stability claim for the simple hard thresholding algorithm, relying on the $\ell_{0,\infty}$ norm. We should note that the analysis conducted in this subsection is for the noisy scenario, and the results for the noiseless case are simply obtained by setting the noise level to zero.

Next, we present a localized ℓ_2 and ℓ_0 measure of a global vector that will prove to be useful in the following analysis.

Definition 6 *Define the* $\|\cdot\|_{2,\infty}^{\mathbf{P}}$ *and* $\|\cdot\|_{0,\infty}^{\mathbf{P}}$ *norm of* $\mathbf{\Gamma}_i$ *to be*

$$\|\mathbf{\Gamma}_i\|_{2,\infty}^{\mathbf{P}} = \max_j \|\mathbf{P}_{i,j}\mathbf{\Gamma}_i\|_2$$

and

$$\|\mathbf{\Gamma}_i\|_{0,\infty}^{\mathbf{P}} = \max_j \|\mathbf{P}_{i,j}\mathbf{\Gamma}_i\|_0,$$

respectively. The operator $\mathbf{P}_{i,j}$ *extracts the* j*-th patch of length* $n_i m_i$ *from the* i*-th sparse vector* $\mathbf{\Gamma}_i$*.*

In the above definition, the letter **p** emphasizes that the norms are computed by sweeping over all patches, rather than stripes. Recall that we have defined $m_0 = 1$, since the number of channels in the input signal $\mathbf{X} = \mathbf{\Gamma}_0$ is equal to one.

Given $\mathbf{Y} = \mathbf{X} + \mathbf{E} = \mathbf{D}_1\mathbf{\Gamma}_1 + \mathbf{E}$, the first stage of the layered hard thresholding algorithm attempts to recover the representation $\mathbf{\Gamma}_1$. Intuitively, assuming that the underlying representation $\mathbf{\Gamma}_1$ is $\ell_{0,\infty}$-sparse, and that the energy of the noise $\mathbf{E}$ is $\ell_{2,\infty}$-bounded; we would expect that the simple hard thresholding algorithm would succeed in recovering a solution $\hat{\mathbf{\Gamma}}_1$, which is both close to $\mathbf{\Gamma}_1$ and has its support. We now present such a claim, the proof of which is found in Appendix C.

Lemma 7 *(Stable recovery of hard thresholding in the presence of noise): Suppose a clean signal* $\mathbf{X}$ *has a convolutional sparse representation* $\mathbf{D}_1\mathbf{\Gamma}_1$*, and that it is contaminated with noise* $\mathbf{E}$ *to create the signal* $\mathbf{Y} = \mathbf{X} + \mathbf{E}$*, such that* $\|\mathbf{E}\|_{2,\infty}^{\mathbf{P}} \leq \epsilon_0$*. Denote by* $|\Gamma_1^{min}|$ *and* $|\Gamma_1^{max}|$ *the lowest and highest entries in absolute value in* $\mathbf{\Gamma}_1$*, respectively. Denote further by* $\hat{\mathbf{\Gamma}}_1$ *the solution obtained by running the hard thresholding algorithm on* $\mathbf{Y}$ *with a constant* β_1*, i.e.* $\hat{\mathbf{\Gamma}}_1 = \mathcal{H}_{\beta_1}(\mathbf{D}_1^T\mathbf{Y})$*. Assuming that*

a) $\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{S}} < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D}_1)}\frac{|\Gamma_1^{min}|}{|\Gamma_1^{max}|}\right) - \frac{1}{\mu(\mathbf{D}_1)}\frac{\epsilon_0}{|\Gamma_1^{max}|}$*; and*

b) The threshold β_1 *is chosen according to Equation* (13) *(see below),*

then the following must hold:

1. *The support of the solution* $\hat{\mathbf{\Gamma}}_1$ *is equal to that of* $\mathbf{\Gamma}_1$*; and*
2. $\|\mathbf{\Gamma}_1 - \hat{\mathbf{\Gamma}}_1\|_{2,\infty}^{\mathbf{P}} \leq \sqrt{\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{P}}}\Big(\epsilon_0 + \mu(\mathbf{D}_1)\left(\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{S}} - 1\right)|\Gamma_1^{max}|\Big)$.

Notice that by plugging $\epsilon_0 = 0$ the above theorem covers the noiseless scenario. Notably, even in such a case, we obtain a deviation from the true representation due to the lack of a Least-Squares step.

We suspect that, both in the noiseless and the noisy case, the obtained bound might be improved, based on the following observation. Given an $\ell_{2,\infty}$-norm bounded noise, the above proof first quantifies the deviation between the true representation and the estimated one in terms of the ℓ_∞ norm, and only then translates the latter into the $\ell_{2,\infty}$ sense. A direct analysis going from an $\ell_{2,\infty}$ input error to an $\ell_{2,\infty}$ output deviation (bypassing the ℓ_∞ norm) might lead to smaller deviations. We leave this for future work.

We now proceed to the next layer. Given $\hat{\mathbf{\Gamma}}_1$, which can be considered as a perturbed version of $\mathbf{\Gamma}_1$, the second stage of the layered hard thresholding algorithm attempts to

recover the representation $\mathbf{\Gamma}_2$. Using the stability of the first layer – guaranteeing that $\mathbf{\Gamma}_1$ and $\hat{\mathbf{\Gamma}}_1$ are close in terms of the $\ell_{2,\infty}$ norm – and relying on the $\ell_{0,\infty}$-sparsity of $\mathbf{\Gamma}_2$, we show next that the second stage of the layered hard thresholding algorithm is stable as well. Applying the same rationale to all the remaining layers, we obtain the theorem below guaranteeing the stability of the complete layered hard thresholding algorithm.

Theorem 8 *(Stability of layered hard thresholding in the presence of noise): Suppose a clean signal* $\mathbf{X}$ *has a decomposition*

$$\begin{aligned}\mathbf{X} &= \mathbf{D}_1\mathbf{\Gamma}_1\\ \mathbf{\Gamma}_1 &= \mathbf{D}_2\mathbf{\Gamma}_2\\ &\vdots\\ \mathbf{\Gamma}_{K-1} &= \mathbf{D}_K\mathbf{\Gamma}_K,\end{aligned}$$

and that it is contaminated with noise $\mathbf{E}$ *to create the signal* $\mathbf{Y} = \mathbf{X}+\mathbf{E}$*, such that* $\|\mathbf{E}\|_{2,\infty}^{\mathbf{P}} \leq \epsilon_0$*. Denote by* $|\Gamma_i^{min}|$ *and* $|\Gamma_i^{max}|$ *the lowest and highest entries in absolute value in the vector* $\mathbf{\Gamma}_i$*, respectively. Let* $\{\hat{\mathbf{\Gamma}}_i\}_{i=1}^K$ *be the set of solutions obtained by running the layered hard thresholding algorithm with thresholds* $\{\beta_i\}_{i=1}^K$*, i.e.* $\hat{\mathbf{\Gamma}}_i = \mathcal{H}_{\beta_i}(\mathbf{D}_i^T\hat{\mathbf{\Gamma}}_{i-1})$ *where* $\hat{\mathbf{\Gamma}}_0 = \mathbf{Y}$*. Assuming that* $\forall\ 1 \leq i \leq K$

a) $\|\mathbf{\Gamma}_i\|_{0,\infty}^{\mathbf{S}} < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D}_i)}\frac{|\Gamma_i^{min}|}{|\Gamma_i^{max}|}\right) - \frac{1}{\mu(\mathbf{D}_i)}\frac{\epsilon_{i-1}}{|\Gamma_i^{max}|}$*; and*

b) The threshold β_i *is chosen according to Equation* (18)*,*

then[5]

1. *The support of the solution* $\hat{\mathbf{\Gamma}}_i$ *is equal to that of* $\mathbf{\Gamma}_i$*; and*
2. $\|\mathbf{\Gamma}_i - \hat{\mathbf{\Gamma}}_i\|_{2,\infty}^{\mathbf{P}} \leq \epsilon_i$*,*

where $\epsilon_i = \sqrt{\|\mathbf{\Gamma}_i\|_{0,\infty}^{\mathbf{P}}}\left(\epsilon_{i-1} + \mu(\mathbf{D}_i)\left(\|\mathbf{\Gamma}_i\|_{0,\infty}^{\mathbf{S}} - 1\right)|\Gamma_i^{max}|\right)$.

The proof for the above is given in Appendix D. We now turn to an analogous theorem for the forward pass of the CNN, prior to discussing the surprising implications of these theorems.

5.4 Stability of the Forward Pass (Layered Soft Thresholding)

In light of the discussion in Section 4, the equivalence between the layered thresholding algorithm and the forward pass of the CNN is achieved assuming that the operator employed is the nonnegative soft thresholding $\mathcal{S}_\beta^+(\cdot)$. However, thus far, we have analyzed the closely related hard version $\mathcal{H}_\beta(\cdot)$ instead. In what follows, we show how the stability

5. Recall that $\|\mathbf{\Gamma}_i\|_{2,\infty}^{\mathbf{P}}$ is defined to be the maximal ℓ_2 norm of a patch extract from $\mathbf{\Gamma}_i$. The size of this patch is defined according to the dictionary $\mathbf{D}_{i+1}$. However, the last sparse vector $\mathbf{\Gamma}_K$ does not have a corresponding dictionary $\mathbf{D}_{K+1}$. As such, the size of a patch in $\mathbf{\Gamma}_K$ can be chosen arbitrarily. Where the choice of the size directly affects the bound on the difference, ϵ_i, due to the term $\sqrt{\|\mathbf{\Gamma}_i\|_{0,\infty}^{\mathbf{P}}}$.

theorem presented in the previous subsection can be modified to the soft version, $\mathcal{S}_\beta(\cdot)$. For simplicity, and in order to stay in line with the vast sparse representation theory, herein we choose not to assume the nonnegative assumption. This implies that we are proposing a slightly different CNN architecture in which the ReLU function is two sided (Kavukcuoglu et al., 2010). We now move to the stable recovery of the soft thresholding algorithm.

Lemma 9 *(Stable recovery of soft thresholding in the presence of noise): Suppose a clean signal* $\mathbf{X}$ *has a convolutional sparse representation* $\mathbf{D}_1\mathbf{\Gamma}_1$*, and that it is contaminated with noise* $\mathbf{E}$ *to create the signal* $\mathbf{Y} = \mathbf{X} + \mathbf{E}$*, such that* $\|\mathbf{E}\|_{2,\infty}^{\mathbf{P}} \leq \epsilon_0$*. Denote by* $|\Gamma_1^{min}|$ *and* $|\Gamma_1^{max}|$ *the lowest and highest entries in absolute value in* $\mathbf{\Gamma}_1$*, respectively. Denote further by* $\hat{\mathbf{\Gamma}}_1$ *the solution obtained by running the soft thresholding algorithm on* $\mathbf{Y}$ *with a constant* β_1*, i.e.* $\hat{\mathbf{\Gamma}}_1 = \mathcal{S}_{\beta_1}(\mathbf{D}_1^T\mathbf{Y})$*. Assuming that*

a) $\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{S}} < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D}_1)}\frac{|\Gamma_1^{min}|}{|\Gamma_1^{max}|}\right) - \frac{1}{\mu(\mathbf{D}_1)}\frac{\epsilon_0}{|\Gamma_1^{max}|}$*; and*

b) The threshold β_1 *is chosen according to Equation* (13)*,*

then the following must hold:

1. *The support of the solution* $\hat{\mathbf{\Gamma}}_1$ *is equal to that of* $\mathbf{\Gamma}_1$*; and*
2. $\|\mathbf{\Gamma}_1 - \hat{\mathbf{\Gamma}}_1\|_{2,\infty}^{\mathbf{P}} \leq \sqrt{\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{P}}}\Big(\epsilon_0 + \mu(\mathbf{D}_1)\left(\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{S}} - 1\right)|\Gamma_1^{max}| + \beta_1\Big)$.

Armed with the above lemma, which is proven in Appendix E, we now proceed to the stability of the forward pass of the CNN.

Theorem 10 *(Stability of the forward pass (layered soft thresholding algorithm) in the presence of noise): Suppose a clean signal* $\mathbf{X}$ *has a decomposition*

$$\begin{aligned}\mathbf{X} &= \mathbf{D}_1\mathbf{\Gamma}_1\\ \mathbf{\Gamma}_1 &= \mathbf{D}_2\mathbf{\Gamma}_2\\ &\vdots\\ \mathbf{\Gamma}_{K-1} &= \mathbf{D}_K\mathbf{\Gamma}_K,\end{aligned}$$

and that it is contaminated with noise $\mathbf{E}$ *to create the signal* $\mathbf{Y} = \mathbf{X} + \mathbf{E}$*, such that* $\|\mathbf{E}\|_{2,\infty}^{\mathbf{P}} \leq \epsilon_0$*. Denote by* $|\Gamma_i^{min}|$ *and* $|\Gamma_i^{max}|$ *the lowest and highest entries in absolute value in the vector* $\mathbf{\Gamma}_i$*, respectively. Let* $\{\hat{\mathbf{\Gamma}}_i\}_{i=1}^K$ *be the set of solutions obtained by running the layered soft thresholding algorithm with thresholds* $\{\beta_i\}_{i=1}^K$*, i.e.* $\hat{\mathbf{\Gamma}}_i = \mathcal{S}_{\beta_i}(\mathbf{D}_i^T\hat{\mathbf{\Gamma}}_{i-1})$ *where* $\hat{\mathbf{\Gamma}}_0 = \mathbf{Y}$*. Assuming that* $\forall\ 1 \leq i \leq K$

a) $\|\mathbf{\Gamma}_i\|_{0,\infty}^{\mathbf{S}} < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D}_i)}\frac{|\Gamma_i^{min}|}{|\Gamma_i^{max}|}\right) - \frac{1}{\mu(\mathbf{D}_i)}\frac{\epsilon_{i-1}}{|\Gamma_i^{max}|}$*; and*

b) The threshold β_i *is chosen according to Equation* (18) *(with the* ϵ_i *defined below),*

then

1. *The support of the solution* $\hat{\mathbf{\Gamma}}_i$ *is equal to that of* $\mathbf{\Gamma}_i$*; and*

2. $\|\mathbf{\Gamma}_i - \hat{\mathbf{\Gamma}}_i\|_{2,\infty}^{\mathbf{P}} \leq \epsilon_i$,

where $\epsilon_i = \sqrt{\|\mathbf{\Gamma}_i\|_{0,\infty}^{\mathbf{P}}} \left(\epsilon_{i-1} + \mu(\mathbf{D}_i)\left(\|\mathbf{\Gamma}_i\|_{0,\infty}^{\mathbf{S}} - 1\right)|\Gamma_i^{max}| + \beta_i\right)$.

The proof for the above is omitted since it is tantamount to that of Theorem 8. As one can see, the layered soft thresholding algorithm is in fact inferior to its hard variant due to the added constant of β_i in the local error level, ϵ_i. This results in a more strict assumption on the $\ell_{0,\infty}$ norm of the various representations and also augments the bound on the distance between the true sparse vector and the one recovered. Following this observation, a natural question arises; why does the deep learning community employ the ReLU, which corresponds to a soft nonnegative thresholding operator instead of another nonlinearity that is more similar to its hard counterpart? One possible explanation could be that the filter training stage of the CNN becomes harder when the ReLU is replaced with a non-convex alternative, which also has discontinuities, such as the hard thresholding operator.

The above theorem guarantees that the distances between the original representations and the ones obtained from the CNN are bounded. Even if we set $\epsilon_0 = 0$, the recovered activations deviate from the true ones, simply because the layered thresholding algorithm does not do a perfect job, even on a noiseless signal. When the signal is noisy, these deviations are strengthened, but still in a controlled way.

This, by itself, might not be surprising. After all, the CNN is a deterministic system of linear operations (convolutions), followed by simple non-linearities that are non-expanding. If we feed a slightly perturbed signal to such a system, it is clear that the activations all along the network will be perturbed as well with a bounded effect. However, the above theorem shows far more than that. There are, in fact, two types of stabilities, the trivial one that considers the sensitivity of the whole feed-forward network to perturbations in its input, and the more intricate one that shows that this system enables a rather accurate recovery of the **generating representations**. The second option is the stability we prove here.

5.5 Guarantees for Fully Connected Networks

One should note that the convolutional structure imposed on the dictionaries in our model could be removed, and the theoretical guarantees we have provided above would still hold. The reason being is that the unconstrained dictionary can be regarded as a convolutional one, constructed from a single shift of a local matrix with no circular boundary. In the context of CNN, this is analogous to a fully connected layer. As such, the theoretical analysis provided here sheds light on both convolutional and fully connected networks. A different point of view on the same matter can also be proposed; fully connected layers can be viewed as convolutional ones with filters that cover their entire input (Long et al., 2015).

6. Layered Basis Pursuit – The Future of Deep Learning?

The stability analysis presented above unveils two significant limitations of the forward pass of the CNN. First, this algorithm is incapable of recovering the unique solution for the $\text{DCP}_{\boldsymbol{\lambda}}$ problem, the existence of which is guaranteed from Theorem 4. This acts against

our expectations, since in the traditional sparsity inspired model it is a well known fact that such a unique representation can be retrieved, assuming certain conditions are met.

The second issue is with the condition for the successful recovery of the true support. The $\ell_{0,\infty}$ norm of the true solution, $\mathbf{\Gamma}_i$, is required to be less than an expression that depends on the term $|\Gamma_i^{\min}|/|\Gamma_i^{\max}|$. The dependence on this ratio is a direct consequence of the forward pass algorithm relying on the simple thresholding operator that is known for having such a theoretical limitation[6]. However, alternative pursuits whose success would not depend on this ratio could be proposed, as indeed was done in the Sparse-Land model; resulting in both theoretical and practical benefits.

A solution for the first problem, already presented throughout this work, is a two-stage approach. First, run the thresholding operator in order to recover the correct support. Then, once the atoms are chosen, their corresponding coefficients can be obtained by solving a linear system of equations. In addition to retrieving the true representation in the noiseless case, this step can also be beneficial in the noisy scenario, resulting in a solution closer to the underlying one. However, since no such step exists in current CNN architectures, we refrain from further analyzing its theoretical implications.

Next, we present an alternative to the layered soft thresholding algorithm, which will tackle both of the aforementioned problems. Recall that the result of the soft thresholding is a simple approximation of the solution for the P_1 problem, previously defined in Equation (4). In every layer, instead of applying a simple thresholding operator that estimates the sparse vector by computing $\hat{\mathbf{\Gamma}}_i = \mathcal{S}_{\beta_i}(\mathbf{D}_i^T \hat{\mathbf{\Gamma}}_{i-1})$; we propose to tackle the full pursuit, i.e. to minimize

$$\hat{\mathbf{\Gamma}}_i = \underset{\mathbf{\Gamma}_i}{\arg\min} \ \|\mathbf{\Gamma}_i\|_1 \ \ \text{s.t.} \ \ \hat{\mathbf{\Gamma}}_{i-1} = \mathbf{D}_i \mathbf{\Gamma}_i. \tag{7}$$

Notice that one could readily obtain the nonnegative sparse coding problem by simply adding an extra constraint in the above equation, forcing the coefficients in $\mathbf{\Gamma}_i$ to be nonnegative. More generally, Equation (7) can be written in its Lagrangian formulation

$$\hat{\mathbf{\Gamma}}_i = \underset{\mathbf{\Gamma}_i}{\arg\min} \ \xi_i \|\mathbf{\Gamma}_i\|_1 + \frac{1}{2}\|\mathbf{D}_i \mathbf{\Gamma}_i - \hat{\mathbf{\Gamma}}_{i-1}\|_2^2, \tag{8}$$

where the constant ξ_i is proportional to the noise level and should tend to zero in the noiseless scenario. We name the above the *layered basis pursuit* (BP) algorithm. In practice, one possible method for solving it is the iterative soft thresholding (IST). Formally, this obtains the minimizer of Equation (8) by repeating the following recursive formula

$$\hat{\mathbf{\Gamma}}_i^t = \mathcal{S}_{\xi_i/c_i}\left(\hat{\mathbf{\Gamma}}_i^{t-1} + \frac{1}{c_i}\mathbf{D}_i^T\left(\hat{\mathbf{\Gamma}}_{i-1} - \mathbf{D}_i \hat{\mathbf{\Gamma}}_i^{t-1}\right)\right), \tag{9}$$

where $\hat{\mathbf{\Gamma}}_i^t$ is the estimate of $\mathbf{\Gamma}_i$ at iteration t. The above can be interpreted as a simple projected gradient descent algorithm, where the constant c_i is inversely proportional to its step size. As a result, if c_i is chosen to be large enough[7], the above algorithm is

6. The dependence on the ratio is also a direct consequence of assuming a worst-case analysis. Perhaps in reality this ratio does not play such a critical role.
7. The constant c_i should satisfy $c_i > 0.5\lambda_{\max}\left(\mathbf{D}_i^T\mathbf{D}_i\right)$, where $\lambda_{\max}\left(\mathbf{D}_i^T\mathbf{D}_i\right)$ is the maximal eigenvalue of the gram matrix $\mathbf{D}_i^T\mathbf{D}_i$ (Combettes and Wajs, 2005).

Algorithm 2 The layered iterative soft thresholding algorithm.

Input:

$\mathbf{X}$ – a signal.
$\{\mathbf{D}_i\}_{i=1}^K$ – convolutional dictionaries.
$\mathcal{P} \in \{\mathcal{S}, \mathcal{S}^+\}$ – a soft thresholding operator.
$\{\xi_i\}_{i=1}^K$ – Lagrangian parameters.
$\{1/c_i\}_{i=1}^K$ – step sizes.
$\{T_i\}_{i=1}^K$ – number of iterations.

Output:

A set of representations $\{\hat{\mathbf{\Gamma}}_i\}_{i=1}^K$.

Process:

$\hat{\mathbf{\Gamma}}_0 \leftarrow \mathbf{X}$
for $i = 1 : K$ **do**
  $\hat{\mathbf{\Gamma}}_i^0 \leftarrow \mathbf{0}$
  for $t = 1 : T_i$ **do**
    $\hat{\mathbf{\Gamma}}_i^t \leftarrow \mathcal{P}_{\xi_i/c_i}\left(\hat{\mathbf{\Gamma}}_i^{t-1} + \frac{1}{c_i}\mathbf{D}_i^T\left(\hat{\mathbf{\Gamma}}_{i-1} - \mathbf{D}_i\hat{\mathbf{\Gamma}}_i^{t-1}\right)\right)$
  end for
  $\hat{\mathbf{\Gamma}}_i \leftarrow \hat{\mathbf{\Gamma}}_i^{T_i}$
end for

guaranteed to converge to its global minimum that is the solution of (8), as was shown in (Daubechies et al., 2004). The method obtained by gradually computing the set of sparse representations, $\{\mathbf{\Gamma}_i\}_{i=1}^K$, via the IST is summarized in Algorithm 2 and named *layered iterative soft thresholding*. Notice that this algorithm coincides with the simple layered soft thresholding if it is run for a single iteration with $c_i = 1$ and initialized with $\hat{\mathbf{\Gamma}}_i^0 = \mathbf{0}$. This implies that the above algorithm is a natural extension to the forward pass of the CNN.

With respect to the computational aspects of the IST algorithm, the work of (Gregor and LeCun, 2010) proposed the LISTA method, showing how the number of iterations required by the IST to convergence can be reduced using neural networks. Analogously, the work of (Xin et al., 2016) presented a generalization of the iterative hard thresholding (IHT), which was shown both theoretically and empirically to be superior to the original IHT.

The original motivation for the layered IST was its theoretical superiority over the forward pass algorithm – one that will be explored in detail in the next subsection. Yet more can be said about this algorithm and the CNN architecture it induces. In (Gregor and LeCun, 2010) it was shown that the IST algorithm can be formulated as a simple recurrent neural network. As such, the same can be said regarding the layered IST algorithm proposed here, with the exception that the induced recurrent network is much deeper. The reader can therefore interpret this part of the work as a theoretical study of a special case of recurrent neural networks.

From another perspective, the underlying architecture of the layered IST algorithm is a cascade of K blocks. Each of these corresponds to a fixed number of unfolded iterations, T_i, of a single IST algorithm. As such, it contains several convolutional layers with shared weights, as well as skip connections in order to compute the residual, $\hat{\mathbf{\Gamma}}_{i-1} - \mathbf{D}_i\hat{\mathbf{\Gamma}}_i^{t-1}$, as

defined in Equation (9). Interestingly, the above description is reminiscent (though not exact) of residual networks (He et al., 2015), which have recently led to state-of-the-art results in image recognition.

6.1 Success of Layered BP Algorithm

In Section 5.1, we established the uniqueness of the solution for the $\text{DCP}_{\boldsymbol{\lambda}}$ problem, assuming that certain conditions on the $\ell_{0,\infty}$ norm of the underlying representations are met. However, as we have seen in the theoretical analysis of the previous section, the forward pass of the CNN is incapable of finding this unique solution; instead, it is guaranteed to be close to it in terms of the $\ell_{2,\infty}$ norm. Herein, we address the question of whether the layered BP algorithm can prevail in a task where the forward pass did not.

Theorem 11 *(Layered BP recovery guarantee using the $\ell_{0,\infty}$ norm): Consider a signal $\mathbf{X}$,*

$$\begin{aligned}\mathbf{X} &= \mathbf{D}_1\boldsymbol{\Gamma}_1\\ \boldsymbol{\Gamma}_1 &= \mathbf{D}_2\boldsymbol{\Gamma}_2\\ &\vdots\\ \boldsymbol{\Gamma}_{K-1} &= \mathbf{D}_K\boldsymbol{\Gamma}_K,\end{aligned}$$

where $\{\mathbf{D}_i\}_{i=1}^K$ is a set of convolutional dictionaries and $\{\mu(\mathbf{D}_i)\}_{i=1}^K$ are their corresponding mutual coherences. Assuming that $\forall\ 1 \leq i \leq K$

$$\|\boldsymbol{\Gamma}_i\|_{0,\infty}^{\mathbf{s}} < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D}_i)}\right)$$

then the layered BP algorithm is guaranteed to recover the set $\{\boldsymbol{\Gamma}_i\}_{i=1}^K$.

The proof for the above can be directly derived from the recovery condition of the BP using the $\ell_{0,\infty}$ norm, as presented in (Papyan et al., 2016a). The implications of this theorem are that the layered BP algorithm can indeed recover the unique solution to the $\text{DCP}_{\boldsymbol{\lambda}}$ problem.

6.2 Stability of Layered BP Algorithm

Having established the guarantee for the success of the layered BP algorithm, we now move to its stability analysis. In particular, in a noisy scenario where obtaining the true underlying representations is impossible, does this algorithm remain stable? If so, how do its guarantees compare to those of the layered thresholding algorithm? The following theorem, which we prove in Appendix F, aims to answer these questions.

Theorem 12 *(Stability of the layered BP algorithm in the presence of noise): Suppose a clean signal $\mathbf{X}$ has a decomposition*

$$\begin{aligned}\mathbf{X} &= \mathbf{D}_1\boldsymbol{\Gamma}_1\\ \boldsymbol{\Gamma}_1 &= \mathbf{D}_2\boldsymbol{\Gamma}_2\\ &\vdots\\ \boldsymbol{\Gamma}_{K-1} &= \mathbf{D}_K\boldsymbol{\Gamma}_K,\end{aligned}$$

and that it is contaminated with noise $\mathbf{E}$ *to create the signal* $\mathbf{Y} = \mathbf{X}+\mathbf{E}$*, such that* $\|\mathbf{E}\|_{2,\infty}^{\mathbf{P}} \leq \epsilon_0$*. Let* $\{\hat{\mathbf{\Gamma}}_i\}_{i=1}^K$ *be the set of solutions obtained by running the layered BP algorithm with parameters* $\{\xi_i\}_{i=1}^K$*. Assuming that* $\forall\ 1 \leq i \leq K$

a) $\|\mathbf{\Gamma}_i\|_{0,\infty}^{\mathbf{S}} < \frac{1}{3}\left(1 + \frac{1}{\mu(\mathbf{D}_i)}\right)$*; and*

b) $\xi_i = 4\epsilon_{i-1}$*,*

then

1. *The support of the solution* $\hat{\mathbf{\Gamma}}_i$ *is contained in that of* $\mathbf{\Gamma}_i$*;*
2. $\|\mathbf{\Gamma}_i - \hat{\mathbf{\Gamma}}_i\|_{2,\infty}^{\mathbf{P}} \leq \epsilon_i$*;*
3. *In particular, every entry of* $\mathbf{\Gamma}_i$ *greater in absolute value than* $\frac{\epsilon_i}{\sqrt{\|\mathbf{\Gamma}_i\|_{0,\infty}^{\mathbf{P}}}}$ *is guaranteed to be recovered; and*
4. *The solution* $\hat{\mathbf{\Gamma}}_i$ *is the unique minimizer of the Lagrangian BP problem (Equation* (8)*),*

where $\epsilon_i = \|\mathbf{E}\|_{2,\infty}^{\mathbf{P}}\ 7.5^i\ \prod_{j=1}^{i}\sqrt{\|\mathbf{\Gamma}_j\|_{0,\infty}^{\mathbf{P}}}$.

Several remarks are due at this point. The condition for the stability of the layered thresholding algorithm, given by

$$\|\mathbf{\Gamma}_i\|_{0,\infty}^{\mathbf{S}} < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D}_i)}\frac{|\Gamma_i^{\min}|}{|\Gamma_i^{\max}|}\right) - \frac{1}{\mu(\mathbf{D}_i)}\frac{\epsilon_{i-1}}{|\Gamma_i^{\max}|},$$

is expected to be more strict than that of the theorem presented above, which is

$$\|\mathbf{\Gamma}_i\|_{0,\infty}^{\mathbf{S}} < \frac{1}{3}\left(1 + \frac{1}{\mu(\mathbf{D}_i)}\right).$$

In the case of the layered BP algorithm, the bound on the $\ell_{0,\infty}$ norm of the underlying sparse vectors does no longer depend on the ratio $|\Gamma_i^{\min}|/|\Gamma_i^{\max}|$ – a term present in all the theoretical results of the thresholding algorithm. Moreover, the $\ell_{0,\infty}$ norm becomes independent of the local noise level of the previous layers, thus allowing more non-zeros per stripe.

In addition, similar to the stability analysis presented in Section 5.2, the above shows the growth (as a function of the depth) of the distance between the recovered representations and the true ones.

7. A Closer Look at the Proposed Model

In this section, we revisit the assumptions of our model by imposing additional constraints on the dictionaries involved and showing their theoretical benefits. These additional assumptions originate from the current common practice of both CNN and sparsity.

7.1 When a Patch Becomes a Stripe

Throughout the analysis presented in this work, we have assumed that the representations in the different layers, $\{\mathbf{\Gamma}_i\}_{i=1}^K$, are $\ell_{0,\infty}$-sparse. Herein, we study the propagation of the $\ell_{0,\infty}$ norm throughout the layers of the network, showing how an assumption on the sparsity of the deepest representation $\mathbf{\Gamma}_K$ reflects on that of the remaining layers. The exact connection between the sparsities will be given in terms of a simple characterization of the dictionaries $\{\mathbf{D}_i\}_{i=1}^K$.

Consider the representation $\mathbf{\Gamma}_{K-1}$, given by

$$\mathbf{\Gamma}_{K-1} = \mathbf{D}_K \mathbf{\Gamma}_K,$$

where $\mathbf{\Gamma}_K$ is $\ell_{0,\infty}$-sparse. Following Figure 4, the i-th patch in $\mathbf{\Gamma}_{K-1}$ can be expressed as

$$\mathbf{P}_{K-1,i}\mathbf{\Gamma}_{K-1} = \mathbf{\Omega}_K \boldsymbol{\gamma}_{K,i},$$

where $\mathbf{\Omega}_K$ is the stripe-dictionary of $\mathbf{D}_K$, the vector $\mathbf{P}_{K-1,i}\mathbf{\Gamma}_{K-1}$ is the i-th patch in $\mathbf{\Gamma}_{K-1}$ and $\boldsymbol{\gamma}_{K,i}$ is its corresponding stripe. Recalling the definition of the $\|\cdot\|_{0,\infty}^{\mathbf{P}}$ norm (Definition 6 in Section 5.3), we have that

$$\|\mathbf{\Gamma}_{K-1}\|_{0,\infty}^{\mathbf{P}} = \max_i \|\mathbf{\Omega}_K \boldsymbol{\gamma}_{K,i}\|_0.$$

Consider the following definition.

Definition 13 *Define the induced ℓ_0 pseudo-norm of a dictionary $\mathbf{D}$, denoted by $\|\mathbf{D}\|_0$, to be the maximal number of non-zeros in any of its atoms*[8]*.*

The multiplication $\mathbf{\Omega}_K \boldsymbol{\gamma}_{K,i}$ can be seen as a linear combination of at most $\|\boldsymbol{\gamma}_{K,i}\|_0$ atoms, each contributing no more than $\|\mathbf{\Omega}_K\|_0$ non-zeros. As such

$$\|\mathbf{\Gamma}_{K-1}\|_{0,\infty}^{\mathbf{P}} \leq \max_i \|\mathbf{\Omega}_K\|_0 \ \|\boldsymbol{\gamma}_{K,i}\|_0.$$

Noticing that $\|\mathbf{\Omega}_K\|_0 = \|\mathbf{D}_K\|_0$ (as can be seen in Figure 4), and using the definition of the $\|\cdot\|_{0,\infty}^{\mathbf{S}}$ norm, we conclude that

$$\|\mathbf{\Gamma}_{K-1}\|_{0,\infty}^{\mathbf{P}} \leq \|\mathbf{D}_K\|_0 \ \|\mathbf{\Gamma}_K\|_{0,\infty}^{\mathbf{S}}. \tag{10}$$

In other words, given $\|\mathbf{\Gamma}_K\|_{0,\infty}^{\mathbf{S}}$ and $\|\mathbf{D}_K\|_0$, we can bound the maximal number of non-zeros in a patch from $\mathbf{\Gamma}_{K-1}$.

The claims in Section 5 and 6 are given in terms of not only $\|\mathbf{\Gamma}_{K-1}\|_{0,\infty}^{\mathbf{P}}$, but also $\|\mathbf{\Gamma}_{K-1}\|_{0,\infty}^{\mathbf{S}}$. According to Table 1, the length of a patch in $\mathbf{\Gamma}_{K-1}$ is $n_{K-1}m_{K-1}$, while the size of a stripe is $(2n_{K-2}-1)m_{K-1}$. As such, we can fit $(2n_{K-2}-1)/n_{K-1}$ patches in a stripe. Assume for simplicity that this ratio is equal to one. As a result, we obtain that a patch in the *signal* $\mathbf{\Gamma}_{K-1}$ extracted from the system

$$\mathbf{\Gamma}_{K-1} = \mathbf{D}_K \mathbf{\Gamma}_K,$$

8. According to the definition of the induced norm $\|\mathbf{D}\|_0 = \max_{\mathbf{v}} \|\mathbf{D}\mathbf{v}\|_0$ s.t. $\|\mathbf{v}\|_0 = 1$. Since $\|\mathbf{v}\|_0 = 1$, the multiplication $\mathbf{D}\mathbf{v}$ is simply equal to one of the atoms in $\mathbf{D}$ times a scalar, and $\|\mathbf{D}\mathbf{v}\|_0$ counts the number of non-zeros in this atom. As a result, $\|\mathbf{D}\|_0$ is equal to the maximal number of non-zeros in any atom from $\mathbf{D}$.

is also a stripe in the *representation* $\mathbf{\Gamma}_{K-1}$ when considering

$$\mathbf{\Gamma}_{K-2} = \mathbf{D}_{K-1}\mathbf{\Gamma}_{K-1},$$

hence the name of this subsection. Leveraging this assumption, we return to Equation (10) and obtain that

$$\|\mathbf{\Gamma}_{K-1}\|_{0,\infty}^{\mathbf{S}} = \|\mathbf{\Gamma}_{K-1}\|_{0,\infty}^{\mathbf{P}} \leq \|\mathbf{D}_K\|_0 \ \|\mathbf{\Gamma}_K\|_{0,\infty}^{\mathbf{S}}.$$

Using the same rationale for the remaining layers, and assuming that once again the patches become stripes, we conclude that

$$\|\mathbf{\Gamma}_i\|_{0,\infty}^{\mathbf{S}} = \|\mathbf{\Gamma}_i\|_{0,\infty}^{\mathbf{P}} \leq \|\mathbf{\Gamma}_K\|_{0,\infty}^{\mathbf{S}} \prod_{j=i+1}^{K} \|\mathbf{D}_j\|_0. \tag{11}$$

We note that our assumption here of having sparse dictionaries is reasonable, since at the training stage of the CNN an ℓ_1 penalty is often imposed on the filters as a regularization, promoting their sparsity. The conclusion thus is that the $\ell_{0,\infty}$ norm is expected to decrease as a function of the depth of the representation. This aligns with the intuition that the higher the depth, the more abstraction one obtains in the filters, and thus the less non-zeros are required to represent the data. Taking this to the extreme, if every input signal could be represented via a single coefficient at the deepest layer, we would obtain that its $\ell_{0,\infty}$ norm is equal to one.

7.2 On the Role of the Spatial-Stride

A common step among practitioners of CNN (Krizhevsky et al., 2012; Simonyan and Zisserman, 2014; He et al., 2015) is to convolve the input to each layer with a set of filters, skipping a fixed number of spatial locations in a regular pattern. One of the primary motivations for this is to reduce the dimensions of the kernel maps throughout the layers, leading to **computational benefits**. In this subsection we unveil some **theoretical benefits** of this common practice, which we coin *spatial-stride*.

Following Figure 5, recall that $\mathbf{D}_i$ is a stride convolutional dictionary that skips m_{i-1} shifts at a time, which correspond to the number of channels in $\mathbf{\Gamma}_{i-1}$. Translating the spatial-stride to our language, the above mentioned works do not consider all spatial shifts of the filters in $\mathbf{D}_i$. Instead, a stride of $m_{i-1}s_{i-1}$ is employed, where m_{i-1} corresponds to the *channel-stride*, while s_{i-1} is due to the *spatial-stride*. The addition of the latter implies that instead of assuming that the i-th sparse vector satisfies $\mathbf{\Gamma}_{i-1} = \mathbf{D}_i\mathbf{\Gamma}_i$, we have that $\mathbf{Q}_{i-1}\mathbf{\Gamma}_{i-1} = \mathbf{D}_i\mathbf{Q}_i^T\mathbf{Q}_i\mathbf{\Gamma}_i$. We denote $\mathbf{Q}_i^T \in \mathbb{R}^{Nm_i \times Nm_i/s_{i-1}}$ as a columns' selection operator that chooses the atoms from $\mathbf{D}_i$ that align with the spatial-stride. The coefficients corresponding to these atoms are extracted from $\mathbf{\Gamma}_i$ (resulting in its subsampled version) via the $\mathbf{Q}_i$ matrix. In light of the above discussion, we modify the DCP$_{\boldsymbol{\lambda}}$ problem, as defined in Definition 1, into the following

$$\begin{aligned}
\text{find} \quad \{\mathbf{\Gamma}_i\}_{i=1}^K \qquad \text{s.t.} \qquad \mathbf{X} &= \mathbf{D}_1\mathbf{Q}_1^T\mathbf{Q}_1\mathbf{\Gamma}_1, && \|\mathbf{Q}_1\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{S}} \leq \lambda_1 \\
\mathbf{Q}_1\mathbf{\Gamma}_1 &= \mathbf{D}_2\mathbf{Q}_2^T\mathbf{Q}_2\mathbf{\Gamma}_2, && \|\mathbf{Q}_2\mathbf{\Gamma}_2\|_{0,\infty}^{\mathbf{S}} \leq \lambda_2 \\
&\vdots && \vdots \\
\mathbf{Q}_{K-1}\mathbf{\Gamma}_{K-1} &= \mathbf{D}_K\mathbf{Q}_K^T\mathbf{Q}_K\mathbf{\Gamma}_K, && \|\mathbf{Q}_K\mathbf{\Gamma}_K\|_{0,\infty}^{\mathbf{S}} \leq \lambda_K.
\end{aligned}$$

Note that while the original $\|\mathbf{\Gamma}_i\|_{0,\infty}^{\mathbf{S}}$ is equal to the maximal number of non-zeros in a stripe of length $(2n_{i-1}-1)m_i$ in $\mathbf{\Gamma}_i$, the term $\|\mathbf{Q}_i\mathbf{\Gamma}_i\|_{0,\infty}^{\mathbf{S}}$ counts the same quantity but for stripes of length $(2\left\lceil n_{i-1}/s_{i-1}\right\rceil-1)m_i$ in $\mathbf{Q}_i\mathbf{\Gamma}_i$.

According to the study in Section 5 and 6, the theoretical advantage of the spatial-stride is twofold. First, consider the mutual coherence of the stride convolutional dictionary $\mathbf{D}_i$. Due to the locality of the filters and their restriction to certain spatial shifts, the mutual coherence of $\mathbf{D}_i\mathbf{Q}_i^T$ is expected to be lower than that of $\mathbf{D}_i$, thus leading to more non-zeros allowed per stripe. Second, the length of a stripe in $\mathbf{Q}_i\mathbf{\Gamma}_i$ is equal to $(2\left\lceil n_{i-1}/s_{i-1}\right\rceil-1)m_i$, while that of $\mathbf{\Gamma}_i$ is $(2n_{i-1}-1)m_i$. As such, our analysis allows a larger number of non-zeros per a smaller-sized stripe. From another perspective, notice that imposing a spatial-stride on the dictionary $\mathbf{D}_i$ is equivalent to forcing a portion of the entries in $\mathbf{\Gamma}_i$ to be zero. As such, the spatial-stride encourages sparser solutions.

8. Experiments: The Generator Behind the CNN

Consider the following question: can we synthesize signals obeying the ML-CSC model? Throughout this work we have posited that the answer to this question is positive; we have assumed the existence of a set of signals $\mathbf{X}$, which satisfy $\forall i\ \mathbf{\Gamma}_i=\mathbf{D}_{i+1}\mathbf{\Gamma}_{i+1}$ where $\{\mathbf{\Gamma}_i\}_{i=1}^K$ are all $\ell_{0,\infty}$ bounded. However, a natural question arises as to whether we can give a simple example of set of dictionaries $\{\mathbf{D}_i\}_{i=1}^K$ and their corresponding signals $\mathbf{X}$ that indeed satisfy our model assumptions.

A naïve attempt would be to choose an arbitrary set of dictionaries, $\{\mathbf{D}_i\}_{i=1}^K$, and a random deepest representation, $\mathbf{\Gamma}_K$, and compute the remaining sparse vectors (and the signal itself) using the set of relations $\mathbf{\Gamma}_i=\mathbf{D}_{i+1}\mathbf{\Gamma}_{i+1}$. However, without further restrictions, this would lead to a set of representations $\{\mathbf{\Gamma}_i\}_{i=1}^K$ with growing $\ell_{0,\infty}$ norm as we propagate towards $\mathbf{\Gamma}_0$. A somewhat better approach would be to impose sparsity on the dictionaries involved, as suggested in Section 7.1, thus leading to sparser representations. However, besides the obvious drawback of forcing a limiting structure on the dictionaries, as can be seen in Equation (11), in the worst case this would also lead to growth in the density of the representations, even if it is more controlled. The spatial-stride – at first glance unrelated to this discussion – is another solution that addresses the same problem. In particular, in Section 7.2 this idea was shown to encourage sparser vectors by forcing zeros in a regular pattern in the set of representations $\{\mathbf{\Gamma}_i\}_{i=1}^K$.

In this section we combine the above notions in order to achieve our goal – generate a set of signals that will satisfy the ML-CSC assumptions. These will then serve as a playground for several experiments, which will compare both theoretically and practically the different pursuits presented in this paper.

8.1 Designing the Dictionaries

We commence by describing the design of the dictionaries, and in the next subsection continue to the actual generation of the signals. In our experiments, the signal is one dimensional and therefore $m_0=1$. Moreover, for simplicity, the dictionary in every layer contains a single atom with its shifts and thus $m_i=1\ \forall 1\leq i\leq K$. We should note that the choice of a single atom simplifies the involved pursuit problem, but as we will see, even

in such a case the suggested layered pursuits (including the forward pass) may fail. This is because the mutual coherence and the amount of non-zeros are still non-trivial.

In the first layer we choose this filter to be the analytically defined discrete Meyer Wavelet of length $n_0 = 29$. In order to obtain sparser representations and improve the coherence of the global dictionary $\mathbf{D}_1$, we employ a stride of $s_0 = 6$, resulting in $\mu(\mathbf{D}_1) = 2.44 \times 10^{-4}$. As a consequence of our choice of $\mathbf{D}_1$, the signals resulting from our model are a superposition of shifted versions of discrete Meyer Wavelets, multiplied by different coefficients.

Recall that in the context of the layered thresholding algorithm, our theoretical study has shown that the stability of the pursuit depends on the ratio $|\Gamma_i^{\min}|/|\Gamma_i^{\max}|$. As such, in addition to requiring $\mu(\mathbf{D}_i)$ to be small, we would also like the ratio $|\Gamma_i^{\min}|/|\Gamma_i^{\max}|$ to be as close as possible to one. Since the sparse vectors satisfy $\mathbf{\Gamma}_i = \mathbf{D}_{i+1}\mathbf{\Gamma}_{i+1}$ $\forall 1 \leq i \leq K-1$, one can control this ratio by forcing the entries in the dictionaries $\{\mathbf{D}_i\}_{i=2}^K$ to be[9] discrete[10]. Following this observation, and motivated by the benefits of a sparse dictionary, we generate a filter of length 20 with 7 non-zero entries belonging to the set $\{-8, -7, ..., 7, 8\}$ (these are the entries before the atom is normalized to a unit ℓ_2 norm). In practice, this is done by sampling random vectors satisfying these constraints and choosing one resulting in a good mutual coherence. For simplicity, all $\{\mathbf{D}_i\}_{i=2}^K$ are created from the very same local atom, i.e. $n_i = 20$ $\forall 1 \leq i \leq K-1$. Moreover, in all the dictionaries this atom is shifted by a stride of $s_i = 6$, leading to $\mu(\mathbf{D}_i) = 4.33 \times 10^{-3}$. Note that in the above description the specific number of layers K was purposely omitted, as this number will vary in the following experiments.

8.2 Noiseless Experiments

We now move to the task of sampling a signal when the number of layers is $K = 3$. First, we draw a random $\mathbf{\Gamma}_3$ of length 100 with an ℓ_0 norm in the range $[20, 66]$ and set each non-zero coefficient in it to ± 1, with equal probability. Given the dictionaries and the sampled sparse vector $\mathbf{\Gamma}_3$, we then compute the representations $\mathbf{\Gamma}_2$, $\mathbf{\Gamma}_1$ and the signal $\mathbf{X}$, which are of length 600, 3,600 and $N = 21,600$, respectively. The obtained sparse vectors satisfy $\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{s}} = 8$, $5 \leq \|\mathbf{\Gamma}_2\|_{0,\infty}^{\mathbf{s}} \leq 6$ and $3 \leq \|\mathbf{\Gamma}_3\|_{0,\infty}^{\mathbf{s}} \leq 7$.

Given the signals, we attempt to retrieve their underlying representations using the layered pursuits presented in this work. Recall that our analysis in Section 5 and 6 indicates that the layered hard thresholding is superior to its soft counterpart, which is equivalent to the forward pass, and that the layered BP is even better than both of these algorithms. We now turn to asserting this claim empirically. While doing so, we aim to study the gap between the theoretical guarantees presented throughout our paper and the empirical performance obtained in practice.

9. Note that we do not force the entries in $\mathbf{D}_1$ to be discrete since $\mathbf{X} = \mathbf{D}_1\mathbf{\Gamma}_1$ and the ratio of the entries in $\mathbf{X}$ is of no significance to the success of the layered thresholding algorithms.
10. In our experiments, the non-zero entries in the deepest representation $\mathbf{\Gamma}_K$ are chosen to be ± 1. As such, the sparse vector $\mathbf{\Gamma}_{K-1} = \mathbf{D}_K\mathbf{\Gamma}_K$ is a superposition of filters (or their negative) taken from the dictionary $\mathbf{D}_K$. If the entries in $\mathbf{D}_K$ are non-discrete then the summation of two filters can result in extremely small values in $\mathbf{\Gamma}_{K-1}$, which in turn would lead to a very small $|\mathbf{\Gamma}_{K-1}^{\min}|$ and a bad ratio. On the other hand, if the atoms are chosen to be discrete, this would not happen since the entries would simply cancel each other.

For every signal $\mathbf{X}$ (termed *realization* below), we employ the layered hard thresholding algorithm. The thresholds are set to be the ones presented in Theorem 8, since the $\ell_{0,\infty}$ norms of the representations of each $\mathbf{X}$ satisfy the assumptions of this theorem. Given the estimated sparse vectors $\{\hat{\mathbf{\Gamma}}_i\}_{i=1}^3$, we then compute the errors $\|\hat{\mathbf{\Gamma}}_i - \mathbf{\Gamma}_i\|_{2,\infty}^{\mathbf{P}}$ and compare these to the theoretical bounds. While doing so, we also verify that the correct support is indeed retrieved, as our theorem guarantees. Next, the same process is repeated for the layered soft thresholding algorithm, with the exception that the thresholds and the bound on the distance are computed according to Theorem 10. We note that the assumptions of this hold as well for every signal $\mathbf{X}$. The results for both algorithms are depicted in Figure 6 in terms of the local signal to noise ratio (SNR), defined as $20\log 10\Big(\frac{\|\mathbf{\Gamma}_i\|_{2,\infty}^{\mathbf{P}}}{\|\hat{\mathbf{\Gamma}}_i-\mathbf{\Gamma}_i\|_{2,\infty}^{\mathbf{P}}}\Big)$. Due to the locality of the analysis, we choose to deviate from the classical definition of the (global) SNR, given by $20\log 10\left(\frac{\|\mathbf{\Gamma}_i\|_2}{\|\hat{\mathbf{\Gamma}}_i-\mathbf{\Gamma}_i\|_2}\right)$.

Several remarks are due here. First and foremost, the theoretical bounds indeed hold, since the blue points are above their corresponding green ones and the correct supports are always recovered. Second, our analysis predicts that the distance between the estimated sparse representation, $\hat{\mathbf{\Gamma}}_i$, and the true ones, $\mathbf{\Gamma}_i$, should increase with the layer. This is evident by the decrease in the values of the green points with the layers. The empirical results presented here (blue dots) corroborate this prognosis, as the error in both algorithms is lowest in the first layer and highest in the last[11]. Third, our analysis suggests that the layered hard thresholding algorithm should be superior to its soft counterpart. Once again, this can be deduced from the figure by comparing the values of the green points in both of the algorithms. The empirical results presented in Figure 6 confirm this behavior, as can be clearly seen by comparing the errors (blue points) obtained by both algorithms in the i-th layer. One should note that the performance gap exhibited here is due to the constant β_i being subtracted from every entry in the soft thresholding algorithm.

The implications of the above discussion might be troubling in the context of CNN, as what this experiment shows is a deterioration of the **empirical** SNR throughout the layers of the network. Is this truly the behavior of CNN? Recall that in practice the biases of the different layers (thresholds) are learned in order to achieve the best possible performance in solving a certain task. As such, it might be possible that the decline in SNR presented here is alleviated when better thresholds are employed in lieu of the theoretical ones used thus far. We demonstrate this by running the layered soft thresholding algorithm with an oracle parameter, chosen to be the minimal threshold that leads to $\|\mathbf{\Gamma}_i\|_0$ non-zeros being chosen in the estimated sparse representation $\hat{\mathbf{\Gamma}}_i$. The results for this are presented in Figure 6 and colored in red. Indeed, we observe that this better choice of parameters improves the empirical performance of the layered soft thresholding algorithm and leads to a slower decline in SNR. Still, the performance of the layered soft thresholding is inferior to that of its hard variant[12], as can be seen by comparing the red points with the blue ones in the subplots below.

11. Interestingly, the error in the layered hard thresholding algorithm is approximately equal in the second and third layers.
12. Note that in the layered hard thresholding, as long as the correct support is chosen, the threshold does not affect the error and as such the oracle version for it is meaningless.

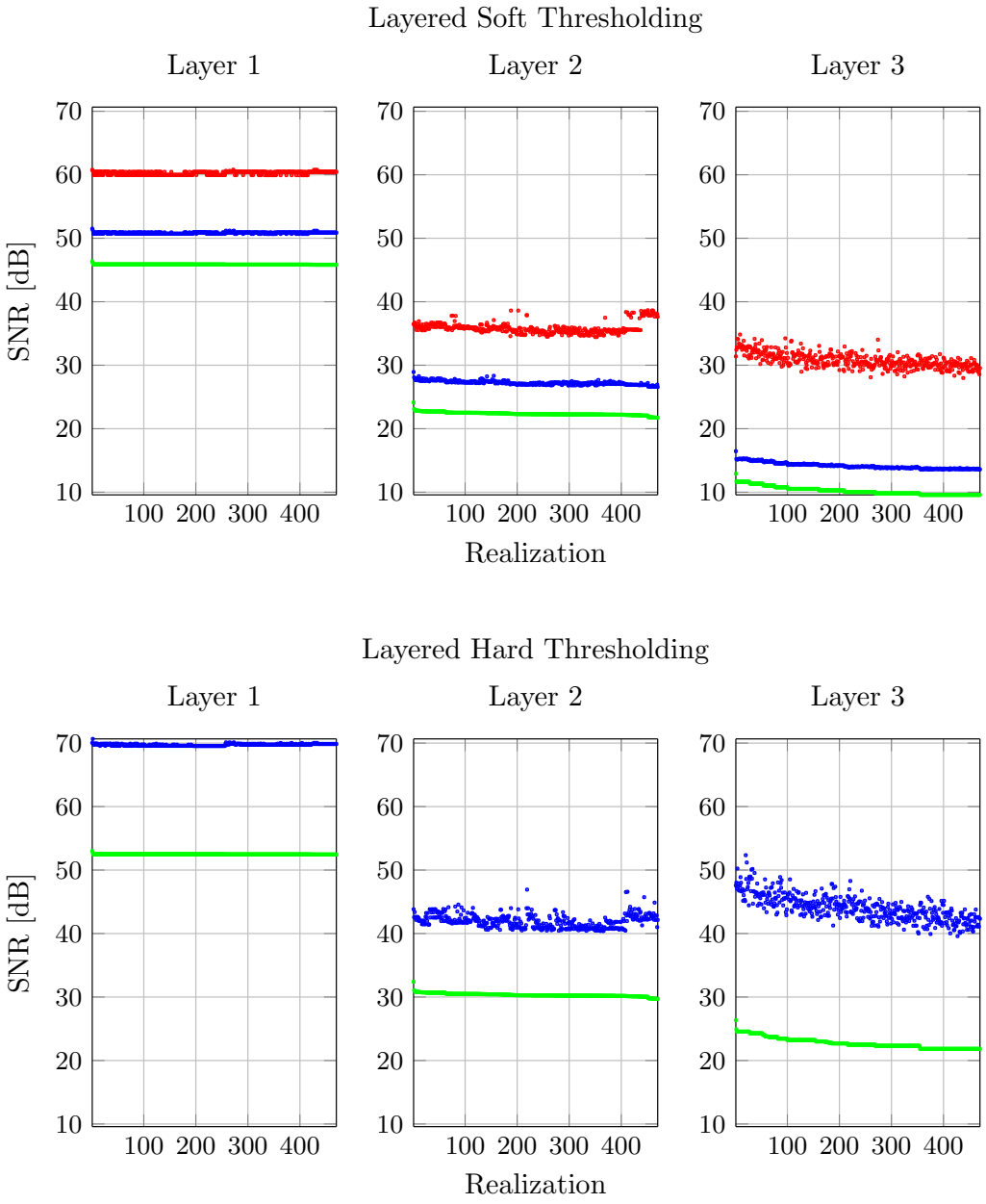


Figure 6: Comparison between the performance of the layered thresholding algorithms in a noiseless setting. All the signals presented here satisfy the assumptions of Theorem 8 and 10 and indeed the correct support of all the representations are recovered. The horizontal axis plots the realization number, while the vertical axis shows the SNR (the higher the better). Blue: layered thresholding algorithm with the theoretically justifiable thresholds. Green: the theoretical bound on the error. Red: layered soft thresholding algorithm with oracle thresholds. Note that the points are sorted according to the theoretical bound.

Next, we proceed our experiments by running the layered BP algorithm, as defined in Section 6, on the same set of signals. Recall that one of the prime motivations for proposing this algorithm was its ability to retrieve the exact underlying representations, as justified theoretically in Theorem 11. In our experiments, we validate this claim by checking that its conditions hold for each signal and that the underlying representations are indeed retrieved. We omit showing a plot for this and comparing it to the layered thresholding algorithms since the errors obtained are simply zeros.

8.3 Noisy Experiments

Having established the stability of our proposed algorithms in a perfect scenario, where $\epsilon_0 = 0$, we now turn to a noisy setting. Naturally, the estimation task becomes now even more challenging – not only does the SNR drop with each layer, as demonstrated previously, but also the input SNR is no longer infinity. In order to facilitate the success of our algorithms, in this section we demonstrate the empirical performance and theoretical bounds on $K = 2$ layers and a small noise level.

Similar to the previous subsection, we begin by sampling a signal $\mathbf{X}$. To this end, we draw a random $\mathbf{\Gamma}_2$ of length 100 where $20 \leq \|\mathbf{\Gamma}_2\|_0 \leq 66$. Each non-zero coefficient in it is then set to ± 1, with equal probability. Given the dictionaries and the sampled sparse vector $\mathbf{\Gamma}_2$, we then compute the representation $\mathbf{\Gamma}_1$ and the signal $\mathbf{X}$, which are of length 600 and $3,600$, respectively. The $\ell_{0,\infty}$ norm of the obtained sparse vectors satisfies $7 \leq \|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{s}} \leq 8$ and $3 \leq \|\mathbf{\Gamma}_2\|_{0,\infty}^{\mathbf{s}} \leq 7$.

Next, we contaminate each signal $\mathbf{X}$ with a zero-mean white additive Gaussian noise $\mathbf{E}$, creating a signal $\mathbf{Y} = \mathbf{X} + \mathbf{E}$. The average SNR of the obtained noisy signals is 68.53 dB. These are then fed into the layered pursuits, resulting in a set of estimated sparse representations, $\hat{\mathbf{\Gamma}}_i$. We note that the $\ell_{0,\infty}$ norms of the representations of each $\mathbf{X}$ satisfy the assumptions of Theorem 8, 10 and 12. As such, the parameters for every algorithm are chosen according to our theoretical study. For each estimated representation we compute the error $\|\hat{\mathbf{\Gamma}}_i - \mathbf{\Gamma}_i\|_{2,\infty}^{\mathbf{p}}$ and its corresponding theoretical bound obtained from the aforementioned theorems. Since the underlying representations satisfy the assumptions of the stability theorem for the layered thresholding algorithms, for each signal we verify that indeed the correct support is found. As for the layered BP, our stability analysis guarantees that the support retrieved should be contained in the true one and coefficients that are large enough in $\mathbf{\Gamma}_i$ should be retrieved. In practice, the layered BP always finds the full support.

We present the obtained results in terms of the local SNR in Figure 7, showing the stability of the different algorithms that is in accordance with our theoretical bounds. Similar to the noiseless experiment, we observe that for all the algorithms the error increases both theoretically (green points) and empirically (blue points) with the layer depth. As previously discussed, a performance gap exists between the soft and hard layered thresholding algorithms. To mitigate this, we run the layered soft thresholding with an oracle parameter and compare the obtained errors (red points) to those of the other algorithms. The results, depicted in the same figure, show a clear improvement in the performance.

Interestingly, although theoretically superior, the layered BP leads to similar performance to that of the layered soft thresholding and worse performance than that of the layered hard thresholding (when comparing the blue points). We attribute this phenomenon

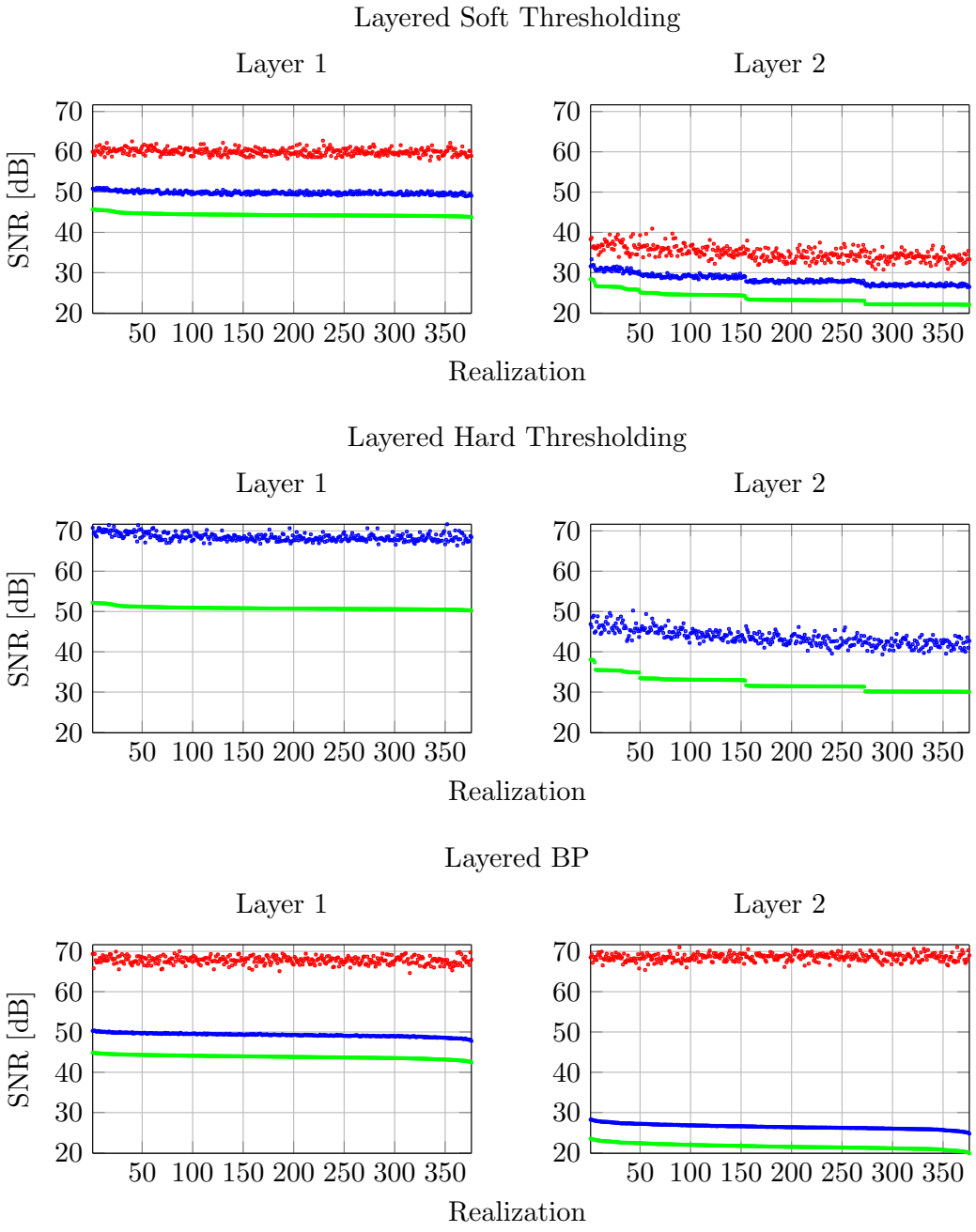


Figure 7: Comparison between the performance of the layered pursuit algorithms in a noisy setting. All the signals presented here satisfy the assumptions of Theorem 8, 10 and 12 and the correct support of all the representations are indeed recovered. The horizontal axis plots the realization number, while the vertical axis shows the SNR (the higher the better). Blue: layered pursuits with the theoretically justifiable thresholds. Green: the theoretical bound on the error. Red: layered soft thresholding with oracle thresholds and layered BP with hand-picked parameters. Note that the points are sorted according to the theoretical bound.

to the suboptimal choice of the parameter ξ_i, which was chosen thus far according to our theoretical analysis. To validate this suspicion, we run the layered BP with hand-picked ξ_i and plot the obtained SNR in red in Figure 7. Not only are the correct supports retrieved for all the signals, but we can also see a clear improvement in terms of the SNR. In the first layer, the layered BP outperforms the layered soft thresholding and leads to similar results to those of the layered hard thresholding, while in the second, the layered BP significantly outperforms both of the other pursuit algorithms.

Thus far, our experiments focused on a setting where the ratio of the coefficients in $\mathbf{\Gamma}_i$ is reasonable. One should note, however, that the superiority of the layered BP becomes conspicuous once this ratio is spoiled. In this case, the layered thresholding algorithms will fail, while the layered BP will still succeed. To illustrate this, we create a signal using the dictionaries delineated in subsection 8.1, where the number of layers is $K = 5$. We first draw a random $\mathbf{\Gamma}_5$ of length 100 where its ℓ_0 norm is in the range $20 \leq \|\mathbf{\Gamma}_5\|_0 \leq 66$, and then set the non-zero coefficients in $\mathbf{\Gamma}_5$, similar to how it was done in the previous experiments. Given the dictionaries and the sampled sparse vector $\mathbf{\Gamma}_5$, we compute the representation $\{\mathbf{\Gamma}_i\}_{i=1}^4$ and the signal $\mathbf{X}$, which is of length $N = 777,600$. The $\ell_{0,\infty}$ norms of the obtained sparse vectors are $\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{S}} = 8$, $\|\mathbf{\Gamma}_2\|_{0,\infty}^{\mathbf{S}} = 6$, $\|\mathbf{\Gamma}_3\|_{0,\infty}^{\mathbf{S}} = 6$, $\|\mathbf{\Gamma}_4\|_{0,\infty}^{\mathbf{S}} = 6$ and $4 \leq \|\mathbf{\Gamma}_5\|_{0,\infty}^{\mathbf{S}} \leq 7$. Besides the depth of the network, the main difference between this experiment and the previous ones is the coefficient ratio. While the ratio of the deepest representation $\mathbf{\Gamma}_5$ is equal to 1, due to the coefficients in it being equal to ± 1, the ratio of $\mathbf{\Gamma}_1$ is equal to 2.44×10^{-4}. As a consequence, the theoretical results we have presented for the layered thresholding algorithms do not hold, while those of the layered BP still do.

Next, each signal $\mathbf{X}$ is contaminated with a zero-mean white additive Gaussian noise $\mathbf{E}$, resulting in a noisy signal $\mathbf{Y} = \mathbf{X} + \mathbf{E}$. The average SNR of the noisy signals obtained is 124.43 dB. Note that this is a weak noise, chosen due to the deterioration of the SNR throughout the layers (one that is worsened when the theoretical parameters are employed). The signals are then fed into the layered BP algorithm, resulting in a set of estimated sparse representations, $\hat{\mathbf{\Gamma}}_i$. The parameters ξ_i employed are the theoretically justified ones, $\xi_i = 4\epsilon_{i-1}$. We should note that in our experiments we attempted to run the layered thresholding algorithms, however, as our theory predicts these failed in recovering the correct supports.

Given the estimated representations, we compute the errors $\|\hat{\mathbf{\Gamma}}_i - \mathbf{\Gamma}_i\|_{2,\infty}^{\mathbf{P}}$ and compare these to their corresponding theoretical bounds, obtained from Theorem 12. In addition, we verify that the retrieved supports are contained in the true one, as the theorem guarantees. In practice, we obtain that the layered BP always finds the full support. The obtained results are depicted in Figure 8 in terms of the local SNR. For comparison, we run the layered BP with hand-picked ξ_i and present the obtained results in the same figure. We conclude that the layered BP remains stable despite the poor coefficient ratio, unlike the layered thresholding algorithms. Moreover, tuning the ξ_i results in a much better performance, similar to what we have seen in the previous experiment.

At this point, one might ponder as to whether the hurdle of poor coefficient ratio is one that the layered soft thresholding (forward pass) can not overcome. We believe that several ideas currently used in CNN, such as Batch Normalization (Ioffe and Szegedy, 2015) or Local Response Normalization (Krizhevsky et al., 2012), are tightly connected to this problem. However, their exact relation to this issue and its theoretical analysis is a matter of future work.

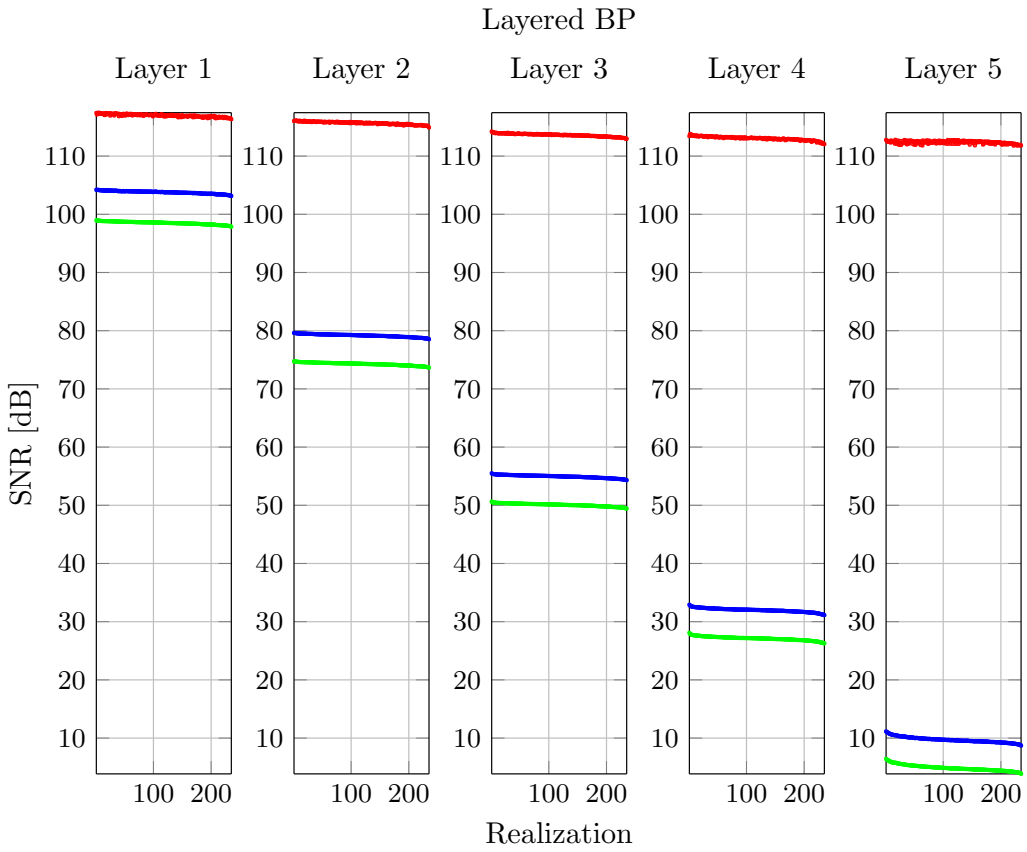


Figure 8: The performance of the layered BP algorithm in a noisy setting. All the signals presented here satisfy the assumptions of Theorem 12 and the correct support of all the representations are recovered. The horizontal axis plots the realization number, while the vertical axis shows the SNR (the higher the better). Blue: layered BP with the theoretically justifiable thresholds. Green: the theoretical bound on the error. Red: layered BP with hand-picked parameters. Note that the points are sorted according to the theoretical bound.

9. Conclusion

Definition: "A guiding question is the fundamental query that directs the search for understanding" (Traver, 1998). In this work our guiding question was who are the signals that the CNN architecture is designed for? To answer this we have defined the ML-CSC model, for which the thresholding pursuit is nothing but the forward pass of the CNN. Although nothing promises that the forward pass will lead to the original representation of a signal emerging from the ML-CSC model, we have shown this is indeed the case. Having established the relevance of our model to CNN, we then turned to its theoretical analysis. In particular, we provided guarantees for the uniqueness of the feature maps CNN aims to recover, and the stability of the problem CNN aims to solve.

Inspired by the evolution of the pursuit methods in the theory of Sparse-Land, we continued our work by proposing the layered BP algorithm. In the noiseless case, this was

theoretically shown to be capable of finding the unique solution of the deep coding problem, the existence of which has been also guaranteed; while in the noisy setting, we have proved the stability of this algorithm.

We analyzed the theoretical benefits of two popular ideas employed in the CNN community, namely the use of sparse filters and the spatial-stride. Leveraging those, we then generated signals satisfying the ML-CSC assumptions and demonstrated the performance of the pursuits presented throughout this work.

We conclude this work by presenting our ongoing research directions:

1. Through this paper we have assumed the worst – an adversary noise. Can our theoretical analysis be extended to a setting where the noise is random?

2. Thus far in tackling the deep coding problem, we have restricted ourself to existing methods, such as the forward pass of the CNN or deconvolutional networks (Zeiler et al., 2010). Can we suggest better approximations for the solution of this problem?

3. Clearly a relation exists between our proposed layered iterative thresholding algorithm and the current throne holder in the task of image recognition – residual networks (He et al., 2015). Can our theory reveal the benefits of introducing skip connections to a CNN?

4. What is the role of common tricks currently employed in CNN in the context of the ML-CSC model? These include but are not limited to, Batch Normalization (Ioffe and Szegedy, 2015), Local Response Normalization (Krizhevsky et al., 2012), Dropout (Srivastava et al., 2014) and Pooling (LeCun et al., 1990; Krizhevsky et al., 2012; Simonyan and Zisserman, 2014).

Acknowledgments

The research leading to these results has received funding from the European Research Council under European Union's Seventh Framework Programme, ERC Grant agreement no. 320649. The authors would like to thank Jeremias Sulam for the inspiring discussions and creative advice.

Appendix A. Uniqueness via the Mutual Coherence (Proof of Theorem 4)

Proof In (Papyan et al., 2016a) a solution $\mathbf{\Gamma}$ to the $\mathrm{P}_{0,\infty}$ problem, as defined in Equation (6), was shown to be unique assuming that $\|\mathbf{\Gamma}\|_{0,\infty}^{\mathbf{s}} < \frac{1}{2}\left(1+\frac{1}{\mu(\mathbf{D})}\right)$. In other words, if the true representation is sparse enough in the $\ell_{0,\infty}$ sense, no other solution is possible. Herein, we leverage this claim in order to prove the uniqueness of the $\mathrm{DCP}_{\boldsymbol{\lambda}}$ problem.

Let $\{\hat{\mathbf{\Gamma}}_i\}_{i=1}^K$ be a set of representations of the signal $\mathbf{X}$, obtained by solving the $\mathrm{DCP}_{\boldsymbol{\lambda}}$ problem. According to our assumptions, $\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{s}} < \frac{1}{2}\left(1+\frac{1}{\mu(\mathbf{D}_1)}\right)$. Moreover, since the set $\{\hat{\mathbf{\Gamma}}_i\}_{i=1}^K$ is a solution of the $\mathrm{DCP}_{\boldsymbol{\lambda}}$ problem, we also have that $\|\hat{\mathbf{\Gamma}}_1\|_{0,\infty}^{\mathbf{s}} \leq \lambda_1 < \frac{1}{2}\left(1+\frac{1}{\mu(\mathbf{D}_1)}\right)$.

As such, in light of the aforementioned uniqueness theorem, both representations are equal. Once we have concluded that $\boldsymbol{\Gamma}_1 = \hat{\boldsymbol{\Gamma}}_1$, we would also like to show that the representations $\boldsymbol{\Gamma}_2$ and $\hat{\boldsymbol{\Gamma}}_2$ are identical. Similarly, the assumptions $\|\boldsymbol{\Gamma}_2\|_{0,\infty}^{\mathbf{s}} < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D}_i)}\right)$ and $\|\hat{\boldsymbol{\Gamma}}_2\|_{0,\infty}^{\mathbf{s}} \leq \lambda_2 < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D}_2)}\right)$ guarantee that $\boldsymbol{\Gamma}_2 = \hat{\boldsymbol{\Gamma}}_2$. The same set of steps can be applied for all $1 \leq i \leq K$, leading to the fact that both sets of representations are identical. ■

Appendix B. Global Stability of the DCP$_{\boldsymbol{\lambda}}^{\boldsymbol{\mathcal{E}}}$ Problem (Proof of Theorem 5)

Proof In (Papyan et al., 2016b), for a signal $\mathbf{Y} = \mathbf{X} + \mathbf{E} = \mathbf{D}_1\boldsymbol{\Gamma}_1 + \mathbf{E}$, it was shown that if the following hold:

1. $\|\boldsymbol{\Gamma}_1\|_{0,\infty}^{\mathbf{s}} < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D}_1)}\right)$ and $\|\mathbf{E}\|_2 = \|\mathbf{Y} - \mathbf{D}_1\boldsymbol{\Gamma}_1\|_2 \leq \mathcal{E}_0$,
2. $\|\hat{\boldsymbol{\Gamma}}_1\|_{0,\infty}^{\mathbf{s}} < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D}_1)}\right)$ and $\|\mathbf{Y} - \mathbf{D}_1\hat{\boldsymbol{\Gamma}}_1\|_2 \leq \mathcal{E}_0$,

then

$$\|\boldsymbol{\Delta}_1\|_2^2 = \|\boldsymbol{\Gamma}_1 - \hat{\boldsymbol{\Gamma}}_1\|_2^2 \leq \frac{4{\mathcal{E}_0}^2}{1 - (2\|\boldsymbol{\Gamma}_1\|_{0,\infty}^{\mathbf{s}} - 1)\mu(\mathbf{D}_1)} = {\mathcal{E}_1}^2.$$

In the above, we have defined $\boldsymbol{\Delta}_1$ as the difference between the true sparse vector, $\boldsymbol{\Gamma}_1$, and the corresponding representation obtained by solving the DCP$_{\boldsymbol{\lambda}}^{\boldsymbol{\mathcal{E}}}$ problem, $\hat{\boldsymbol{\Gamma}}_1$. In item 2 we have used the fact that the solution for the DCP$_{\boldsymbol{\lambda}}^{\boldsymbol{\mathcal{E}}}$ problem, $\hat{\boldsymbol{\Gamma}}_1$, must satisfy $\|\mathbf{Y} - \mathbf{D}_1\hat{\boldsymbol{\Gamma}}_1\|_2 \leq \mathcal{E}_0$ and $\|\hat{\boldsymbol{\Gamma}}_1\|_{0,\infty}^{\mathbf{s}} \leq \lambda_1$; and our assumption that $\lambda_1 < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D}_1)}\right)$. Next, notice that $\hat{\boldsymbol{\Gamma}}_1 = \boldsymbol{\Gamma}_1 + \boldsymbol{\Delta}_1 = \mathbf{D}_2\boldsymbol{\Gamma}_2 + \boldsymbol{\Delta}_1$, and that the following hold:

1. $\|\boldsymbol{\Gamma}_2\|_{0,\infty}^{\mathbf{s}} < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D}_2)}\right)$ and $\|\boldsymbol{\Delta}_1\|_2 = \|\hat{\boldsymbol{\Gamma}}_1 - \mathbf{D}_2\boldsymbol{\Gamma}_2\|_2 \leq \mathcal{E}_1$,
2. $\|\hat{\boldsymbol{\Gamma}}_2\|_{0,\infty}^{\mathbf{s}} < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D}_2)}\right)$ and $\|\hat{\boldsymbol{\Gamma}}_1 - \mathbf{D}_2\hat{\boldsymbol{\Gamma}}_2\|_2 \leq \mathcal{E}_1$.

The second item relies on the fact that both $\hat{\boldsymbol{\Gamma}}_1$ and $\hat{\boldsymbol{\Gamma}}_2$, obtained by solving the DCP$_{\boldsymbol{\lambda}}^{\boldsymbol{\mathcal{E}}}$ problem, must satisfy $\|\hat{\boldsymbol{\Gamma}}_1 - \mathbf{D}_2\hat{\boldsymbol{\Gamma}}_2\|_2 \leq \mathcal{E}_1$ and $\|\hat{\boldsymbol{\Gamma}}_2\|_{0,\infty}^{\mathbf{s}} \leq \lambda_2$. In addition, the second expression uses the assumption that $\lambda_2 < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D}_2)}\right)$. Employing once again the aforementioned stability theorem, we are guaranteed that

$$\|\boldsymbol{\Delta}_2\|_2^2 = \|\boldsymbol{\Gamma}_2 - \hat{\boldsymbol{\Gamma}}_2\|_2^2 \leq \frac{4{\mathcal{E}_1}^2}{1 - (2\|\boldsymbol{\Gamma}_2\|_{0,\infty}^{\mathbf{s}} - 1)\mu(\mathbf{D}_2)} = {\mathcal{E}_2}^2.$$

Using the same set of steps presented above, we conclude that

$$\forall\ 1 \leq i \leq K \qquad \|\boldsymbol{\Gamma}_i - \hat{\boldsymbol{\Gamma}}_i\|_2^2 \leq {\mathcal{E}_i}^2,$$

as claimed. ■

Appendix C.
Stable Recovery of Hard Thresholding in the Presence of Noise (Proof of Lemma 7)

Proof Denote by $\mathcal{T}_1$ the support of $\mathbf{\Gamma}_1$. Denote further the i-th atom from $\mathbf{D}_1$ by $\mathbf{d}_{1,i}$. The success of the hard thresholding algorithm with threshold β_1 in recovering the correct support is guaranteed if the following holds

$$\min_{i\in\mathcal{T}_1}\left|\mathbf{d}_{1,i}^T\mathbf{Y}\right| > \beta_1 > \max_{j\notin\mathcal{T}_1}\left|\mathbf{d}_{1,j}^T\mathbf{Y}\right|.$$

Using the same set of steps as those used in proving Theorem 4 in (Papyan et al., 2016b), we can lower bound the left-hand-side by

$$\min_{i\in\mathcal{T}_1}\left|\mathbf{d}_{1,i}^T\mathbf{Y}\right| \geq |\mathbf{\Gamma}_1^{\min}| - (\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{s}}-1)\mu(\mathbf{D}_1)|\mathbf{\Gamma}_1^{\max}| - \epsilon_0$$

and upper bound the right-hand-side via

$$\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{s}}\mu(\mathbf{D}_1)|\mathbf{\Gamma}_1^{\max}| + \epsilon_0 \geq \max_{j\notin\mathcal{T}_1}\left|\mathbf{d}_{1,j}^T\mathbf{Y}\right|.$$

Next, by requiring

$$\begin{aligned}\min_{i\in\mathcal{T}_1}\left|\mathbf{d}_{1,i}^T\mathbf{Y}\right| \geq& |\mathbf{\Gamma}_1^{\min}| - (\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{s}}-1)\mu(\mathbf{D}_1)|\mathbf{\Gamma}_1^{\max}| - \epsilon_0 \\ >& \beta_1 \\ >& \|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{s}}\mu(\mathbf{D}_1)|\mathbf{\Gamma}_1^{\max}| + \epsilon_0 \\ \geq& \max_{j\notin\mathcal{T}_1}\left|\mathbf{d}_{1,j}^T\mathbf{Y}\right|,\end{aligned} \tag{12}$$

we ensure the success of the thresholding algorithm. This condition can be equally written as

$$\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{s}} < \frac{1}{2}\left(1+\frac{1}{\mu(\mathbf{D}_1)}\frac{|\Gamma_1^{\min}|}{|\Gamma_1^{\max}|}\right) - \frac{1}{\mu(\mathbf{D}_1)}\frac{\epsilon_0}{|\Gamma_1^{\max}|}.$$

Equation (12) also implies that the threshold β_1 that should be employed must satisfy

$$|\mathbf{\Gamma}_1^{\min}| - (\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{s}}-1)\mu(\mathbf{D}_1)|\mathbf{\Gamma}_1^{\max}| - \epsilon_0 > \beta_1 > \|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{s}}\mu(\mathbf{D}_1)|\mathbf{\Gamma}_1^{\max}| + \epsilon_0. \tag{13}$$

Thus far, we have considered the successful recovery of the support of $\mathbf{\Gamma}_1$. Next, assuming this correct support was recovered, we shall dwell on the deviation of the thresholding result, $\hat{\mathbf{\Gamma}}_1$, from the true $\mathbf{\Gamma}_1$. Denote by $\mathbf{\Gamma}_{1,\mathcal{T}_1}$ and $\hat{\mathbf{\Gamma}}_{1,\mathcal{T}_1}$ the vectors $\mathbf{\Gamma}_1$ and $\hat{\mathbf{\Gamma}}_1$ restricted to the support $\mathcal{T}_1$, respectively. We have that

$$\begin{aligned}\|\mathbf{\Gamma}_1 - \hat{\mathbf{\Gamma}}_1\|_\infty &= \|\mathbf{\Gamma}_{1,\mathcal{T}_1} - \hat{\mathbf{\Gamma}}_{1,\mathcal{T}_1}\|_\infty \\ &= \left\|\left(\mathbf{D}_{1,\mathcal{T}_1}^T\mathbf{D}_{1,\mathcal{T}_1}\right)^{-1}\mathbf{D}_{1,\mathcal{T}_1}^T\mathbf{X} - \mathbf{D}_{1,\mathcal{T}_1}^T\mathbf{Y}\right\|_\infty \\ &= \left\|\left(\left(\mathbf{D}_{1,\mathcal{T}_1}^T\mathbf{D}_{1,\mathcal{T}_1}\right)^{-1}\mathbf{D}_{1,\mathcal{T}_1}^T - \mathbf{D}_{1,\mathcal{T}_1}^T\right)\mathbf{X} - \mathbf{D}_{1,\mathcal{T}_1}^T(\mathbf{Y}-\mathbf{X})\right\|_\infty,\end{aligned}$$

where the Gram $\mathbf{D}_{1,\mathcal{T}_1}^T\mathbf{D}_{1,\mathcal{T}_1}$ is invertible according to Lemma 1 in (Papyan et al., 2016a). Using the triangle inequality of the ℓ_∞ norm and the relation $\mathbf{X}=\mathbf{D}_{1,\mathcal{T}_1}\mathbf{\Gamma}_{1,\mathcal{T}_1}$, we obtain

$$\begin{aligned}\|\mathbf{\Gamma}_1-\hat{\mathbf{\Gamma}}_1\|_\infty &\leq \left\|\left(\left(\mathbf{D}_{1,\mathcal{T}_1}^T\mathbf{D}_{1,\mathcal{T}_1}\right)^{-1}\mathbf{D}_{1,\mathcal{T}_1}^T-\mathbf{D}_{1,\mathcal{T}_1}^T\right)\mathbf{D}_{1,\mathcal{T}_1}\mathbf{\Gamma}_{1,\mathcal{T}_1}\right\|_\infty+\left\|\mathbf{D}_{1,\mathcal{T}_1}^T(\mathbf{Y}-\mathbf{X})\right\|_\infty \\ &= \left\|\left(\mathbf{I}-\mathbf{D}_{1,\mathcal{T}_1}^T\mathbf{D}_{1,\mathcal{T}_1}\right)\mathbf{\Gamma}_{1,\mathcal{T}_1}\right\|_\infty+\left\|\mathbf{D}_{1,\mathcal{T}_1}^T(\mathbf{Y}-\mathbf{X})\right\|_\infty,\end{aligned}$$

where $\mathbf{I}$ is an identity matrix. Relying on the definition of the induced ℓ_∞ norm, the above is equal to

$$\|\mathbf{\Gamma}_1-\hat{\mathbf{\Gamma}}_1\|_\infty \leq \left\|\mathbf{I}-\mathbf{D}_{1,\mathcal{T}_1}^T\mathbf{D}_{1,\mathcal{T}_1}\right\|_\infty\cdot\|\mathbf{\Gamma}_{1,\mathcal{T}_1}\|_\infty+\left\|\mathbf{D}_{1,\mathcal{T}_1}^T(\mathbf{Y}-\mathbf{X})\right\|_\infty. \tag{14}$$

In what follows, we shall upper bound both of the expressions in the right hand side of the inequality.

Beginning with the first term in the above inequality, $\left\|\mathbf{I}-\mathbf{D}_{1,\mathcal{T}_1}^T\mathbf{D}_{1,\mathcal{T}_1}\right\|_\infty$, recall that the induced infinity norm of a matrix is equal to its maximum absolute row sum. The diagonal entries of $\mathbf{I}-\mathbf{D}_{1,\mathcal{T}_1}^T\mathbf{D}_{1,\mathcal{T}_1}$ are equal to zero, due to the normalization of the atoms, while the off diagonal entries can be bounded by relying on the locality of the atoms and the definition of the $\ell_{0,\infty}$ norm. As such, each row has at most $\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{S}}-1$ non-zeros, where each is bounded by $\mu(\mathbf{D}_1)$ based on the definition of the mutual coherence. We conclude that the maximum absolute row sum can be bounded by

$$\|\mathbf{D}_{1,\mathcal{T}_1}^T\mathbf{D}_{1,\mathcal{T}_1}-\mathbf{I}\|_\infty \leq (\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{S}}-1)\mu(\mathbf{D}_1). \tag{15}$$

Next, moving to the second expression, define $\mathbf{R}_{1,i}\in\mathbb{R}^{n_0\times N}$ to be the operator that extracts a filter of length n_0 from $\mathbf{d}_{1,i}$. Consequently, the operator $\mathbf{R}_{1,i}^T$ pads a local filter of length n_0 with zeros, resulting in a global atom of length N. Notice that, due to the locality of the atoms $\mathbf{R}_{1,i}^T\mathbf{R}_{1,i}\mathbf{d}_{1,i}=\mathbf{d}_{1,i}$. Using this together with the Cauchy-Schwarz inequality, the normalization of the atoms, and the local bound on the error $\|\mathbf{Y}-\mathbf{X}\|_{2,\infty}^{\mathbf{P}}\leq\epsilon_0$, we have that

$$\begin{aligned}\left\|\mathbf{D}_{1,\mathcal{T}_1}^T(\mathbf{Y}-\mathbf{X})\right\|_\infty &= \max_{i\in\mathcal{T}_1}\left|\mathbf{d}_{1,i}^T(\mathbf{Y}-\mathbf{X})\right| \\ &= \max_{i\in\mathcal{T}_1}\left|\left(\mathbf{R}_{1,i}\mathbf{d}_{1,i}\right)^T\mathbf{R}_{1,i}(\mathbf{Y}-\mathbf{X})\right| \\ &\leq \max_{i\in\mathcal{T}_1}\|\mathbf{R}_{1,i}\mathbf{d}_{1,i}\|_2\cdot\|\mathbf{R}_{1,i}(\mathbf{Y}-\mathbf{X})\|_2 \\ &\leq 1\cdot\|\mathbf{Y}-\mathbf{X}\|_{2,\infty}^{\mathbf{P}} \\ &\leq \epsilon_0.\end{aligned} \tag{16}$$

In the second to last inequality we have used Definition 6, denoting the maximal ℓ_2 norm of a *patch* extracted from $\mathbf{Y}-\mathbf{X}$ by $\|\mathbf{Y}-\mathbf{X}\|_{2,\infty}^{\mathbf{P}}$. Plugging (15) and (16) into Equation (14), and using the fact that $\|\mathbf{\Gamma}_{1,\mathcal{T}_1}\|_\infty=|\mathbf{\Gamma}_1^{\max}|$, we obtain that

$$\|\mathbf{\Gamma}_1-\hat{\mathbf{\Gamma}}_1\|_\infty \leq (\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{S}}-1)\mu(\mathbf{D}_1)|\mathbf{\Gamma}_1^{\max}|+\epsilon_0. \tag{17}$$

In the remainder of this proof we will localize the above bound into one that is posed in terms of patch-errors. Note that $\|\mathbf{\Gamma}_1-\hat{\mathbf{\Gamma}}_1\|_{2,\infty}^{\mathbf{P}}$ is equal to the maximal energy of an n_1m_1-dimensional patch taken from it, where the i-th patch can be extracted using the operator

$\mathbf{P}_{1,i}$. Relying on this and the relation $\|\mathbf{V}\|_2 \leq \sqrt{\|\mathbf{V}\|_0}\,\|\mathbf{V}\|_\infty$, we have that

$$\begin{aligned}\|\mathbf{\Gamma}_1 - \hat{\mathbf{\Gamma}}_1\|_{2,\infty}^{\mathbf{P}} &= \max_i \left\|\mathbf{P}_{1,i}\left(\mathbf{\Gamma}_1 - \hat{\mathbf{\Gamma}}_1\right)\right\|_2 \\ &\leq \max_i \sqrt{\left\|\mathbf{P}_{1,i}\left(\mathbf{\Gamma}_1 - \hat{\mathbf{\Gamma}}_1\right)\right\|_0}\ \left\|\mathbf{P}_{1,i}\left(\mathbf{\Gamma}_1 - \hat{\mathbf{\Gamma}}_1\right)\right\|_\infty.\end{aligned}$$

Recalling that, based on Definition 6, $\|\mathbf{\Gamma}_1 - \hat{\mathbf{\Gamma}}_1\|_{0,\infty}^{\mathbf{P}}$ denotes the maximal number of non-zeros in a patch of length $n_1 m_1$ extracted from this vector, we obtain that

$$\begin{aligned}\|\mathbf{\Gamma}_1 - \hat{\mathbf{\Gamma}}_1\|_{2,\infty}^{\mathbf{P}} &\leq \sqrt{\|\mathbf{\Gamma}_1 - \hat{\mathbf{\Gamma}}_1\|_{0,\infty}^{\mathbf{P}}}\ \|\mathbf{\Gamma}_1 - \hat{\mathbf{\Gamma}}_1\|_\infty \\ &\leq \sqrt{\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{P}}}\ \|\mathbf{\Gamma}_1 - \hat{\mathbf{\Gamma}}_1\|_\infty.\end{aligned}$$

In the last inequality we have used the success of the first stage in recovering the correct support, resulting in $\|\mathbf{\Gamma}_1 - \hat{\mathbf{\Gamma}}_1\|_{0,\infty}^{\mathbf{P}} \leq \|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{P}}$. Plugging inequality (17) into the above equation, we conclude that

$$\|\mathbf{\Gamma}_1 - \hat{\mathbf{\Gamma}}_1\|_{2,\infty}^{\mathbf{P}} \leq \sqrt{\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{P}}}\Big(\epsilon_0 + \mu(\mathbf{D}_1)\left(\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{S}} - 1\right)|\Gamma_1^{\max}|\Big),$$

as claimed. ■

Appendix D.
Stability of the Layered Hard Thresholding in the Presence of Noise (Proof of Theorem 8)

Proof The stability of the first stage of the layered hard thresholding algorithm is obtained from Lemma 7. Denoting by $\mathbf{\Delta}_1 = \hat{\mathbf{\Gamma}}_1 - \mathbf{\Gamma}_1$, notice that $\hat{\mathbf{\Gamma}}_1 = \mathbf{\Gamma}_1 + \mathbf{\Delta}_1 = \mathbf{D}_2\mathbf{\Gamma}_2 + \mathbf{\Delta}_1$. In other words, $\mathbf{\Gamma}_1$ is a signal that admits a convolutional sparse representation $\mathbf{D}_2\mathbf{\Gamma}_2$, which is contaminated with noise $\mathbf{\Delta}_1$, resulting in $\hat{\mathbf{\Gamma}}_1$. Next, we would like to employ Lemma 7 for the signal $\hat{\mathbf{\Gamma}}_1 = \mathbf{\Gamma}_1 + \mathbf{\Delta}_1 = \mathbf{D}_2\mathbf{\Gamma}_2 + \mathbf{\Delta}_1$, with the local noise level

$$\|\mathbf{\Delta}_1\|_{2,\infty}^{\mathbf{P}} \leq \sqrt{\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{P}}}\Big(\epsilon_0 + \mu(\mathbf{D}_1)\left(\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{S}} - 1\right)|\Gamma_1^{\max}|\Big) = \epsilon_1,$$

to obtain the stability of the second stage. To this end, we require its conditions to hold; in particular, the $\ell_{0,\infty}$ norm of $\mathbf{\Gamma}_2$ to obey

$$\|\mathbf{\Gamma}_2\|_{0,\infty}^{\mathbf{S}} < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D}_2)}\frac{|\Gamma_2^{\min}|}{|\Gamma_2^{\max}|}\right) - \frac{1}{\mu(\mathbf{D}_2)}\frac{\epsilon_1}{|\Gamma_2^{\max}|},$$

and the threshold β_2 to satisfy

$$|\mathbf{\Gamma}_2^{\min}| - (\|\mathbf{\Gamma}_2\|_{0,\infty}^{\mathbf{S}} - 1)\mu(\mathbf{D}_2)|\mathbf{\Gamma}_2^{\max}| - \epsilon_1 > \ \beta_2 > \|\mathbf{\Gamma}_2\|_{0,\infty}^{\mathbf{S}}\mu(\mathbf{D}_2)|\mathbf{\Gamma}_2^{\max}| + \epsilon_1.$$

Assuming the above hold, Lemma 7 guarantees that the support of $\hat{\mathbf{\Gamma}}_2$ is equal to that of $\mathbf{\Gamma}_2$, and also that

$$\|\mathbf{\Gamma}_2 - \hat{\mathbf{\Gamma}}_2\|_{2,\infty}^{\mathbf{P}} \leq \sqrt{\|\mathbf{\Gamma}_2\|_{0,\infty}^{\mathbf{P}}}\Big(\epsilon_1 + \mu(\mathbf{D}_2)\left(\|\mathbf{\Gamma}_2\|_{0,\infty}^{\mathbf{S}} - 1\right)|\Gamma_2^{\max}|\Big) = \epsilon_2.$$

Using the same steps as above, we obtain the desired claim for all the remaining layers, assuming that

$$\|\mathbf{\Gamma}_i\|_{0,\infty}^{\mathbf{s}} < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D}_i)}\frac{|\Gamma_i^{\min}|}{|\Gamma_i^{\max}|}\right) - \frac{1}{\mu(\mathbf{D}_i)}\frac{\epsilon_{i-1}}{|\Gamma_i^{\max}|}$$

and that the thresholds β_i are chosen to satisfy

$$|\mathbf{\Gamma}_i^{\min}| - (\|\mathbf{\Gamma}_i\|_{0,\infty}^{\mathbf{s}} - 1)\mu(\mathbf{D}_i)|\mathbf{\Gamma}_i^{\max}| - \epsilon_{i-1} > \ \beta_i > \|\mathbf{\Gamma}_i\|_{0,\infty}^{\mathbf{s}}\mu(\mathbf{D}_i)|\mathbf{\Gamma}_i^{\max}| + \epsilon_{i-1}. \tag{18}$$

This completes our proof. ■

Appendix E. Stable Recovery of Soft Thresholding in the Presence of Noise (Proof of Lemma 9)

Proof The success of the soft thresholding algorithm with threshold β_1 in recovering the correct support is guaranteed if the following holds

$$\min_{i\in\mathcal{T}_1}\left|\mathbf{d}_{1,i}^T\mathbf{Y}\right| > \beta_1 > \max_{j\notin\mathcal{T}_1}\left|\mathbf{d}_{1,j}^T\mathbf{Y}\right|.$$

Since the soft thresholding operator chooses all atoms with correlations greater than β_1, the above implies that the true support $\mathcal{T}_1$ will be chosen. This condition is equal to that of the hard thresholding algorithm, and thus using the same steps as in Lemma 7, we are guaranteed that the correct support will be chosen under Assumptions (a) and (b).

The difference between the hard thresholding algorithm and its soft counterpart becomes apparent once we consider the estimated sparse vector. While the former estimates the non-zero entries in $\hat{\mathbf{\Gamma}}_1$ by computing $\mathbf{D}_{1,\mathcal{T}_1}^T\mathbf{Y}$, the latter subtracts or adds a constant β_1 from these, obtaining $\mathbf{D}_{1,\mathcal{T}_1}^T\mathbf{Y} - \beta_1\mathbf{B}$, where $\mathbf{B}$ is a vector of ± 1. As a result, the distance between the true sparse vector and the estimated one is given by

$$\begin{aligned}
\|\mathbf{\Gamma}_1 - \hat{\mathbf{\Gamma}}_1\|_\infty &= \left\|\mathbf{\Gamma}_{1,\mathcal{T}_1} - \hat{\mathbf{\Gamma}}_{1,\mathcal{T}_1}\right\|_\infty \\
&= \left\|\left(\mathbf{D}_{1,\mathcal{T}_1}^T\mathbf{D}_{1,\mathcal{T}_1}\right)^{-1}\mathbf{D}_{1,\mathcal{T}_1}^T\mathbf{X} - \left(\mathbf{D}_{1,\mathcal{T}_1}^T\mathbf{Y} - \beta_1\mathbf{B}\right)\right\|_\infty \\
&\le \left\|\left(\mathbf{D}_{1,\mathcal{T}_1}^T\mathbf{D}_{1,\mathcal{T}_1}\right)^{-1}\mathbf{D}_{1,\mathcal{T}_1}^T\mathbf{X} - \mathbf{D}_{1,\mathcal{T}_1}^T\mathbf{Y}\right\|_\infty + \|\beta_1\mathbf{B}\|_\infty,
\end{aligned}$$

where in the last step we have used the triangle inequality for the ℓ_∞ norm. Notice that $\|\beta_1\mathbf{B}\|_\infty = \beta_1$, since β_1 must be positive according to Equation (13). Combining this together with the same steps as those used in proving Lemma 7, the above can be bounded by

$$\|\mathbf{\Gamma}_1 - \hat{\mathbf{\Gamma}}_1\|_{2,\infty}^{\mathbf{p}} \le \sqrt{\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{p}}}\Big(\epsilon_0 + \mu(\mathbf{D}_1)\left(\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{s}} - 1\right)|\Gamma_1^{\max}| + \beta_1\Big),$$

resulting in the desired claim. ■

Appendix F. Stability of the Layered BP Algorithm in the Presence of Noise (Proof of Theorem 12)

Proof In (Papyan et al., 2016b), for a signal $\mathbf{Y} = \mathbf{X} + \mathbf{E} = \mathbf{D}_1\mathbf{\Gamma}_1 + \mathbf{E}$, it was shown that if the following hold:

a) $\|\mathbf{Y} - \mathbf{X}\|_{2,\infty}^{\mathbf{P}} \leq \epsilon_0$; and

b) $\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{S}} < \frac{1}{3}\left(1 + \frac{1}{\mu(\mathbf{D}_1)}\right)$,

then the solution $\hat{\mathbf{\Gamma}}_1$ for the Lagrangian formulation of the BP problem with $\beta_1 = 4\epsilon_0$ (see Equation (8)) satisfies that

1. The support of the solution $\hat{\mathbf{\Gamma}}_1$ is contained in that of $\mathbf{\Gamma}_1$;
2. $\|\mathbf{\Delta}_1\|_\infty = \|\mathbf{\Gamma}_1 - \hat{\mathbf{\Gamma}}_1\|_\infty \leq 7.5\ \epsilon_0$;
3. In particular, every entry of $\mathbf{\Gamma}_1$ greater in absolute value than $7.5\ \epsilon_0$ is guaranteed to be recovered; and
4. The solution $\hat{\mathbf{\Gamma}}_1$ is the unique minimizer of the Lagrangian BP problem (Equation (8)).

Using similar steps to those employed in the proof of Theorem 8, we obtain that

$$\|\mathbf{\Delta}_1\|_{2,\infty}^{\mathbf{P}} \leq \sqrt{\|\mathbf{\Delta}_1\|_{0,\infty}^{\mathbf{P}}}\ \|\mathbf{\Delta}_1\|_\infty.$$

Plugging above the inequality $\|\mathbf{\Delta}_1\|_\infty \leq 7.5\ \epsilon_0$, we get

$$\|\mathbf{\Delta}_1\|_{2,\infty}^{\mathbf{P}} \leq \sqrt{\|\mathbf{\Delta}_1\|_{0,\infty}^{\mathbf{P}}}\ 7.5\ \epsilon_0.$$

Since the support of $\hat{\mathbf{\Gamma}}_1$ is contained in that of $\mathbf{\Gamma}_1$, we have that $\|\mathbf{\Delta}_1\|_{0,\infty}^{\mathbf{P}} = \|\mathbf{\Gamma}_1 - \hat{\mathbf{\Gamma}}_1\|_{0,\infty}^{\mathbf{P}} \leq \|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{P}}$, leading to

$$\|\mathbf{\Delta}_1\|_{2,\infty}^{\mathbf{P}} \leq \sqrt{\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{P}}}\ 7.5\ \epsilon_0 = \epsilon_1.$$

We conclude that the first stage of the layered BP is stable and the following must hold

1. The support of the solution $\hat{\mathbf{\Gamma}}_1$ is contained in that of $\mathbf{\Gamma}_1$;
2. $\|\mathbf{\Delta}_1\|_{2,\infty}^{\mathbf{P}} \leq\ \epsilon_1$;
3. In particular, every entry of $\mathbf{\Gamma}_1$ greater in absolute value than $\frac{\epsilon_1}{\sqrt{\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{P}}}}$ is guaranteed to be recovered; and
4. The solution $\hat{\mathbf{\Gamma}}_1$ is the unique minimizer of the Lagrangian BP problem (Equation (8)).

Next, we turn to the stability of the second stage of the layered BP algorithm. Notice that $\hat{\mathbf{\Gamma}}_1 = \mathbf{\Gamma}_1 + \mathbf{\Delta}_1 = \mathbf{D}_2\mathbf{\Gamma}_2 + \mathbf{\Delta}_1$. Put differently, $\mathbf{\Gamma}_1$ is a signal that admits a convolutional sparse representation $\mathbf{D}_2\mathbf{\Gamma}_2$ that is perturbed by $\mathbf{\Delta}_1$, resulting in $\hat{\mathbf{\Gamma}}_1$. As such, we can invoke once again the same theorem from (Papyan et al., 2016b). Since we have that

a) $\|\mathbf{\Delta}_1\|_{2,\infty}^{\mathbf{P}} \leq \epsilon_1$; and

b) $\|\mathbf{\Gamma}_2\|_{0,\infty}^{\mathbf{S}} < \frac{1}{3}\left(1 + \frac{1}{\mu(\mathbf{D}_2)}\right)$,

we are guaranteed that the solution $\hat{\mathbf{\Gamma}}_2$ for the Lagrangian formulation of the BP problem with parameter $\beta_2 = 4\epsilon_1$ satisfies

1. The support of the solution $\hat{\mathbf{\Gamma}}_2$ is contained in that of $\mathbf{\Gamma}_2$;
2. $\|\mathbf{\Delta}_2\|_\infty = \|\mathbf{\Gamma}_2 - \hat{\mathbf{\Gamma}}_2\|_\infty \leq 7.5\ \epsilon_1$;
3. In particular, every entry of $\mathbf{\Gamma}_2$ greater in absolute value than $7.5\ \epsilon_1$ is guaranteed to be recovered; and
4. The solution $\hat{\mathbf{\Gamma}}_2$ is the unique minimizer of the Lagrangian BP problem (Equation (8)).

Using similar steps to those used above, the inequality that relies on the ℓ_∞ norm can be translated into another one that depends on the $\ell_{2,\infty}$. This results in

$$\|\mathbf{\Delta}_2\|_{2,\infty}^{\mathbf{P}} \leq \sqrt{\|\mathbf{\Gamma}_2\|_{0,\infty}^{\mathbf{P}}}\ 7.5\ \epsilon_1 = \epsilon_2.$$

We conclude that, similar to the first one, the second stage of the layered BP is stable and the following must hold

1. The support of the solution $\hat{\mathbf{\Gamma}}_2$ is contained in that of $\mathbf{\Gamma}_2$;
2. $\|\mathbf{\Delta}_2\|_{2,\infty}^{\mathbf{P}} \leq\ \epsilon_2$;
3. In particular, every entry of $\mathbf{\Gamma}_2$ greater in absolute value than $\frac{\epsilon_2}{\sqrt{\|\mathbf{\Gamma}_2\|_{0,\infty}^{\mathbf{P}}}}$ is guaranteed to be recovered; and
4. The solution $\hat{\mathbf{\Gamma}}_2$ is the unique minimizer of the Lagrangian BP problem (Equation (8)).

Using the same set of steps, we obtain similarly the stability of the remaining layers. ■

References

Sanjeev Arora, Aditya Bhaskara, Rong Ge, and Tengyu Ma. Provable bounds for learning some deep representations. In *ICML*, pages 584–592, 2014.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *journal of machine learning research*, 3(Feb):1137–1155, 2003.

Hilton Bristow, Anders Eriksson, and Simon Lucey. Fast convolutional sparse coding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 391–398, 2013.

Joan Bruna and Stéphane Mallat. Invariant scattering convolution networks. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1872–1886, 2013.

Emmanuel Candes and Terence Tao. The dantzig selector: Statistical estimation when p is much larger than n. *The Annals of Statistics*, pages 2313–2351, 2007.

Emmanuel J Candes, Justin K Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on pure and applied mathematics*, 59(8):1207–1223, 2006.

Scott Shaobing Chen, David L Donoho, and Michael A Saunders. Atomic Decomposition by Basis Pursuit. *SIAM Review*, 43(1):129–159, 2001.

Sheng Chen, Stephen A Billings, and Wan Luo. Orthogonal least squares methods and their application to non-linear system identification. *International Journal of control*, 50 (5):1873–1896, 1989.

Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. In *AISTATS*, 2015.

Patrick L Combettes and Valérie R Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling & Simulation*, 4(4):1168–1200, 2005.

Ingrid Daubechies, Michel Defrise, and Christine De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on pure and applied mathematics*, 57(11):1413–1457, 2004.

Ingrid Daubechies et al. *Ten lectures on wavelets*, volume 61. SIAM, 1992.

Yann N Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in neural information processing systems*, pages 2933–2941, 2014.

Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2016.

Weisheng Dong, Lei Zhang, Guangming Shi, and Xiaolin Wu. Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization. *IEEE Trans. on Image Process.*, 20(7):1838–1857, 2011.

David L Donoho and Michael Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via 1 minimization. *Proceedings of the National Academy of Sciences*, 100(5):2197–2202, 2003.

David L Donoho, Michael Elad, and Vladimir N Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Transactions on information theory*, 52(1):6–18, 2006.

M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. Image Process.*, 15(12):3736–3745, December 2006. ISSN 1057-7149.

Michael Elad. *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. Springer Publishing Company, Incorporated, 1st edition, 2010. ISBN 144197010X, 9781441970107.

Michael Elad and Alfred M Bruckstein. A generalized uncertainty principle and sparse representation in pairs of bases. *IEEE Transactions on Information Theory*, 48(9):2558–2567, 2002.

Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. Learning hierarchical features for scene labeling. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1915–1929, 2013.

Alhussein Fawzi, Mike Davies, and Pascal Frossard. Dictionary learning for fast classification based on soft-thresholding. *International Journal of Computer Vision*, 114(2-3):306–321, 2015.

Simon Foucart and Holger Rauhut. *A mathematical introduction to compressive sensing*, volume 1. Springer, 2013.

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

Raja Giryes, Guillermo Sapiro, and Alex M Bronstein. Deep neural networks with random gaussian weights: A universal classification strategy? *IEEE Transactions on Signal Processing*, 64(13):3444–3457, 2015.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Aistats*, volume 15, page 275, 2011.

Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 399–406, 2010.

Shuhang Gu, Wangmeng Zuo, Qi Xie, Deyu Meng, Xiangchu Feng, and Lei Zhang. Convolutional sparse coding for image super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1823–1831, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

Felix Heide, Wolfgang Heidrich, and Gordon Wetzstein. Fast and flexible convolutional sparse coding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5135–5143. IEEE, 2015.

Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Zhuolin Jiang, Zhe Lin, and Larry S Davis. Learning a discriminative dictionary for sparse coding via label consistent K-SVD. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1697–1704. IEEE, 2011.

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. *arXiv preprint arXiv:1603.08155*, 2016.

Koray Kavukcuoglu, Pierre Sermanet, Y-Lan Boureau, Karol Gregor, Michaël Mathieu, and Yann L Cun. Learning convolutional feature hierarchies for visual recognition. In *Advances in neural information processing systems*, pages 1090–1098, 2010.

Bailey Kong and Charless C Fowlkes. Fast convolutional sparse coding (fcsc). *Department of Computer Science, University of California, Irvine, Tech. Rep*, 2014.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

B Boser LeCun, John S Denker, D Henderson, Richard E Howard, W Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*. Citeseer, 1990.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553): 436–444, 2015.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

Julien Mairal, Francis Bach, and Jean Ponce. Task-driven dictionary learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4):791–804, 2012.

Julien Mairal, Francis Bach, and Jean Ponce. Sparse modeling for image and vision processing. *arXiv preprint arXiv:1411.3230*, 2014.

Stéphane Mallat. Group invariant scattering. *Communications on Pure and Applied Mathematics*, 65(10):1331–1398, 2012.

Stéphane G Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on signal processing*, 41(12):3397–3415, 1993.

Stphane Mallat. *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*. Academic Press, 3rd edition, 2008. ISBN 0123743702, 9780123743701.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

Vardan Papyan, Jeremias Sulam, and Michael Elad. Working locally thinking globally-part I: Theoretical guarantees for convolutional sparse coding. *arXiv preprint arXiv:1607.02005*, 2016a.

Vardan Papyan, Jeremias Sulam, and Michael Elad. Working locally thinking globally-part II: Stability and algorithms for convolutional sparse coding. *arXiv preprint arXiv:1607.02009*, 2016b.

Yagyensh Chandra Pati, Ramin Rezaiifar, and PS Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers*, pages 40–44. IEEE, 1993.

Yunchen Pu, Xin Yuan, Andrew Stevens, Chunyuan Li, and Lawrence Carin. A deep generative deconvolutional image model. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pages 741–750, 2016.

Yaniv Romano and Michael Elad. Patch-disagreement as a way to improve K-SVD denoising. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1280–1284. IEEE, 2015.

R. Rubinstein, M. Zibulevsky, and M. Elad. Double sparsity : Learning sparse dictionaries for sparse signal approximation. *IEEE Transactions on Signal Processing*, 58(3):1553–1564, 2010.

Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

Karin Schnass and Pierre Vandergheynst. Average performance analysis for thresholding. *IEEE Signal Processing Letters*, 14(11):828–831, 2007.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Jeremias Sulam and Michael Elad. Expected patch log likelihood with a sparse prior. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 99–111. Springer, 2015.

Jeremias Sulam, Boaz Ophir, Michael Zibulevsky, and Michael Elad. Trainlets: Dictionary learning in high dimensions. *IEEE Transactions on Signal Processing*, 64(12):3180–3193, 2016.

Rob Traver. What is a good guiding question?. *Educational Leadership*, 55(6):70–73, 1998.

J. A. Tropp. Just Relax : Convex Programming Methods for Identifying Sparse Signals in Noise. *IEEE Transactions on In*, 52(3):1030–1051, 2006.

Joel A Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Transactions on Information theory*, 50(10):2231–2242, 2004.

Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. *arXiv preprint arXiv:1603.03417*, 2016.

Lloyd R Welch. Lower bounds on the maximum cross correlation of signals (corresp.). *IEEE Transactions on Information Theory*, 20(3):397–399, 1974.

Brendt Wohlberg. Efficient convolutional sparse coding. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7173–7177. IEEE, 2014.

Bo Xin, Yizhou Wang, Wen Gao, and David Wipf. Maximal sparsity with deep networks? *arXiv preprint arXiv:1605.01636*, 2016.

Matthew D Zeiler, Dilip Krishnan, Graham W Taylor, and Rob Fergus. Deconvolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2528–2535. IEEE, 2010.

Qiang Zhang and Baoxin Li. Discriminative K-SVD for dictionary learning in face recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2691–2698. IEEE, 2010.