SIAM J. Math. Data Sci.
Vol. 1, No. 1, pp. 46–77

# Multi-Layer Sparse Coding: The Holistic Way[*]

Aviad Aberdam[†], Jeremias Sulam[‡], and Michael Elad[§]

**Abstract.** The recently proposed multi-layer sparse model has raised insightful connections between sparse representations and convolutional neural networks (CNNs). In its original conception, this model was restricted to a cascade of *convolutional synthesis* representations. In this paper, we start by addressing a more general model, revealing interesting ties to fully connected networks. We then show that this multi-layer construction admits a brand new interpretation in a unique symbiosis between synthesis and analysis models: while the deepest layer indeed provides a synthesis representation, the midlayer decompositions provide an analysis counterpart. This new perspective exposes the suboptimality of previously proposed pursuit approaches, as they do not fully leverage all the information comprised in the model constraints. Armed with this understanding, we address fundamental theoretical issues, revisiting previous analysis and expanding it. Motivated by the limitations of previous algorithms, we then propose an integrated—*holistic*—alternative that estimates all representations in the model simultaneously, and we analyze all these different schemes under stochastic noise assumptions. Inspired by the synthesis-analysis duality, we further present a Holistic Pursuit algorithm, which alternates between synthesis- and analysis-sparse coding steps, eventually solving for the entire model as a whole, with provable improved performance. Finally, we present numerical results that demonstrate the practical advantages of our approach.

**Key words.** sparse representations, multi-layer representations, sparse coding, analysis and synthesis priors, neural networks

**AMS subject classifications.** 65F10, 65F20, 65F22, 68W25, 62H35, 47A52, 65F50, 62M45

**DOI.** 10.1137/18M1183352

## 1. Introduction.

Sparse representation is one of the most popular priors in signal and image processing, leading to remarkable results in various applications [10, 13, 20, 23, 40]. In its most popular interpretation, this model assumes that a natural signal, represented by the vector $\mathbf{x} \in \mathbb{R}^n$, can be *synthesized* as a linear combination of only a few columns, or atoms, from a matrix $\mathbf{D} \in \mathbb{R}^{n \times m}$, termed a dictionary. In other words, $\mathbf{x} = \mathbf{D}\boldsymbol{\gamma}$, where the vector $\boldsymbol{\gamma} \in \mathbb{R}^m$ is sparse: only a few of its entries are nonzeros, which is indicated by its low $\ell_0$ (pseudo)norm, $\|\boldsymbol{\gamma}\|_0 \ll n$.

In [3, 17, 18, 21, 29], this general model was deployed in a Convolutional Sparse Coding

[*] Received by the editors April 26, 2018; accepted for publication (in revised form) September 28, 2018; published electronically February 12, 2019.

http://www.siam.org/journals/simods/1-1/M118335.html

**Funding:** The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme, ERC grant agreement 320649, and in part by Israel Science Foundation (ISF) grant 335/18.

[†] Electrical Engineering Department, Technion Israel Institute of Technology, Haifa 3200003, Israel (aaberdam@campus.technion.ac.il).

[‡] Biomedical Engineering Department, Johns Hopkins University, Baltimore, MD 21205 (jsulam1@jhu.edu).

[§] Computer Science Department, Technion Israel Institute of Technology, Haifa 3200003, Israel (elad@cs.technion.ac.il).

(CSC) form, in which the dictionary $\mathbf{D}$ is given by a union of banded and circulant matrices. More recently, this CSC model has been extended to a multi-layer version in [27]. This construction, termed Multi-Layer Convolutional Sparse Coding (ML-CSC), raises particular interest because of its tight connection to deep learning. Somewhat surprisingly, under this model assumption, the forward pass of a CNN can be interpreted as a pursuit algorithm searching for the respective sparse representations of a given input signal [27]. As a result, this provides a promising framework for a theoretical study and analysis of deep learning architectures.

In its original formulation [27, 37], this multi-layer model was interpreted as a cascade of synthesis-sparse representations. More precisely, every intermediate layer in this model wears two hats: it provides a sparse representation for the previous layer, while also acting as a signal for the subsequent layer. This perception has led the authors of [27] to propose synthesis-oriented pursuit algorithms that proceed in a layer-by-layer fashion, propagating the signal from the input to the deepest layer [27]. Alternatively, one may adopt a projection approach by seeking the deepest representation and then propagating it back towards shallower layers [37]. In this paper, we revisit these algorithms more broadly, adopting fully connected layers, providing more general constructions that are not restricted to the convolutional case.

As we will carefully show, the above pursuit algorithms suffer from several caveats as neither approach can fully leverage all the information in the model. The layerwise approach provides representations with increasing deviations from the input signal. The projection variant resolves this issue, though it condenses to a traditional global synthesis model that fails to explicitly employ the information represented in the intermediate layers. In addition, the analysis and performance of these methods rely on the intermediate dictionaries being sparse, thus enabling sparse intermediate decompositions. We will show that this does not have to be the case, and one can indeed consider more general dictionaries while still allowing for signals in the model. Alas, the above algorithms collapse in such cases and fail to retrieve the corresponding representations even in noiseless (ideal) cases.

Motivated by these observations, and aiming to effectively resolve these problems, we give the multi-layer sparse model a brand new interpretation. The key observation is that the Multi-Layer Sparse Coding (ML-SC) model provides a unique integration between two types of sparse models: synthesis and analysis [14, 24, 25, 34, 35]. As explained above, the synthesis-sparse model assumes that a signal $\mathbf{x}$ can be expressed as $\mathbf{D}\boldsymbol{\gamma}$, with $\boldsymbol{\gamma}$ being sparse. The analysis counterpart—a somewhat more recent and less glaring variant—states that a signal $\mathbf{x}$ provides a sparse representation after being multiplied by an analysis dictionary, $\boldsymbol{\Omega} \in \mathbb{R}^{m\times n}$. In this way, $\|\boldsymbol{\Omega}\mathbf{x}\|_0 = m - \ell$, where the number of zeros, $\ell$, refers to the cardinality of the *cosupport* of $\mathbf{x}$, termed *cosparsity*. With this understanding, we will show that while the outer or global shell of the ML-SC model provides a synthesis representation for a given signal, the intermediate representations enforce an analysis prior on the deepest representation.

This new synthesis-analysis conception leads us to reformulate and expand on several theoretical questions: When is the model valid? More precisely, are there signals that admit all model constraints? What is the ML-SC signal space? How can we synthesize an ML-SC signal under this interpretation? How can uniqueness guarantees be improved based on these extra constraints? The answers to these questions will shed light on the achievable sparsity bounds of the intermediate representations. Interestingly, our analysis shows that these should, in fact, not be *too sparse*. This will eliminate the restriction of employing sparse matrices as the intermediate dictionaries, on the one hand, and provide a closer behavior to what is observed in practical deep neural networks, on the other.

Driven by the limitations of previous pursuit algorithms and the offered analysis-synthesis perspective, we turn to define a novel pursuit approach. Our proposed method seeks to estimate all representations in the model simultaneously, for which reason we dubbed it a *Holistic* pursuit. We first analyze the performance of its *Oracle* estimator (i.e., having knowledge of the underlying representation supports) and compare it with the corresponding estimators of the previous layerwise and projection alternatives.

We then propose a practical Holistic Pursuit algorithm, which iteratively builds on the intermediate layer supports while refining the global synthesis representation, improving on it at every step. To this end, this algorithm alternates between a synthesis-type sparse coding of the deepest layer, and an analysis-like estimation of the midlayer supports using the deepest layer estimation. This holistic approach leverages the synergy across layers, resulting in improved provable recovery guarantees and performance bounds.

This paper is organized as follows: In section 2 we review the basics of the sparse representations model. Section 3 then introduces previous theoretical claims for the ML-CSC model and adapts them to a more general (nonconvolutional) case. In section 4 we demonstrate the limitations of the existing multi-layer synthesis interpretation and introduce an analysis perspective to these constructions. We undertake the study of uniqueness guarantees in light of the synthesis-analysis duality in section 5 and present a holistic approach for the pursuit of these representations. Section 6 provides the analysis of the Oracle estimators for the different pursuit approaches under random noise assumptions, while in section 7 we present a new pursuit algorithm for the ML-SC model, the *Holistic Pursuit*, which we demonstrate with numerical experiments in section 8. We finally conclude in section 9, delineating further research directions.

### 1.1. Notation.

Vectors and matrices are denoted by bold lowercase and uppercase letters, respectively. $\mathbf{x}^{\Lambda}$ denotes the vector in $\mathbb{R}^{|\Lambda|}$ that carries the entries of $\mathbf{x}$ indexed by $\Lambda$. Naturally, $\Lambda^c$ will denote the complement of the set $\Lambda$. Similarly, when $\mathbf{D}$ is a matrix in $\mathbb{R}^{n\times m}$ and $\Lambda_c \subseteq \{1,\dots,m\}$, $\mathbf{D}^{\Lambda_c}$ is the submatrix in $\mathbb{R}^{n\times|\Lambda_c|}$ whose columns are those of $\mathbf{D}$ indexed by $\Lambda_c$. If $\Lambda_r \subseteq \{1,\dots,n\}$, then the matrix $\mathbf{D}^{\Lambda_r,\Lambda_c}$ is the matrix in $\mathbb{R}^{|\Lambda_r|\times|\Lambda_c|}$ whose rows are those of $\mathbf{D}^{\Lambda_c}$ indexed by $\Lambda_r$. We will further denote by $\mathbf{D}^{\Lambda_r,\mathcal{I}}$ the matrix containing the rows indexed by $\Lambda_r$ across all $m$ columns of $\mathbf{D}$, where $\mathcal{I}$ denotes the index set $\mathcal{I}=[1,m]$.

## 2. Sparse representation modeling background.

Many natural images and signals have been observed to be inherently low-dimensional despite their possibly very high ambient dimension. Sparse representation offers an elegant and clear way to model such inherent low-dimensionality by assuming that the signal $\mathbf{x}\in\mathbb{R}^n$ belongs to a finite (yet huge) union of $s$ ($\ll n$) dimensional subspaces [22]. This general idea comes in two forms: the traditional and very popular synthesis approach [12], and the newer and complementary analysis-sparse model [24, 25], which we review next.

### 2.1. The synthesis model.

Synthesis-sparse representations is a signal model that assumes that natural signals can be represented, or well approximated, by a linear combination of a few basic components, termed atoms. Formally, such a signal $\mathbf{x} \in \mathbb{R}^n$ can be expressed as $\mathbf{x} = \mathbf{D}\boldsymbol{\gamma}$, where $\mathbf{D} \in \mathbb{R}^{n \times m}$ is a dictionary containing signal atoms as its columns, and the vector $\boldsymbol{\gamma} \in \mathbb{R}^m$ contains only a few nonzero entries. The cardinality of a vector is measured by the $\ell_0$ pseudonorm, $\|\boldsymbol{\gamma}\|_0$. Typically, we are interested in the case of redundant dictionaries, i.e., $m > n$, allowing for very sparse representations.

The synthesis inverse problem aims to recover the sparse representation $\boldsymbol{\gamma}$ from a noisy signal observation $\mathbf{y} = \mathbf{x} + \mathbf{e} = \mathbf{D}\boldsymbol{\gamma} + \mathbf{e}$, where $\mathbf{e}$ is a noise vector and the dictionary $\mathbf{D}$ is assumed given. This task is often called sparse coding, or simply pursuit, and can be formally written[1] as [11, 12, 38]

$$(P_0): \quad \min_{\boldsymbol{\gamma}} \|\boldsymbol{\gamma}\|_0 \text{ s.t. } \|\mathbf{D}\boldsymbol{\gamma} - \mathbf{y}\|_2 \leq \epsilon. \tag{2.1}$$

Since solving the problem in (2.1) is NP-hard in general [16], one can use greedy strategies like Orthogonal Matching Pursuit (OMP) [30] or the thresholding algorithm [12, 38] to approximate its solution. Alternatively, one can also use a convex relaxation of this pursuit by replacing the $\ell_0$ norm with the convex $\ell_1$. In the latter case, the resulting problem, termed Basis-Pursuit, is defined formally as [8, 11, 39]

$$(P_1): \quad \min_{\boldsymbol{\gamma}} \|\boldsymbol{\gamma}\|_1 \text{ s.t. } \|\mathbf{D}\boldsymbol{\gamma} - \mathbf{y}\|_2 \leq \epsilon. \tag{2.2}$$

In the noiseless case, where $\epsilon = 0$, one of the fundamental questions is whether and when one can be sure that the result of these approximation algorithms is in fact the unique sparsest representation of the signal. Equivalently, from a *transformation* perspective, these questions explore the conditions under which the sparse synthesis operation is invertible. A key property for the study of uniqueness is the *spark* of the dictionary, $\sigma(\mathbf{D})$, defined as the smallest number of columns from $\mathbf{D}$ that are linearly dependent. If there exists a representation $\boldsymbol{\gamma}$ for a signal $\mathbf{x}$ such that $\|\boldsymbol{\gamma}\|_0 < \frac{\sigma(\mathbf{D})}{2}$, then this solution is necessarily the sparsest possible [11]; in other words, such a condition is sufficient for the representation to be unique. However, the spark is at least as difficult to evaluate as solving $(P_0)$, and thus it is common to lower-bound it with the mutual coherence, $\mu(\mathbf{D})$. This value is simply the maximal correlation between atoms in the dictionary:

$$\mu(\mathbf{D}) = \max_{i \neq j} \frac{\left|\mathbf{d}_i^T \mathbf{d}_j\right|}{\|\mathbf{d}_i\|_2 \|\mathbf{d}_j\|_2}, \tag{2.3}$$

where we have denoted by $\mathbf{d}_j$ the $j$th column of the matrix $\mathbf{D}$. One may then bound the spark with the mutual coherence [11], as $\sigma(\mathbf{D}) \geq 1 + \frac{1}{\mu(\mathbf{D})}$. Then, a sufficient condition for uniqueness is to require that

$$\|\boldsymbol{\gamma}\|_0 < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D})}\right). \tag{2.4}$$

[1] For the sake of simplicity and to avoid introducing more notation, we will employ hereafter the same variable to denote the ground truth and the running variable we are optimizing over; these are not to be mixed up.

### 2.2. The analysis model.

The above model has an analysis counterpart, in which the assumption is that one can linearly transform the signal into a sparse representation [24, 25]. Formally, for a fixed analysis operator $\mathbf{\Omega} \in \mathbb{R}^{m \times n}$, a signal $\mathbf{x} \in \mathbb{R}^n$ belongs to the analysis model with cosparsity $\ell$ if $\|\mathbf{\Omega x}\|_0 = m - \ell$. When $\mathbf{\Omega}$ is a squared unitary matrix, the analysis model is identical to the synthesis one with dictionary $\mathbf{D} = \mathbf{\Omega}^T$. However, in the overcomplete case where $m > n$, there is no simple connection between the two as they lead to generally different constructions. Note that, unlike the synthesis counterpart, the pursuit in the analysis model is trivial in the noiseless case as it simply amounts to a multiplication by the analysis dictionary $\mathbf{\Omega}$. In the noisy case, the process of recovering $\mathbf{x}$ from the corrupted measurements $\mathbf{y} = \mathbf{x} + \mathbf{e}$ is done by solving the following minimization problem [14]:

$$\left(P_0^\ell\right): \ \min_{\mathbf{x}} \|\mathbf{\Omega x}\|_0 \ \text{ s.t. } \ \|\mathbf{x} - \mathbf{y}\|_2 \leq \epsilon. \tag{2.5}$$

Just as in the $(P_0)$ problem, this objective is NP-hard in general, and so one must resort to greedy approaches [15, 25] or $\ell_1$ relaxation alternatives [7, 14, 25].

Finally, a third type of sparse model is that of Sparsifying Transforms [32, 33]. This analysis-type model seeks a (typically square) dictionary $\mathbf{W}$—the transform—that approximately sparsifies a signal $\mathbf{y}$, so that $\mathbf{Wy} = \boldsymbol{\gamma} + \mathbf{e}$, where $\|\boldsymbol{\gamma}\|_0 \ll n$ and $\mathbf{e}$ is some nuisance (dense) vector. The optimization problems related to this model present interesting advantages, as the pursuit is no longer an NP-hard problem but rather a simple thresholding operation. We will not dwell on Transform Learning any further in this paper, but we believe that many of the ideas raised in our work could be adapted to this model form as well.

## 3. The Multi-Layer Sparse Coding model.

While the above sparse models have been around for nearly two decades, a multi-layer extension was only recently introduced. This was done in a convolutional setting, thus termed Multi-Layer Convolutional Sparse Coding (ML-CSC) [27, 37]. This model is an extension of the convolutional sparse model [3, 29], which addresses the modeling of high-dimensional signals through local shift-invariant sparse decompositions. It is our intention in this work to consider a more general case and not to restrict ourselves to the convolutional scenario. We refer the interested reader to [28, 29] for a thorough review of convolutional sparse representations, their associated results, and algorithms.

### 3.1. Model and pursuit definitions.

The synthesis-sparse model assumes that a signal $\mathbf{x} \in \mathbb{R}^n$ can be decomposed into a multiplication of a dictionary $\mathbf{D}_1 \in \mathbb{R}^{n \times m_1}$ and a sparse vector $\boldsymbol{\gamma}_1 \in \mathbb{R}^{m_1}$. In the multi-layer model we extend this by assuming that $\boldsymbol{\gamma}_1$, and in fact every sparse representation, $\boldsymbol{\gamma}_i$, can also be decomposed as $\boldsymbol{\gamma}_i = \mathbf{D}_{i+1}\boldsymbol{\gamma}_{i+1}$, where $\mathbf{D}_{i+1} \in \mathbb{R}^{m_i \times m_{i+1}}$ is the dictionary of layer $i+1$ and $\boldsymbol{\gamma}_{i+1} \in \mathbb{R}^{m_{i+1}}$ is the corresponding sparse representation. We name this the Multi-Layer Sparse Coding (ML-SC) model and formalize its definition as follows.

Definition 3.1 (ML-SC signal). *Given a set of dictionaries $\{\mathbf{D}_i\}_{i=1}^k$, of appropriate dimensions, a signal $\mathbf{x} \in \mathbb{R}^n$ admits a representation in terms of the ML-SC model if*

$$\begin{aligned} \mathbf{x} &= \mathbf{D}_1\boldsymbol{\gamma}_1, \quad \|\boldsymbol{\gamma}_1\|_0 \le s_1, \\ \boldsymbol{\gamma}_1 &= \mathbf{D}_2\boldsymbol{\gamma}_2, \quad \|\boldsymbol{\gamma}_2\|_0 \le s_2, \\ &\vdots \\ \boldsymbol{\gamma}_{k-1} &= \mathbf{D}_k\boldsymbol{\gamma}_k, \quad \|\boldsymbol{\gamma}_k\|_0 \le s_k. \end{aligned} \tag{3.1}$$

For the purpose of the following derivations, define $\mathbf{D}_{(i,j)}$ to be the effective dictionary from the $i$th layer to the $j$th layer, i.e., $\mathbf{D}_{(i,j)} = \mathbf{D}_i\mathbf{D}_{i+1}\cdots\mathbf{D}_j$. This way, one can concisely write $\boldsymbol{\gamma}_i = \mathbf{D}_{(i,j)}\boldsymbol{\gamma}_j$. For effective dictionaries from the first layer to the $j$th layer, we simplify the notation and denote $\mathbf{D}_{(j)} = \mathbf{D}_{(1,j)}$, so that $\mathbf{x} = \mathbf{D}_{(i)}\boldsymbol{\gamma}_i$. The ML-SC can then be interpreted as a global synthesis model, $\mathbf{x} = \mathbf{D}_{(k)}\boldsymbol{\gamma}_k$, with additional intermediate layer constraints. As we will see, these two observations are laying the foundation for the current pursuit algorithms proposed by recent works [27, 37]. We now formalize the pursuit for the ML-SC model, referred to as Deep Pursuit.

Definition 3.2 (Deep Pursuit). *For a signal $\mathbf{y} = \mathbf{x} + \mathbf{e}$, where $\mathbf{x}$ is an ML-CS signal and $\mathbf{e}$ is an additive noise, assume that the set of dictionaries $\{\mathbf{D}_i\}_{i=1}^k$, the cardinality vector $\mathbf{s}$, and the noise energy $\epsilon$, are all known. Define the Deep Pursuit ($DP_\mathbf{s}$) problem as*

$$(DP_\mathbf{s}): \ \textit{find } \{\boldsymbol{\gamma}_i\}_{i=1}^k \ \ \textit{s.t. } \begin{aligned}[t] &\|\mathbf{y} - \mathbf{D}_1\boldsymbol{\gamma}_1\|_2 \le \epsilon, \\ &\boldsymbol{\gamma}_{i-1} = \mathbf{D}_i\boldsymbol{\gamma}_i \ \forall\ 2 \le i \le k, \\ &\|\boldsymbol{\gamma}_i\|_0 \le s_i \ \forall\ 1 \le i \le k, \end{aligned} \tag{3.2}$$

*where the scalar $s_i$ is the ith entry of $\mathbf{s}$.*

We are interested in the following questions: Is the solution of $(DP_\mathbf{s})$ unique in a noiseless $(\epsilon = 0)$ setting? Is the solution stable to noise contamination? And under which conditions would these be true?

**3.2. Uniqueness.** Consider a set of dictionaries $\{\mathbf{D}_i\}_{i=1}^k$ and a signal $\mathbf{x}$ admitting a multi-layer sparse representation defined by the set $\{\boldsymbol{\gamma}_i\}_{i=1}^k$. The claim of uniqueness answers the question of whether another set of sparse vectors can represent the same signal $\mathbf{x}$. We present here the uniqueness theorem from [27], with necessary changes that make it suitable for the general (i.e., nonconvolutional) multi-layer sparse model.

Theorem 3.3 (uniqueness via the mutual coherence). *Consider a noiseless ML-SC signal $\mathbf{x}$, its set of dictionaries $\{\mathbf{D}_i\}_{i=1}^k$, and their corresponding mutual coherence constants $\mu(\mathbf{D}_i)$ $\forall 1 \le i \le k$. If*

$$\forall\ 1 \le i \le k, \quad \|\boldsymbol{\gamma}_i\|_0 = s_i < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D}_i)}\right), \tag{3.3}$$

*then the set $\{\boldsymbol{\gamma}_i\}_{i=1}^k$ is the unique solution to the $DP_\mathbf{s}$ problem.*

The simple modus operandi behind this result is to propagate the uniqueness guarantees progressively through the layers. In other words, it first demands the first layer to be the unique representation of the signal, then it demands the second layer to be the unique representation of the first layer, and so on. In [37], the authors suggested an improvement based on a projection approach. Instead of propagating the uniqueness conditions layer by layer, one can project the signal directly to the deepest representation layer, and demand uniqueness using the effective model, $\mathbf{D}_{(k)}$, requiring

$$\|\boldsymbol{\gamma}_k\|_0 < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D}_{(k)})}\right). \tag{3.4}$$

An immediate way of improving over these uniqueness conditions is to maximize between (3.3) and (3.4). One can leverage this idea in order to maximize over the entire combination of the mid-effective dictionaries, $\mathbf{D}_{(i,j)}$, and if there is any partition of $\{1, \ldots, k\}$ such that $\boldsymbol{\gamma}_k$ is guaranteed to be unique, then the entire representation set $\{\boldsymbol{\gamma}_i\}_{i=1}^k$ is thus also unique. However, as we will see in the next section, all these variants of uniqueness guarantees are in fact too restrictive and do not capture the true essence of the ML-SC model. In section 5 we will revisit this matter and provide a better study of this property, with far tighter bounds.

### 3.3. Stability.

Real signals might contain noise or deviations from the idealistic model assumptions presented above. In these cases, one would like to know what the error in the estimated representation is, and how sensitive the different pursuit formulations are to different levels of noise. In other words, we would like to analyze the stability of the solutions to the pursuit problems. The theorem below is an adaptation of a result from [27].

Theorem 3.4 (stability of the $DP_{\mathbf{s}}$ problem). *Suppose an ML-SC signal $\mathbf{x}$ is contaminated with energy-bounded noise $\mathbf{e}$, $\|\mathbf{e}\|_2 \le \epsilon$, resulting in $\mathbf{y} = \mathbf{x} + \mathbf{e}$, and suppose the set of solutions $\{\hat{\boldsymbol{\gamma}}_i\}_{i=1}^k$ is obtained by solving the $DP_{\mathbf{s}}$ problem. If the true representation set satisfies the uniqueness conditions in (3.3), then*

$$\|\boldsymbol{\gamma}_i - \hat{\boldsymbol{\gamma}}_i\|_2^2 \le 4\epsilon^2 \prod_{j=1}^{i} \frac{1}{1 - (2s_j - 1)\,\mu(\mathbf{D}_j)}. \tag{3.5}$$

Clearly, one could use the same improvements suggested above regarding the uniqueness analysis in order to further strengthen this result.

Note that the above result refers to the solution of the $DP_{\mathbf{s}}$ problem—though without specifying how such solutions could be estimated in practice. We now turn to address this aspect and present the existing pursuit algorithms as introduced in [27] and later in [37], which aim to solve the $DP_{\mathbf{s}}$ problem.

### 3.4. The Layer-by-Layer Pursuit.

Definition 3.5 (Deep Pursuit). *For a signal $\mathbf{y} = \mathbf{x} + \mathbf{e}$, where $\mathbf{x}$ is an ML-CS signal and $\mathbf{e}$ is an additive noise, assume that the set of dictionaries $\{\mathbf{D}_i\}_{i=1}^k$, the cardinality vector $\mathbf{s}$, and* *the noise energy $\epsilon$ are all known. Define the Deep Pursuit* $(DP_{\mathbf{s}})$ *problem as*

$$(DP_{\mathbf{s}}):\ \textit{find}\ \{\boldsymbol{\gamma}_i\}_{i=1}^k\ \ \textit{s.t.}\ \begin{aligned}&\|\mathbf{y}-\mathbf{D}_1\boldsymbol{\gamma}_1\|_2\le\epsilon,\\ &\boldsymbol{\gamma}_{i-1}=\mathbf{D}_i\boldsymbol{\gamma}_i\ \forall\ 2\le i\le k,\\ &\|\boldsymbol{\gamma}_i\|_0\le s_i\ \forall\ 1\le i\le k,\end{aligned} \tag{3.6}$$

*where the scalar $s_i$ is the ith entry of* $\mathbf{s}$.

In Algorithm 3.1 we present the Layered Pursuit algorithm, which was introduced and theoretically analyzed in [27], and which was shown to be connected to the forward pass of neural networks. Note that two such variants were suggested in [27]: one relying on the Thresholding algorithm, and the other on the Basis-Pursuit alternative. Algorithm 3.1 presents these two options together, where we denote by $\mathcal{H}(\cdot)$ a thresholding operator, and $P_1(\mathbf{D},\mathbf{y},\lambda)\triangleq\operatorname{argmin}_{\boldsymbol{\gamma}}\|\mathbf{D}\boldsymbol{\gamma}-\mathbf{y}\|_2^2\ \text{s. t.}\ \|\boldsymbol{\gamma}\|_1\le\lambda$.

**Algorithm 3.1.** The Layered Pursuit algorithm.

```
Input
    $\mathbf{y}$ - a signal.
    $\{\mathbf{D}_i\}_{i=1}^k$ - a set of dictionaries.
Output
    $\{\hat{\boldsymbol{\gamma}}_i\}_{i=1}^k$ - a set of representations.
Process
    $\hat{\boldsymbol{\gamma}}_0 \leftarrow \mathbf{y}$
    for $i = 1 : k$ do
        $\hat{\boldsymbol{\gamma}}_i = \begin{cases} \mathcal{H}(\mathbf{D}_i^T\hat{\boldsymbol{\gamma}}_{i-1}) & \text{Thres.} \\ P_1(\mathbf{D}_i,\hat{\boldsymbol{\gamma}}_{i-1},\lambda_i) & \text{BP} \end{cases}$
    return $\{\hat{\boldsymbol{\gamma}}_i\}_{i=1}^k$
```

**Algorithm 3.2.** The Basic Projection Pursuit algorithm.

```
Input
    $\mathbf{y}$ - a signal.
    $\{\mathbf{D}_i\}_{i=1}^k$ - a set of dictionaries.
Output
    $\{\hat{\boldsymbol{\gamma}}_i\}_{i=1}^k$ - a set of representations.
Process
    $\hat{\boldsymbol{\gamma}}_k = \begin{cases} \mathcal{H}(\mathbf{D}_{(k)}^T\mathbf{y}) & \text{Thres.} \\ P_1(\mathbf{D}_{(k)},\mathbf{y},\lambda_k) & \text{BP} \end{cases}$
    for $i = k-1 : -1 : 1$ do
        $\hat{\boldsymbol{\gamma}}_i \leftarrow \mathbf{D}_{i+1}\boldsymbol{\gamma}_{i+1}$
    return $\{\hat{\boldsymbol{\gamma}}_i\}_{i=1}^k$
```

It is important to note that, while providing approximations, this algorithm does not recover a valid ML-SC signal as it only guarantees that $\boldsymbol{\gamma}_{i-1}\approx\mathbf{D}_i\boldsymbol{\gamma}_i$. Another drawback is the recovery error, which grows as a function of the network's depth, contradicting the intuition that additional information should decrease the error.

### 3.5. The Projection Pursuit.

An alternative to the Layered Pursuit is the projection approach presented in [37]. In this algorithm, one first finds the deepest representation using the effective dictionary $\mathbf{D}_{(k)}$, and then propagates this solution all the way back to the first layer. In Algorithm 3.2 we present the simplified version of the Projection Pursuit algorithm, noting that in [37] the authors suggested an improvement that iteratively backtracks if the propagated midlayer representations violate the model constraints and attempts to find an alternative sparser and feasible representation, $\boldsymbol{\gamma}_k$, in a greedy manner.

This algorithm, if successful, provides an estimation which, unlike the previous case, satisfies the ML-SC constraints. Similar to the behavior for the uniqueness guarantees appearing in (3.4), this approach provides a looser condition, and, as presented in [37], the recovery error is reduced. However, this algorithm is essentially a single-layer effective model, and therefore it does not explicitly use all the available information.

## 4. The synthesis-analysis interpretation.

So far, the multi-layer model was interpreted as an extension of the general synthesis model. We present here several concerns that follow from this understanding:

1. Sparse dictionaries: The approaches presented above had no choice but to enforce sparsity on the intermediate dictionaries in order to ensure sparse intermediate representations. In the more general case, where the intermediate dictionaries are dense, the two presented algorithms simply fail: The layerwise approach would cause a very high error since dense dictionaries do not span sparse signals well. As a result, every midlayer pursuit, except the first one, would result in a very low SNR which further decreases as we go deeper into the model layers. The projection alternative, on the other hand, would converge to the zero representation, because even if it would attain a reasonable deepest layer estimation, the corresponding intermediate representations would become fully dense due to the estimation noise. Following the backtracking in the proposed algorithm, the deepest layer cardinality would have to reduce in an attempt to decrease the intermediate cardinalities, eventually resulting in zero representations.
2. Spanned space: Under the current interpretation, it is unclear what the space spanned by the ML-CS model is. This is related to the following questions:
   - Empty model. Are there signals, and their corresponding representations, that satisfy the model constraints, or is the model empty?
   - Model sampling. If the model is not empty, how can we synthesize signals satisfying the model constraints? What are the restrictions on the model parameters?
3. Recovery error: The ML-SC signal belongs to a model that is far more constrained than the single-layer version, as additional conditions are introduced in the form of the sparsity of the intermediate representations. Correspondingly, it is expected that the recovery error will be significantly better given these increased constraints. As we have shown above, however, the error in the layerwise method increases across the layers, and the projection method provides error bounds that are basically *single-layer* type estimates. This indicates that the current approaches provide suboptimal solutions in attempting to solve the multi-layer pursuit problem.

Motivated by these unsolved issues, we now propose to look at the ML-SC model as one piece rather than as a collection of single-layer constructions. We interpret this model as a unique combination between the synthesis and the analysis paradigms. While the outer shell maintains a synthesis interpretation, $\mathbf{x} = \mathbf{D}_{(k)}\boldsymbol{\gamma}_k$, $\|\boldsymbol{\gamma}_k\|_0 \leq s_k$, the intermediate constraints can be understood as analysis constraints on the deepest representation: $\|\boldsymbol{\gamma}_i\|_0 = \left\|\mathbf{D}_{(i+1,k)}\boldsymbol{\gamma}_k\right\|_0 \leq s_i \; \forall 1 \leq i < k$.

Armed with this observation, we can begin re-examining the ML-SC model. First, we would like to identify the true space spanned by the signals satisfying the model constraints. Relying on the effective dictionary (synthesis) interpretation, we know that these signals lie in a union of subspaces composed of all the options of choosing $s_k$ columns from $\mathbf{D}_{(k)}$, each spanning a subspace of dimension $s_k$. However, invoking the intermediate analysis constraints must further influence this construction. For known cosupports $\Lambda_1^c, \Lambda_2^c, \ldots, \Lambda_{k-1}^c$, and support

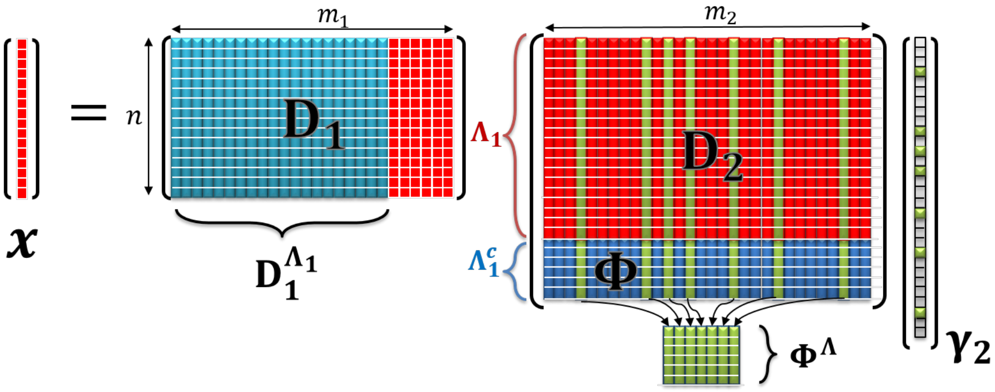


**Figure 4.1.** *Illustration of the ML-SC model for a two-layer decomposition.*

$\Lambda_k$, we define the following two matrices:

$$\boldsymbol{\Phi} \triangleq \begin{bmatrix} \mathbf{D}_{(2,k)}^{\Lambda_1^c,\mathcal{I}} \\ \mathbf{D}_{(3,k)}^{\Lambda_2^c,\mathcal{I}} \\ \vdots \\ \mathbf{D}_{(k,k)}^{\Lambda_{k-1}^c,\mathcal{I}} \end{bmatrix} \text{ and } \boldsymbol{\Phi}^{\Lambda_k} \triangleq \begin{bmatrix} \mathbf{D}_{(2,k)}^{\Lambda_1^c,\Lambda_k} \\ \mathbf{D}_{(3,k)}^{\Lambda_2^c,\Lambda_k} \\ \vdots \\ \mathbf{D}_{(k,k)}^{\Lambda_{k-1}^c,\Lambda_k} \end{bmatrix}. \tag{4.1}$$

The rows in $\boldsymbol{\Phi}$ define directions to which $\boldsymbol{\gamma}_k$ must be orthogonal, as they refer to the zeros in all the intermediate representations. Clearly, the definition of the above two matrices depends not only on the support of $\boldsymbol{\gamma}_k$, but also on the cosupports of the intermediate representations.

Considering the fact that $\boldsymbol{\gamma}_k$ is $s_k$ sparse and its support is $\Lambda_k$, the matrix $\boldsymbol{\Phi}^{\Lambda_k}$ defines a null-space to which the nonzeros in $\boldsymbol{\gamma}_k$ belong. Thus, the degrees of freedom in choosing the deepest representation reduce from $s_k$ to $s_k - \text{rank}\{\boldsymbol{\Phi}^{\Lambda_k}\}$. In other words, these signals no longer live in a union of subspaces of dimension $s_k$, but rather in a union of $s_k - \text{rank}\{\boldsymbol{\Phi}^{\Lambda_k}\}$ dimensional subspaces. Denoting by $\ell_i$ the cosparsity of $\boldsymbol{\gamma}_i$, the number of such subspaces hence grows from $\binom{m_k}{s_k}$ to $\binom{m_k}{s_k}\prod_{i=1}^{k-1}\binom{m_i}{\ell_i}$, which is the number of the supports options. These elements in the ML-SC model are depicted in Figure 4.1 for a two-layer model.

Several theoretical corollaries can be derived from this analysis, answering some of the questions and issues posed in the previous section:

1. Empty model: An immediate corollary is that as long as $s_k > \text{rank}\{\boldsymbol{\Phi}^{\Lambda_k}\}$, the model is not empty, because using the rank-nullity theorem, we know that
$$\dim\left\{\ker\left(\boldsymbol{\Phi}^{\Lambda_k}\right)\right\} = s_k - \text{rank}\{\boldsymbol{\Phi}^{\Lambda_k}\}. \tag{4.2}$$
Then, as long as $s_k > \text{rank}\{\boldsymbol{\Phi}^{\Lambda_k}\}$, there exists a $\boldsymbol{\gamma}_k$ that satisfies the model constraints.
2. *Not so sparse* intermediate layers: A very interesting benefit is that the midlayer representations must not be too sparse. In fact, contrary to previous works that limited

the midlayers' cardinality, in the analysis-synthesis view one limits the intermediate cocardinality, i.e., the number of zeros, such that $s_k > \text{rank}\{\mathbf{\Phi}^{\Lambda_k}\}$. This property mimics the typical behavior of deep neural networks, in which the intermediate representations (or activations [27]) are sparse but not extremely so.

3. Model sampling: One can now devise a systematic way to sample a signal from the model. The first step is to randomly choose the representations' support, or equivalently, select one of the subspaces. Then, one can multiply the matrix $\mathbf{K}$, which spans the null space of $\mathbf{\Phi}^{\Lambda_k}$, with a random vector $\boldsymbol{\alpha} \in \mathbb{R}^{s_k - \text{rank}\{\mathbf{\Phi}^{\Lambda_k}\}}$, resulting in the nonzero coefficient in $\boldsymbol{\gamma}_k = \mathbf{K}\boldsymbol{\alpha}$. Finally, the multiplication of $\boldsymbol{\gamma}_k$ by the effective dictionary, $\mathbf{D}_{(k)}$, produces the desired ML-SC signal.
4. Sparse dictionaries: Recall the need of previous works to consider intermediate sparse dictionaries. Such a construction can be understood only as a particular case of this model, where the obtained representations are indeed sparse (due to the sparse atoms) but not because nontrivial orthogonality was enforced between the deepest representation and the intermediate dictionaries. This way, the matrix $\mathbf{\Phi}^{\Lambda_k}$ becomes the zero matrix almost surely, and therefore, the signal lies in a subspace of dimension $s_k$, and the intermediate constraints are passive. One might think that in this case there is no extra advantage in the multi-layer model over the single-layer one. However, the matrix $\mathbf{\Phi}$ *does* contain information on $\boldsymbol{\gamma}_k$, and therefore it is expected to improve the recovery of the support (as in the correcting-support version of the projection algorithm). This can be understood by enforcing the constraint that every new nonzero that is to be added to $\boldsymbol{\gamma}_k$ should be orthogonal to the matrix $\mathbf{\Phi}$. Indeed, this will be exploited by the algorithm presented in section 7, and its benefits will become a lot clearer then. As we see, the matrix $\mathbf{\Phi}$ has two functions: aiding the detection of the true support, for which one should employ the whole matrix $\mathbf{\Phi}$, and estimating the values in $\boldsymbol{\gamma}_{\Lambda_k}$, for which the submatrix $\mathbf{\Phi}^{\Lambda_k}$ is the one of interest.
5. Random dictionaries: In the other extreme, when the dictionaries are fully dense and sampled from a continuous distribution, the rank of $\mathbf{\Phi}^{\Lambda_k}$ is equal to $\sum_{i=1}^{k-1} \ell_i$ with probability 1. Therefore, there are $s_k - \sum_{i=1}^{k-1} \ell_i$ degrees of freedom in choosing $\boldsymbol{\gamma}_k$, implying that the signal dimension is significantly reduced.
6. Recovery error: Projecting the signal on a smaller dimensional space reduces the recovery error, and therefore, the ML-SC model is expected to give a significant improvement over the single-layer model. We will elaborate on this matter later, but we anticipate that this error is proportional to the degrees of freedom, enabling a significant improvement in the ML-SC model.
7. A Holistic alternative: The new interpretation points to the fact that the various representations in all the layers should be estimated jointly as opposed to (relatively) independently or sequentially. This will motivate the derivation of our Holistic Pursuit, to be presented in section 7.

## 5. Uniqueness revisited.

The new analysis perspective motivates us to derive a new uniqueness theorem. The following result reflects the underlying benefits of the ML-SC model, exemplifying the gain one can obtain in return for more constrained assumptions. In the proof below we combine the spark—a synthesis characterization [4, 11]—with the union of subspaces interpretation [22, 25]. In addition, we will require the dictionaries to be in *general position*, as in [25], and we defer the precise definition of this characterization to Appendix A.

Theorem 5.1. *Consider an ML-SC signal* $\mathbf{x}$ *and a set of dictionaries* $\{\mathbf{D}_i\}_{i=1}^k$ *in general position. If there exists a set of representations* $\{\boldsymbol{\gamma}_i\}_{i=1}^k$ *satisfying*

$$s_k \leq \frac{\sigma(\mathbf{D}_{(k)}) - 1}{2} + r, \tag{5.1}$$

*where* $r = \operatorname{rank}\{\boldsymbol{\Phi}^{\Lambda_k}\}$ *and* $s_k$ *is number of nonzero coefficients in the deepest layer, then this set is the* unique *ML-SC representation for* $\mathbf{x}$ *such that its deepest layer has no more than* $s_k$ *nonzeros and the rank of the corresponding* $\boldsymbol{\Phi}^{\Lambda_k}$ *is no greater than* $r$.

Before moving forward, a comment is needed. The traditional uniqueness claims in [27, 37] provide uniqueness guarantees only when $s_k \leq \frac{\sigma(\mathbf{D}_{(k)})-1}{2}$, since such results use only the synthesis-type interpretation. Now, the above result efficiently leverages the analysis prior imposed on $\boldsymbol{\gamma}_k$, resulting in less restrictive conditions for uniqueness. For the sake of brevity, we defer the proof to Appendix A.

Finally, note that the above guarantees exclude dictionaries that are not in general position. In this way, some relevant and practical dictionaries, such as certain wavelet frames and total variation, are not covered by the above theorem. We believe that this result can in fact be extended to consider linear dependencies and dictionaries not in general position, and this might be studied in detail in future work.

## 6. The Oracle estimator.

The above result begins to show the benefit of considering all representations simultaneously. In this section, we aim to quantify this precisely by analyzing the performance of the Oracle estimator: suppose one knows the true supports across all layers; what is then the optimal (Oracle) estimator for the all representations? The answer to this question is of great importance since the Oracle estimator is the cornerstone in every pursuit, and it provides an idealistic understanding of the capabilities of a given algorithm. In order to provide a complete picture, we first analyze the Oracle estimators for the layerwise and projection approaches and then proceed to analyze the holistic alternative proposed in this work. We note that the previous work [27, 37] on multi-layer sparse models has only considered bounded noise assumptions, adopting a worst-case point of view. Not only does this lead to very loose bounds, but it also blurs the real connection between the model features and the error these induce. Thus, we present a novel analysis of the Oracle estimator performance for all approaches under stochastic noise assumptions.

Let us start by recalling the average performance bounds for the general single-layer sparse model. Consider a signal $\mathbf{y} = \mathbf{x} + \mathbf{e}$, where $\mathbf{x} = \mathbf{D}\boldsymbol{\gamma}$, $\mathbf{e} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ is a Gaussian noise, the representation true support is $\Lambda$ with cardinality $s$, and $\delta_s^{\mathbf{D}}$ is the restricted isometry property (RIP) constant of the dictionary $\mathbf{D}$ [6]. Then, the Oracle estimator is obtained via simple least-squares, $\hat{\boldsymbol{\gamma}}^\Lambda = \mathbf{D}^{\Lambda^\dagger}\mathbf{y}$. The above estimate can be equivalently expressed as $\hat{\boldsymbol{\gamma}}^\Lambda = \boldsymbol{\gamma}^\Lambda + \tilde{\mathbf{e}} = \boldsymbol{\gamma}^\Lambda + \sigma(\mathbf{D}^{\Lambda^T}\mathbf{D}^\Lambda)^{-1/2}\mathbf{z}$, where $\tilde{\mathbf{e}} \sim \mathcal{N}(\mathbf{0}, \sigma^2(\mathbf{D}^{\Lambda^T}\mathbf{D}^\Lambda)^{-1})$, and $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$. Therefore, in expectation, one has that

$$\mathbb{E}\,\|\boldsymbol{\gamma} - \hat{\boldsymbol{\gamma}}\|_2^2 = \sigma^2 \operatorname{Trace}\left(\mathbf{D}^{\Lambda^T}\mathbf{D}^\Lambda\right)^{-1}, \tag{6.1}$$

and the bounds on the recovery error can be shown to be (see [1, 5])

$$\frac{\sigma^2 s}{1+\delta_s^{\mathbf{D}}} \leq \mathbb{E}\left\|\boldsymbol{\gamma}-\hat{\boldsymbol{\gamma}}\right\|_2^2 \leq \frac{\sigma^2 s}{1-\delta_s^{\mathbf{D}}}. \tag{6.2}$$

As we see, the recovery error of the Oracle estimator is proportional to the representation cardinality.

With these tools we now analyze the Oracle estimator performance for the different approaches for the multi-layer model. Consider a signal $\mathbf{y}=\mathbf{x}+\mathbf{e}$, but now $\mathbf{x}=\mathbf{D}_1\boldsymbol{\gamma}_1=\cdots=\mathbf{D}_{(k)}\boldsymbol{\gamma}_k$ is an ML-SC signal, the true support of layer $i$ is $\Lambda_i$ with cardinality $s_i$, and we denote by $\delta_{s_i}^{\mathbf{D}_i}$ the RIP constant of the dictionary $\mathbf{D}_i$ for cardinality $s_i$.

In addition, we will need to define an appropriate extension of the RIP, which we named Subset RIP, for those cases where not only the representation is sparse, but the signal is also sparse. Recall that if a matrix satisfies the RIP, the constant $\delta_s^{\mathbf{D}}$ provides a bound to the deviation of the singular values of every subdictionary of $s$ columns from 1. The proposed extension provides a similar interpretation, but for subdictionaries obtained by removing not just columns, corresponding to the support of a certain $\boldsymbol{\gamma}_i$, but also rows, corresponding to the support of $\boldsymbol{\gamma}_{i-1}$.

For normalized Gaussian random matrices, the singular values of a subdictionary are indeed expected to be centered at 1. If now certain $s$ out of $n$ rows are removed, one would expect the singular values of those subdictionaries to be centered around $\frac{s}{n}$. Note that most matrices that satisfy the RIP (like sub-Gaussian matrices) would also satisfy the Subset RIP definition. We define the Subset RIP formally as follows.

Definition 6.1. *For any subset of $s_R$ rows, $\Lambda_R$, and any subset of $s_C$ columns, $\Lambda_C$, the matrix $\mathbf{D}\in\mathbb{R}^{n\times m}$ satisfies the Subset RIP with constant $\delta_{s_R,s_C}^{\mathbf{D}}$ if this is the minimum constant so that*

$$\left(\frac{s_R}{n}-\delta_{s_R,s_C}^{\mathbf{D}}\right)\|\mathbf{e}\|_2^2 \leq \left\|\mathbf{D}^{\Lambda_R,\Lambda_C}\mathbf{e}\right\|_2^2 \leq \left(\frac{s_R}{n}+\delta_{s_R,s_C}^{\mathbf{D}}\right)\|\mathbf{e}\|_2^2 \tag{6.3}$$

*holds for all vectors* $\mathbf{e}$.

The Subset RIP will become useful in the Oracle estimator analysis that we are about to present.

### 6.1. Oracle performance for Layerwise Pursuit.

In the layerwise approach, we start the recovery process by estimating $\boldsymbol{\gamma}_1$ and obtaining $\hat{\boldsymbol{\gamma}}_1$. Then, we use $\hat{\boldsymbol{\gamma}}_1^{\Lambda_1}$ to estimate $\boldsymbol{\gamma}_2$, and so on to the deepest layer. In an oracle setting, the estimation of each layer is performed using the corresponding oracle estimators for each layer. One should wonder if the oracle estimation of $\boldsymbol{\gamma}_2$ should be carried out using the subdictionary $\mathbf{D}_2^{\Lambda_2}$ or the more restricted version $\mathbf{D}_2^{\Lambda_1,\Lambda_2}$. As we will show towards the end of this section, the former is preferred, as the latter leads to a biased error.

In this manner, we employ the zero extension of the previous layer, $\hat{\boldsymbol{\gamma}}_{i-1}$, in order to estimate $\boldsymbol{\gamma}_i$, which results in

$$\begin{aligned}\hat{\boldsymbol{\gamma}}_i^{\Lambda_i} &= \mathbf{D}_i^{\Lambda_i\dagger}\hat{\boldsymbol{\gamma}}_{i-1} = \left(\mathbf{D}_i^{\Lambda_i T}\mathbf{D}_i^{\Lambda_i}\right)^{-1}\mathbf{D}_i^{\Lambda_{i-1},\Lambda_i T}\hat{\boldsymbol{\gamma}}_{i-1}^{\Lambda_{i-1}} \\ &= \left(\mathbf{D}_i^{\Lambda_i T}\mathbf{D}_i^{\Lambda_i}\right)^{-1}\mathbf{D}_i^{\Lambda_{i-1},\Lambda_i T}\cdots\left(\mathbf{D}_1^{\Lambda_1 T}\mathbf{D}_1^{\Lambda_1}\right)^{-1}\mathbf{D}_1^{\Lambda_1 T}\mathbf{y} = \mathcal{U}_{(i,1)}\mathbf{y}.\end{aligned} \tag{6.4}$$

In this form, one can concisely express, for every $i$th layer, $\hat{\boldsymbol{\gamma}}_i^{\Lambda_i} = \boldsymbol{\gamma}_i^{\Lambda_i} + \mathcal{U}_{(i,1)}\mathbf{e} = \boldsymbol{\gamma}_i + \sigma\mathbf{W}_i\mathbf{z}$, where $\mathbf{W}_i = (\mathcal{U}_{(i,1)}\mathcal{U}_{(i,1)}^T)^{1/2}$, and as before $\mathbf{z} \sim \mathcal{N}(0,\mathbf{I})$. As we prove in Appendix B.1, the recovery error bounds are

$$\sigma^2 s_i \frac{1}{1+\delta_{s_1}^{\mathbf{D}_1}} \prod_{j=2}^{i} \frac{\frac{s_{j-1}}{n_{j-1}} - \delta_{s_{j-1},s_j}^{\mathbf{D}_j}}{(1+\delta_{s_j}^{\mathbf{D}_j})^2} \leq \mathbb{E}\,\|\boldsymbol{\gamma}_i - \hat{\boldsymbol{\gamma}}_i\|_2^2 \leq \sigma^2 s_i \frac{1}{1-\delta_{s_1}^{\mathbf{D}_1}} \prod_{j=2}^{i} \frac{\frac{s_{j-1}}{n_{j-1}} + \delta_{s_{j-1},s_j}^{\mathbf{D}_j}}{(1-\delta_{s_j}^{\mathbf{D}_j})^2}. \tag{6.5}$$

These bounds show that, for a given layer, its Oracle estimator error depends linearly on its sparsity level; i.e., it is proportional to $\sigma^2 s_i$, just like the single-layer case. It is worthwhile to note that the above constants at each layer depend on the particular setting of the model parameters, such as the ratio of nonzero elements in other layers. In the nonoracle case (where the supports are unknown) as shown in [27, 37], these bounds become looser with the depth of the network.

### 6.2. Oracle performance for the Projection Pursuit.

In the projection approach, we start the recovery by using the effective model to estimate the deepest sparse representation, $\boldsymbol{\gamma}_k$: $\hat{\boldsymbol{\gamma}}_k = \mathbf{D}_{(k)}^{\dagger}\mathbf{y}$, for which we can provide the single-layer error bounds from (6.2):

$$\frac{\sigma^2 s_k}{1+\delta_{s_k}^{\mathbf{D}_{(k)}}} \leq \mathbb{E}\,\|\boldsymbol{\gamma}_k - \hat{\boldsymbol{\gamma}}_k\|_2^2 \leq \frac{\sigma^2 s_k}{1-\delta_{s_k}^{\mathbf{D}_{(k)}}}. \tag{6.6}$$

In the next steps, we use the known midlayer supports to backtrack $\hat{\boldsymbol{\gamma}}_k$ to shallower representations:

$$\hat{\boldsymbol{\gamma}}_i = \mathbf{D}_{i+1}^{\Lambda_i,\Lambda_{i+1}}\hat{\boldsymbol{\gamma}}_{i+1} = \mathbf{D}_{i+1}^{\Lambda_i,\Lambda_{i+1}} \cdots \mathbf{D}_{k}^{\Lambda_{k-1},\Lambda_k}\hat{\boldsymbol{\gamma}}_k. \tag{6.7}$$

As we prove in Appendix B.2, this process results in the following midlayer error bounds:

$$\sigma^2 s_k \frac{c_{k_1}}{1+\delta_{s_k}^{\mathbf{D}_{(k)}}} \leq \mathbb{E}\,\|\boldsymbol{\gamma}_i - \hat{\boldsymbol{\gamma}}_i\|_2^2 \leq \sigma^2 s_k \frac{c_{k_2}}{1-\delta_{s_k}^{\mathbf{D}_{(k)}}}, \tag{6.8}$$

where $c_{k_1} = \prod_{j=i+1}^{k}(\frac{s_{j-1}}{m_{j-1}} - \delta_{s_{j-1},s_j}^{\mathbf{D}_j})$ and $c_{k_2} = \prod_{j=i+1}^{k}(\frac{s_{j-1}}{m_{j-1}} + \delta_{s_{j-1},s_j}^{\mathbf{D}_j})$.

Interestingly, we can see that the recovery error of the intermediate layers no longer depends on their cardinality but rather on that of the deepest layer—which typically results in a lower error. Just as in the layerwise case, the particular constants depend on the model parameters. However, in this case, these values are given by the multiplication of terms from the deeper to shallower layers, as opposed to from shallower to deeper. In the nonoracle case, this fact causes the error to grow accordingly [37].

One might think that the deepest layer estimator in the projection method must be optimal, as it results from a global least-squares. We will show that this is in fact not the case, because the least-squares estimator is only optimal when no additional information can be exploited. As we present next, there is an alternative for estimating $\boldsymbol{\gamma}_k$ which explicitly exploits the additional supports information and leads to a significantly better error.

Before proceeding, we would like to demonstrate that employing the intermediate dictionaries, in their row-restricted versions, introduces a bias in the oracle estimators. For the sake of simplicity, consider a two-layer model, and separate the effective dictionary into two parts:

$$\mathbf{y} = \mathbf{D}_{(2)}^{\Lambda_2}\boldsymbol{\gamma}_2^{\Lambda_2} + \mathbf{e} = \mathbf{D}_1^{\Lambda_1}\mathbf{D}_2^{\Lambda_1,\Lambda_2}\boldsymbol{\gamma}_2^{\Lambda_2} + \mathbf{D}_1^{\Lambda_1^c}\mathbf{D}_2^{\Lambda_1^c,\Lambda_2}\boldsymbol{\gamma}_2^{\Lambda_2} + \mathbf{e}. \tag{6.9}$$

The above expression clearly shows that employing only $\mathbf{D}_1^{\Lambda_1}\mathbf{D}_2^{\Lambda_1,\Lambda_2}$ to estimate $\boldsymbol{\gamma}_2$ simply ignores the second term, leading to a bias in the estimate. In this simple two-layer example, this bias can be expressed as

$$\mathbf{b}(\hat{\boldsymbol{\gamma}}_2) = \mathbb{E}\left[\hat{\boldsymbol{\gamma}}_2\right] - \boldsymbol{\gamma}_2 = \left(\mathbf{D}_1^{\Lambda_1}\mathbf{D}_2^{\Lambda_1,\Lambda_2}\right)^{\dagger}\mathbf{D}_1^{\Lambda_1^c}\mathbf{D}_2^{\Lambda_1^c,\Lambda_2}\boldsymbol{\gamma}_2^{\Lambda_2}, \tag{6.10}$$

and generally, for $k$ layers, the bias becomes

$$\mathbf{b}(\hat{\boldsymbol{\gamma}}_k) = \left(\mathbf{D}_1^{\Lambda_1}\mathbf{D}_2^{\Lambda_1,\Lambda_2}\cdots\mathbf{D}_k^{\Lambda_{k-1},\Lambda_k}\right)^{\dagger}\left(\mathbf{D}_{(k)}^{\Lambda_k} - \mathbf{D}_1^{\Lambda_1}\mathbf{D}_2^{\Lambda_1,\Lambda_2}\cdots\mathbf{D}_k^{\Lambda_{k-1},\Lambda_k}\right)\boldsymbol{\gamma}_2^{\Lambda_2}. \tag{6.11}$$

### 6.3. Oracle performance for a Holistic Pursuit.

We have seen that the layerwise and the projection approaches cannot be optimal as they both ignore some information. But how can one use all the model information simultaneously? In this section we provide the answer to this question when the true supports are known. The solution is based on the synthesis-analysis understanding we have presented above and the corresponding holistic approach. Analyzing the recovery error of this strategy will result in a significantly improved result that would confirm that the whole is more than merely the sum of its parts.

The synthesis-analysis dual interpretation provides a way to use the midlayer supports when estimating the last layer. Indeed, as we have shown, $\boldsymbol{\gamma}_k$ does not lie in $\mathbb{R}^{s_k}$ but rather in the kernel of $\boldsymbol{\Phi}^{\Lambda_k}$, which we define in (4.1). Therefore, the optimal Oracle estimator is

$$\hat{\boldsymbol{\gamma}}_k^{\Lambda_k} = \underset{\boldsymbol{\gamma}_k^{\Lambda_k}}{\operatorname{argmin}} \left\|\mathbf{y} - \mathbf{D}_{(k)}^{\Lambda_k}\boldsymbol{\gamma}_k^{\Lambda_k}\right\|_2 \quad \text{s. t.} \;\; \boldsymbol{\gamma}_k^{\Lambda_k} \in \ker\{\boldsymbol{\Phi}^{\Lambda_k}\}. \tag{6.12}$$

While this problem might look somewhat challenging, it admits a surprisingly simple solution. Let us define $\mathbf{K}$ to be an orthogonal matrix that spans the null space of $\boldsymbol{\Phi}^{\Lambda_k}$. Such a matrix can be obtained by computing the singular-value decomposition (SVD) of $\boldsymbol{\Phi}^{\Lambda_k}$ and choosing the $s_k - r$ right-singular vectors corresponding to the zero singular values, where $r$ is the rank of $\boldsymbol{\Phi}^{\Lambda_k}$. Using the matrix $\mathbf{K}$, we might rewrite the objective simply as

$$\hat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \left\|\mathbf{y} - \mathbf{D}_{(k)}^{\Lambda_k}\mathbf{K}\boldsymbol{\alpha}\right\|_2, \tag{6.13}$$

where $\boldsymbol{\alpha}$ is of length $s_k - r$, and then choose $\hat{\boldsymbol{\gamma}}_k^{\Lambda_k} = \mathbf{K}\hat{\boldsymbol{\alpha}}$. The corresponding Oracle estimator for this problem (once more, given the support $\Lambda_k$) is given by

$$\hat{\boldsymbol{\gamma}}_k^{\Lambda_k} = \mathbf{K}\hat{\boldsymbol{\alpha}} = \mathbf{K}\left(\mathbf{D}_{(k)}^{\Lambda_k}\mathbf{K}\right)^{\dagger}\mathbf{y}. \tag{6.14}$$

Since the columns of $\mathbf{K}$ are orthonormal, the error in $\hat{\boldsymbol{\gamma}}_k^{\Lambda_k}$ is simple to analyze:

$$\mathbb{E}\left\|\boldsymbol{\gamma}_k - \hat{\boldsymbol{\gamma}}_k\right\|_2^2 = \mathbb{E}\left\|\mathbf{K}(\boldsymbol{\alpha} - \hat{\boldsymbol{\alpha}})\right\|_2^2 = \mathbb{E}\left\|\boldsymbol{\alpha} - \hat{\boldsymbol{\alpha}}\right\|_2^2. \tag{6.15}$$

The error in $\hat{\boldsymbol{\alpha}}$ will be the single-layer Oracle error employed above in (6.2), where the dictionary is given by $\mathbf{D}_{(k)}^{\Lambda_k}\mathbf{K}$. Using again the orthonormality of $\mathbf{K}$, one can bound the singular values of $\mathbf{D}_{(k)}^{\Lambda_k}\mathbf{K}$ employing the RIP of $\mathbf{D}_{(k)}^{\Lambda_k}$,

$$(6.16)\qquad (1-\delta_{s_k}^{\mathbf{D}_{(k)}})\,\|\boldsymbol{\alpha}\|_2^2 =(1-\delta_{s_k}^{\mathbf{D}_{(k)}})\,\|\mathbf{K}\boldsymbol{\alpha}\|_2^2 \leq \left\|\mathbf{D}_{(k)}^{\Lambda_k}\mathbf{K}\boldsymbol{\alpha}\right\|_2^2$$

$$(6.17)\qquad \leq(1+\delta_{s_k}^{\mathbf{D}_{(k)}})\,\|\mathbf{K}\boldsymbol{\alpha}\|_2^2 = (1+\delta_{s_k}^{\mathbf{D}_{(k)}})\,\|\boldsymbol{\alpha}\|_2^2 .$$

This way, the recovery error bounds for the Holistic Oracle estimator are[2]

$$(6.18)\qquad \sigma^2\,(s_k-r)\,\frac{c_{k_1}}{1+\delta_{s_k}^{\mathbf{D}_{(k)}}} \leq \mathbb{E}\,\|\boldsymbol{\gamma}_i-\hat{\boldsymbol{\gamma}}_i\|_2^2 \leq \sigma^2\,(s_k-r)\,\frac{c_{k_2}}{1-\delta_{s_k}^{\mathbf{D}_{(k)}}},$$

where $c_{k_1}$ and $c_{k_2}$ are the same as the constants defined for the Projection Oracle case.

As can be seen, the error is now proportional to the dimension of the kernel space of $\mathbf{\Phi}^{\Lambda_k}$: $s_k-r$. This reveals the significant advantage of this multi-layer sparse construction. When employing this estimator, the recovery error decreases by a factor of $\frac{s_k-r}{s_k}$ compared to previous approaches. In addition, this demonstrates the crucial role of the matrix $\mathbf{\Phi}^{\Lambda_k}$, as it determines to what extent the holistic version is better than the projection approach. For example, when the intermediate dictionaries are sparse and the nonzero coefficients of $\boldsymbol{\gamma}_k$ are randomly chosen (as done in [37]), $\mathbf{\Phi}^{\Lambda_k}$ is the zero matrix with high probability and its rank is zero. In such a case, the performance of the Holistic Oracle estimator is the same as that of Projection Oracle estimators. However, when the dictionaries are dense(r) and random, $\mathbf{\Phi}^{\Lambda_k}$ has a full row rank, and thus $r=\sum_{i=1}^{k-1}\ell_i$, resulting in a significantly performance improvement.

## 7. The Holistic Pursuit.

Given the understanding of the benefits of exploiting the mid-layer cosupports, in this section we undertake the following perhaps more interesting question: How can we design a pursuit algorithm that can implement these ideas in practice? Note that the estimation of the intermediate layers' zeros, or cosupports, should be done with care, as their wrong estimation would cause a biased error by possibly projecting onto the wrong subspace.

In what follows we present an algorithm to estimate the intermediate cosupports and the corresponding matrix $\mathbf{K}$, resulting in a significant performance improvement. We call this approach the "Holistic Pursuit" as it gives the solution for the whole system simultaneously. For simplicity, we shall assume that one has access to the knowledge of the number of cosupport elements at each layer, $\ell_i$. In addition, one could of course devise strategies in order to estimate these values if they are unknown, and we will comment on this towards the end of this section. We depict this algorithm in Algorithm 7.1.

### 7.1. The Holistic Pursuit algorithm.

The Holistic Pursuit consists of two main steps, which are to be iterated. In the first step, one aims to estimate the sparse innermost $\boldsymbol{\gamma}_k$ given the intermediate layers' cosupport elements that have been found in previous iterations—initialized as the empty set. This estimation amounts to solving a constrained sparse coding

[2] We omit the Holistic Oracle estimator performance proof since it is similar to the Projection recovery error bounds proof, which appears in Appendix B.2.

**Algorithm 7.1.** The Holistic Pursuit.

```
Input
    • Signal $\mathbf{y}$ and dictionaries $\{\mathbf{D}_i\}_{i=1}^{k}$.
    • ML-SC parameters: the midlayer cosparsity levels $\{\ell_i\}_{i=1}^{k-1}$ and the deepest layer sparsity $s_k$.
    • The midlayers' minimum absolute value $\{\gamma_i^{\min}\}_{i=1}^{k-1}$.
Output
    • Set of representations $\{\hat{\gamma}_i\}_{i=1}^{k}$.
Initialization
    • Initialize the midlayer cosupports: $\hat{\Lambda}_i^c = \emptyset \ \forall 1 \leq i \leq k-1$.
    • Initialize the matrix that spans $\gamma_k$'s subspace: $\mathbf{K} = \mathbf{I}_{m_k \times m_k}$.
Holistic iterations: perform the following steps for $\ell^{tot} = \sum_{i=1}^{k-1} \ell_i$ times in order to estimate $\mathbf{K}$:
    1. Estimate $\hat{\gamma}_k$ using the current estimation of $\mathbf{K}$ with (7.4):
       (7.1)   $\min_{\boldsymbol{\alpha}, \gamma_k} \frac{1}{2} \left\| \mathbf{y} - \mathbf{D}_{(k)} \mathbf{K} \boldsymbol{\alpha} \right\|_2^2 + \eta \left\| \gamma_k \right\|_1 \text{ s.t. } \gamma_k = \mathbf{K} \boldsymbol{\alpha}.$
    2. Select a layer $g$ on which to add a cosupport element, using (7.6) (see below).
    3. Find a new element $j$ which is the minimum absolute value in layer $g$:
       $j \leftarrow \text{argmin} \left| \mathbf{D}_{(g+1,k)}^{\hat{\Lambda}_g} \hat{\gamma}_k \right|$.
    4. Update the cosupport estimation $\hat{\Lambda}_g^c$, and the corresponding $\mathbf{K}$:
       $\mathbf{K} \leftarrow \bigcup_{i=1}^{k-1} \ker\{\mathbf{D}_{(i+1,k)}^{\hat{\Lambda}_i^c}\}$.
Holistic final step: Use the found subspace to estimate $\gamma_k$:
    1. Estimate $\hat{\gamma}_k$ using the current $\mathbf{K}$, with (7.4).
    2. Propagate $\hat{\gamma}_k$ to the midlayer representations:
       (7.2)   $\hat{\gamma}_i \leftarrow \mathbf{D}_{(i+1,k)} \hat{\gamma}_k \ \forall \ 1 \leq i \leq k-1.$
```

problem, in essence searching for a sparse $\gamma_k$ such that it is orthogonal to the rows of the previously found midlayer cosupport. In other words, we are interested in solving the following subproblem:

$$\min_{\gamma_k} \frac{1}{2} \left\| \mathbf{y} - \mathbf{D}_{(k)} \gamma_k \right\|_2^2 + \eta \left\| \gamma_k \right\|_1 \quad \text{s.t. } \ \gamma_k \in \ker\{\boldsymbol{\Phi}\}. \tag{7.3}$$

Note that this problem is almost the same as the one solved by the projection approach from [37], except in the latter there is no constraint as to the subspace on which $\gamma_k$ should live. Here, we are explicitly requiring that $\gamma_k \in \ker\{\boldsymbol{\Phi}\}$, therefore incorporating more information to help find $\gamma_k$. Such information is still relevant in cases where the dictionaries (from layer 2 to $k$) are sparse, providing an advantage over the Layered and the Projection approaches in those cases as well.

The constrained problem in (7.3) is convex and can be solved with a variety of methods. We propose to address it by employing the Alternating Direction Method of Multipliers (ADMM) [2], for which we introduce a variable split. In addition, we enforce the subspace constraint by means of the matrix $\mathbf{K}$, which spans the $\ker\{\mathbf{\Phi}\}$, resulting in

$$\min_{\boldsymbol{\alpha},\boldsymbol{\gamma}_k} \frac{1}{2}\left\|\mathbf{y}-\mathbf{D}_{(k)}\mathbf{K}\boldsymbol{\alpha}\right\|_2^2+\eta\left\|\boldsymbol{\gamma}_k\right\|_1 \text{ s.t. } \boldsymbol{\gamma}_k=\mathbf{K}\boldsymbol{\alpha}. \tag{7.4}$$

In order to minimize this new constrained problem, we construct the (normalized) augmented Lagrangian penalty by introducing the dual variable $\mathbf{u}$,

$$\min_{\boldsymbol{\alpha},\boldsymbol{\gamma}_k,\mathbf{u}} \frac{1}{2}\left\|\mathbf{y}-\mathbf{D}_{(k)}\mathbf{K}\boldsymbol{\alpha}\right\|_2^2+\eta\left\|\boldsymbol{\gamma}_k\right\|_1+\frac{\rho}{2}\left\|\mathbf{K}\boldsymbol{\alpha}-\boldsymbol{\gamma}_k+\mathbf{u}\right\|_2^2. \tag{7.5}$$

This can now be minimized iteratively by alternating minimization with respect to $\boldsymbol{\alpha}$, $\boldsymbol{\gamma}_k$, and the dual variable $\mathbf{u}$, and we detail this process in Algorithm 7.2. As can be seen, this minimization reduces to the iteration of simple operations: the problem in line (2) is solved in closed form by an entrywise thresholding operator, while the step in line (3) reduces to a least-squares estimate. Moreover, the matrix that needs to be inverted for this step[3] can be precomputed in advance, saving computations.

**Algorithm 7.2.** ADMM for constraint Lasso.

```
1: while not converged do
2:     $\boldsymbol{\gamma}_k \leftarrow \text{argmin}_{\boldsymbol{\gamma}}\ \eta\|\boldsymbol{\gamma}\|_1+\frac{\rho}{2}\|\mathbf{K}\boldsymbol{\alpha}-\boldsymbol{\gamma}+\mathbf{u}\|_2^2$ ;
3:     $\boldsymbol{\alpha} \leftarrow \text{argmin}_{\boldsymbol{\alpha}}\frac{1}{2}\left\|\mathbf{y}-\mathbf{D}_{(k)}\mathbf{K}\boldsymbol{\alpha}\right\|_2^2+\frac{\rho}{2}\|\mathbf{K}\boldsymbol{\alpha}-\boldsymbol{\gamma}+\mathbf{u}\|_2^2$ ;
4:     $\mathbf{u} \leftarrow \mathbf{u}+\rho(\mathbf{K}\boldsymbol{\alpha}-\boldsymbol{\gamma}_k)$ ;
```

The second stage in the Holistic Pursuit is the estimation of the cosupport from the intermediate layers, using the obtained $\hat{\gamma}_k$ from the previous step. A number of options are available for such a process. One could attempt, for instance, to estimate the entire $\ell^{tot}$ cosupport elements at once. However, this is prone to yielding mistakes that would cause the projection of $\hat{\gamma}_k$ onto an incorrect subspace. For this reason, we limit the search of these elements to one cosupport element at a time. As there are multiple layers to choose from, the chosen layer should maximize the chances of obtaining a correct element of the cosupport. In this spirit, one prefers a layer with a high value of $\boldsymbol{\gamma}_i^{\min}$ and with many cosupport elements yet to be found, $\ell_i-|\hat{\Lambda}_i^c|$. We propose choosing the layer by

$$g \leftarrow \underset{i:\ |\hat{\Lambda}_i^c|<\ell_i}{\text{argmax}}\ \boldsymbol{\gamma}_i^{\min}\left(1+\frac{1+\mu_R^i\left(\ell_i-\left|\hat{\Lambda}_i^c\right|-1\right)}{\sqrt{\ell_i-\left|\hat{\Lambda}_i^c\right|}}\right)^{-1}, \tag{7.6}$$

where $\mu_R^i$ is the *row mutual coherence* of dictionary $\mathbf{D}_{(i+1,k)}$:

$$\mu_R \triangleq \max_{p\neq j}\left|\mathbf{d}_p\mathbf{d}_j^T\right|. \tag{7.7}$$

[3] This matrix is given by $\mathbf{H}=\left((\mathbf{D}_{(k)}\mathbf{K})^T\mathbf{D}_{(k)}\mathbf{K}\right)^{\dagger}$.

The choice of this particular expression will become clear later in the analysis of the algorithm. After choosing the layer $g$, the entry to be updated is chosen as the minimum absolute value of the inner products between $\boldsymbol{\gamma}_k$ and the rows of $\mathbf{D}_{(g,k)}$.

Once a new element has been found and added to the cosupport, one must update the matrix $\mathbf{K}$—spanning a subspace that now includes the new added row—which is to be used in the next iteration in the estimation of $\boldsymbol{\gamma}_k$. This way, the estimation of the innermost representation becomes more and more accurate as the iterations proceed, and the algorithm continues until all the intermediate layers' cosupport elements are found. Finally, once all cosupport rows in $\boldsymbol{\Phi}$ have been identified, one estimates $\boldsymbol{\gamma}_k$ one last time by solving the problem in (7.4) and then computes the intermediate representations as $\forall\, 1 \leq i \leq k-1,\ \ \hat{\gamma}_i \leftarrow \mathbf{D}_{(i+1,k)}\hat{\gamma}_k$.

Before moving to the analysis of this algorithm, we would like to stress that the Holistic Pursuit suggests a general framework for the search of sparse representations under this dual synthesis-analysis model. Indeed, while we have made each iteration specific, the general components of the algorithm can be changed to either more or less accurate steps. For example, one could estimate $\boldsymbol{\gamma}_k$ in a greedy way instead of employing a Basis-Pursuit formulation. Alternatively, one might prefer to estimate blocks of the cosupport elements of layer $g$ in each iteration, or even re-evaluate part of the cosupport found and possibly replace elements, somewhat along the lines of the CoSaMP [26] and Subspace Pursuit [9] algorithms.

As stated in the beginning of this section, the presented algorithm assumes that the set of minimal entries, $\{\boldsymbol{\gamma}_i^{\min}\}_{i=1}^{k-1}$, and the set of cosparsity levels, $\{\ell_i\}_{i=1}^{k-1}$, are assumed given. With small modifications, however, one could adapt the Holistic Pursuit to handle those cases in which this information is not available. For example, if the set of minimal entries in the representations is unknown, one could remove $\boldsymbol{\gamma}_i^{\min}$ from the rule that determines which layer to address next—essentially assuming that all layers use the same minimal value. On the other hand, if the set of the intermediate layer cosparsity levels, $\{\ell_i\}_{i=1}^{k-1}$, is not given, one could search for the minimum absolute value across all layers, or even search for the entry that has the smallest overall effect on the residual. This process should be iterated until some condition or threshold is met, such as having reached a total number of cosupport elements, *global cosparsity* level $\ell^{tot}$, or a residual noise level.

**7.2. Theoretical analysis.** In what follows, we demonstrate and theoretically analyze how the Holistic Pursuit leverages the constraints in the model, providing better estimates than previous approaches. We divide our derivations according to the two steps of the algorithm: the synthesis-pursuit, and the analysis-pursuit.

Let us start by equivalently rewriting the objective function of the first step appearing in (7.3):

$$\tilde{\gamma}_k \leftarrow \underset{\boldsymbol{\gamma}_k}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{y} - \mathbf{D}_{(k)}\boldsymbol{\gamma}_k\|_2^2 + \eta\|\boldsymbol{\gamma}_k\|_1 \ \text{ s.t. } \boldsymbol{\Phi}\boldsymbol{\gamma}_k = \mathbf{0}. \tag{7.8}$$

This is an interesting constrained Lasso problem that has been recently studied in [19]. We bring here the performance guarantees provided in that work, while allowing ourselves to omit tedious details that deviate from our main message. These can be easily found in [19].

Lemma 7.1 (Corollary 1 in [19]). *Under mild assumptions on the dictionary $\mathbf{D}_{(k)} \in \mathbb{R}^{n \times m_k}$, on the constrained matrix $\mathbf{\Phi} \in \mathbb{R}^{\ell \times m_k}$, and on the true representation $\boldsymbol{\gamma}_k \in \mathbb{R}^{m_k}$ (see [19]), if $\eta = (4\sqrt{2}-2)\sigma\sqrt{\frac{\log m_k}{n}}$, $\boldsymbol{\gamma}_k$ obeys the constraint $\mathbf{\Phi}\boldsymbol{\gamma}_k = \mathbf{0}$, and $\mathbf{y} = \mathbf{D}_{(k)}\boldsymbol{\gamma}_k + \mathbf{e}$, where $\mathbf{e} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$ is a random noise vector, then with probability at least $1 - 2/m_k$,*

$$\|\tilde{\boldsymbol{\gamma}}_k - \boldsymbol{\gamma}_k\|_2^2 \leq \max\{s_k - \ell\ ,\ \ell\}\frac{64\sigma^2 \log m_k}{\kappa_L^2 n}, \tag{7.9}$$

*where $\kappa_L$ is a constant related to the dictionary and the constrained matrix.*

In comparison, solving the same problem in (7.8) but without the constraint,

$$\tilde{\boldsymbol{\gamma}}_k^{Unconst} \leftarrow \underset{\boldsymbol{\gamma}_k}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{y} - \mathbf{D}_{(k)}\boldsymbol{\gamma}_k\|_2^2 + \eta\|\boldsymbol{\gamma}_k\|_1 \tag{7.10}$$

with $\eta = 4\sigma\sqrt{\frac{\log m_k}{n}}$ leads to the following recovery error bound [19]:

$$\left\|\tilde{\boldsymbol{\gamma}}_k^{Unconst} - \boldsymbol{\gamma}_k\right\|_2^2 \leq s_k \frac{64\sigma^2 \log m_k}{\kappa_L^2 n}. \tag{7.11}$$

As can be observed, the additional intermediate layer information reduces the recovery error from being proportional to $s_k$ to being proportional[4] to $\max\{s_k - \ell\ ,\ \ell\}$. This is precisely the motivation behind the iterations in the Holistic Pursuit: the recovery error of $\boldsymbol{\gamma}_k$ is reduced at every iteration, leading to a higher probability to estimate a new cosupport element from the intermediate layers.

The next lemma analyzes the second step of the algorithm: the pursuit of a new cosupport element. This result is based on the derivations in [31]. However, we restrict the analysis to the probability of finding one true cosupport element, resulting in looser conditions. Its proof can be found in Appendix C.

Lemma 7.2. *Let $\hat{\boldsymbol{\gamma}}_k = \boldsymbol{\gamma}_k + \mathbf{e}$ be the estimation of the deepest layer, let $\hat{\Lambda}_i^c$ be the estimated cosupport in the ith layer, where $\sum_{i=1}^{k-1} |\hat{\Lambda}_i^c| = j - 1$, and let $\mu_R^i$ be the* row-wise *mutual-coherence defined in (7.7). If*

$$\|\mathbf{e}\|_2 \leq \max_{i:\left|\hat{\Lambda}_i^c\right| < \ell_i} \gamma_i^{\min} \left( 1 + \frac{1 + \mu_R^i \left( \ell_i - \left|\hat{\Lambda}_i^c\right| - 1 \right)}{\sqrt{\ell_i - \left|\hat{\Lambda}_i^c\right|}} \right)^{-1}, \tag{7.12}$$

*then the Holistic Pursuit algorithm succeeds in its jth iteration in recovering a new element from the midlayers' cosupport.*

We now combine both lemmas above in the form of an overall theorem, providing a theoretical guarantee for the Holistic Pursuit.

[4] We conjecture that a stronger claim could be formulated in terms of $s_k - \ell$, as opposed to $\max\{s_k - \ell, \ell\}$.

Theorem 7.3. *Consider an ML-SC signal* $\mathbf{x}$ *with intermediate layer cosparsity levels* $\{\ell_i\}_{i=1}^{k-1}$ *and a deepest layer sparsity of* $s_k$*, contaminated by random noise* $\mathbf{e} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$*, resulting in the observation* $\mathbf{y} = \mathbf{x} + \mathbf{e}$*. Under the mild assumptions appearing in Corollary* 1 *in* [19]*, and if for every* $j$*th iteration*

$$\sqrt{\max\{s_k - j\ ,\ j\}}\frac{8\sigma}{\kappa_L}\sqrt{\frac{\log m_k}{n}} \leq \max_{i:\left|\hat{\Lambda}_i^c\right|<\ell_i} \gamma_i^{\min}\left(1 + \frac{1+\mu_R^i\left(\ell_i - \left|\hat{\Lambda}_i^c\right| - 1\right)}{\sqrt{\ell_i - \left|\hat{\Lambda}_i^c\right|}}\right)^{-1} \tag{7.13}$$

*holds, then, with probability exceeding* $(1 - 2/m_k)^{\ell^{tot}}$*, where* $\ell^{tot} = \sum_{i=1}^{k-1}\ell_i$*, the Holistic Pursuit succeeds in recovering the support of all sparse representations* $\boldsymbol{\gamma}_i$*, and the recovery error is bounded by*

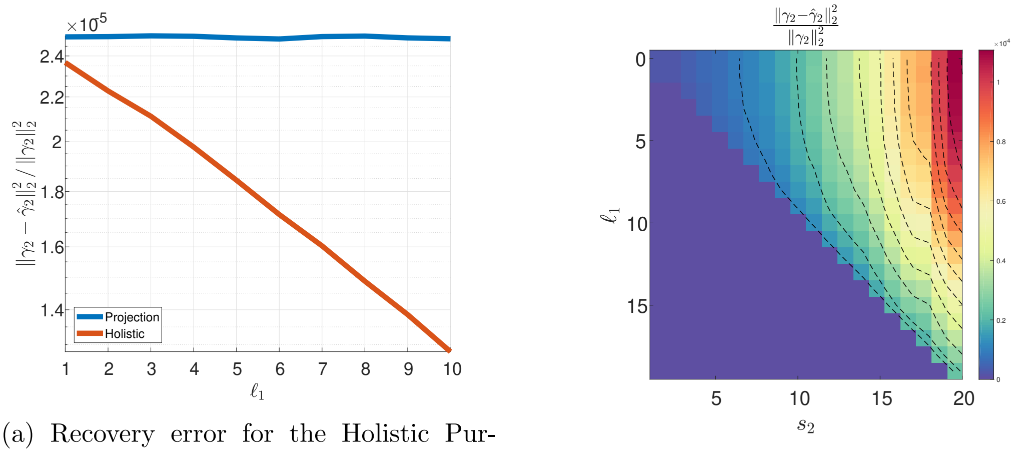
$$\left\|\hat{\boldsymbol{\gamma}}_k^{Holistic} - \boldsymbol{\gamma}_k\right\|_2^2 \leq \max\{s_k - \ell^{tot}\ ,\ \ell^{tot}\}\frac{64\sigma^2 \log m_k}{\kappa_L^2 n}. \tag{7.14}$$

While the above theorem does succeed in posing clear conditions for success of the Holistic Pursuit, the terms of success are somewhat disappointing. On the positive side, we count the fact that the error is shown to be proportional to $s_k - \ell$, which agrees with the Oracle analysis from section 6. On the negative side of the scales, we must mention that the probability for success seems to be weak, and the condition in (7.13) is too convoluted and unclear. Still, in the spirit of this work, which aims to propose a first-of-its-kind Holistic Pursuit for the ML-SC model, we find this theorem encouraging. Further work should be invested in both devising new such algorithms and improving their theoretical study.

## 8. Numerical results.

In this section we present numerical results that demonstrate the Holistic Pursuit algorithm while comparing it with previous approaches. We consider a two-layer sparse model with a signal dimension of $n = 50$, layer dimensions of $m_1 = 100$ and $m_2 = 50$, and dictionaries that were sampled from a Gaussian distribution, $d_1(i,j) \sim \mathcal{N}(0, \frac{1}{n})$ and $d_2(i,j) \sim \mathcal{N}(0, \frac{1}{m_2})$. We set the deepest-layer sparsity to be $s_2$ and the desired cosparsity in the first layer to be $\ell_1$. We synthesize signals from this model by first randomly choosing the supports of $\boldsymbol{\gamma}_2$ and then multiplying by the corresponding matrix $\mathbf{K}$ (computed with the SVD decomposition of the matrix $\boldsymbol{\Phi}$) with a random vector $\boldsymbol{\alpha}$ sampled from a Gaussian distribution, as explained in section 4. Finally, we add white Gaussian noise to the signals, $\mathbf{e} \sim \mathcal{N}(0, \mathbf{I}\sigma^2)$, to obtain the measurement vector $\mathbf{y}$. We study the recovery error of $\boldsymbol{\gamma}_2$ as a function of $\ell_1$ and present the results in Figure 8.1a.

As discussed at length in section 4, the layerwise and projections approaches will not succeed in finding feasible estimates for $\boldsymbol{\gamma}_1$ and $\boldsymbol{\gamma}_2$. The layer-by-layer algorithm will clearly fail due to the very high cardinality of the first layer which is even bigger than the signal dimension, $s_1 > n$. Moreover, $\boldsymbol{\gamma}_1$ is not the unique representation for $\mathbf{x}$ if one ignores the remaining layers. The projection algorithm, on the other hand, would do a somewhat better job of estimating $\boldsymbol{\gamma}_2$; alas it will obtain a dense estimate for $\boldsymbol{\gamma}_1$ when computing $\hat{\boldsymbol{\gamma}}_1 = \mathbf{D}_2\hat{\boldsymbol{\gamma}}_2$. The complete version of this algorithm, which attempts to correct the support in $\hat{\boldsymbol{\gamma}}_2$ if some constraints are not met, will eventually result in the zero-vector. This discussion exposes



(a) Recovery error for the Holistic Pursuit algorithm and the outer-layer projection where the difference $s_2 - \ell_1 = 1$.

(b) Recovery error for the Holistic Pursuit algorithm as function of $\ell_1$ and $s_2$.

**Figure 8.1.** *Empirical recovery error for the Holistic Pursuit algorithm.*

once more the fact that existing pursuit algorithms for the ML-SC model are suitable only for sparse dictionaries. All in all, we refrain from depicting the results from the layerwise approach (as they do not provide competitive results), and we compare the Holistic Pursuit with the outer shell projection, the main part in the projection pursuit (i.e., without backtracking and correcting its solution).

Figure 8.1a presents the performance of both algorithms with a signal-to-noise ratio (SNR) of 25 dB. The penalty parameters, $\eta$, were set as the maximum value such that $\left\|\mathbf{y} - \mathbf{D}_{(2)}\gamma_2\right\|_2^2 \leq n\sigma^2$. One can see that, in both cases, as $\ell_1$ grows, the advantage of the Holistic Pursuit increases. This is to be expected, as the information of the cosupport of $\boldsymbol{\gamma}_1$ becomes more significant and the effective dimension of the signal becomes smaller.

In Figure D.1 we include further results for an SNR of 15 dB. Interestingly, the improvement of the Holistic Pursuit over the projection alternative is more significant in the high SNR scenario. For instance, when $\ell_1 = 10$ the Holistic algorithm reduces the error in 20% in the 15 dB SNR case, and up to 50% in the 25 dB SNR case. This behavior is explained by the fact that when the SNR is low, false detection in the estimation of the midlayer cosupports is more likely, which in turn causes the selection of a wrong subspace on which to project $\hat{\gamma}_2$. This also points to possible improvements for the proposed approach: in such cases, one should not try to estimate the complete cosupport but rather stop before its full recovery.

Before concluding, we depict in Figure 8.1b the recovery error for the Holistic Pursuit as a function of the enforced cosparsity in the intermediate layer, $\ell_1$. In this case, signals were constructed exactly as described above, and the Holistic pursuit was run with different levels of the intermediate cosupport. Recall that, as explained in section 4, in order to sample signals satisfying the model constraints, one must require $s_2 > \ell_1$. This figure clearly shows that, by explicitly leveraging the sparsity of $\boldsymbol{\gamma}_1$, our approach enables us to recover denser representations with the same error as what the projection algorithm would have offered for a sparser signal. These iso-error curves are depicted in dashed black lines, which show the scaling of $s_2 - \ell_1$, as predicted by Theorem 7.3.

## 9. Conclusions.

In this work we have revisited the multi-layer sparse model and analyzed it in its most general (nonconvolutional) form, providing the first known results for recovery of the model's sparse representations under random noise assumptions. The limitations of previous methods led us to propose a new interpretation of this multi-layer construction: this model can now be seen as a global synthesis construction with added analysis-sparse priors. This understanding opened the door to both a tighter theoretical analysis (such as uniqueness guarantees and oracle estimator performance) and the development of better pursuit algorithms to estimate the corresponding representations. This model has demonstrated, for the first time, the symbiotic effect of employing both synthesis and analysis priors on signals. While the Holistic Pursuit proposed above is a first implementation of these ideas, we envision several improvements to boost this algorithm. Naturally, future work should address the performance of this model on real data, for which a proper dictionary learning algorithm should be proposed.

We have opted to consider the general sparse model and not dwell on the convolutional setting of [27, 37] in order keep the technical derivations simpler. While doing so, we have also considered the more general case of fully connected layers, as opposed to convolutional ones. While we see no fundamental problems that would prevent us from applying this dual synthesis-analysis interpretation to the convolutional setting; we also believe that this will call for a careful analysis and necessary adaptations. Some of the tools already presented in previous works [29] (such as the $\ell_{0,\infty}$ norm as a sparsity measure, shifted mutual coherence for dictionary characterization, etc.) would be required, but others would likely be needed as well. For instance, a convolutional "$\ell_\infty$" version of the cosparsity measure would probably come into play. All these points indeed constitute interesting directions for future work.

More broadly, the connection to deep neural networks has motivated much of our analysis. In this respect, just as the work in [27, 37] provided a multi-layer sparse model for convolutional neural networks, in this work we have presented and analyzed a multi-layer model for fully connected ones. More precisely, the forward pass in such fully connected networks can be seen as a pursuit algorithm for signals in the ML-SC model. This work goes further, however, as we have deepened the analysis of this model, demonstrating that it is not empty and presenting a clear way to sample from it. In addition, we have shown (by deriving an Oracle estimator) that neural networks can hope for far better performance in terms of the estimation of the representations if a holistic pursuit is carried out, delineating exciting prospects for the deep learning community. It is true, however, that the algorithm that we presented in order to exploit this synthesis-analysis interpretation does not speak the language of neural networks, as it is greedy in nature. Exploring how similar ideas can be implemented in terms of appropriate network architectures remains a promising and interesting open question. Indeed, some initial ideas have already appeared in the recent work [36], and we believe several others will follow.

## Appendix A. Uniqueness revisited.

In this section, we first restate and later prove improved uniqueness guarantees for the ML-SC model. For the following result, we will require the dictionaries to be in *general position*, as in [25], in the sense that for any set of supports, $\{\Lambda_i\}_{i=1}^k$, if there exists a vector $\boldsymbol{\alpha}$ satisfying $\mathbf{D}_{(k)}^{\Lambda_k}\boldsymbol{\alpha} = \mathbf{0}$ and $\boldsymbol{\Phi}^{\Lambda_k}\boldsymbol{\alpha} = \mathbf{0}$, then necessarily $\boldsymbol{\alpha} = \mathbf{0}$. Moreover, and in order to avoid trivial solutions, one should require that

$|\Lambda_k| \leq \text{rank}\{\mathbf{\Phi}^{\Lambda_k}\} + \text{rank}\{\mathbf{D}_{(k)}^{\Lambda_k}\}$. In other words, a nonzero vector—of particular size—cannot be in both null-spaces simultaneously. Note that this is a fair requirement as it holds for almost all sets of dictionaries in a Lebesgue measure sense.

Theorem A.1. *Consider an ML-SC signal* $\mathbf{x}$*, and a set of dictionaries* $\{\mathbf{D}_i\}_{i=1}^k$ *in general position. If there exists a set of representations* $\{\boldsymbol{\gamma}_i\}_{i=1}^k$ *satisfying*

$$s_k \leq \frac{\sigma(\mathbf{D}_{(k)}) - 1}{2} + r, \tag{A.1}$$

*where* $r = \text{rank}\{\mathbf{\Phi}^{\Lambda_k}\}$ *and* $s_k$ *is the number of nonzero coefficients in the deepest layer, then this set is the* unique *ML-SC representation for* $\mathbf{x}$ *such that its deepest layer has no more than* $s_k$ *nonzeros and the rank of the corresponding* $\mathbf{\Phi}^{\Lambda_k}$ *is no greater than* $r$*.*

*Proof.* Let us assume that $\{\boldsymbol{\gamma}_{i_a}\}_{i=1}^k$ and $\{\boldsymbol{\gamma}_{i_b}\}_{i=1}^k$ are two different representation sets for the ML-SC signal $\mathbf{x}$, such that

$$\left\|\boldsymbol{\gamma}_{k_a}\right\|_0 = \left\|\boldsymbol{\gamma}_{k_b}\right\|_0 = s_k \tag{A.2}$$

and

$$\text{rank}\{\mathbf{\Phi}_a^{\Lambda_{k_a}}\} \leq r, \quad \text{rank}\{\mathbf{\Phi}_b^{\Lambda_{k_b}}\} \leq r, \tag{A.3}$$

where $\mathbf{\Phi}_a$ denotes the matrix $\mathbf{\Phi}$ (as in (4.1)) for representation set $a$, and similarly for $\mathbf{\Phi}_b$. In addition, we remind the reader that $\mathbf{\Phi}^{\Lambda_k}$ restricts $\mathbf{\Phi}$ to the support of $\boldsymbol{\gamma}_k$. Define next the union of the supports $\Lambda_{k_a}$ and $\Lambda_{k_b}$, the deepest layer supports of $a$ and $b$, respectively, to be $\Lambda_{k_U}$, i.e.,

$$\Lambda_{k_U} \triangleq \Lambda_{k_a} \cup \Lambda_{k_b}. \tag{A.4}$$

Define $h_k$ to be the cardinality of the intersection of the deepest layer supports,

$$h_k \triangleq \left|\Lambda_{k_a} \cap \Lambda_{k_b}\right|, \tag{A.5}$$

and define $u$ to be the cardinality of their union:

$$u \triangleq |\Lambda_{k_U}| = 2s_k - h_k. \tag{A.6}$$

We may now use the union subdictionary $\mathbf{D}_{(k)}^{\Lambda_{k_U}}$ to express

$$\mathbf{x} = \mathbf{D}_{(k)}^{\Lambda_{k_U}} \boldsymbol{\gamma}_{k_a}^{\Lambda_{k_U}} = \mathbf{D}_{(k)}^{\Lambda_{k_U}} \boldsymbol{\gamma}_{k_b}^{\Lambda_{k_U}}, \tag{A.7}$$

where the extra elements added to these supports are simply zeros. Following [25], let us now define the subspaces where $\boldsymbol{\gamma}_{k_a}^{\Lambda_{k_U}}, \boldsymbol{\gamma}_{k_b}^{\Lambda_{k_U}}$ lie:

$$\mathcal{W}_a \triangleq \left\{\boldsymbol{\alpha} : \boldsymbol{\alpha} \in \ker\{\mathbf{\Phi}_a^{\Lambda_{k_U}}\},\ Supp(\boldsymbol{\alpha}) = \Lambda_{k_a}\right\}, \tag{A.8}$$

$$\mathcal{W}_b \triangleq \left\{\boldsymbol{\alpha} : \boldsymbol{\alpha} \in \ker\{\mathbf{\Phi}_b^{\Lambda_{k_U}}\},\ Supp(\boldsymbol{\alpha}) = \Lambda_{k_b}\right\}. \tag{A.9}$$

We now consider a difference vector $\mathbf{\Delta} = \boldsymbol{\gamma}_{k_a} - \boldsymbol{\gamma}_{k_b}$. In what follows, we shall find a condition under which $\mathbf{\Delta}$ cannot be other than the zero vector, implying that the representation for $\mathbf{x}$ must be in fact unique.

The difference vector restricted to the union of support, $\mathbf{\Delta}^{\Lambda_{k_U}}$, must satisfy two conditions:

1. It must lie in the null space of $\mathbf{D}_{(k)}^{\Lambda_{k_U}}$, since

$$\mathbf{D}_{(k)}^{\Lambda_{k_U}} \mathbf{\Delta}^{\Lambda_{k_U}} = \mathbf{0}. \tag{A.10}$$

2. It must lie in the union of $\mathcal{W}_a$ and $\mathcal{W}_b$:[5]

$$\mathbf{\Delta}^{\Lambda_{k_U}} \in (\mathcal{W}_a + \mathcal{W}_b)\,. \tag{A.11}$$

Therefore, $\mathbf{\Delta}$ ought to be zero if

$$(\mathcal{W}_a + \mathcal{W}_b) \cap \ker\{\mathbf{D}_{(k)}^{\Lambda_{k_U}}\} = \{\mathbf{0}\} \tag{A.12}$$

for all $\{\boldsymbol{\gamma}_{i_a}\}_{i=1}^{k}, \{\boldsymbol{\gamma}_{i_b}\}_{i=1}^{k}$ satisfying (A.2) and (A.3).

Following [25], as the dictionaries are in general position, then as long as

$$\dim\{(\mathcal{W}_a + \mathcal{W}_b)\} + \dim\{\ker(\mathbf{D}_{(k)}^{\Lambda_{k_U}})\} \leq u, \tag{A.13}$$

(A.12) also holds. Note that the assumption about the dictionaries being in general position is a fair one, as it holds for almost every dictionary set in Lebesgue measure [25].

We shall now elaborate on this condition by upper-bounding the two elements in the left term and developing a corresponding sufficient condition for uniqueness. We start by injecting the effective dictionary spark to upper-bound $\dim\{\ker(\mathbf{D}_{(k)}^{\Lambda_{k_U}})\}$. We know that the rank of $\mathbf{D}_{(k)}^{\Lambda_{k_U}}$ is no less than $\min\{u, \sigma(\mathbf{D}_{(k)}) - 1\}$, because $u$ is the number of columns in this matrix, and every $\sigma(\mathbf{D}_{(k)}) - 1$ of its columns are linearly independent by definition of the spark. Recall that, due to the rank-nullity theorem, we have

$$\dim\{\ker(\mathbf{D}_{(k)}^{\Lambda_{k_U}})\} = u - \operatorname{rank}\{\mathbf{D}_{(k)}^{\Lambda_{k_U}}\}, \tag{A.14}$$

and so the dimension of the kernel of $\mathbf{D}_{(k)}^{\Lambda_{k_U}}$ is upper-bounded by

$$\dim\{\ker(\mathbf{D}_{(k)}^{\Lambda_{k_U}})\} \leq u - \min\{u, \sigma(\mathbf{D}_{(k)}) - 1\} = \max\{0, u - \sigma(\mathbf{D}_{(k)}) + 1\}. \tag{A.15}$$

On the other hand, to upper-bound $\dim\{(\mathcal{W}_a + \mathcal{W}_b)\}$, we recall that $\dim(\mathcal{W}_a)$ and $\dim(\mathcal{W}_b)$ are less than or equal to $s_k - r$ (from (A.3)), resulting in

$$\dim\{(\mathcal{W}_a + \mathcal{W}_b)\} \leq 2s_k - 2r = u + h_k - 2r. \tag{A.16}$$

Therefore, the corresponding sufficient condition for (A.13) is

$$\max\{0, u - \sigma(\mathbf{D}_{(k)}) + 1\} + u + h_k - 2r \leq u. \tag{A.17}$$

[5] Since $\boldsymbol{\gamma}_{k_a} \in \mathcal{W}_a$ and $\boldsymbol{\gamma}_{k_b} \in \mathcal{W}_b$, any linear combination of them must reside in $(\mathcal{W}_a + \mathcal{W}_b)$.

Given the max in the above expression, let us analyze both cases separately. First, if $u \leq \sigma(\mathbf{D}_{(k)}) - 1$, then, by definition of the spark, $\mathbf{\Delta}$ must be zero in order to obtain

$$\mathbf{0} = \mathbf{D}_{(k)}^{\Lambda_{k_U}} \mathbf{\Delta}. \tag{A.18}$$

As one can see, this boils down to the traditional condition for uniqueness of sparse (synthesis) representations, namely that $s_k < \sigma(\mathbf{D}_{(k)})/2$.

On the other end, when $u > \sigma(\mathbf{D}_{(k)}) - 1$, one obtains the condition

$$u + h_k \leq \sigma(\mathbf{D}_{(k)}) + 2r - 1, \tag{A.19}$$

which leads to

$$s_k \leq \frac{\sigma(\mathbf{D}_{(k)}) - 1}{2} + r. \tag{A.20}$$

In other words, as long as this condition holds, $\mathbf{\Delta} = \mathbf{0}$, and so $\boldsymbol{\gamma}_{i_a} = \boldsymbol{\gamma}_{i_b}$ $\forall i$. ■

Before concluding this section, note that the above result was derived for the case of $\|\boldsymbol{\gamma}_k\|_0 = s_k$ for simplicity, though the same results hold for every $\boldsymbol{\gamma}_k : \|\boldsymbol{\gamma}_k\|_0 \leq s_k$.

## Appendix B. The Oracle estimator performance proof.

### B.1. Layerwise: Oracle performance.

*Proof.* Recall from (6.4) that the Oracle estimator for the layerwise approach is

$$\begin{aligned}
\hat{\boldsymbol{\gamma}}_i^{\Lambda_i} &= \mathbf{D}_i^{\Lambda_i \dagger} \hat{\boldsymbol{\gamma}}_{i-1} = \left(\mathbf{D}_i^{\Lambda_i T} \mathbf{D}_i^{\Lambda_i}\right)^{-1} \mathbf{D}_i^{\Lambda_{i-1},\Lambda_i T} \hat{\boldsymbol{\gamma}}_{i-1}^{\Lambda_{i-1}} \\
&= \left(\mathbf{D}_i^{\Lambda_i T} \mathbf{D}_i^{\Lambda_i}\right)^{-1} \mathbf{D}_i^{\Lambda_{i-1},\Lambda_i T} \cdots \left(\mathbf{D}_1^{\Lambda_1 T} \mathbf{D}_1^{\Lambda_1}\right)^{-1} \mathbf{D}_1^{\Lambda_1 T} \mathbf{y} = \mathcal{U}_{(i,1)} \mathbf{y},
\end{aligned} \tag{B.1}$$

where

$$\mathcal{U}_{(i,j)} \triangleq \left(\mathbf{D}_i^{\Lambda_i T} \mathbf{D}_i^{\Lambda_i}\right)^{-1} \mathbf{D}_i^{\Lambda_{i-1},\Lambda_i T} \cdots \left(\mathbf{D}_j^{\Lambda_j T} \mathbf{D}_j^{\Lambda_j}\right)^{-1} \mathbf{D}_j^{\Lambda_{j-1},\Lambda_j T}, \tag{B.2}$$

and $\Lambda_0 \triangleq \{1, \ldots, n\}$. Using the fact that $\hat{\boldsymbol{\gamma}}_i^{\Lambda_i} = \boldsymbol{\gamma}_i^{\Lambda_i} + \mathcal{U}_{(i,1)}\mathbf{e} = \boldsymbol{\gamma}_i + \sigma \mathbf{W}_i \mathbf{z}$, we can write

$$\begin{aligned}
\mathbb{E}\, \|\boldsymbol{\gamma}_i - \hat{\boldsymbol{\gamma}}_i\|_2^2 &= \mathbb{E}\, \|\sigma \mathbf{W}_i \mathbf{z}\|_2^2 \\
&= \sigma^2 \operatorname{Trace} \mathcal{U}_{(i,1)} \mathcal{U}_{(i,1)}^T \\
&= \sigma^2 \operatorname{Trace} \mathcal{U}_{(i,2)} \left(\mathbf{D}_1^{\Lambda_1 T} \mathbf{D}_1^{\Lambda_1}\right)^{-1} \mathcal{U}_{(i,2)}^T \\
&= \sigma^2 \operatorname{Trace} \left(\mathbf{D}_1^{\Lambda_1 T} \mathbf{D}_1^{\Lambda_1}\right)^{-1} \mathcal{U}_{(i,2)}^T \mathcal{U}_{(i,2)}.
\end{aligned} \tag{B.3}$$

We shall now use the fact that for every pair of symmetric positive definite matrices $\mathbf{A}, \mathbf{B}$ the following holds:

$$\operatorname{Trace} \mathbf{AB} = \operatorname{Trace} \lambda_{\mathbf{A}}^{\min} \mathbf{B} + \operatorname{Trace} \left(\mathbf{A} - \lambda_{\mathbf{A}}^{\min} \mathbf{I}\right) \mathbf{B} \geq \lambda_{\mathbf{A}}^{\min} \operatorname{Trace} \mathbf{B}, \tag{B.4}$$

where $\lambda_{\mathbf{A}}^{\min}$ is the minimal eigenvalue of $\mathbf{A}$. The inequality is true because $\mathbf{A}-\lambda_{\mathbf{A}}^{\min}\mathbf{I}$ is a semipositive symmetric matrix, resulting in that the matrix $\left(\mathbf{A}-\lambda_{\mathbf{A}}^{\min}\mathbf{I}\right)\mathbf{B}$ is a semipositive symmetric matrix, and therefore, its trace, which equals the eigenvalues' sum, is bigger than 0. In our case, $\mathbf{A}$ is the matrix $(\mathbf{D}_1^{\Lambda_1}{}^T\mathbf{D}_1^{\Lambda_1})^{-1}$, and $\mathbf{B}$ is the matrix $\mathcal{U}_{(i,2)}^T\mathcal{U}_{(i,2)}$. In addition, using the RIP of dictionary $\mathbf{D}_1$, we know that the eigenvalues of $(\mathbf{D}_1^{\Lambda_1}{}^T\mathbf{D}_1^{\Lambda_1})^{-1}$ are no less than $\frac{1}{1+\delta_{s_1}^{\mathbf{D}_1}}$. Therefore, we can lower-bound the recovery error,

$$\begin{aligned}
&\mathbb{E}\left\|\gamma_i-\hat{\gamma}_i\right\|_2^2 \geq \sigma^2\frac{1}{1+\delta_{s_1}^{\mathbf{D}_1}}\,\mathrm{Trace}\,\mathcal{U}_{(i,2)}^T\mathcal{U}_{(i,2)}\\
&=\sigma^2\frac{1}{1+\delta_{s_1}^{\mathbf{D}_1}}\,\mathrm{Trace}\,\mathcal{U}_{(i,3)}^T\left(\mathbf{D}_2^{\Lambda_2}{}^T\mathbf{D}_2^{\Lambda_2}\right)^{-1}\cdot\mathbf{D}_2^{\Lambda_1,\Lambda_2}{}^T\mathbf{D}_2^{\Lambda_1,\Lambda_2}\left(\mathbf{D}_2^{\Lambda_2}{}^T\mathbf{D}_2^{\Lambda_2}\right)^{-1}\mathcal{U}_{(i,3)}\\
&=\sigma^2\frac{1}{1+\delta_{s_1}^{\mathbf{D}_1}}\,\mathrm{Trace}\,\mathbf{D}_2^{\Lambda_1,\Lambda_2}{}^T\mathbf{D}_2^{\Lambda_1,\Lambda_2}\left(\mathbf{D}_2^{\Lambda_2}{}^T\mathbf{D}_2^{\Lambda_2}\right)^{-1}\cdot\mathcal{U}_{(i,3)}\mathcal{U}_{(i,3)}^T\left(\mathbf{D}_2^{\Lambda_2}{}^T\mathbf{D}_2^{\Lambda_2}\right)^{-1}.
\end{aligned}\tag{B.5}$$

Using again the fact which was introduced in (B.4) and the Subset RIP of $\mathbf{D}_2^{\Lambda_1,\Lambda_2}$, we can write

$$\begin{aligned}
&\mathbb{E}\left\|\gamma_i-\hat{\gamma}_i\right\|_2^2\\
&\geq\sigma^2\frac{1}{1+\delta_{s_1}^{\mathbf{D}_1}}\left(\frac{s_1}{n_1}-\delta_{s_1,s_2}^{\mathbf{D}_2}\right)\mathrm{Trace}\left(\mathbf{D}_2^{\Lambda_2}{}^T\mathbf{D}_2^{\Lambda_2}\right)^{-1}\cdot\mathcal{U}_{(i,3)}\mathcal{U}_{(i,3)}^T\left(\mathbf{D}_2^{\Lambda_2}{}^T\mathbf{D}_2^{\Lambda_2}\right)^{-1}\\
&\geq\sigma^2\frac{1}{1+\delta_{s_1}^{\mathbf{D}_1}}\frac{\frac{s_1}{n_1}-\delta_{s_1,s_2}^{\mathbf{D}_2}}{\left(1+\delta_{s_2}^{\mathbf{D}_2}\right)^2}\,\mathrm{Trace}\,\mathcal{U}_{(i,3)}\mathcal{U}_{(i,3)}^T\\
&\;\vdots\\
&\geq\sigma^2\frac{1}{1+\delta_{s_1}^{\mathbf{D}_1}}\prod_{j=2}^{i-1}\frac{\frac{s_{j-1}}{n_{j-1}}-\delta_{s_{j-1},s_j}^{\mathbf{D}_j}}{\left(1+\delta_{s_j}^{\mathbf{D}_j}\right)^2}\,\mathrm{Trace}\,\mathcal{U}_{(i,i)}\mathcal{U}_{(i,i)}^T\\
&\geq\sigma^2\frac{1}{1+\delta_{s_1}^{\mathbf{D}_1}}\prod_{j=2}^{i-1}\frac{\frac{s_{j-1}}{n_{j-1}}-\delta_{s_{j-1},s_j}^{\mathbf{D}_j}}{\left(1+\delta_{s_j}^{\mathbf{D}_j}\right)^2}\left(\frac{s_{i-1}}{n_{i-1}}-\delta_{s_{i-1},s_i}^{\mathbf{D}_i}\right)\cdot\mathrm{Trace}\left(\mathbf{D}_i^{\Lambda_i}{}^T\mathbf{D}_i^{\Lambda_i}\right)^{-2}\\
&\geq\sigma^2 s_i\frac{1}{1+\delta_{s_1}^{\mathbf{D}_1}}\prod_{j=2}^{i}\frac{\frac{s_{j-1}}{n_{j-1}}-\delta_{s_{j-1},s_j}^{\mathbf{D}_j}}{\left(1+\delta_{s_j}^{\mathbf{D}_j}\right)^2}.
\end{aligned}\tag{B.6}$$

We omit the upper bound proof since it is equivalent to the above lower bound proof apart from changing signs (− to + and opposite). ■

**B.2. Projection: Oracle performance.** We next derive the performance bounds for the projection Oracle estimator equation (6.8).

*Proof.* The bound for the first step of the projection algorithm which is the deepest layer estimation is the single-layer bound of the effective model. In order to bound the recovery error of the intermediate layers, $\{\gamma_i\}_{i=1}^{k-1}$, we use the connection which was present in (6.7). For the sake of convenience we define

$$\overline{\mathbf{D}}_i = \mathbf{D}_i^{\Lambda_{i-1},\Lambda_i} \cdots \mathbf{D}_k^{\Lambda_{k-1},\Lambda_k} \quad \forall\ 2 \le i \le k. \tag{B.7}$$

Therefore, we can write

$$\begin{aligned} \mathbb{E}\,\|\boldsymbol{\gamma}_i - \hat{\boldsymbol{\gamma}}_i\|_2^2 &= \mathbb{E}\left\|\sigma \overline{\mathbf{D}}_{i+1} \mathbf{D}_{(k)}^{\Lambda_k \dagger} \mathbf{z}\right\|_2^2 \\ &= \sigma^2 \operatorname{Trace} \mathbf{D}_{(k)}^{\Lambda_k \dagger T} \overline{\mathbf{D}}_{i+1}{}^T \overline{\mathbf{D}}_{i+1} \mathbf{D}_{(k)}^{\Lambda_k \dagger} \\ &= \sigma^2 \operatorname{Trace} \left(\mathbf{D}_{(k)}^{\Lambda_k T} \mathbf{D}_{(k)}^{\Lambda_k}\right)^{-1} \overline{\mathbf{D}}_{i+1}{}^T \overline{\mathbf{D}}_{i+1}. \end{aligned} \tag{B.8}$$

Using (B.4) and the RIP of the matrix $\mathbf{D}_{(k)}^{\Lambda_k}$, we can lower-bound the recovery error:

$$\begin{aligned} \mathbb{E}\,\|\boldsymbol{\gamma}_i - \hat{\boldsymbol{\gamma}}_i\|_2^2 &\ge \frac{1}{1+\delta_{s_k}^{\mathbf{D}_{(k)}}} \operatorname{Trace} \overline{\mathbf{D}}_{i+1}{}^T \overline{\mathbf{D}}_{i+1} \\ &= \frac{1}{1+\delta_{s_k}^{\mathbf{D}_{(k)}}} \operatorname{Trace} \mathbf{D}_{i+1}^{\Lambda_i,\Lambda_{i+1} T} \mathbf{D}_{i+1}^{\Lambda_i,\Lambda_{i+1}} \\ &\quad \cdot \overline{\mathbf{D}}_{i+2} \overline{\mathbf{D}}_{i+2}{}^T. \end{aligned} \tag{B.9}$$

Using again the connection from (B.4) and the Subset RIP of $\mathbf{D}_{i+1}^{\Lambda_i,\Lambda_{i+1}}$, we can write

$$\begin{aligned} \mathbb{E}\,\|\boldsymbol{\gamma}_i - \hat{\boldsymbol{\gamma}}_i\|_2^2 &\ge \frac{1}{1+\delta_{s_k}^{\mathbf{D}_{(k)}}} \left(\frac{s_i}{m_i} - \delta_{s_i,s_{i+1}}^{\mathbf{D}_{i+1}}\right) \operatorname{Trace} \overline{\mathbf{D}}_{i+2}^T \overline{\mathbf{D}}_{i+2} \\ &\ \ \vdots \\ &\ge \sigma^2 \frac{\prod_{j=i+1}^{k-1} \left(\frac{s_{j-1}}{m_{j-1}} - \delta_{s_{j-1},s_j}^{\mathbf{D}_j}\right)}{1+\delta_{s_k}^{\mathbf{D}_{(k)}}} \cdot \operatorname{Trace} \mathbf{D}_k^{\Lambda_{k-1},\Lambda_k T} \mathbf{D}_k^{\Lambda_{k-1},\Lambda_k} \\ &\ge \sigma^2 s_k \frac{\prod_{j=i+1}^{k} \left(\frac{s_{j-1}}{m_{j-1}} - \delta_{s_{j-1},s_j}^{\mathbf{D}_j}\right)}{1+\delta_{s_k}^{\mathbf{D}_{(k)}}}. \end{aligned} \tag{B.10}$$

The upper bound proof is the same as the lower bound proof with changing signs (− to + and opposite). ■

## Appendix C. Holistic Pursuit algorithm analysis.

Lemma C.1. *Let* $\hat{\boldsymbol{\gamma}}_k = \boldsymbol{\gamma}_k + \mathbf{e}$ *be the estimation of the deepest layer, let* $\hat{\Lambda}_i^c$ *be the estimated cosupport in the ith layer, where*

$$\sum_{i=1}^{k-1} \left|\hat{\Lambda}_i^c\right| = j - 1, \tag{C.1}$$

*and let* $\mu_R^i$ *be the* row-wise *mutual-coherence defined in* (7.7). *If*

$$\|\mathbf{e}\|_2 \leq \max_{i:\left|\hat{\Lambda}_i^c\right|<\ell_i} \gamma_i^{\min}\left(1+\frac{1+\mu_R^i\left(\ell_i-\left|\hat{\Lambda}_i^c\right|-1\right)}{\sqrt{\ell_i-\left|\hat{\Lambda}_i^c\right|}}\right)^{-1}, \tag{C.2}$$

*then the Holistic Pursuit algorithm succeeds in its* $j$*th iteration in recovering a new element from the midlayers' cosupport.*

*Proof.* Let $i$ be

$$i=\underset{i:\ \left|\hat{\Lambda}_i^c\right|<\ell_i}{\operatorname{argmax}}\ \gamma_i^{\min}\left(1+\frac{1+\mu_R^i\left(\ell_i-\left|\hat{\Lambda}_i^c\right|-1\right)}{\sqrt{\ell_i-\left|\hat{\Lambda}_i^c\right|}}\right)^{-1}. \tag{C.3}$$

The algorithm succeeds in its $j$th iteration if

$$\min\left|\mathbf{D}_{(i+1,k)}^{\Lambda_i^c/\hat{\Lambda}_i^c}\hat{\boldsymbol{\gamma}}_k\right| < \min\left|\mathbf{D}_{(i+1,k)}^{\Lambda_i}\hat{\boldsymbol{\gamma}}_k\right|, \tag{C.4}$$

where $\Lambda_i^c/\hat{\Lambda}_i^c$ is the set of the unfounded cosupport elements in layer $i$. Since the left term is only for cosupport rows, we can simplify the left term:

$$\min\left|\mathbf{D}_{(i+1,k)}^{\Lambda_i^c/\hat{\Lambda}_i^c}\hat{\boldsymbol{\gamma}}_k\right| = \min\left|\mathbf{D}_{(i+1,k)}^{\Lambda_i^c/\hat{\Lambda}_i^c}\mathbf{e}\right|. \tag{C.5}$$

In order to upper-bound (C.5), we look for the error vector $\mathbf{e}$ that maximizes this term. Let us first assume that the rows in $\mathbf{D}_{(i+1,k)}^{\Lambda_i^c/\hat{\Lambda}_i^c}$ are orthonormal. In this simplified case, the error that maximizes (C.5) is the vector obtained by the average distance to every row of $\mathbf{D}_{(i+1,k)}^{\Lambda_i^c/\hat{\Lambda}_i^c}$, having $\sqrt{\ell_i-|\hat{\Lambda}_i^c|}$ of these. In other words, we look for the error vector $\tilde{\mathbf{e}}$ such that

$$\tilde{\mathbf{e}}=\frac{\|\mathbf{e}\|_2}{\sqrt{\ell_i-\left|\hat{\Lambda}_i^c\right|}}(\mathbf{D}_{(i+1,k)}^{\Lambda_i^c/\hat{\Lambda}_i^c})^T\mathbf{1}. \tag{C.6}$$

Such an error vector leads to the following upper bound:

$$\min\left|\mathbf{D}_{(i+1,k)}^{\Lambda_i^c/\hat{\Lambda}_i^c}\mathbf{e}\right| \leq \min\left|\mathbf{D}_{(i+1,k)}^{\Lambda_i^c/\hat{\Lambda}_i^c}\tilde{\mathbf{e}}\right| = \frac{\|\mathbf{e}\|_2}{\sqrt{\ell_i-\left|\hat{\Lambda}_i^c\right|}}. \tag{C.7}$$

Consider now the more general case, where the rows of $\mathbf{D}_{(i+1,k)}^{\Lambda_i^c/\hat{\Lambda}_i^c}$ are not orthogonal, but rather the correlation between every two rows is upper-bounded by $\mu_R^i$. Now, in order to bound the minimal correlation between the noise vector and every atom, we might consider the worst-case scenario where the inner product between any pair of atoms is equal to $\mu_R^i$. In such a case, $\mathbf{e}$ in (C.6) maximizes (C.5), and one thus obtains the following upper bound:

$$\min \left|\mathbf{D}^{\Lambda_i^c/\hat{\Lambda}_i^c}_{(i+1,k)}\mathbf{e}\right| \le \|\mathbf{e}\|_2 \frac{\left(1+\mu_R^i\left(\ell_i - \left|\hat{\Lambda}_i^c\right| - 1\right)\right)}{\sqrt{\ell_i - \left|\hat{\Lambda}_i^c\right|}}. \tag{C.8}$$

For the right term of (C.4) we use the same derivations as in [31], resulting in

$$\min \left|\mathbf{D}^{\Lambda_i}_{(i+1,k)}\hat{\gamma}_k\right| \ge \gamma_i^{\min} - \max \left|\mathbf{D}^{\Lambda_i}_{(i+1,k)}\mathbf{e}\right| \ge \gamma_i^{\min} - \|\mathbf{e}\|_2 , \tag{C.9}$$

where the last inequality follows from the Cauchy–Schwarz inequality and the fact that all rows in $\mathbf{D}_{(i+1,k)}$ are assumed to be normalized. Using (C.8) and (C.9), one arrives at a sufficient condition for the success of the algorithm, in the form of

$$\|\mathbf{e}\|_2 \le \gamma_i^{\min} \left(1 + \frac{1+\mu_R^i\left(\ell_i - \left|\hat{\Lambda}_i^c\right| - 1\right)}{\sqrt{\ell_i - \left|\hat{\Lambda}_i^c\right|}}\right)^{-1}. \tag{C.10}$$

Combining (C.3) with (C.10) provides the claimed lemma. ■

**Appendix D. Numerical experiments.** Figure D.1 expands on the experimental results presented in section 8.

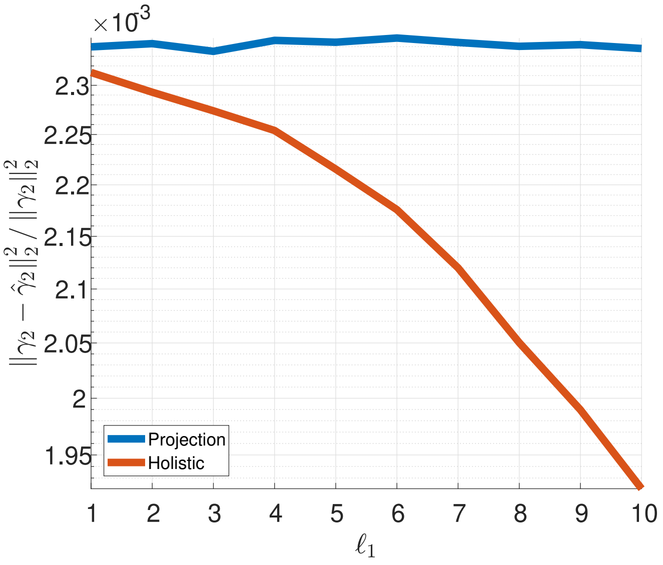


**Figure D.1.** *Recovery error for the Holistic Pursuit algorithm and the outer-layer projection where the difference $s_2 - \ell_1 = 1$.*

## REFERENCES

[1] Z. Ben-Haim, Y. C. Eldar, and M. Elad, *Coherence-based performance guarantees for estimating a sparse vector under random noise*, IEEE Trans. Signal Process., 58 (2010), pp. 5030–5043.

[2] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Found. Trends Machine Learning, 3 (2011), pp. 1–122.

[3] H. Bristow, A. Eriksson, and S. Lucey, *Fast convolutional sparse coding*, in Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Washington, DC, 2013, pp. 391–398.

[4] A. M. Bruckstein, D. L. Donoho, and M. Elad, *From sparse solutions of systems of equations to sparse modeling of signals and images*, SIAM Rev., 51 (2009), pp. 34–81, https://doi.org/10.1137/060657704.

[5] E. Candès and T. Tao, *The Dantzig selector: Statistical estimation when p is much larger than n*, Ann. Statist., 35 (2007), pp. 2313–2351.

[6] E. J. Candès, *The restricted isometry property and its implications for compressed sensing*, C. R. Math. Acad. Sci. Paris, 346 (2008), pp. 589–592.

[7] E. J. Candès, Y. C. Eldar, D. Needell, and P. Randall, *Compressed sensing with coherent and redundant dictionaries*, Appl. Comput. Harmon. Anal., 31 (2011), pp. 59–73.

[8] S. S. Chen, D. L. Donoho, and M. A. Saunders, *Atomic decomposition by basis pursuit*, SIAM Rev., 43 (2001), pp. 129–159, https://doi.org/10.1137/S003614450037906X.

[9] W. Dai and O. Milenkovic, *Subspace pursuit for compressive sensing signal reconstruction*, IEEE Trans. Inform. Theory, 55 (2009), pp. 2230–2249.

[10] W. Dong, L. Zhang, G. Shi, and X. Wu, *Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization*, IEEE Trans. Image Process., 20 (2011), pp. 1838–1857.

[11] D. L. Donoho and M. Elad, *Optimally sparse representation in general (nonorthogonal) dictionaries via $\ell_1$ minimization*, Proc. Natl. Acad. Sci. USA, 100 (2003), pp. 2197–2202.

[12] M. Elad, *From exact to approximate solutions*, in Sparse and Redundant Representations, Springer, New York, 2010, pp. 79–109.

[13] M. Elad and M. Aharon, *Image denoising via sparse and redundant representations over learned dictionaries*, IEEE Trans. Image Process., 15 (2006), pp. 3736–3745.

[14] M. Elad, P. Milanfar, and R. Rubinstein, *Analysis versus synthesis in signal priors*, Inverse Problems, 23 (2007), pp. 947–968.

[15] R. Giryes, S. Nam, M. Elad, R. Gribonval, and M. E. Davies, *Greedy-like algorithms for the cosparse analysis model*, Linear Algebra Appl., 441 (2014), pp. 22–60.

[16] R. Gribonval and M. Nielsen, *Sparse representations in unions of bases*, IEEE Trans. Inform. Theory, 49 (2003), pp. 3320–3325.

[17] S. Gu, W. Zuo, Q. Xie, D. Meng, X. Feng, and L. Zhang, *Convolutional sparse coding for image super-resolution*, in Proceedings of the IEEE International Conference on Computer Vision, IEEE, Washington, DC, 2015, pp. 1823–1831.

[18] F. Heide, W. Heidrich, and G. Wetzstein, *Fast and flexible convolutional sparse coding*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Washington, DC, 2015, pp. 5135–5143.

[19] G. M. James, C. Paulson, and P. Rusmevichientong, *Penalized and Constrained Regression*, Tech. report, mimeo, Marshall School of Business, University of Southern California, Los Angeles, CA, 2013.

[20] Z. Jiang, Z. Lin, and L. S. Davis, *Learning a discriminative dictionary for sparse coding via label consistent K-SVD*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Washington, DC, 2011, pp. 1697–1704.

[21] K. Kavukcuoglu, M. Ranzato, and Y. LeCun, *Fast Inference in Sparse Coding Algorithms with Applications to Object Recognition*, preprint, https://arxiv.org/abs/1010.3467, 2010.

[22] Y. M. Lu and M. N. Do, *A theory for sampling signals from a union of subspaces*, IEEE Trans. Signal Process., 56 (2008), pp. 2334–2345.

[23] J. Mairal, F. Bach, and J. Ponce, *Sparse modeling for image and vision processing*, Found. Trends Comput. Graphics Vision, 8 (2014), pp. 85–283.

[24] S. Nam, M. E. Davies, M. Elad, and R. Gribonval, *Cosparse analysis modeling-uniqueness and algorithms*, in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, Washington, DC, 2011, pp. 5804–5807.

[25] S. Nam, M. E. Davies, M. Elad, and R. Gribonval, *The cosparse analysis model and algorithms*, Appl. Comput. Harmon. Anal., 34 (2013), pp. 30–56.

[26] D. Needell and J. A. Tropp, *CoSaMP: Iterative signal recovery from incomplete and inaccurate samples*, Appl. Comput. Harmon. Anal., 26 (2009), pp. 301–321.

[27] V. Papyan, Y. Romano, and M. Elad, *Convolutional neural networks analyzed via convolutional sparse coding*, J. Mach. Learn. Res., 18 (2017), pp. 2887–2938.

[28] V. Papyan, Y. Romano, J. Sulam, and M. Elad, *Convolutional dictionary learning via local processing*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Washington, DC, 2017, pp. 5296–5304.

[29] V. Papyan, J. Sulam, and M. Elad, *Working locally thinking globally: Theoretical guarantees for convolutional sparse coding*, IEEE Trans. Signal Process., 65 (2017), pp. 5687–5701.

[30] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, *Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition*, in 1993 Conference Record of the Twenty-Seventh Asilomar Conference on Signals, Systems and Computers, IEEE, Washington, DC, 1993, pp. 40–44.

[31] T. Peleg and M. Elad, *Performance guarantees of the thresholding algorithm for the cosparse analysis model*, IEEE Trans. Inform. Theory, 59 (2013), pp. 1832–1845.

[32] S. Ravishankar and Y. Bresler, *Learning sparsifying transforms*, IEEE Trans. Signal Process., 61 (2013), pp. 1072–1086.

[33] S. Ravishankar, B. Wen, and Y. Bresler, *Online sparsifying transform learning—part* I: *Algorithms*, IEEE J. Sel. Topics Signal Process., 9 (2015), pp. 625–636.

[34] R. Rubinstein, T. Peleg, and M. Elad, *Analysis K-SVD: A dictionary-learning algorithm for the analysis sparse model*, IEEE Trans. Signal Process., 61 (2013), pp. 661–677.

[35] I. W. Selesnick and M. A. Figueiredo, *Signal restoration with overcomplete wavelet transforms: Comparison of analysis and synthesis priors*, in Wavelets XIII, Proc. SPIE 7446, International Society for Optics and Photonics, SPIE, Bellingham, WA, 2009, 74460D.

[36] J. Sulam, A. Aberdam, and M. Elad, *On Multi-Layer Basis Pursuit, Efficient Algorithms and Convolutional Neural Networks*, preprint, https://arxiv.org/abs/1806.00701, 2018.

[37] J. Sulam, V. Papyan, Y. Romano, and M. Elad, *Multilayer convolutional sparse modeling: Pursuit and dictionary learning*, IEEE Trans. Signal Process., 66 (2018), pp. 4090–4104, https://doi.org/10.1109/TSP.2018.2846226.

[38] J. A. Tropp, *Greed is good: Algorithmic results for sparse approximation*, IEEE Trans. Inform. Theory, 50 (2004), pp. 2231–2242.

[39] J. A. Tropp, *Just relax: Convex programming methods for identifying sparse signals in noise*, IEEE Trans. Inform. Theory, 52 (2006), pp. 1030–1051.

[40] Q. Zhang and B. Li, *Discriminative K-SVD for dictionary learning in face recognition*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Washington, DC, 2010, pp. 2691–2698.