DECONET: an Unfolding Network for Analysis-based Compressed Sensing with Generalization Error Bounds

Vicky Kouni and Yannis Panagakis

***Abstract*—We present a new deep unfolding network for analysis-sparsity-based Compressed Sensing. The proposed network coined Decoding Network (DECONET) jointly learns a decoder that reconstructs vectors from their incomplete, noisy measurements and a redundant sparsifying analysis operator, which is shared across the layers of DECONET. Moreover, we formulate the hypothesis class of DECONET and estimate its associated Rademacher complexity. Then, we use this estimate to deliver meaningful upper bounds for the generalization error of DECONET. Finally, the validity of our theoretical results is assessed and comparisons to state-of-the-art unfolding networks are made, on both synthetic and real-world datasets. Experimental results indicate that our proposed network outperforms the baselines, consistently for all datasets, and its behaviour complies with our theoretical findings.**

***Index Terms*—Compressed Sensing, analysis sparsity, unfolding network, generalization error, Rademacher complexity.**

I. Introduction

Compressed Sensing (CS) [1] is a modern technique to reconstruct signals $x \in \mathbb{R}^n$ from few linear and possibly corrupted observations $y = Ax + e \in \mathbb{R}^m$, $m < n$. Iterative methods applied on CS are by now widely used [2, 3, 4]. Nevertheless, deep neural networks (DNNs) have become popular for tackling sparse recovery problems like CS [5, 6], since they significantly reduce the time complexity and increase the quality of the reconstruction. A new line of research lies on merging DNNs and optimization algorithms, leading to the so-called *deep unfolding/unrolling* [7, 8]. The latter pertains to interpreting the iterations of well-known iterative algorithms as layers of a DNN, which reconstructs x from y.

Deep unfolding networks have become increasingly popular in the last few years [9], [10], [11], [12] because – in contrast with traditional DNNs – they are interpretable [13], integrate prior knowledge about the signal structure [14], and have relatively few trainable parameters [15]. Especially in the case of CS, unfolding networks have proven to work particularly well. For example, [16], [17], [18], [19], [20], [21], [22] interpret the iterations of well-studied optimization algorithms as layers of a neural network, which learns a *decoder* for CS, i.e., a function that reconstructs x from y. Additionally, some of these networks jointly learn a *sparsifying transform* for x. This sparsifier may either be a nonlinear transform [18] or an orthogonal matrix [22] – integrating that way a dictionary learning technique. The latter has shown promising results when employed in model-based CS [23, 24, 25]; hence, it looks appealing to combine it with unfolding networks. Furthermore, research community focuses lately on the *generalization error* [26, 27] of deep unfolding networks [28], [22], [29], [30]. Despite recent results of this kind, estimating the generalization error of unfolding networks for CS is still in its infancy.

In fact, generalization error bounds are only provided for unfolding networks that promote *synthesis sparsity* in CS, by means of the dictionary learning technique. On the other hand, the *analysis sparsity model* differs significantly [31] from its synthesis counterpart and it can be more advantageous for CS [32]. For example, the redundancy of so-called analysis operators can lead to a more flexible sparse representation of the signals of interest, compared to orthogonal sparsifying transforms (see Sections II-A and II-B for a detailed comparison between the two sparsity models). To the best of our knowledge, only one unfolding network [33] takes advantage of analysis sparsity in CS, in terms of learning a redundant sparsifying analysis operator. Nevertheless, the generalization ability of [33] is not mathematically explained.

In this paper, we are inspired by the articles [20], [22], [28], [29], [33]. These publications propose ISTA-based [2], reweighted FISTA-based [34] and ADMM-based [4] unfolding networks, which jointly learn a decoder for CS and a sparsifying transform. Particularly, the learnable sparsifiers of [20, 22, 28, 29] promote synthesis sparsity, while [33] employs its handier analysis counterpart. The deficiency of [20, 33] lies on the fact that their proposed frameworks are not accompanied by a generalization analysis, whereas [22, 28, 29] provide generalization error bounds for the proposed networks. Similarly, we develop a new unfolding network based on an optimal analysis-l_1 algorithm [35] and call it *Decoding Network* (DECONET). The latter jointly learns a decoder for CS and a redundant sparsifying analysis operator; thus, we address the CS problem under the analysis sparsity model. Our novelty lies on estimating the generalization error of the

V. Kouni acknowledges financial support for the implementation of this paper by Greece and the European Union (European Social Fund-ESF) through the Operational Program "Human Resources Development, Education and Lifelong Learning" in the context of the Act "Enhancing Human Resources Research Potential by undertaking a Doctoral Research" Sub-action 2: IKY Scholarship Program for PhD candidates in the Greek Universities

V. Kouni and Y. Panagakis are with Dep. of Informatics & Telecommunications, National & Kapodistrian University of Athens, Athens, Greece. Emails: vicky-kouni@di.uoa.gr and yannisp@di.uoa.gr

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors. The material includes a pdf file, which contains proofs of some theoretical results and link to a code repository for reproducibility purposes. This material is 134KB in size

DUN	Iterative Scheme	Sparsity Model	Gen. Error Bounds
ISTA-net [22]	Iterative Soft Thresholding Algorithm [2]	Synthesis	Yes
SISTA-RNN [19]	Sequential Iterative Soft Thresholding Algorithm [39]	Synthesis	No
Reweighted-RNN [29]	Reweighted $l_1 - l_1$ algorithm [29]	Synthesis	Yes
AMP-Net [16]	Approximate Message Passing [40]	Synthesis	No
ADMM-net [20]	Alternating Direction Method of Multipliers [4]	Synthesis	No
ADMM-DAD [33]	Alternating Direction Method of Multipliers [4]	Analysis	No
DECONET (proposed)	Analysis-l_1 [35]	Analysis	Yes

Table I: Comparisons among some example unfolding networks for CS. Categorizations are based on a) the associated optimization algorithm b) the employed sparsity model c) the study of generalization error.

proposed analysis-based unfolding network. To that end, we upper bound the generalization error of DECONET in terms of the Rademacher complexity [36] of the associated hypothesis class. In the end, we numerically test the validity of our theory and compare our proposed network to the state-of-the-art (SotA) unfolding networks of [22] and [33], on real-world and synthetic data. In all datasets, our proposed neural architecture outperforms the baselines in terms of generalization error, which scales in accordance with our theoretical results.

Our key contributions are listed below.

1) After differentiating synthesis from analysis sparsity in CS and presenting example unfolding networks in Section II, we develop in Section III a new unfolding network dubbed DECONET. The latter jointly learns a decoder that solves the analysis-based CS problem and a redundant sparsifying analysis operator $W \in \mathbb{R}^{N\times n}$, $n < N$, that is shared across the layers of DECONET.
2) We introduce in Section IV the hypothesis class – parameterized by W – of all the decoders DECONET can realize and restrict W to be bounded in this class, so that we impose a realistic structural constraint on the operator.
3) Later in Section IV, we estimate the generalization error of DECONET using a chaining technique. Our results showcase that the redundancy N of W and the number of layers L affect the generalization ability of DECONET; roughly speaking, the generalization error scales like $\sqrt{NL}$ (see Theorem IV.12 and Corollary IV.13). To the best of our knowledge, we are the first to study the generalization ability of an unfolding network for analysis-based CS.
4) We confirm the validity of our theoretical guarantees in Section V, by testing DECONET on a synthetic dataset and two real-world image datasets, i.e., MNIST [37] and CIFAR10 [38]. We also compare DECONET to two SotA unfolding networks: a recent variant of ISTA-net [22] and ADMM-DAD net [33]. Our experiments demonstrate that a) the generalization error of DECONET scales correctly with our theoretical findings b) DECONET outperforms both baselines, consistently for all datasets.

Notation. We denote the set of real, positive numbers by $\mathbb{R}_+$. For a sequence a_n that is upper bounded by $M > 0$, we write $\{a_n\} \leq M$. For a matrix $A \in \mathbb{R}^{n\times n}$, we write $\|A\|_{2\to 2}$ for its operator/spectral norm and $\|A\|_F$ for its Frobenius norm. Moreover, we write $\|A\|_{2\to 2} \approx 1$ if $\|A\|_{2\to 2} \geq 1$, but there exists $C > 0$ such that $C\|A\|_{2\to 2} \leq 1$. For a family of vectors $(w_i)_{i=1}^N$ in $\mathbb{R}^n$, its associated analysis operator is given by $Wf := \{\langle f, w_i\rangle\}_{i=1}^N$, where $f \in \mathbb{R}^n$. For square matrices $A_1, A_2 \in \mathbb{R}^{N\times N}$, we denote by $[A_1; A_2] \in \mathbb{R}^{2N\times N}$ their concatenation with respect to the first dimension, while we denote by $[A_1 \,|\, A_2] \in \mathbb{R}^{N\times 2N}$ their concatenation with respect to the second dimension. Similarly, for non-square matrices $A_1 \in \mathbb{R}^{m_1\times n}$, $A_2 \in \mathbb{R}^{m_2\times n}$, $A_3 \in \mathbb{R}^{m\times n_1}$, $A_4 \in \mathbb{R}^{m\times n_2}$, we denote by $[A_1; A_2] \in \mathbb{R}^{(m_1+m_2)\times n}$ the concatenation of A_1 and A_2 with respect to the first dimension, while we denote by $[A_3 \,|\, A_4] \in \mathbb{R}^{m\times(n_1+n_2)}$ the concatenation of A_3 and A_4 with respect to the second dimension. We write $O_{N\times N}$ for a real-valued $N \times N$ matrix filled with zeros and $I_{N\times N}$ for the $N \times N$ identity matrix. We denote by $\mathrm{diag}(\alpha)$ a square diagonal matrix having $\alpha \in \mathbb{R}$ in its main diagonal and zero elsewhere. For $x \in \mathbb{R}$, $\tau > 0$, the soft thresholding operator $\mathcal{S}_\tau : \mathbb{R} \mapsto \mathbb{R}$ is defined as

$$\mathcal{S}_\tau(x) = \mathcal{S}(x,\tau) = \begin{cases} \mathrm{sign}(x)(|x| - \tau), & |x| \geq \tau \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

or in closed form $\mathcal{S}(x,\tau) = \mathrm{sign}(x)\max(0, |x| - \tau)$. For $x \in \mathbb{R}^n$, the soft thresholding operator acts component-wise, i.e. $(\mathcal{S}_\tau(x))_i = \mathcal{S}_\tau(x_i)$, and is 1-Lipschitz with respect to x. For $y \in \mathbb{R}^n$, $\tau > 0$, the mapping

$$P_G(\tau; y) = \mathrm{argmin}_{x\in\mathbb{R}^n}\left\{\tau G(x) + \frac{1}{2}\|x - y\|_2^2\right\}, \tag{2}$$

is the *proximal mapping associated to the convex function G*. For $G(\cdot) = \|\cdot\|_1$, (2) coincides with (1). For $x \in \mathbb{R}$, $\tau > 0$, the truncation operator $\mathcal{T}_\tau : \mathbb{R} \mapsto \mathbb{R}$ is defined as

$$\mathcal{T}_\tau(x) = \mathcal{T}(x,\tau) = \begin{cases} \tau\mathrm{sign}(x), & |x| \geq \tau \\ x, & \text{otherwise} \end{cases}, \tag{3}$$

or in closed form $\mathcal{T}(x,\tau) = \mathrm{sign}(x)\min\{|x|, \tau\}$. For $x \in \mathbb{R}^n$, the truncation operator acts component-wise and is 1-Lipschitz with respect to x. For two functions $f, g : \mathbb{R}^n \mapsto \mathbb{R}^n$, we write their composition as $f \circ g : \mathbb{R}^n \mapsto \mathbb{R}^n$ and if there exists some constant $C > 0$ such that $f(x) \leq Cg(x)$, then we write $f(x) \lesssim g(x)$. For the ball of radius $t > 0$ in $\mathbb{R}^n$ with respect to some norm $\|\cdot\|$, we write $B^n_{\|\cdot\|}(t)$. The covering number $\mathcal{N}(T, \|\cdot\|, t)$ of a space T, equipped with a norm $\|\cdot\|$, at level $t > 0$, is defined as the smallest number of balls $B^n_{\|\cdot\|}(t)$ required to cover T. The set of all matrices $W \in \mathbb{R}^{N\times n}$ with operator norm bounded by some $0 < \Lambda < \infty$, is defined as $\mathcal{B}_\Lambda = \{W \in \mathbb{R}^{N\times n} : \|W\|_{2\to 2} \leq \Lambda\}$.

II. Related work:

FROM MODEL-BASED TO DATA-DRIVEN CS

The main idea of CS is to reconstruct a vector $x \in \mathbb{R}^n$ from measurements $y = Ax + e \in \mathbb{R}^m$, $m < n$, where A is the so-called *measurement matrix* [41] and $e \in \mathbb{R}^m$, with $\|e\| \leq \varepsilon$, corresponds to noise. In order to ensure exact/approximate reconstruction of x, CS relies on two principles. First, A must meet some conditions, for example the restricted isometry property or the null space property [41]. In particular, random Gaussian matrices $A \in \mathbb{R}^{m \times n}$ have proven to be nice candidates for CS, since they satisfy such conditions [41]. Second, we assume x is (approximately) sparse. Sparse data models are split in synthesis and analysis sparsity [42].

A. Synthesis Sparsity in CS and ISTA-based Unfolding

Under the synthesis sparsity model [41, 43, 44, 45], signals are considered to be sparse when *synthesized* by a few column vectors taken from a large matrix called *dictionary*, which is typically assumed to be orthogonal, i.e. $D \in \mathbb{R}^{n \times n}$, with $DD^T = I_{n \times n}$ (e.g. D may be the discrete cosine transform), so that $x = Dz$. Employing synthesis sparsity in CS, we aim to recover x from y. A common way to do so is by solving the l_1*-minimization* problem

$$\min_{z \in \mathbb{R}^n} \|z\|_1 \quad \text{s. t.} \quad \|y - ADz\|_2 \leq \varepsilon. \tag{4}$$

Towards that end, numerous iterative algorithms [2, 3, 46] have emerged. Typically, they consist of an iterative scheme that incorporates a proximal mapping and after a number of iterations and under certain conditions, they converge to a minimizer $\hat{x}$ of (4). For example, ISTA uses the proximal mapping (2) to yield the following iterative scheme

$$\begin{aligned} z_{k+1} =& \mathcal{S}_{\tau\lambda}(z_k + \tau(AD)^T(y - ADz)) \\ =& \mathcal{S}_{\tau\lambda}((I - D^T A^T AD)z + \tau(AD)^T y), \end{aligned} \tag{5}$$

for $k = 0, 1, \dots$, $z_0 = 0$, with $\tau, \lambda > 0$ being parameters of the algorithm. If $\tau\|AD\|_{2\to 2}^2 \leq 1$ [2], z_k converges to a minimizer $\hat{z}$ of (4), so that the reconstructed $\hat{x}$ is simply given by $\hat{x} = D\hat{z}$. As stated in [22], under the assumption that D is learned from a training set, the iterative scheme of (5) can be interpreted as a layer of a neural network (whose trainable parameters are the entries of D) with weight matrix $I - D^T A^T AD$, bias $\tau(AD)^T y$ and activation function $\mathcal{S}_{\tau\lambda}$. Then, the composition of a given number of layers and the consequent application of D constitutes the decoder implemented by the ISTA-based network, which outputs $\hat{x} \approx x$.

B. Analysis Sparsity in CS and ADMM-based Unfolding

Despite its success, synthesis sparsity has a “twin”, i.e., the *analysis sparsity model* [47, 48, 49], in which one assumes that there exists a *redundant analysis operator* $W \in \mathbb{R}^{N \times n}$, $n < N$, so that Wx is sparse. For example, W may be the analysis operator associated to a *frame* [50, 51] or a finite difference operator [52]. The associated optimization problem for CS is the *analysis* l_1*-minimization* problem

$$\min_{x \in \mathbb{R}^n} \|Wx\|_1 \quad \text{s. t.} \quad \|Ax - y\|_2 \leq \varepsilon. \tag{6}$$

From now on, whenever we speak about the *redundancy* of an analysis operator, we mean the number of its rows N. Analysis sparsity has become popular, due to some benefits it has compared to its synthesis counterpart. For example, it is computationally more appealing to solve the optimization algorithm of analysis-based CS, since the actual optimization takes place in the ambient space [53] and the algorithm involved may need less measurements for perfect reconstruction, if one uses a redundant transform instead of an orthogonal one [48]. Nevertheless, choosing the appropriate iterative algorithm for solving (6) may be a tricky task. The reason is that most thresholding algorithms cannot handle analysis sparsity, since the proximal mapping associated to $\|W(\cdot)\|_1$ does not have a closed-form type. To tackle this issue and solve (6), one may employ the so-called ADMM [4] algorithm, which uses the following iterative scheme

$$x^{k+1} = (A^T A + \rho W^T W)^{-1}(A^T y + \rho W^T(z^k - u^k)) \tag{7}$$

$$z^{k+1} = \mathcal{S}_{\lambda/\rho}(Wx^{k+1} - u^k) \tag{8}$$

$$u^{k+1} = u^k + Wx^{k+1} - z^{k+1}, \tag{9}$$

with $k \in \mathbb{N}$, dual variables $z, u \in \mathbb{R}^N$, initial points $(x^0, z^0, u^0) = (0, 0, 0)$, penalty parameter $\rho > 0$ and regularization parameter $\lambda > 0$. As shown in [4], the iterates (7) – (9) converge to a solution $p^\star$ of

$$\min_{x \in \mathbb{R}^n} \frac{1}{2}\|Ax - y\|_2^2 + \lambda\|z\|_1 \quad \text{s. t.} \quad Wx - z = 0. \tag{10}$$

In [33], the updates (7) - (9) are formulated as a neural network (whose trainable parameters are the entries of W) with L layers, defined as

$$g_1(y) = I_1 b(y) + I_2 \mathcal{S}_{\lambda/\rho}(b(y)), \tag{11}$$

$$g_k(v) = \tilde{\Theta} v + I_1 b + I_2 \mathcal{S}_{\lambda/\rho}(\Theta v + b), \quad k = 2, \dots, L, \tag{12}$$

with $v \in \mathbb{R}^{2N \times 1}$ being an *intermediate variable*, weight matrices $\tilde{\Theta}$, Θ (depending on A, W, ρ), bias b (depending on A, W, ρ, y) and activation function $\mathcal{S}_{\lambda/\rho}$. The resulting learned decoder that reconstructs x from y emerges by applying an affine map on the composition of a given number of layers.

III. DECONET: A NEW UNFOLDING NETWORK FOR ANALYSIS-BASED CS

The generic ADMM-based unfolding network of [33] seems promising, but it involves the costly computation of $(A^T A + \rho W^T W)^{-1}$. On the other hand, ADMM-based unfolding networks may deal with the bottleneck of the inverses' computation by leveraging some structure of the problem [54] or imposing specific constraints [55]. Nevertheless, such unfolding networks are synthesis-based and thus they cannot treat analysis sparsity. Therefore, in order to address both aspects, we opt for solving (6) with the optimal first-order *analysis-*l_1 algorithm described in [35], which uses transposes – instead of inverses – of the involved matrices. We will briefly describe the steps leading to the derivation of the aforementioned algorithm, as these are stated in [35]. First, an equivalent to (6) *smoothed* formulation is given as follows:

$$\min_{x \in \mathbb{R}^n} \|Wx\|_1 + \frac{\mu}{2}\|x - x_0\|_2^2 \quad \text{s. t.} \quad \|y - Ax\|_2 \leq \varepsilon, \tag{13}$$

where $\mu \in \mathbb{R}_+$ is the so-called *smoothing parameter* and $x_0 \in \mathbb{R}^n$ is an initial guess on x. Second, the dual of (13) is determined to be

$$\begin{aligned} \max_{z^2\in\mathbb{R}^m} \quad & \langle y, z^2\rangle - \varepsilon\|z^2\|_2 \\ \text{s. t.} \quad & A^T z^2 - W^T z^1 = 0, \ \|z^1\|_\infty \leq 1, \end{aligned} \tag{14}$$

where $z^1 \in \mathbb{R}^N$, $z^2 \in \mathbb{R}^m$ are dual variables. The afore-described formulations – combined with a collection of arguments and computations – lead to Algorithm 1, which constitutes a variant of an optimal first-order method. For reasons of convenience, we call this algorithm *analysis conic form* (ACF) from now on. ACF also involves step sizes $\{t_k^1\}$, $\{t_k^2\} > 0$ and a step size multiplier $0 < \{\theta_k\}$. A standard setup for ACF employs update rules such that $0 < \{t_k^1\}_{k\geq 0}$, $\{t_k^2\}_{k\geq 0} \leq 1$, $0 < \{\theta_k\}_{k\geq 0} \leq 1$.

```
Algorithm 1: ACF
Input  : $x_0 \in \mathbb{R}^n$, $z_0^1 \in \mathbb{R}^N$, $z_0^2 \in \mathbb{R}^m$, $\mu \in \mathbb{R}_+$, step
         sizes $\{t_k^1\}, \{t_k^2\}$
Output: solution $\hat{x}_\mu$ of (13)
$\theta_0 \leftarrow 1$, $u_0^1 = z_0^1$, $u_0^2 = z_0^2$;
for iterations $k = 0, 1, \ldots$ do
    $x_k \leftarrow x_0 + \mu^{-1}((1-\theta_k)W^T u_k^1 + \theta_k W^T z_k^1 - (1-\theta_k)A^T u_k^2 - \theta_k A^T z_k^2)$;
    $z_{k+1}^1 \leftarrow \mathcal{T}((1-\theta_k)u_k^1 + \theta_k z_k^1 - \theta_k^{-1}t_k^1 W x_k, \theta_k^{-1}t_k^1)$;
    $z_{k+1}^2 \leftarrow$
        $\mathcal{S}((1-\theta_k)u_k^2 + \theta_k z_k^2 - \theta_k^{-1}t_k^2(y - Ax_k), \theta_k^{-1}t_k^2\varepsilon)$;
    $u_{k+1}^1 \leftarrow (1-\theta_k)u_k^1 + \theta_k z_{k+1}^1$;
    $u_{k+1}^2 \leftarrow (1-\theta_k)u_k^2 + \theta_k z_{k+1}^2$;
    $\theta_{k+1} \leftarrow 2/(1 + (1 + 4/(\theta_k)^2)^{1/2})$;
end
```

The dual function g_μ corresponding to (14) has a Lipschitz continuous gradient, hence ACF converges [35] to a solution $\hat{x}_\mu$ of (13), for which we have $\hat{x}_\mu \xrightarrow{\mu\to 0} \hat{x}$, where $\hat{x}$ is an optimal solution of (6). The authors of [35] clarify that when they speak about the optimal solution $\hat{x}$, they refer to this uniquely determined value. Additionally, they argue that there are situations where $\hat{x}$ and $\hat{x}_\mu$ coincide. Henceforward, we stick to their formulation and speak about the solution $\hat{x}$.

We consider a standard scenario for the ACF, where $z_0^1 = u_0^1 = 0$, $z_0^2 = u_0^2 = 0$, $t_0^1 = t_0^2 = \theta_0 = 1$, $0 < \{t_k^1\}, \{t_k^2\}, \{\theta_k\} \leq 1$, $\mu > 1$, $x_0 = A^T y$ and $A \in \mathbb{R}^{m\times n}$ is an appropriately normalized random matrix (which constitutes a typical choice for CS), with $\|A\|_{2\to 2} \approx 1$. We substitute first $x-$update into z^1- and z^2-updates and second z^1- and z^2- into u^1- and u^2-updates, respectively, concatenate $z_k^1, z_k^2, u_k^1, u_k^2$ in one vector v_k, i.e. $v_k^T = (z_k^1, z_k^2, u_k^1, u_k^2)^T \in \mathbb{R}^{(1\times p)}$, $p = 2N + 2m$, for $k \geq 0$, with $v_0 = 0$, and do the calculations, so that

$$v_{k+1} = D_k v_k + \Theta_k \begin{pmatrix} \mathcal{T}(G_k^1 v_k - b_k^1, \theta_k^{-1}t_k^1) \\ \mathcal{S}(G_k^2 v_k - b_k^2, \theta_k^{-1}t_k^2\varepsilon) \\ \mathcal{T}(G_k^1 v_k - b_k^1, \theta_k^{-1}t_k^1) \\ \mathcal{S}(G_k^2 v_k - b_k^2, \theta_k^{-1}t_k^2\varepsilon) \end{pmatrix}, \tag{15}$$

where

$$b_k^1 = \theta_k^{-1}t_k^1 W x_0 \in \mathbb{R}^N, \tag{16}$$

$$b_k^2 = \theta_k^{-1}t_k^2 (y - Ax_0) \in \mathbb{R}^m, \tag{17}$$

$$\Theta_k = \operatorname{diag}(\underbrace{\theta_0, \ldots, \theta_0}_{(N+m)\ \text{times}}, \underbrace{\theta_k, \ldots, \theta_k}_{(N+m)\ \text{times}}) \in \mathbb{R}^{p\times p}, \tag{18}$$

$$D_k = (I_{p\times p} - \Theta_k) \in \mathbb{R}^{p\times p}, \tag{19}$$

$$\begin{aligned} G_k^1 =& (\theta_k(I - \theta_k^{-1}t_k^1\mu^{-1}WW^T) \mid t_k^1\mu^{-1}WA^T \\ & \mid (1-\theta_k)(I - \theta_k^{-1}t_k^1\mu^{-1}WW^T) \\ & \mid (1-\theta_k)\theta_k^{-1}t_k^1\mu^{-1}WA^T) \in \mathbb{R}^{N\times p}, \end{aligned} \tag{20}$$

$$\begin{aligned} G_k^2 =& (t_k^2\mu^{-1}AW^T \mid \theta_k(I - \theta_k^{-1}t_k^2\mu^{-1}AA^T) \\ & \mid (1-\theta_k)\theta_k^{-1}t_k^2\mu^{-1}AW^T \mid (1-\theta_k) \\ & \cdot (I - \theta_k^{-1}t_k^2\mu^{-1}AA^T)) \in \mathbb{R}^{m\times p}. \end{aligned} \tag{21}$$

We observe that (15) can be interpreted as a layer of a neural network, with weights G^1, G^2, biases b^1, b^2 and activation functions $\mathcal{T}(\cdot,\cdot)$, $\mathcal{S}(\cdot,\cdot)$. Nevertheless, this interpretation of ACF as a DNN does not account for any trainable parameters. We cope with this issue by considering W to be unknown and learned from a training sequence $\mathbf{S} = \{(x_i, y_i)\}_{i=1}^s$ with i.i.d. samples drawn from an unknown distribution[1] $\mathcal{D}^s$. Hence, the trainable parameters are the entries of W. Additionally, we make the realistic assumption that W is bounded with respect to the operator norm, i.e., $W \in \mathcal{B}_\Lambda$. Based on (15), we formulate ACF as a neural network with L layers/iterations, defined as

$$f_1(y) = \sigma_1(y) \tag{22}$$

$$f_k(v) = D_{k-1}v + \Theta_{k-1}\sigma_{k-1}(v), \quad k = 2, \ldots, L, \tag{23}$$

where

$$\begin{aligned} \sigma_1(y)^T =& (\mathcal{T}(-t_0^1 W x_0, t_0^1), \mathcal{S}(t_0^2(y - Ax_0), t_0^2\varepsilon), \\ & \mathcal{T}(-t_0^1 W x_0, t_0^1), \mathcal{S}(t_0^2(y - Ax_0), t_0^2\varepsilon))^T, \end{aligned} \tag{24}$$

$$\begin{aligned} \sigma_k(v)^T =& (\mathcal{T}(G_k^1 v - b_k^1), \mathcal{S}(G_k^2 v - b_k^2), \\ & \mathcal{T}(G_k^1 v - b_k^1), \mathcal{S}(G_k^2 v - b_k^2))^T, \end{aligned} \tag{25}$$

for $k = 2, \ldots, L$. We denote the composition of L such layers (all having the same W) as

$$f_W^L(y) = f_L \circ f_{L-1} \circ \cdots \circ f_1(y). \tag{26}$$

The latter constitutes the realization of a neural network with L layers, that reconstructs the intermediate variable v from y. Thus, we call (26) *intermediate decoder*. Motivated by the x-update of ACF, we apply an affine map $\phi : \mathbb{R}^{p\times 1} \mapsto \mathbb{R}^{n\times 1}$ after the last layer L, yielding the desired solution $\hat{x}$:

$$\hat{x} := \phi(v) = \Phi v + x_0, \tag{27}$$

where

$$\begin{aligned} \Phi = & (\mu^{-1}\theta_L W^T | - \mu^{-1}\theta_L A^T | \ \mu^{-1}(1-\theta_L)W^T \\ & | - \mu^{-1}(1-\theta_L)A^T) \in \mathbb{R}^{n\times p}. \end{aligned} \tag{28}$$

[1] Formally speaking, this is a distribution over the x_i and then $y_i = Ax_i + e$, with fixed A, e

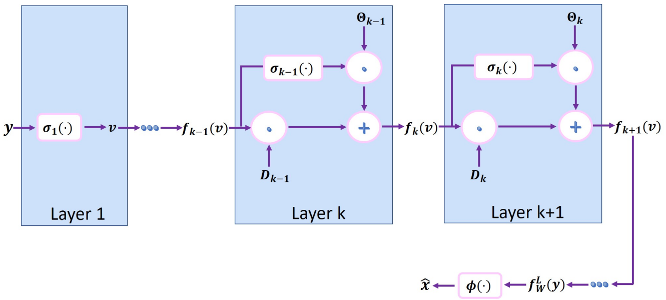


Fig. 1: Schematic illustration of (22), (23), (26) and (27). Note that y is passed as input into every subsequent layer $k = 2, \ldots, L$, but for notational simplification, we only write the intermediate variable v. The operators " $\cdot$ " and " $+$ " denote matrix-vector multiplication and vector-vector addition, respectively, while the activation functions $\sigma_1(\cdot)$ and $\sigma_k(\cdot)$ ($k = 2, \ldots, L$) are defined as in (24) and (25), respectively.

Moreover, in order to clip the output $\phi(f_W^L(y))$ in case its norm falls out of a reasonable range, we apply an extra function $\psi : \mathbb{R}^n \to \mathbb{R}^n$ after ϕ and define it as

$$\psi(x) = \left\{ \begin{array}{cc} x, & \|x\|_2 \leq B_{\text{out}} \\ B_{\text{out}} \frac{x}{\|x\|_2}, & \text{otherwise} \end{array} \right. , \tag{29}$$

for some constant $B_{\text{out}} > 0$. For fixed L, the desired learnable decoder is written as

$$\text{dec}_W^L(y) = \psi(\phi(f_W^L(y))). \tag{30}$$

We call Decoding Network (DECONET) the network that implements such a decoder, which is parameterized by W.

IV. Generalization Analysis of DECONET

In this Section, we deliver meaningful – in terms of L and N – upper bounds on the generalization error of DECONET. We do so in a series of steps presented in the next subsections.

A. Hypothesis Class of DECONET and Associated Rademacher Complexity

We introduce the hypothesis class

$$\mathcal{H}^L = \{h : \mathbb{R}^m \mapsto \mathbb{R}^n : \, h(y) = \psi(\phi(f_W^L(y))), \, W \in \mathcal{B}_\Lambda\}, \tag{31}$$

parameterized by W and consisting of all the functions/decoders DECONET can implement. Given (31) and the training set $\mathbf{S}$, DECONET yields a function $h_{\mathbf{S}} \in \mathcal{H}^L$ that aims at reconstructing x from y. For a loss function $\ell : \mathcal{H}^L \times \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}_+$, the empirical loss of a hypothesis $h \in \mathcal{H}^L$ is the reconstruction error on the training set, i.e.

$$\hat{\mathcal{L}}_{train}(h) = \frac{1}{s} \sum_{i=1}^{s} \ell(h, x_i, y_i). \tag{32}$$

In this paper, we choose as loss function ℓ the squared l_2-norm; hence, (32) takes the form of the training *mean-squared error* (MSE):

$$\hat{\mathcal{L}}_{train}(h) = \frac{1}{s} \sum_{j=1}^{s} \|h(y_j) - x_j\|_2^2. \tag{33}$$

The true loss is

$$\mathcal{L}(h) = \mathbb{E}_{(x,y)\sim\mathcal{D}}(\|h(y) - x\|_2^2). \tag{34}$$

The generalization error is given as the difference[2] between the empirical and true loss

$$\text{GE}(h) = |\hat{\mathcal{L}}_{train}(h) - \mathcal{L}(h)|. \tag{35}$$

A typical way to estimate (35) consists in upper bounding it in terms of the *Rademacher complexity* [56, Definition 2]. The *empirical Rademacher complexity* is defined as

$$\mathcal{R}_{\mathbf{S}}(\ell \circ \mathcal{H}^L) = \mathbb{E} \sup_{h \in \mathcal{H}^L} \frac{1}{s} \sum_{i=1}^{s} \epsilon_i \|h(y_i) - x_i\|_2^2, \tag{36}$$

where ϵ is a Rademacher vector, that is, a vector with i.i.d. entries taking the values ± 1 with equal probability. Then, the Rademacher complexity is defined as

$$\mathcal{R}_s(\ell \circ \mathcal{H}^L) = \mathbb{E}_{\mathbf{S}\sim\mathcal{D}^s}(\mathcal{R}_{\mathbf{S}}(\ell \circ \mathcal{H}^L)). \tag{37}$$

In this paper, we solely work with (36). We rely on the following Theorem that estimates (35) in terms of (36).

Theorem IV.1 ([26, Theorem 26.5]). *Let $\mathcal{H}$ be a family of functions, $\mathbf{S}$ the training set drawn from $\mathcal{D}^s$, and ℓ a real-valued bounded loss function satisfying $|\ell(h,z)| \leq c$, for all $h \in \mathcal{H}, z \in Z$. Then, for $\delta \in (0,1)$, with probability at least $1 - \delta$, we have for all $h \in \mathcal{H}$*

$$\mathcal{L}(h) \leq \hat{\mathcal{L}}(h) + 2\mathcal{R}_{\mathbf{S}}(\ell \circ \mathcal{H}) + 4c\sqrt{\frac{2\log(4\delta)}{s}}. \tag{38}$$

In order to use the previous Theorem, the loss function must be bounded. Towards this end, we make two typical – for the machine learning literature – assumptions regarding the training set $\mathbf{S} = \{(x_i, y_i)\}_{i=1}^s$. Let us suppose that with overwhelming probability we have

$$\|y_i\|_2 \leq \text{B}_{\text{in}}, \tag{39}$$

for some constant $\text{B}_{\text{in}} > 0$, $i = 1, \ldots, s$. Moreover, we assume that for any $h \in \mathcal{H}^L$, with overwhelming probability over y_i chosen from $\mathcal{D}$, the following holds

$$\|h(y_i)\|_2 \leq B_{\text{out}}, \tag{40}$$

by definition of ψ, for some constant $B_{\text{out}} > 0$, for all $i = 1, \ldots, s$. Hence, $\| \cdot \|_2^2$ is bounded as $\|h(y_i) - x_i\|_2^2 \leq (B_{\text{in}} + B_{\text{out}})^2$, $i = 1, \ldots, s$. Following the previous assumptions, it is easy to check that $\| \cdot \|_2^2$ is a Lipschitz continuous function, with Lipschitz constant $\text{Lip}_{\|\cdot\|_2^2} = 2\text{B}_{\text{in}} + 2\text{B}_{\text{out}}$; hence, we can employ the so-called (vector-valued) contraction principle, which allows us to study $\mathcal{R}_{\mathbf{S}}(\mathcal{H})$ alone:

Lemma IV.2 ([57, Corollary 4]). *Let $\mathcal{H}$ be a set of function $h : \mathcal{X} \mapsto \mathbb{R}^n$, $f : \mathbb{R}^n \mapsto \mathbb{R}^n$ a K-Lipschitz function and $\mathbf{S} = \{x_i\}_{i=1}^s$. Then*

$$\mathbb{E} \sup_{h \in \mathcal{H}} \sum_{i=1}^{s} \epsilon_i f \circ h(x_i) \leq \sqrt{2} K \mathbb{E} \sup_{h \in \mathcal{H}} \sum_{i=1}^{s} \sum_{k=1}^{n} \epsilon_{ik} h_k(x_i), \tag{41}$$

[2] Some of the existing literature denotes the true loss as the generalization error, but the definition we give in (35) is more convenient for our purposes

where $(\epsilon_i), (\epsilon_{ik})$ are both Rademacher sequences.

Applying Lemma IV.2 in $\mathcal{R}_{\mathbf{S}}(\ell \circ \mathcal{H})$ yields:

$$\begin{aligned}\mathcal{R}_{\mathbf{S}}(l \circ \mathcal{H}^L) \leq& \sqrt{2}\mathrm{Lip}_{\|\cdot\|_2^2}\mathcal{R}_{\mathbf{S}}(\mathcal{H}^L)\\ =&\sqrt{2}\mathrm{Lip}_{\|\cdot\|_2^2}\mathbb{E}\sup_{h\in\mathcal{H}^L}\sum_{i=1}^{s}\sum_{k=1}^{n}\epsilon_{ik}h_k(x_i)\\ =&\mathrm{B}_{\mathrm{in}}^{\mathrm{out}}\mathbb{E}\sup_{h\in\mathcal{H}^L}\sum_{i=1}^{s}\sum_{k=1}^{n}\epsilon_{ik}h_k(x_i),\end{aligned}\tag{42}$$

where $\mathrm{B}_{\mathrm{in}}^{\mathrm{out}} = \sqrt{2}(2\mathrm{B}_{\mathrm{in}} + 2\mathrm{B}_{\mathrm{out}})$. We return to estimating (42) in Section IV-D, after presenting the adequate mathematical tools in Sections IV-B and IV-C.

B. Boundedness of DECONET's Outputs

We take into account the number of training samples and pass to matrix notation. Due to (39) and the Cauchy-Schwartz inequality, we get $\|Y\|_F \leq \sqrt{s}\mathrm{B}_{\mathrm{in}}$. Similarly, the application of the Cauchy-Schwartz inequality in (40) yields

$$\|h(Y)\|_F = \|\psi(\phi(f_W^L(Y)))\|_F \leq \sqrt{s}\mathrm{B}_{\mathrm{out}}. \tag{43}$$

Assumption IV.3. *Since there exist redundant analysis operators W for which Λ may be relatively small [50, Corollary 6.2.3.], [58, Section V], [59, Proposition 5.1], [60, Lemma 1], we may reasonably assume that $c_{1,k}\Lambda \leq 1$ and $c_{1,k}\Lambda^2 \leq 1$, for all $k \geq 0$. This simplifying assumption will hold for the remainder of this paper.*

The next Lemma presents bounds on a quantity we will encounter more often in the sequel.

Lemma IV.4 (Proof in the supplementary material). *Let $k \geq 0$. For any $W \in \mathcal{B}_\Lambda$, step sizes $0 < \{t_k^1\}, \{t_k^2\} \leq 1$ with $t_0^1 = t_0^2 = 1$, step size multiplier $0 < \{\theta_k\} \leq 1$ with $\theta_0 = 1$, and smoothing parameter $\mu > 1$, the following holds for the matrices G_k^1, G_k^2 defined in (20), (21), respectively:*

$$2\|G_k^1\|_{2\to2} + 2\|G_k^2\|_{2\to2} + 1 \leq \Gamma_k, \tag{44}$$

where

$$\begin{aligned}\Gamma_k = 2\big[&c_{1,k}\Lambda^2 + c_{2,k}\|A\|_{2\to2}^2\\ &+ 2\|A\|_{2\to2}\Lambda(c_{1,k} + c_{2,k})\big] + 1,\end{aligned}\tag{45}$$

with $\{c_{1,k}\} = \{\theta_k^{-1}\mu^{-1}t_k^1\} \leq 1$, $\{c_{2,k}\} = \{\theta_k^{-1}\mu^{-1}t_k^2\} \leq 1$, for all $k \geq 0$. Moreover, if $c_{1,k}\Lambda \leq 1$, $c_{1,k}\Lambda^2 \leq 1$, $c_{2,k}\|A\|_{2\to2}^2 \leq 1$, then $\Gamma_k \leq \gamma$ for all $k \geq 0$, with $\gamma = 4(\Lambda + \|A\|_{2\to2} + 1) + 1$.

Apart from (43), we can upper-bound the output $f_W^k(Y)$ with respect to the Frobenius norm, after any number of layers k and especially for $k < L$, so that ϕ and ψ are not applied after the final layer L.

Lemma IV.5. *Let $k \in \mathbb{N}$. For any $W \in \mathcal{B}_\Lambda$, step sizes $0 < \{t_k^1\}, \{t_k^2\} \leq 1$ with $t_0^1 = t_0^2 = 1$, $t_{-1}^1 = t_{-1}^2 = 0$, step size multiplier $0 < \{\theta_k\} \leq 1$ with $\theta_0 = \theta_{-1} = 1$, and smoothing parameter $\mu > 1$, the following holds for the output of the functions f_W^k defined in (22) - (23):*

$$\|f_W^k(Y)\|_F \leq 2\mu\|Y\|_F\left(\sum_{i=0}^{k-1}\left(Q_{i-1}\prod_{j=i}^{k-1}\Gamma_j\right) + Q_{k-1}\right), \tag{46}$$

where $Q_k = \|A\|_{2\to2}(c_{1,k}\Lambda + c_{2,k}\|A\|_{2\to2}) + c_{2,k}$, $k \geq 0$, with $Q_{-1} = 0$ and $\{\Gamma_k\}_{k\geq0}$, $\{c_{1,k}\}_{k\geq0}$, $\{c_{2,k}\}_{k\geq0} \leq 1$ are defined as in Lemma IV.4. Moreover, if $c_{1,k}\Lambda \leq 1$, $c_{1,k}\Lambda^2 \leq 1$, $c_{2,k}\|A\|_{2\to2}^2 \leq 1$, then we have the simplified upper bound

$$\|f_W^k(Y)\|_F \leq 2\mu\|Y\|_F(\|A\|_{2\to2} + 1)(\zeta_k + 1), \tag{47}$$

where $\zeta_k = \frac{\gamma^k - 1}{\gamma - 1}$, with γ defined as in Lemma IV.4.

Proof. By definition of (1) and (3), $\mathcal{S}(\cdot,\cdot)$ and $\mathcal{T}(\cdot,\cdot)$ are 1-Lipschitz functions with respect to their first parameter, respectively. For the matrices Θ_k and D_k in (18) and (19) respectively, we have $\|D_k\|_{2\to2} \leq 1$ and $\|\Theta_k\|_{2\to2} = 1$, for any $k \geq 1$, since $0 < \{\theta_k\} \leq 1$ and $\theta_0 = 1$. We use the previous statements, along with (22) and (23), to prove (46) via induction. For $k = 1$, we have

$$\begin{aligned}\|f_W^1(Y)\|_F \leq& 2t_0^1\Lambda\|X_0\|_F + 2t_0^2(\|Y\|_F + \|A\|_{2\to2}\|X_0\|_F)\\ =&2\mu c_{1,0}\Lambda\|A\|_{2\to2}\|Y\|_F\\ &+ 2\mu c_{2,0}(\|Y\|_F + \|A\|_{2\to2}^2\|Y\|_F)\\ =&2\mu\|Y\|_F\big(\|A\|_{2\to2}(c_{1,0}\Lambda + c_{2,0}\|A\|_{2\to2}) + c_{2,0}\big).\end{aligned}$$

Suppose (46) holds for k. Then, for $k+1$:

$$\begin{aligned}\|f_W^{k+1}(Y)\|_F <& \|f_W^k(Y)\|_F + 2\|G_k^1 f_W^k(Y) - B_k^1\|_F\\ &+ 2\|G_k^2 f_W^k(Y) - B_k^2\|_F\\ \leq& \|f_W^k(Y)\|_F(2\|G_k^1\|_{2\to2} + 2\|G_k^2\|_{2\to2} + 1)\\ &+ 2(\|B_k^1\|_F + \|B_k^2\|_F)\\ \leq& \Gamma_k\|f_W^k(Y)\|_F + 2\mu\|X_0\|_F(c_{1,k}\Lambda\\ &+ c_{2,k}\|A\|_{2\to2}) + 2\mu c_{2,k}\|Y\|_F\\ \leq& \Gamma_k 2\mu\|Y\|_F\sum_{i=0}^{k-1}\left(\left(Q_{i-1}\right)\prod_{j=i}^{k-1}\Gamma_j\right)\\ &+ \Gamma_k 2\mu\|Y\|_F Q_{k-1} + 2\mu\|Y\|_F Q_k\\ =& 2\mu\|Y\|_F\left[\sum_{i=0}^{k}\left(Q_{i-1}\prod_{j=i}^{k}\Gamma_j\right) + Q_k\right],\end{aligned}$$

where in the forth inequality we set $Q_k = \|A\|_{2\to2}(c_{1,k}\Lambda + c_{2,k}\|A\|_{2\to2}) + c_{2,k}$ for all $k \geq 0$ and applied Lemma IV.4. Therefore, we proved that (46) holds for any $k \in \mathbb{N}$. Under the additional assumptions $c_{1,k}\Lambda \leq 1$, $c_{1,k}\Lambda^2 \leq 1$, $c_{2,k}\|A\|_{2\to2}^2 \leq 1$, for any $k \geq 0$, we may apply Lemma IV.4 on (46), yielding

$$\begin{aligned}\|f_W^k(Y)\|_F &\leq 2\mu\|Y\|_F(\|A\|_{2\to2} + 1)\left(\sum_{i=0}^{k-1}\prod_{j=i}^{k-1}\gamma + 1\right)\\ &= 2\mu\|Y\|_F(\|A\|_{2\to2} + 1)\left(\sum_{i=1}^{k}\gamma^i + 1\right)\\ &= 2\mu\|Y\|_F(\|A\|_{2\to2} + 1)(\zeta_k + 1),\end{aligned}$$

with $\zeta_k = \frac{\gamma^k - 1}{\gamma - 1}$ and γ defined as in Lemma IV.4. □

C. Lipschitzness Results

In the next Theorem, we prove that the intermediate decoder (26) is Lipschitz continuous with respect to W and explicitly calculate the Lipschitz constants, which depend on L.

Theorem IV.6 (Proof in the supplementary material)**.** *Let f_W^L defined as in (26), $L \geq 2$, dictionary $W \in \mathcal{B}_\Lambda$, step sizes $0 < \{t_k^1\}_{k\geq 0}, \{t_k^2\}_{k\geq 0} \leq 1$ with $t_0^1 = t_0^2 = 1$, $t_{-1}^1 = t_{-1}^2 = 0$, step size multiplier $0 < \{\theta_k\}_{k\geq 0} \leq 1$ with $\theta_0 = \theta_{-1} = 1$, and smoothing parameter $\mu > 1$. Then, for any $W_1, W_2 \in \mathcal{B}_\Lambda$, we have*

$$\|f_{W_1}^L(Y) - f_{W_2}^L(Y)\|_F \leq K_L \|W_1 - W_2\|_{2\to 2}, \tag{48}$$

where

$$\begin{aligned} K_L =& 2\mu\|Y\|_F \Bigg[\mu^{-1}\|A\|_{2\to 2} + \sum_{k=2}^{L}\Bigg(\bigg(\max_{0\leq l\leq L-1}\Gamma_l\bigg)^{L-k} \\ &\cdot \sum_{i=0}^{k-2} 2\bigg(Q_{i-1}\prod_{j=i}^{k-2}\Gamma_j\bigg) + 2Q_{k-1}(2\Lambda c_{1,k-1} + \|A\|_{2\to 2} \\ &\cdot (c_{1,k-1} + c_{2,k-1})) + c_{1,k-1}\|A\|_{2\to 2}\Bigg)\Bigg], \end{aligned} \tag{49}$$

with $\{Q_k\}_{k\geq 0}$ ($Q_{-1} = 0$) defined as in Lemma IV.5, $\{\Gamma_k\}_{k\geq 0}$, $\{c_{1,k}\}_{k\geq 0}$, $\{c_{2,k}\}_{k\geq 0}$ defined as in Lemma IV.4 and $c_{1,-1} = c_{2,-1} = 0$. Moreover, if $c_{1,k}\Lambda \leq 1$, $c_{1,k}\Lambda^2 \leq 1$, $c_{2,k}\|A\|_{2\to 2}^2 \leq 1$, for all $k \geq 0$, then we have the simplified upper bound

$$\begin{aligned} K_L \leq& 2\mu\|Y\|_F\Big[\|A\|_{2\to 2}(L - 1 + \mu^{-1}) \\ &+ 2(\|A\|_{2\to 2} + 1)(\|A\|_{2\to 2} + 3)\kappa_L\Big], \end{aligned} \tag{50}$$

where

$$\kappa_L = \gamma^L\left(\frac{L-1}{\gamma(\gamma-1)} + \frac{\gamma(\gamma-2)}{(\gamma-1)^2}\right) - \frac{\gamma^2(\gamma-2)}{(\gamma-1)^2}, \tag{51}$$

with γ as in Lemma IV.4.

We also prove below the Lipschitzness of the main decoder defined in (30).

Corollary IV.7. *Let $h \in \mathcal{H}^L$ defined as in (31), $L \geq 2$, and dictionary $W \in \mathcal{B}_\Lambda$. Then, for any $W_1, W_2 \in \mathcal{B}_\Lambda$, we have:*

$$\begin{aligned} &\|\psi(\phi(f_{W_2}^L(Y))) - \psi(\phi(f_{W_1}^L(Y)))\|_F \\ &\qquad \leq \mu^{-1}(\Lambda + \|A\|_{2\to 2})K_L\|W_2 - W_1\|_F, \end{aligned} \tag{52}$$

with K_L as in Theorem IV.6.

Proof. By definition, ψ is a 1-Lipschitz function. Moreover, as an affine map, ϕ is Lipschitz continuous with Lipschitz constant $\mathrm{Lip}_\phi = \|\Phi\|_{2\to 2}$, with Φ defined as in (28). We evaluate $\|\Phi\|_{2\to 2}$:

$$\begin{aligned} \|\Phi\|_{2\to 2} \leq& \mu^{-1}\theta_L\|W\|_{2\to 2} + \mu^{-1}\theta_L\|A\|_{2\to 2} \\ &+ \mu^{-1}(1-\theta_L)\|W\|_{2\to 2} + \mu^{-1}(1-\theta_L)\|A\|_{2\to 2} \\ \leq& \mu^{-1}(\Lambda + \|A\|_{2\to 2}). \end{aligned}$$

Combining the previous estimate with Theorem IV.6, we get

$$\begin{aligned} &\|\psi(\phi(f_{W_2}^L(Y))) - \psi(\phi(f_{W_1}^L(Y)))\|_F \\ &\quad \leq \|\phi(f_{W_2}^L(Y)) - \phi(f_{W_1}^L(Y))\|_F \\ &\quad \leq \|\Phi\|_{2\to 2}\|f_{W_2}^L(Y) - f_{W_1}^L(Y)\|_F \\ &\quad \leq \mu^{-1}(\Lambda + \|A\|_{2\to 2})K_L\|W_2 - W_1\|_F. \qquad \square \end{aligned}$$

D. Covering Numbers and Dudley's Inequality

For a fixed number of layers $L \in \mathbb{N}$, we define the set $\mathcal{M} \subset \mathbb{R}^{n\times s}$ corresponding to the hypothesis class $\mathcal{H}^L$ to be

$$\begin{aligned} \mathcal{M} :&= \{(h(y_1)|h(y_2)|\dots|h(y_s)) \in \mathbb{R}^{n\times s}:\ h \in \mathcal{H}^L\} \\ &= \{\psi(\phi((f_W^L(Y))) \in \mathbb{R}^{n\times s}:\ W \in \mathcal{B}_\Lambda\}. \end{aligned} \tag{53}$$

The column elements of each matrix in $\mathcal{M}$ are the reconstructions given by a decoder $h \in \mathcal{H}^L$ when applied to the measurements y_i. Since $\mathcal{M}$ is parameterized by W like $\mathcal{H}^L$ is, we may rewrite (42) as

$$\mathcal{R}_{\mathbf{S}}(l \circ \mathcal{H}^L) \leq \mathrm{B}_{\text{in}}^{\text{out}}\mathbb{E}\sup_{M\in\mathcal{M}}\frac{1}{s}\sum_{i=1}^{s}\sum_{k=1}^{n}\epsilon_{ik}M_{ik}. \tag{54}$$

Thus, we are left with estimating the Rademacher process in the right hand side of (54). The latter has subgaussian increments, hence we use Dudley's inequality [41, Theorem 8.23], [61, Theorem 5.23] to upper bound it in terms of the covering numbers of $\mathcal{M}$. Towards that end, we calculate the radius of $\mathcal{M}$, that is,

$$\begin{aligned} \Delta(\mathcal{M}) &= \sup_{h\in\mathcal{H}^L}\sqrt{\mathbb{E}\left(\sum_{i=1}^{s}\sum_{k=1}^{n}\epsilon_{ik}h_k(y_i)\right)^2} \\ &\leq \sup_{h\in\mathcal{H}^L}\sqrt{\mathbb{E}\sum_{i=1}^{s}\sum_{k=1}^{n}\epsilon_{ik}(h_k(y_i))^2} \\ &\leq \sup_{h\in\mathcal{H}^L}\sqrt{\sum_{i=1}^{s}\|h(y_i)\|_2^2} \overset{(43)}{\leq} \sqrt{s}B_{\text{out}}. \end{aligned} \tag{55}$$

With (55) in hand, applying Dudley's inequality to (54) yields

$$\begin{aligned} \mathcal{R}_{\mathbf{S}}(l \circ \mathcal{H}^L) \leq& \frac{16(\mathrm{B}_{\text{in}} + \mathrm{B}_{\text{out}})}{s} \\ &\cdot \int_0^{\frac{\sqrt{s}B_{\text{out}}}{2}}\sqrt{\log\mathcal{N}(\mathcal{M}, \|\cdot\|_F, \varepsilon)}d\varepsilon. \end{aligned} \tag{56}$$

Lemma IV.8.

$$\mathcal{N}(B^{N\times n}_{\|\cdot\|_{2\to 2}}(\Lambda), \|\cdot\|_{2\to 2}, \varepsilon) \leq \left(1 + \frac{2\Lambda}{\varepsilon}\right)^{Nn}. \tag{57}$$

Proof. For $|\cdot|$ denoting the volume in $\mathbb{R}^{N\times n}$, the following is an adaptation of a well-known result [62, Proposition 4.2.12], connecting covering numbers and volume in $\mathbb{R}^{N\times n}$:

$$\begin{aligned} \mathcal{N}(B^{N\times n}_{\|\cdot\|_{2\to 2}}(\Lambda), \|\cdot\|_{2\to 2}, \varepsilon) &\leq \frac{|B^{N\times n}_{\|\cdot\|_{2\to 2}}(\Lambda) + (\frac{\varepsilon}{2})B^{N\times n}_{\|\cdot\|_{2\to 2}}(1)|}{|(\frac{\varepsilon}{2})B^{N\times n}_{\|\cdot\|_{2\to 2}}(1)|} \\ &= \frac{|(\Lambda + \frac{\varepsilon}{2})B^{N\times n}_{\|\cdot\|_{2\to 2}}(1)|}{|(\frac{\varepsilon}{2})B^{N\times n}_{\|\cdot\|_{2\to 2}}(1)|}. \end{aligned}$$

Hence,

$$\mathcal{N}(B^{N\times n}_{\|\cdot\|_{2\to 2}}(\Lambda), \|\cdot\|_{2\to 2}, \varepsilon) \leq \left(1 + \frac{2\Lambda}{\varepsilon}\right)^{Nn}. \qquad \square$$

We employ the previous Lemma, in order to estimate (56).

Proposition IV.9. *The following estimate holds for the covering numbers of $\mathcal{M}$:*

$$\mathcal{N}(\mathcal{M}, \|\cdot\|_F, \varepsilon) \leq \left(1 + \frac{2\Lambda(\Lambda + \|A\|_{2\to 2})K_L}{\mu\varepsilon}\right)^{Nn}. \tag{58}$$

Proof. Due to Lemma IV.8, we can upper bound the covering numbers of $\mathcal{B}_\Lambda$ as follows:

$$\mathcal{N}(\mathcal{B}_\Lambda, \|\cdot\|_{2\to 2}, \varepsilon) \leq \left(1 + \frac{2\Lambda}{\varepsilon}\right)^{Nn}. \tag{59}$$

Therefore, for the covering numbers of $\mathcal{M}$ we have

$$\begin{aligned}\mathcal{N}(\mathcal{M}, \|\cdot\|_F, \varepsilon) &\leq \mathcal{N}(\rho K_L \mathcal{B}_\Lambda, \|\cdot\|_{2\to 2}, \varepsilon)\\ &= \mathcal{N}(\mathcal{B}_\Lambda, \|\cdot\|_{2\to 2}, \varepsilon/\rho K_L)\\ &\leq \left(1 + \frac{2\Lambda\rho K_L}{\varepsilon}\right)^{Nn},\end{aligned}$$

where $\rho = \mu^{-1}(\Lambda + \|A\|_{2\to 2})$. □

E. Generalization Error Bounds

We are now in position to deliver generalization error bounds for DECONET.

Theorem IV.10. *Let $\mathcal{H}^L$ be the hypothesis class defined in (31). With probability at least $1-\delta$, for all $h \in \mathcal{H}^L$, the generalization error is bounded as*

$$\begin{aligned}\mathcal{L}(h) \leq& \hat{\mathcal{L}}(h) + 8(B_{\text{in}} + B_{\text{out}})B_{\text{out}}\sqrt{\frac{Nn}{s}}\\ &\cdot\sqrt{\log\left(e\left(1 + \frac{4\mu^{-1}\Lambda(\Lambda + \|A\|_{2\to 2})K_L}{\sqrt{s}B_{\text{out}}}\right)\right)}\\ &+ 4(B_{\text{in}} + B_{\text{out}})^2\sqrt{\frac{2\log(4/\delta)}{s}},\end{aligned} \tag{60}$$

with K_L defined in (49).

Proof. We apply Proposition IV.9 to (56), yielding:

$$\begin{aligned}&\mathcal{R}_{\mathbf{S}}(l \circ \mathcal{H}^L)\\ &\leq \frac{16(\mathrm{B}_{\text{in}} + \mathrm{B}_{\text{out}})}{s}\int_0^{\frac{\sqrt{s}B_{\text{out}}}{2}}\sqrt{\log\mathcal{N}(\mathcal{M}, \|\cdot\|_F, \varepsilon)}d\varepsilon\\ &\leq \frac{16(\mathrm{B}_{\text{in}} + \mathrm{B}_{\text{out}})}{s}\\ &\cdot\int_0^{\frac{\sqrt{s}B_{\text{out}}}{2}}\sqrt{Nn\log\left(1 + \frac{2\Lambda(\Lambda + \|A\|_{2\to 2})K_L}{\mu\varepsilon}\right)}d\varepsilon\\ &\leq 8(\mathrm{B}_{\text{in}} + \mathrm{B}_{\text{out}})B_{\text{out}}\sqrt{\frac{Nn}{s}}\\ &\cdot\sqrt{\log\left(e\left(1 + \frac{4\Lambda(\Lambda + \|A\|_{2\to 2})K_L}{\mu\sqrt{s}B_{\text{out}}}\right)\right)},\end{aligned}$$

where in the last step we used the inequality[3]

$$\int_0^a \sqrt{\log\left(1 + \frac{b}{t}\right)}dt \leq a\sqrt{\log(e(1 + b/a))}, \qquad a, b > 0.$$

The proof follows by employing Theorem IV.1 with the upper bound $c = (B_{\text{in}} + B_{\text{out}})^2$ for the loss function $\|\cdot\|_2^2$. □

If we further assume that it holds $\{c_{1,k}\Lambda\}_{k\geq 0} \leq 1$, $\{c_{1,k}\Lambda^2\}_{k\geq 0} \leq 1$, $\{c_{2,k}\|A\|^2_{2\to 2}\}_{k\geq 0} \leq 1$, for $\{c_{1,k}\}_{k\geq 0} = \{t^1_k/\mu\theta_k\}_{k\geq 0} \leq 1$, $\{c_{2,k}\}_{k\geq 0} = \{t^2_k/\mu\theta_k\}_{k\geq 0} \leq 1$, we obtain:

Corollary IV.11. *Let $\mathcal{H}^L$ be the hypothesis class defined in (31) and assume that $c_{1,k}\Lambda \leq 1$, $c_{1,k}\Lambda^2 \leq 1$, $c_{2,k}\|A\|^2_{2\to 2} \leq 1$, for all $k \geq 0$, with $\{c_{1,k}\}$, $\{c_{1,k}\} \leq 1$ defined as in Lemma IV.4. With probability at least $1-\delta$, for all $h \in \mathcal{H}^L$, the generalization error is bounded as*

$$\begin{aligned}\mathcal{L}(h) \leq& \hat{\mathcal{L}}(h) + 8(B_{\text{in}} + B_{\text{out}})\Bigg(B_{\text{out}}\sqrt{\frac{Nn}{s}}\\ &\cdot\sqrt{\log\left(e\left(1 + \frac{\|Y\|_F(p + qL + r\kappa_L)}{\sqrt{s}B_{\text{out}}}\right)\right)}\\ &+ \sqrt{\frac{2\log(4/\delta)}{s}}\Bigg),\end{aligned} \tag{61}$$

with κ_L as in Theorem IV.6 and $p, q, r > 0$ constants depending on $\|A\|_{2\to 2}, \Lambda, \mu$.

Proof. The estimate easily follows from Theorems IV.6 and IV.10, if we set $p := \Lambda(\Lambda + \|A\|_{2\to 2})\|A\|_{2\to 2}$, $q := p(\mu^{-1} - 1)$ and $r := 2p(\|A\|_{2\to 2} + 1)(\|A\|_{2\to 2} + 3)$. □

All the previous results are summarized in

Theorem IV.12. *Let $\mathcal{H}^L$ be the hypothesis class defined in (31). Assume there exist pair-samples $\{(x_i, y_i)\}_{i=1}^s$, with $y_i = Ax_i + e$, $\|e\|_2 \leq \varepsilon$, for some $\varepsilon > 0$, that are drawn i.i.d. according to an unknown distribution $\mathcal{D}$, and that it holds $\|y_i\|_2 \leq \mathrm{B}_{\text{in}}$ almost surely with $\mathrm{B}_{\text{in}} = \mathrm{B}_{\text{out}}$ in (29). Let us further assume that for step sizes $0 < \{t^1_k\}_{k\geq 0}$, $\{t^2_k\}_{k\geq 0} \leq 1$, step size multiplier $0 < \{\theta_k\}_{k\geq 0} \leq 1$ and smoothing parameter $\mu > 1$, we have $\mu^{-1}\theta_k^{-1}t^1_k\Lambda \leq 1$, $\mu^{-1}\theta_k^{-1}t^1_k\Lambda^2 \leq 1$, $\mu^{-1}\theta_k^{-1}t^2_k\|A\|_{2\to 2} \leq 1$, for all $k \geq 0$. Then with probability at least $1-\delta$, for all $h \in \mathcal{H}^L$, the generalization error is bounded as*

$$\begin{aligned}\mathcal{L}(h) \leq& \hat{\mathcal{L}}(h) + 16B^2_{\text{out}}\sqrt{\frac{Nn}{s}}\\ &\cdot\sqrt{\log\left(e\left(1 + \frac{\|Y\|_F(p + qL + r\kappa_L)}{\sqrt{s}B_{\text{out}}}\right)\right)}\\ &+ 16B_{\text{out}}\sqrt{\frac{2\log(4/\delta)}{s}},\end{aligned} \tag{62}$$

with κ_L as in Theorem IV.6 and constants $p, q, r > 0$ as in Corollary IV.11.

We also state below a key remark, regarding the generalization error bound as a function of L, N and s.

[3] The interested reader may refer to [41, Lemma C.9] for a detailed proof of this inequality

Corollary IV.13 (Informal)**.** *According to* (51)*, we have that* L *enters at most exponentially in the definition of* κ_L*. If we consider the dependence of the generalization error bound* (62) *only on* L, N, s *and treat all other terms as constants, we roughly have*

$$|\mathcal{L}(h) - \hat{\mathcal{L}}(h)| \lesssim \sqrt{\frac{NL}{s}}. \tag{63}$$

Comparison with related work: Similarly to Theorem IV.12, the result obtained in [22, Theorem 2] demonstrates that the generalization error of the proposed synthesis-sparsity-based ISTA-net roughly scales like $\sqrt{(n \log L)(n+m)/s}$. The latter estimate is slightly better in terms of L than our theoretical results. On the other hand, as depicted in Theorem IV.12, our bound does not depend on m, which makes it tighter in terms of the number of measurements.

V. Numerical Experiments

In this Section, we examine whether our theory regarding the generalization error of DECONET is consistent to real-world applications of our framework.

A. Experimental Setup

We train and test DECONET on a synthetic dataset of random vectors, drawn from the normal distribution (70000 training and 10000 test examples), and two real-world image datasets: MNIST (60000 training and 10000 test 28 × 28 image examples) and CIFAR10 (50000 training and 10000 test 32 × 32 coloured image examples). For the CIFAR10 dataset, we transform the images into grayscale ones. For both datasets, we consider the vectorized form of the images. We examine DECONET with varying number of layers L. We consider two CS ratios, i.e. $m/n = 25\%$ and $m/n = 50\%$. We choose a random Gaussian measurement matrix $A \in \mathbb{R}^{m \times n}$ and appropriately normalize it, i.e., $\tilde{A} = A/\sqrt{m}$. We add zero-mean Gaussian noise e with standard deviation $\text{std} = 10^{-4}$ to the measurements y, so that $y = \tilde{A}x + e$. We set $\varepsilon = \|y - \tilde{A}x\|_2$ and $x_0 = A^T y$, which are standard algorithmic setups. We take different values of N and perform two different initializations for $W \in \mathbb{R}^{N \times n}$: normal initialization and initialization based on Beta distribution [63], with varying values of Beta's parameters a and b. We set $\mu = 100$, initial step sizes $t_0^1 = t_0^2 = 1$ and step size multiplier $\theta_0 = 1$. For t_k^1, t_k^2, θ_k, we apply the following update rules: $t_k^1 = \alpha t_{k-1}^1$, $t_k^2 = \beta t_{k-1}^2$, $\theta_k = \theta_{k-1} \cdot \theta'$, $k = 1, \dots, L$, where $(\alpha, \beta) \in (0,1) \times (0,1)$, $\theta' = \frac{1-\sqrt{\mu/\tilde{L}}}{1+\sqrt{\mu/\tilde{L}}}$, respectively, and $\tilde{L}$ is an upper bound on the smoothing parameter μ; we set $\tilde{L} = 1000$. All networks are implemented in PyTorch [64] and trained using *Adam* [65] algorithm, with batch size 128. Adam constitutes a stochastic optimization method which can adaptively estimate lower-order moments of the gradient of (33). We employ the Pytorch implementation of Adam and set the initial learning rates to $\eta_{\text{M}} = 10^{-2}$, $\eta_{\text{C}} = 10^{-3}$ and $\eta_{\text{S}} = 10^{-4}$, for the MNIST, CIFAR10 and synthetic datasets, respectively; the rest of Adam's parameters are set to their default values. For our experiments, we report the *test MSE* defined by

$$\mathcal{L}_{test} = \frac{1}{d} \sum_{i=1}^{d} \|h(\tilde{y}_i) - \tilde{x}_i\|_2^2, \tag{64}$$

where $\mathbf{D} = \{(\tilde{y}_i, \tilde{x}_i)\}_{i=1}^d$ is a set of d test data, not used in the training phase. We also report the *empirical generalization error* (EGE) defined by

$$\mathcal{L}_{gen} = |\mathcal{L}_{test} - \mathcal{L}_{train}|, \tag{65}$$

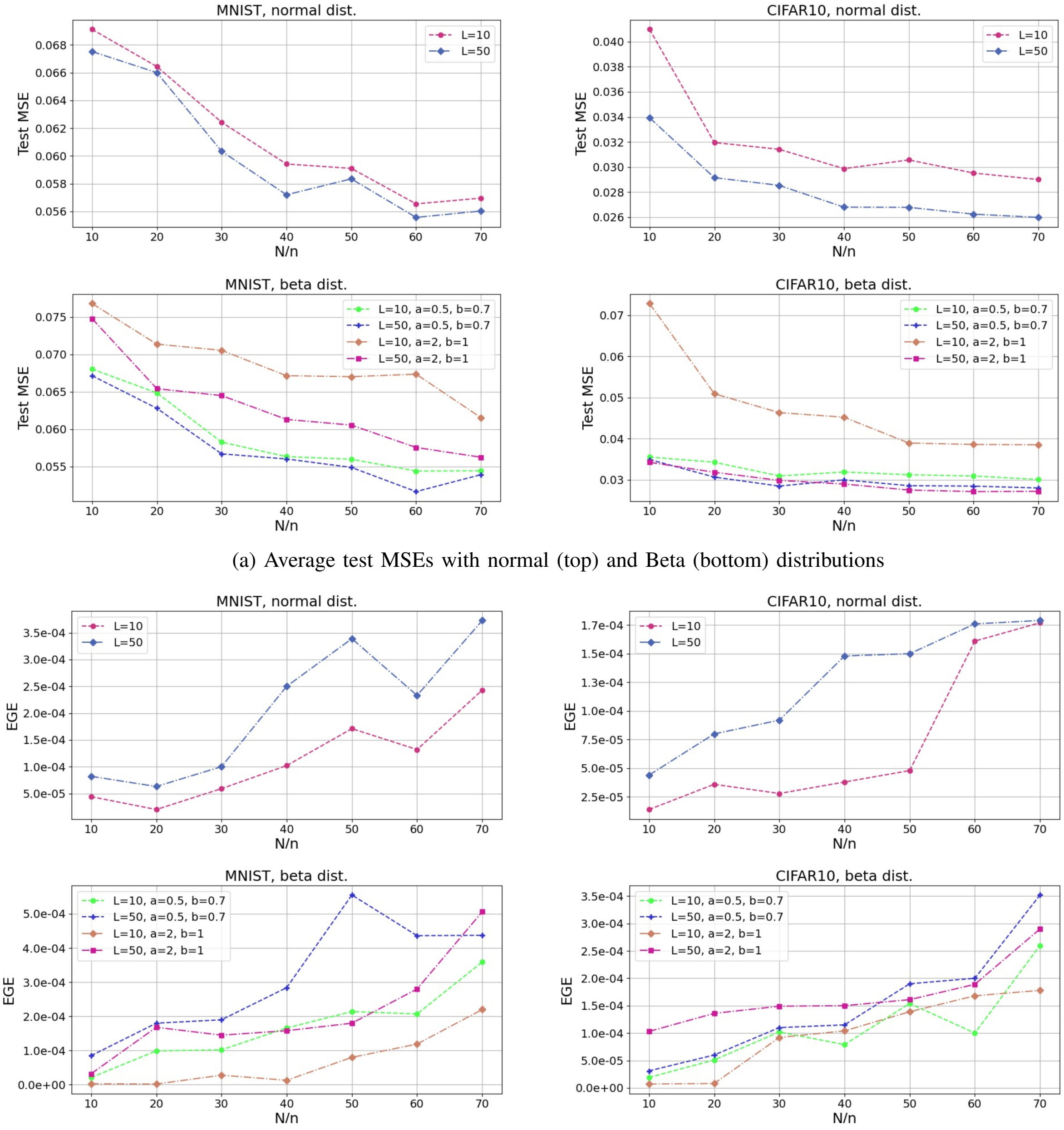
where $\mathcal{L}_{train}$ is the train MSE defined in (33). Since test MSE approximates the true loss, we use (65) – which can be explicitly computed – to approximate the generalization error of (35). We train all networks, on all datasets, employing an early stopping technique [66] with respect to (65). We repeat all the experiments at least 10 times and average the results over the runs. We compare the reconstruction quality offered by DECONET, to the MSE achieved by the original optimization algorithm[4], namely ACF (see Section III), with a similar structure. To that end, we choose as predefined sparsifiers a redundant Haar wavelet transform and a finite difference operator. The latter is associated to the popular method of total variation [52], while redundant wavelets constitute standard choices of sparsifying transforms for analysis-based inverse problems [48], [42]. Finally, we compare the EGE of DECONET to the EGE achieved by two SotA unfolding networks serving as baselines: a recent variant of ISTA-net [22] and ADMM-DAD net [33]. Both baselines jointly learn a decoder for CS and a sparsifying transform. Nevertheless, ISTA-net solves the CS problem employing synthesis sparsity, since the learnable sparsifier is orthogonal, while ADMM-DAD involves analysis sparsity, by learning a redundant analysis operator. Our choice of the aforementioned baselines is attributed to our interest of considering one SotA unfolding network from each sparsity "category", so that we examine a) how the EGE is affected by each of the two sparsity models and b) if DECONET outperforms[5] ADMM-DAD. For both baselines, we set the best hyper-parameters proposed by the original authors. From a complexity perspective, the most costly computation of both DECONET and ISTA-net consists in the multiplication of a matrix with its transpose ($N \times n$

Dataset / Method	Test MSE MNIST	Test MSE CIFAR10	Test MSE Synthetic
ACF (Wavelet)	0.2515	0.3174	0.1585
ACF (TV)	0.1978	0.3141	0.1051
DECONET (learnable)	**0.0571**	**0.0298**	$\mathbf{0.7523 \cdot 10^{-2}}$

Table II: Comparison of MSEs achieved by DECONET with its learnable analysis operator, ACF with a redundant Haar wavelet transform and ACF with a total variation operator, on all datasets, with 10 layers/iterations. Bold letters indicate the best performance among three methods.

[4] One of the motivations for unrolling iterative schemes to corresponding neural networks relies on the fact that the latter yield a smaller reconstruction error, than their original iterative schemes [16]

[5] To the best of authors' knowledge, ADMM-DAD is the only unfolding network, apart from DECONET, that entails analysis sparsity for solving the CS problem



(a) Average test MSEs with normal (top) and Beta (bottom) distributions

(b) Empirical generalization errors with normal (top) and Beta (bottom) distributions

Fig. 2: Performance plots for 10- and 50-layer DECONET, $m = n/4$, tested on MNIST (left) and CIFAR10 (right) datasets.

with $n \times N$ for DECONET, $n \times m$ with $m \times n$ for ISTA-net), so that the dominant part per layer L is of the order of $O(N^2 n)$ and $O(n^2 m)$, respectively. ADMM-DAD has a cubic complexity – with respect to n – per layer L, since its "heavier" computation involves the inversion of a $n \times n$ matrix. Overall, the computational complexities of DECONET, ISTA-net and ADMM-DAD are of the order of $O(LN^2 n)$, $O(Ln^2 m)$ and $O(Ln^3)$, respectively.

B. Experimental Results

We test DECONET on all datasets under multiple experimental scenarios.

1) Fixed CS ratio with varying N/n and L, for different initializations, on image datasets: We examine the performance of 10- and 50-layer DECONET for a fixed 25% CS ratio, varying redundancy ratio N/n and both normal and Beta initializations for W. We report the results for MNIST and CIFAR10 in Fig. 2. As illustrated in Fig. 2a, the test MSEs, achieved by 10- and 50-layer DECONET on both datasets, drop as L and N/n increase, for both types of initialization. The decays seem reasonable, if one considers a standard analysis CS scenario: the reconstruction quality and performance of the analysis-l_1 algorithm, typically benefit from the (high) redundancy offered by the involved analysis operator, especially as the number of iterations/layers

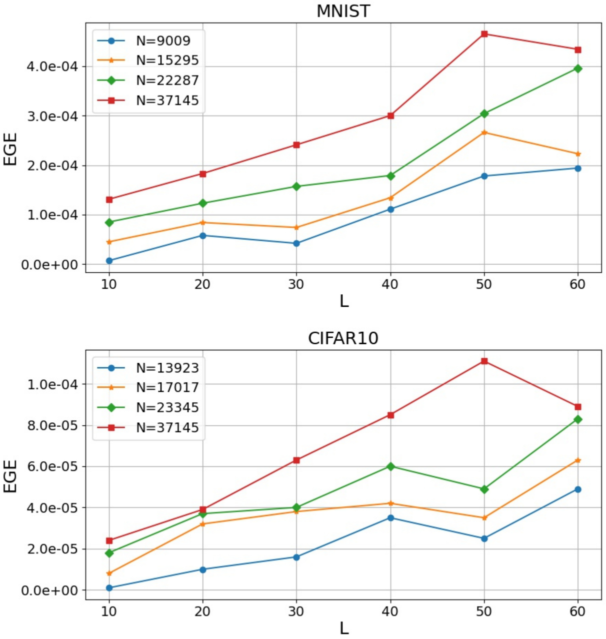


Fig. 3: Performance plots for DECONET with 50% CS ratio, tested on MNIST (top) and CIFAR10 (bottom) datasets.

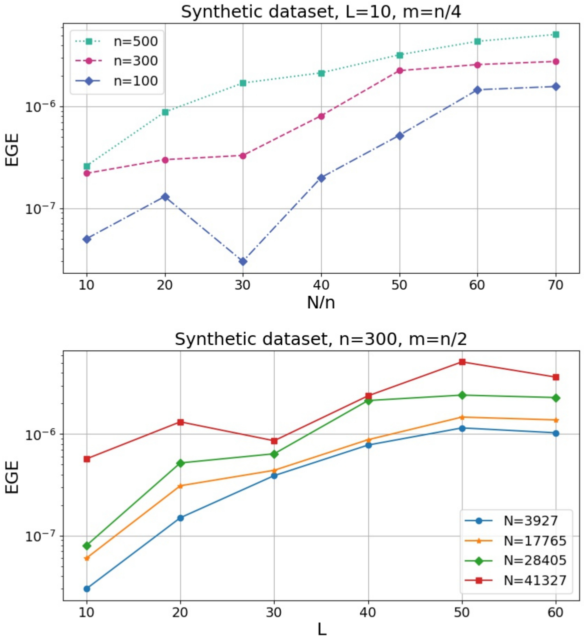


Fig. 4: Performance plots for DECONET, tested on a synthetic dataset, under different settings.

increases. Furthermore, Fig. 2b demonstrates that the EGE of DECONET increases as both L and N/n increase, for both normal and Beta initialization. For the latter, we observe that the different values of its parameters affect the generalization ability of DECONET on both image datasets. The overall performance of DECONET confirms our theoretical results depicted in Section IV-E, since EGEs seem to scale like $\sqrt{NL}$.

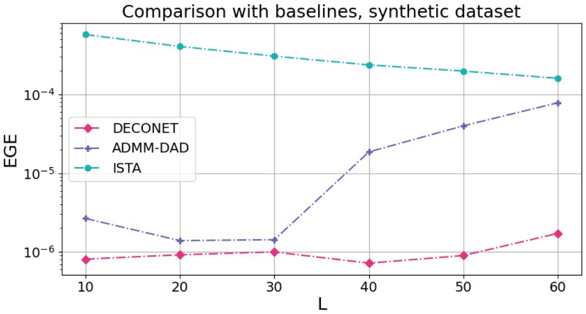


Fig. 5: Performance plot for all decoders, with fixed m, n, N, on a synthetic dataset.

2) Fixed CS ratio, with varying N and L, on image datasets: We examine the generalization ability of DECONET for $m = n/2$, with increasing number of layers L, under different choices of N and normal initialization. Inspired by frames with redundancy ratio $N/n \notin \mathbb{N}$ [67], we consider N of the form

$$N = pn + q, \qquad p, q \in \mathbb{N}. \tag{66}$$

We report the results in Fig. 3 for MNIST and CIFAR10. Similarly to Section V-B1, we observe that the empirical generalization error increases in L and N, for both datasets. Even though the upper bound in (62) depends on other terms too, the empirical generalization error appears to grow at the rate of $\sqrt{NL}$. The behaviour of DECONET again conforms with our theoretical results presented in Section IV. One may also notice that – in general – we choose different N for each of the two datasets. This is simply due to (66), i.e., N depends on the vectorized ambient dimension n, which is different for each of the two datasets.

3) Fixed CS ratio with varying n, N, L, on synthetic dataset: Similarly to Sections V-B1 and V-B2, we examine the generalization error that DECONET achieves on a synthetic dataset of random vectors, with a normally initialized W. We present the results in Fig. 4 (and note that all plots regarding the synthetic data are made in logarithmic scale). Particularly, the top plot of Fig. 4 demonstrates how the EGE for 10-layer DECONET scales, for different values of the ambient dimension n, with 25% CS ratio, as N/n increases. On the other hand, as depicted in the bottom plot of Fig. 4, we revisit the experimental reasoning of Section V-B2, with fixed $n = 300$ and $m = n/2$. In both subplots, we see that the EGEs achieved by DECONET comply with our theory.

4) Comparison between DECONET and ACF: We investigate the reconstruction quality – on all datasets – offered by DECONET with a learnable analysis operator satisfying $N/n = 50$, to the test MSE achieved by ACF, with two different redundant analysis operators: wavelets and total variation operator. We set $m = n/4$, $L = 10$ (for all methods) and fix $n = 100$ for the synthetic dataset. We present our results in Table II, which showcases that the test MSE achieved by both instances of ACF is larger than DECONET's, consistently for all three datasets. This behaviour complies with the motivation for interpreting iterative methods as neural networks,

	25% CS ratio					
Dataset	MNIST			CIFAR10		
Layers Decoder	$L = 10$	$L = 20$	$L = 30$	$L = 10$	$L = 20$	$L = 30$
DECONET	**0.000033**	**0.000110**	**0.000314**	**0.000066**	**0.000058**	**0.000100**
ADMM-DAD	0.000212	0.000284	0.000394	0.000086	0.000102	0.000168
ISTA-net	0.013408	0.015099	0.007874	0.007165	0.005045	0.004120

	50% CS ratio					
Dataset	MNIST			CIFAR10		
Layers Decoder	$L = 10$	$L = 20$	$L = 30$	$L = 10$	$L = 20$	$L = 30$
DECONET	**0.000131**	**0.000183**	**0.000241**	**0.000024**	**0.000039**	**0.000063**
ADMM-DAD	0.000224	0.000455	0.000332	0.000094	0.000046	0.000087
ISTA-net	0.016547	0.011624	0.009311	0.009274	0.006739	0.005576

Table III: Empirical generalization errors for 10-, 20- and 30-layer decoders, with 25% and 50% CS ratios, and fixed $N = 37145$ for DECONET's and ADMM-DAD's sparsifiers. Bold letters indicate the best performance among the three decoders.

since the latter are able to achieve a smaller reconstruction error than their iterative schemes, for the same number of layers/iterations.

5) Comparison to baselines: We examine how analysis and synthesis sparsity models affect the generalization ability of CS-oriented unfolding networks. Towards this end, we compare the proposed DECONET's decoder to ISTA-net's and ADMM-DAD's decoders. For the image datasets, the comparisons are made for 10, 20 and 30 layers, with 25% and 50% CS ratio, and fixed $N = 37145$ for both DECONET's and ADMM-DAD's sparsifiers. For the synthetic dataset, we fix $N = 12000$ (for both DECONET's and ADMM-DAD's sparsifiers), $n = 300$, $m = n/2$, and vary L. We report the empirical generalization errors for the image datasets in Table III and for the synthetic dataset in Fig. 5. We conjecture that DECONET should outperform ISTA-net in terms of EGE. Our speculation is attributed to similar (albeit limited) experimental findings in [33, Section 4], which indicate that analysis sparsity can affect the generalization ability of unfolding networks. This is indeed the case according to Table III and Fig. 5, which demonstrate that our proposed decoder outperforms both baseline decoders, consistently for all datasets. In fact, both DECONET and ADMM-DAD outperform the synthesis-sparsity-based ISTA-net. This behaviour showcases that learning a redundant sparsifier instead of an orthogonal one, improves the performance of a CS-oriented unfolding network. Additionally, our proposed network outperforms ADMM-DAD, which is considered to be a SotA unfolding network for analysis-based CS. Finally, our theoretical results on the generalization error of DECONET seem to align with the experiments, since the EGE of DECONET increases as L also increases. Interestingly, we observe that the EGE of DECONET on the CIFAR10 dataset decreases as the number of measurements m increases. This behaviour is not reflected by our theoretical results (which are independent of m), but it could serve as a potential line of future work.

VI. Conclusion and Future Work

In this paper we derived DECONET, a new deep unfolding network for solving the analysis-sparsity-based Compressed Sensing problem. DECONET jointly learns a decoder for CS and a redundant sparsifying analysis operator. Furthermore, we estimated the generalization error of DECONET, in terms of the Rademacher complexity of the associated hypothesis class. Our generalization error bounds roughly scale like the square root of the product between the number of layers and the redundancy of the learnable sparsifier. To the best of our knowledge, this is the first result of its kind for unfolding networks solving the analysis-based CS problem. Furthermore, we conducted experiments that confirmed the validity of our theoretical results and compared DECONET to state-of-the-art CS-oriented unfolding networks. Our proposed network outperformed the baselines, consistently for synthetic and real-world datasets. As a future direction, we would like to examine the performance of our proposed framework on speech datasets and experiment with different values of the non-learnable parameters. Additionally, it would be interesting to further characterize (e.g. in terms of structure) the sparsifying transform that DECONET learns. Last but not least, it would be intriguing to investigate the tightness of our delivered generalization error bounds. For example, we could employ techniques presented in [68], [69], in order to examine whether our bounds could be independent of the number of layers and/or the redundancy of the learnable sparsifying transform.

Acknowledgements

V. Kouni would like to thank Holger Rauhut for his advice and helpful conversations around the subject covered in this paper.

References

[1] E. J. Candès, J. Romberg, and T. Tao. "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information". In: *IEEE Trans. Inf. Theory* 52.2 (2006), pp. 489–509.

[2] I. Daubechies, M. Defrise, and C. De Mol. "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint". In: *Commun. Pure and Appl. Math.* 57.11 (2004), pp. 1413–1457.

[3] R. Chartrand and W. Yin. "Iteratively reweighted algorithms for compressive sensing". In: *Int. Conf. Acoust., Speech and Signal Process.* IEEE. 2008, pp. 3869–3872.

[4] S. Boyd, N. Parikh, and E. Chu. *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.

[5] G. Yang et al. "DAGAN: Deep de-aliasing generative adversarial networks for fast compressed sensing MRI reconstruction". In: *IEEE Trans. Med. Imag.* 37.6 (2017), pp. 1310–1321.

[6] H. Yao et al. "Dr2-net: Deep residual reconstruction network for image compressive sensing". In: *Neurocomputing* 359 (2019), pp. 483–493.

[7] V. Monga, Y. Li, and Y. C Eldar. "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing". In: *IEEE Signal Process. Mag.* 38.2 (2021), pp. 18–44.

[8] J. R. Hershey, J. Le Roux, and F. Weninger. "Deep unfolding: Model-based inspiration of novel deep architectures". In: *arXiv preprint arXiv:1409.2574* (2014).

[9] K. Gregor and Y. LeCun. "Learning fast approximations of sparse coding". In: *Proc. 27th Int. Conf. Mach. Learn.* 2010, pp. 399–406.

[10] J. Zhang et al. "Deep Unfolding With Weighted l_2 Minimization for Compressive Sensing". In: *IEEE Internet of Things J.* 8.4 (2020), pp. 3027–3041.

[11] C. Bertocchi et al. "Deep unfolding of a proximal interior point method for image restoration". In: *Inverse Problems* 36.3 (2020), p. 034005.

[12] J. Scarlett et al. "Theoretical perspectives on deep learning methods in inverse problems". In: *arXiv preprint arXiv:2206.14373* (2022).

[13] H. Van Luong, B. Joukovsky, and N. Deligiannis. "Designing interpretable recurrent neural networks for video reconstruction via deep unfolding". In: *IEEE Trans. Image Process.* 30 (2021), pp. 4099–4113.

[14] Y. Yang et al. "A Robust Deep Unfolded Network for Sparse Signal Recovery from Noisy Binary Measurements". In: *Eur. Signal Process. Conf.* IEEE. 2021, pp. 2060–2064.

[15] A. P. Sabulal and S. Bhashyam. "Joint Sparse Recovery Using Deep Unfolding With Application to Massive Random Access". In: *Int. Conf. Acoust., Speech and Signal Process.* 2020, pp. 5050–5054.

[16] M. Borgerding, P. Schniter, and S. Rangan. "AMP-Inspired Deep Networks for Sparse Linear Inverse Problems". In: *IEEE Trans. Signal Process.* 65.16 (2017), pp. 4293–4308.

[17] W. Pu, Y. C. Eldar, and M. RD. Rodrigues. "Optimization Guarantees for ISTA and ADMM Based Unfolded Networks". In: *Int. Conf. Acoust., Speech and Signal Process.* IEEE. 2022, pp. 8687–8691.

[18] J. Zhang and B. Ghanem. "ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing". In: *Proc. IEEE Comput. Vision and Pattern Recognit.* 2018, pp. 1828–1837.

[19] Y. Khalifa, Z. Zhang, and E. Sejdić. "Sparse recovery of time-frequency representations via recurrent neural networks". In: *22nd Int. Conf. Digit. Signal Process.* IEEE. 2017, pp. 1–5.

[20] J. Sun et al. "Deep ADMM-Net for compressive sensing MRI". In: *Advances Neural Inf. Process. Syst.* 29 (2016).

[21] P. Xiao, B. Liao, and N. Deligiannis. "Deepfpc: A deep unfolded network for sparse signal recovery from 1-bit measurements with application to doa estimation". In: *Signal Process.* 176 (2020), p. 107699.

[22] A. Behboodi, H. Rauhut, and E. Schnoor. "Compressive sensing and neural networks from a statistical learning perspective". In: *Compressed Sensing in Information Processing*. Springer, 2022, pp. 247–277.

[23] Y. Shen et al. "Image reconstruction algorithm from compressed sensing measurements by dictionary learning". In: *Neurocomputing* 151 (2015), pp. 1153–1162.

[24] Z. Li, H. Huang, and S. Misra. "Compressed sensing via dictionary learning and approximate message passing for multimedia Internet of Things". In: *IEEE Internet of Things J.* 4.2 (2016), pp. 505–512.

[25] H. Zayyani, M. Korki, and F. Marvasti. "Dictionary learning for blind one bit compressed sensing". In: *IEEE Signal Process. Lett.* 23.2 (2015), pp. 187–191.

[26] S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[27] A. Xu and M. Raginsky. "Information-theoretic analysis of generalization capability of learning algorithms". In: *Advances Neural Inf. Process. Syst.* 30 (2017).

[28] E. Schnoor, A. Behboodi, and H. Rauhut. "Generalization Error Bounds for Iterative Recovery Algorithms Unfolded as Neural Networks". In: *arXiv preprint arXiv:2112.04364* (2021).

[29] Huynh Van L., B. Joukovsky, and N. Deligiannis. "Interpretable Deep Recurrent Neural Networks via Unfolding Reweighted $\ell_1 - \ell_1$ Minimization: Architecture Design and Generalization Analysis". In: *arXiv preprint arXiv:2003.08334* (2020).

[30] B. Joukovsky et al. "Generalization error bounds for deep unfolding RNNs". In: *Uncertainty in Artif. Intell.* PMLR. 2021, pp. 1515–1524.

[31] S. Nam et al. "The cosparse analysis model and algorithms". In: *Appl. and Comput. Harmon. Anal.* 34.1 (2013), pp. 30–56.

[32] H. Cherkaoui et al. "Analysis vs synthesis-based regularization for combined compressed sensing and parallel MRI reconstruction at 7 tesla". In: *Eur. Signal Process. Conf.* IEEE. 2018, pp. 36–40.

[33] V. Kouni et al. "ADMM-DAD Net: A Deep Unfolding Network for Analysis Compressed Sensing". In: *Int. Conf. Acoust., Speech and Signal Process.* IEEE. 2022, pp. 1506–1510.

[34] A. Beck and M. Teboulle. "A fast iterative shrinkage-thresholding algorithm for linear inverse problems". In: *J. Imag. Sci.* 2.1 (2009), pp. 183–202.

[35] S. R. Becker, E. J. Candès, and M. C. Grant. "Templates for convex cone problems with applications to sparse signal recovery". In: *Math. Prog. Comput.* 3.3 (2011), pp. 165–218.

[36] C. Ma et al. "Rademacher complexity and the generalization error of residual networks". In: *Commun. Math. Sci.* 18.6 (2020), pp. 1755–1774.

[37] Y. LeCun et al. "Gradient-based learning applied to document recognition". In: *Proc. IEEE* 86.11 (1998), pp. 2278–2324.

[38] A. Krizhevsky et al. "Learning multiple layers of features from tiny images". In: (2009).

[39] S. Wisdom et al. "Building recurrent networks by unfolding iterative thresholding for sequential sparse recovery". In: *Int. Conf. Acoust., Speech and Signal Process.* IEEE. 2017, pp. 4346–4350.

[40] D. L. Donoho, A. Maleki, and A. Montanari. "Message-passing algorithms for compressed sensing". In: *Proc. Nat. Acad. of Sci.* 106.45 (2009), pp. 18914–18919.

[41] S. Foucart and H. Rauhut. "An invitation to compressive sensing". In: *A mathematical introduction to compressive sensing*. Springer, 2013, pp. 1–39.

[42] I. W. Selesnick and M. A. Figueiredo. "Signal restoration with overcomplete wavelet transforms: Comparison of analysis and synthesis priors". In: *Wavelets XIII*. Vol. 7446. SPIE. 2009, pp. 107–121.

[43] S. Pejoski, V. Kafedziski, and D. Gleich. "Compressed sensing MRI using discrete nonseparable shearlet transform and FISTA". In: *IEEE Signal Process. Lett.* 22.10 (2015), pp. 1566–1570.

[44] C. Li and B. Adcock. "Compressed sensing with local structure: uniform recovery guarantees for the sparsity in levels class". In: *Appl. and Comput. Harmon. Anal.* 46.3 (2019), pp. 453–477.

[45] P. T. Dao, A. Griffin, and X. J. Li. "Compressed sensing of EEG with Gabor dictionary: Effect of time and frequency resolution". In: *40th Annu. Int. Conf. IEEE Eng Med. and Biol. Soc.* IEEE. 2018, pp. 3108–3111.

[46] S.-J. Kim et al. "An Efficient Method for Compressed Sensing". In: *Int. Conf. Image Process.* Vol. 3. 2007, pp. III - 117-III –120.

[47] M. Kabanava and H. Rauhut. "Analysis l_1-recovery with frames and gaussian measurements". In: *Acta Appl. Math.* 140.1 (2015), pp. 173–195.

[48] M. Genzel, G. Kutyniok, and M. März. "l_1-Analysis minimization and generalized (co-) sparsity: When does recovery succeed?" In: *Appl. and Comput. Harmon. Anal.* 52 (2021), pp. 82–140.

[49] E. J. Candes et al. "Compressed sensing with coherent and redundant dictionaries". In: *Appl. and Comput. Harmon. Anal.* 31.1 (2011), pp. 59–73.

[50] K. Gröchenig. *Foundations of time-frequency analysis*. Springer Science & Business Media, 2001.

[51] H. G Feichtinger and T. Strohmer. *Advances in Gabor analysis*. Springer Science & Business Media, 2003.

[52] F. Krahmer, C Kruschel, and M. Sandbichler. "Total variation minimization in compressed sensing". In: *Compressed sensing and its applications*. Springer, 2017, pp. 333–358.

[53] V. Kouni and H. Rauhut. "Spark Deficient Gabor Frame Provides a Novel Analysis Operator for Compressed Sensing". In: *Int. Conf. Neural Inf. Process.* Springer. 2021, pp. 700–708.

[54] J. Ma et al. "Deep tensor admm-net for snapshot compressive imaging". In: *Proc. Int. Conf. Comput. Vision*. IEEE. 2019, pp. 10223–10232.

[55] Y. Yang et al. "ADMM-CSNet: A deep learning approach for image compressive sensing". In: *Trans. Pattern Anal. and Mach. Intell.* 42.3 (2018), pp. 521–538.

[56] P. L. Bartlett and S. Mendelson. "Rademacher and Gaussian complexities: Risk bounds and structural results". In: *J. of Mach. Learn. Res.* 3.Nov (2002), pp. 463–482.

[57] A. Maurer. "A vector-contraction inequality for rademacher complexities". In: *Int. Conf. Algorithmic Learn. Theory*. Springer. 2016, pp. 3–17.

[58] L. Chai et al. "Efficient computation of frame bounds using LMI-based optimization". In: *IEEE Trans. Signal Process.* 56.7 (2008), pp. 3029–3033.

[59] M. Faulhuber. "Minimal frame operator norms via minimal theta functions". In: *J. Fourier Anal. and Appl.* 24.2 (2018), pp. 545–559.

[60] J. E. Fowler. "The redundant discrete wavelet transform and additive noise". In: *IEEE Signal Process. Lett.* 12.9 (2005), pp. 629–632.

[61] Richard M Dudley. "The sizes of compact subsets of Hilbert space and continuity of Gaussian processes". In: *J. Funct. Anal.* 1.3 (1967), pp. 290–330.

[62] R. Vershynin. *High-dimensional probability: An introduction with applications in data science*. Vol. 47. Cambridge university press, 2018.

[63] L. Lu et al. "Dying relu and initialization: Theory and numerical examples". In: *arXiv preprint arXiv:1903.06733* (2019).

[64] N. Ketkar. "Introduction to pytorch". In: *Deep learning with python*. Springer, 2017, pp. 195–208.

[65] D. P. Kingma and J. Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[66] L. Prechelt. "Early stopping-but when?" In: *Neural Networks: Tricks of the trade*. Springer, 1998, pp. 55–69.

[67] P. Kittipoom, G. Kutyniok, and W. Q. Lim. "Construction of compactly supported shearlet frames". In: *Constructive Approximations* 35.1 (2012), pp. 21–72.

[68] S. Park, U. Şimşekli, and M. A. Erdogdu. "Generalization Bounds for Stochastic Gradient Descent via Localized ε-Covers". In: *arXiv preprint arXiv:2209.08951* (2022).

[69] A. Asadi, E. Abbe, and S. Verdú. "Chaining mutual information and tightening generalization bounds". In: *Advances Neural Inf. Process. Sys.* 31 (2018).

Vicky Kouni received her B.Sc. and M.Sc. in Mathematics, from the Dep. of Mathematics, National & Kapodistrian University of Athens, Greece. She is currently a PhD student at the Dep. of Informatics & Telecommunications, National & Kapodistrian University of Athens, Greece. She has received scholarships and awards for her studies and research, including the recent *Scholarship for Senior Doctorate Students* by the greek State Scholarships Foundation (IKY). Her research interests are mainly focused on deep unfolding, learning theory, compressed sensing, sparse representations, applied harmonic analysis, and their applications in audio and image processing.

Yannis Panagakis received his PhD and MSc from the Dep. of Informatics, Aristotle University of Thessaloniki, and his B.Sc. in Informatics & Telecommunications from the National & Kapodistrian University of Athens, Greece. He previously held research and academic positions at the Samsung AI Centre, Cambridge, U.K., Middlesex University London, and Imperial College London, U.K. He is currently an Associate Professor of machine learning and signal processing in the Dep. of Informatics & Telecommunications, National & Kapodistrian University of Athens, Greece. He has received prestigious scholarships and awards for his studies and research, including the Marie-Curie Fellowship in 2013. His research interests include: machine learning and its interface with tensor methods, deep learning, computer vision, signal processing and mathematical optimization. He has published over 80 articles in leading journals and conferences.