Theoretical Perspectives on Deep Learning Methods in Inverse Problems

Jonathan Scarlett, *Member, IEEE*, Reinhard Heckel, *Member, IEEE*, Miguel R. D. Rodrigues, *Fellow, IEEE*, Paul Hand, and Yonina C. Eldar, *Fellow, IEEE*

***Abstract*—In recent years, there have been significant advances in the use of deep learning methods in inverse problems such as denoising, compressive sensing, inpainting, and super-resolution. While this line of works has predominantly been driven by practical algorithms and experiments, it has also given rise to a variety of intriguing theoretical problems. In this paper, we survey some of the prominent theoretical developments in this line of works, focusing in particular on generative priors, untrained neural network priors, and unfolding algorithms. In addition to summarizing existing results in these topics, we highlight several ongoing challenges and open problems.**

***Index Terms*—Inverse problems, generative priors, untrained neural networks, unfolding algorithms, compressive sensing, denoising, theoretical guarantees, information-theoretic limits.**

I. Introduction

THE STUDY of inverse problems spans several research communities, covering problems such as inpainting, denoising, super-resolution, medical imaging, and more. Over the years, research on inverse problems has seen a series of paradigm shifts and new perspectives; for instance, the incorporation of low-dimensional structure such as sparsity led to extensive research on *compressive sensing* [36], [41], [44].

Manuscript received 16 November 2022; accepted 8 January 2023. Date of publication 2 February 2023; date of current version 16 March 2023. The work of Jonathan Scarlett was supported by the Singapore National Research Foundation (NRF) under Grant R-252-000-A74-281. The work of Reinhard Heckel was supported by the Institute of Advanced Studies at the Technical University of Munich and the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Grant 456465471 and Grant 464123524. The work of Miguel R. D. Rodrigues was supported in part by The Weizmann-UK Making Connections Programme under Grant 129589, and in part by the Alan Turing Institute. The work of Yonina C. Eldar was supported by The Weizmann-UK Making Connections Programme under Grant 129589. The work of Paul Hand was supported by the National Science Foundation (NSF) under Award DMS-1848087, Award DMS-2022205, and Award DMS-2053448. *(Corresponding author: Jonathan Scarlett.)*

Jonathan Scarlett is with the Department of Computer Science, the Department of Mathematics, and the Institute of Data Science, National University of Singapore, Singapore (e-mail: scarlett@comp.nus.edu.sg).

Reinhard Heckel is with the Department of Electrical and Computer Engineering, Technical University of Munich, 80333 Munich, Germany (e-mail: reinhard.heckel@tum.de).

Miguel R. D. Rodrigues is with the Department of Electronic and Electrical Engineering, University College London, WC1E 6BT London, U.K. (e-mail: m.rodrigues@ucl.ac.uk).

Paul Hand is with the College of Science and the Khoury College of Computer Sciences, Northeastern University, Boston, MA 02115 USA (e-mail: p.hand@northeastern.edu).

Yonina C. Eldar is with the Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot 7610001, Israel (e-mail: yonina.eldar@weizmann.ac.il).

Digital Object Identifier 10.1109/JSAIT.2023.3241123

The most prominent new trend in inverse problems is the incorporation of *deep learning* methods, which have been utilized for signal modeling, decoder design, measurement design, and more. These methods frequently attain state-of-the-art performance in domains such as imaging, signal processing, and communications. While research in this direction has predominantly been practically-oriented and relied on experiments for evaluation, it has also given rise to a wide variety of interesting theoretical developments and challenges. In this paper, we provide an introductory overview of theoretical frameworks and results relating to deep learning methods in inverse problems, and highlight their strengths, limitations, and directions for further research.

A. Background: Inverse Problems

The goal of an inverse problem is to recover (either exactly or approximately) an unknown signal $\mathbf{x}^* \in \mathbb{R}^n$ from a set of measurements $\mathbf{y} \in \mathbb{R}^m$ (often referred to as observations),[1] which are related via a *measurement model* $\mathcal{A}$ (often referred to as the *forward model*):

$$\mathbf{y} = \mathcal{A}(\mathbf{x}^*) + \boldsymbol{\eta}, \tag{1}$$

where $\boldsymbol{\eta}$ represents possible additive noise.[2] The measurement model $\mathcal{A}$ may be known, unknown, or partially known.

An important special case is the class of *linear models*, in which $\mathcal{A}$ is a linear operation:

$$\mathbf{y} = \mathbf{A}\mathbf{x}^* + \boldsymbol{\eta} \tag{2}$$

for some *measurement matrix* $\mathbf{A} \in \mathbb{R}^{m \times n}$. We focus on the case that $\mathbf{A}$ is known (unless stated otherwise), and the goal is to design an algorithm that recovers $\mathbf{x}^*$ from $(\mathbf{A}, \mathbf{y})$.

Linear models already capture numerous important problems, including denoising, inpainting, deblurring, and super-resolution. Among these, we highlight the seemingly simplest problem of denoising, in which $\mathbf{A}$ is the identity matrix:

$$\mathbf{y} = \mathbf{x}^* + \boldsymbol{\eta}. \tag{3}$$

While this problem may appear limited in scope compared to general linear or non-linear measurement models, it turns out that effective solutions to the denoising problem can be used

[1] The reals can also be replaced by the complex numbers or other mathematical types, depending on the application.

[2] More generally, we could write $\mathcal{A}(\mathbf{x}^*, \boldsymbol{\eta})$ to model noise that need not be additive.

as a powerful building block to solve more general inverse problems via plug-and-play methods [94], [111], [136].

A crucial component of inverse problems and their associated algorithms/theory is the assumed prior knowledge on the underlying signal $\mathbf{x}^*$. Such prior knowledge typically amounts to an assumption that $\mathbf{x}^*$ lies in or near some restricted set $\mathcal{X}$, which may be intrinsically low-dimensional despite $\mathbb{R}^n$ being a high-dimensional space. A ubiquitous example is the set of sparse signals:

$$\mathcal{X}_s = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\|_0 \leq s\}, \tag{4}$$

where $\|\mathbf{x}\|_0$ denotes the number of non-zero entries in $\mathbf{x}$, and s is a suitably-chosen sparsity level, typically with $s \ll n$. Related notions include structured sparsity [11], [40], low-rankness [107], and manifold structure [12], [61].

B. Deep Learning Methods for Solving Inverse Problems

Advances in neural networks and deep learning have reshaped the field of machine learning, and are increasingly impacting other domains throughout academia and industry. As hinted above, inverse problems are no exception to this trend. Previous surveys on deep learning methods in inverse problems can be found in [91], [101], and the key distinction of our survey is our focus on mathematical theory. The reader is assumed to be familiar with basic neural network concepts such as depth, width, training, empirical risk minimization, gradient descent, generalization, convolutional neural networks, and recurrent neural networks; an introduction to these concepts can be found in [148], among many others.

There are many different ways in which deep learning can play a role in designing methods for inverse problems. We will focus on the following three themes in this survey:

1) Generative Priors: One of the tremendous successes of deep learning has been *deep generative modeling*, in which a neural network is trained on a large data set of signals/images, and the resulting network $G : \mathbb{R}^k \to \mathbb{R}^n$ (typically with $k \ll n$) serves as a model for the underlying class of signals, i.e., for each input $\mathbf{z} \in \mathbb{R}^k$, the output $G(\mathbf{z})$ corresponds to some signal (or image in vectorized form). The network is *generative* in the sense that it can generate new images different to those used for training.

Building on practically-oriented works such as [38], [80], [145], Bora et al. [17] introduced a theoretical framework for studying generative model based priors in inverse problems. In comparison to sparse modeling, the idea is to replace the set $\mathcal{X}_s$ in (4) by the set

$$\mathcal{X}_G = \text{Range}(G). \tag{5}$$

By doing so, the prior knowledge can be much more specifically geared to the task at hand. For instance, while a sparse prior in a suitably-chosen basis could model nearly all natural images, a generative prior could specifically target a particular type of image (e.g., brain scans in medical imaging), thus providing a much more precise form of prior information, and leading to improved reconstruction accuracy and/or fewer required measurements. We survey several relevant theoretical results in Section II.

2) Untrained Neural Network Priors: It has recently been observed that even neural networks with *no prior training* can serve as excellent priors for inverse problems [57], [134]. In this approach, the prior information is implicitly encoded in the neural network architecture, and decoding is done by tuning the weights to produce a single image that fits the measurements well.

Despite using neural networks, these methods are perhaps more closely related to sparse priors, in the sense that the priors are "broad" (e.g., capturing general natural images) and are not targeted at specific data sets. On the other hand, their empirical performance often significantly improves on that of sparsity-based methods. We survey several relevant theoretical developments in Section III.

3) Unfolding Methods: Another component of inverse problems amenable to deep learning methods is the design of the decoder, e.g., the algorithm for reconstructing $\mathbf{x}^*$ from $(\mathbf{A}, \mathbf{y})$ in the case of linear measurements. A variety of deep learning approaches have been devised for this task, consisting of trainable components that are optimized for the task at hand, e.g., see [91], [101], [120], [121] for recent surveys.

In Section IV, we consider sparse signal priors and survey the prominent approach of *algorithm unfolding* [50], [96], which frequently provides state-of-the-art practical performance. Briefly, the idea is to select a (recurrent) neural network structure that directly matches a classical iterative algorithm, but to replace the fixed weights of that algorithm with learnable weights. A detailed survey of algorithm unfolding techniques can be found in [96], and our survey is again distinguished by the focus on theory.

These three topics are by no means exhaustive; for instance, there are many deep learning based decoders beyond unfolding methods [91], [101], [120], [121] (as mentioned above), and there are other aspects of inverse problems that also admit deep learning methods, such as designing the measurement matrix [98], [142]. In Section V, we will briefly discuss some further relevant topics beyond the three that we focus on.

C. Theoretical Guarantees for Sparse Recovery

To set the stage for the results that we overview in this paper, it is useful to summarize some of the related results in the literature on sparse recovery. For concreteness, we focus on linear models of the form (2), and signals that are exactly or approximate sparse according to (4), though many results are known beyond this setting (e.g., see [44]). Among the wide range of concepts and results in the literature, we focus on a small sample that are particularly relevant to this survey, and for which we consider closely-related notions for deep learning methods throughout Sections II–IV.

Recovery guarantees: Theoretical results on sparse recovery can differ considerably depending on the presence/absence of noise, whether the signal is exactly or approximately sparse, and the desired recovery guarantee. Particularly relevant to this survey is the ℓ_2/ℓ_2 *for-each guarantee*, which states that there exists a randomized measurement matrix $\mathbf{A}$ such that given $\mathbf{y} = \mathbf{A}\mathbf{x}^*$ (and $\mathbf{A}$), the decoder outputs some $\hat{\mathbf{x}}$ satisfying the

following with high probability:

$$\left\|\hat{\mathbf{x}}-\mathbf{x}^*\right\|_2 \leq C \min_{\mathbf{x}\in\mathcal{X}_s}\left\|\mathbf{x}-\mathbf{x}^*\right\|_2 \tag{6}$$

for some $C > 1$. That is, the estimation error is within a constant factor of the best possible sparse approximation. This guarantee can be achieved with constant probability and $m = O(s\log\frac{n}{s})$ [29], or more generally, with probability $1-\rho$ and $m = O(s\log\frac{n}{s}+\log\frac{1}{\rho})$ [45].

To highlight the impact of the recovery criteria, we note that deterministically attaining (6) (for all $\mathbf{x}^*$) with fixed $\mathbf{A}$ is only possible when $m = \Omega(n)$ [29], though analogous guarantees are possible by using different norms on the left and right sides of (6), known as ℓ_p/ℓ_q guarantees (e.g., $p = q = 1$). In contrast, when $\mathbf{x}^*$ is exactly sparse and the measurements are noisy (i.e., $\mathbf{y} = \mathbf{A}\mathbf{x}^*+\boldsymbol{\eta}$), the preceding difficulty is alleviated, and one can attain a deterministic guarantee of the form

$$\left\|\hat{\mathbf{x}}-\mathbf{x}^*\right\|_2 \leq C\|\boldsymbol{\eta}\|_2 \tag{7}$$

for some constant C, with $m = O(s\log\frac{n}{s})$ [22].

Importantly, the guarantees (6) and (7) (as well as other related guarantees) with the above-mentioned bounds on m are not only information-theoretically achievable, but are known to be attained by *practical* decoding algorithms coupled with suitably-chosen $\mathbf{A}$. Some common choices of $\mathbf{A}$ and decoding algorithms are discussed below.

Measurement matrix design and properties: The measurement matrix $\mathbf{A}$ is often constrained by the application (e.g., subsampled Fourier matrices in medical imaging), but can sometimes be designed freely. In theoretical studies, the most widely-considered type of measurement matrix is *i.i.d. Gaussian*, in which each entry of $\mathbf{A}$ is independently drawn from $\mathcal{N}(0,1)$, $\mathcal{N}(0,\frac{1}{m})$, or similar (the choice of normalization varies for convenience of the analysis). For probabilistic guarantees such as (6), such designs are often analyzed directly. For deterministic guarantees such as (7) the typical approach is to (i) establish deterministic conditions on $\mathbf{A}$ that suffice to obtain the desired recovery guarantee, and (ii) establish that i.i.d. Gaussian (or other randomized) measurements satisfy those conditions with high probability.

We highlight in particular the *restricted isometry property* (RIP) [20]: The matrix $\mathbf{A}$ satisfies the RIP with parameters (s, δ_s) if, for every $\mathbf{x}\in\mathcal{X}_s$, it holds that

$$(1-\delta_s)\|\mathbf{x}\|_2^2 \leq \|\mathbf{A}\mathbf{x}\|_2^2 \leq (1+\delta_s)\|\mathbf{x}\|_2^2. \tag{8}$$

Intuitively, this property states that $\mathbf{A}$ is nearly orthonormal when restricted to sparse vectors. Certain works instead only required the lower bound on $\|\mathbf{A}\mathbf{x}\|_2^2$ in (8), and this variant is known as the *restricted eigenvalue condition* (REC) [16].

Information-theoretic lower bounds: In the above discussion, we highlighted that various upper bounds on the number of measurements have been obtained for attaining recovery guarantees such as (6) and (7). These are complemented by *information-theoretic lower bounds*, which state that any sparse recovery algorithm attaining a certain guarantee must have a minimum number of measurements. Such results are crucial in certifying the degree of optimality of practical algorithms, and steering research towards cases where the greatest improvements are possible.

Lower bounds for sparse recovery have been obtained for a variety of recovery criteria (e.g., see [8], [21], [45], [103]), often with scaling laws that match existing upper bounds. Among these, we highlight the fact that any algorithm attaining the ℓ_2/ℓ_2 guarantee in (6) with constant probability must have $m = \Omega(s\log\frac{n}{s})$, thus matching the above-mentioned upper bound to within a constant factor. A proof of this result is given in [103], based on a reduction to a communication problem over a Gaussian channel.

Practical decoding techniques: Recovery guarantees, often with a near-optimal number of measurements, have been attained for a wide range of practical decoding techniques. For instance, the RIP and/or REC have been used as a tool for studying guarantees of convex relaxation algorithms, thresholding algorithms, and greedy algorithms (e.g., see [44, Ch. 6] and [16]). The class of convex relaxation algorithms can roughly be viewed as trying to find $\mathbf{x}$ such that both $\|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2$ and $\|\mathbf{x}\|_0$ (the number of non-zeros in $\mathbf{x}$) are small, but to circumvent the combinatorial nature of the latter, the convex proxy $\|\mathbf{x}\|_1$ is used. A famous example is the least absolute shrinkage and selection operator (Lasso) method, in which $\hat{\mathbf{x}}$ is the solution to

$$\min_{\mathbf{x}}\|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2^2+\lambda\|\mathbf{x}\|_1 \tag{9}$$

for some regularization parameter $\lambda > 0$. This is a convex optimization problem for which numerous solvers are available that converge to the optimal solution.

In principle, (9) could be solved using off-the-shelf convex optimization solvers, but due to the ubiquity of Lasso, several special-purpose iterative algorithms have also been devised. In Section IV, one such algorithm called the iterative shrinkage thresholding algorithm (ISTA) [35] will play a major role.

D. Overview of the Paper

Our goal is to provide an introduction to several theoretical results on deep learning methods in inverse problems. In addition, we seek to highlight interesting connections between these results, and to discuss ongoing challenges and open problems. We provide intuition behind several of the associated proofs, but avoid going into significant technical detail.

The structure of the paper is as follows.

- In Section II, we overview several theoretical developments concerning generative priors in inverse problems, including statistical guarantees, information-theoretic limits, and optimization guarantees.
- In Section III, we overview theoretical developments regarding neural network priors with no prior training, including provable recovery guarantees for denoising and compressive sensing.
- In Section IV, we overview theoretical developments regarding unfolding algorithms, focusing on sparse signal priors and neural network structures that are based on the classical ISTA algorithm.
- In Section V, we discuss other uses of deep learning in inverse problems, highlighting additional relevant existing theory, as well as scenarios where theory is currently lacking but may be of interest. Several directions for

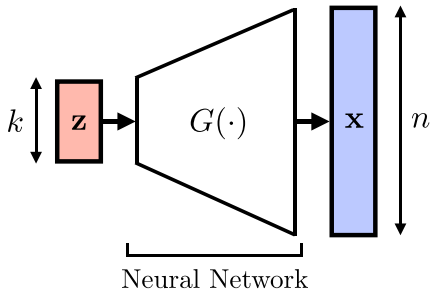

Fig. 1. High-level structure of a typical deep generative model. In the case of 2D images, the length-n vector represents the vectorized form.

future research are additionally mentioned throughout Sections II–IV.

We emphasize that our goal is not to be exhaustive or near-exhaustive in covering the existing literature. While we seek to cover a diverse set of perspectives and results, the ones that we focus on are naturally heavily influenced by our own backgrounds and interests.

Notation: We make frequent use of the standard asymptotic notation $O(\cdot)$ and $\Omega(\cdot)$ (note that $f_n = \Omega(g_n) \iff g_n = O(f_n)$). The ReLU function is given by $\text{relu}(z) = \max\{0, z\}$, and is applied element-wise when applied to vectors. Further notation will be introduced throughout the relevant sections.

II. Generative Priors

In this section, we overview a recent line of works studying theoretical guarantees for inverse problems with generative priors. We begin by outlining the relevant background, and then state some statistical upper and lower bounds. We then turn to guarantees for specific optimization procedures.

A. Background

As outlined in Section I-B, the idea of this line of works is to replace conventional priors (e.g., sparse or low-rank models) by *data-driven generative priors* that can be much more specifically targeted to the task at hand. Given a generative network $G : \mathbb{R}^k \to \mathbb{R}^n$ that accurately models the signals we are interested in, it is natural to decode by outputting a signal in $\text{Range}(G)$ that best matches the measurements in some sense (e.g., $\|\mathbf{y} - \mathbf{A}G(\mathbf{z})\|_2$ is small). This idea is captured by equations (11)–(12) and (14)–(15) to follow.

The structure of a typical generative model is depicted in Figure 1. The function G maps a low-dimensional input $\mathbf{z} \in \mathbb{R}^k$ to a high-dimensional signal $\mathbf{x} \in \mathbb{R}^n$, with the internal structure of G typically being a neural network. As a toy example, with $k = 1$ and $n = 2$, the function

$$G(z) = \big[\sin(z), \cos(z)\big]^T \tag{10}$$

maps $z \in [-\pi, \pi]$ to points on the unit circle in $\mathbb{R}^2$. As a more realistic example, for a relatively simple data set such as MNIST, G might consist of k in the tens and produce 28×28 images (i.e., $n = 784$), whereas a generative model for face images might have k in the hundreds, and a number of pixels in the thousands or more.

Broadly, the use of generative priors in inverse problems consists of two main steps that are typically decoupled.

(i) Given suitably representative training data, train the generative model G (or find a pre-trained one).

(ii) Given the generative model G and compressed measurements such as $\mathbf{y} = \mathbf{A}\mathbf{x}^*$, run an optimization procedure (e.g., (11)–(12) below) to produce an estimate $\hat{\mathbf{x}}$ of $\mathbf{x}^*$ lying in (or near) the range of G.

Step (i) has been widely studied in the machine learning literature, with prominent methods including generative adversarial networks [48], variational autoencoders [76], and so on.

At first glance, performing a theoretical analysis for signal recovery in this setup may appear to be daunting. A typical neural network induces a highly complicated non-linear mapping; the network architecture and training algorithm may play a major role; and using training data inevitably leads to challenges relating to generalization error.

The pioneering work of Bora et al. [17] circumvented these challenges by identifying simple properties of typical generative models that suffice to give meaningful recovery guarantees. As a result, more fine-grained issues centered around training, generalization, and representation error are essentially abstracted away (though their further study would still be of significant interest).

Specifically, the following two mathematical classes of generative models were proposed in [17].

(i) G is a Lipschitz continuous function, with Lipschitz constant denoted by L;

(ii) G is a neural network with ReLU activations,[3] and the width and depth of the network are denoted by w and d.

The Lipschitz assumption can easily be shown to be satisfied by neural networks with Lipschitz activation functions (e.g., ReLU, sigmoid, and more) and bounded weights, and the ReLU network assumption is also natural in view of the ubiquity of ReLU networks in practice. While the second class is essentially encompassed by the first, it is still of interest to study it separately, since doing so yields slightly stronger results, as well as further insights via a distinct analysis.

B. Statistical Upper Bounds on the Reconstruction Error

The following two theorems give upper bounds on the reconstruction error (in terms of the number of measurements m) under the Lipschitz and ReLU assumptions, respectively, considering a (possibly impractical) decoding rule based on solving a constrained ℓ_2-minimization problem.

Theorem 1 (Upper Bound for Lipschitz Generative Models [17, Th. 1.2]): Let $G : \mathbb{R}^k \to \mathbb{R}^n$ be an L-Lipschitz generative model, and let the measurement matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ have i.i.d. $\mathcal{N}(0, \frac{1}{m})$ entries. Suppose that, upon observing $\mathbf{y} = \mathbf{A}\mathbf{x}^* + \boldsymbol{\eta}$ for some noise vector $\boldsymbol{\eta}$, the decoder forms the estimate

$$\hat{\mathbf{x}} = G(\hat{\mathbf{z}}), \quad \text{where} \tag{11}$$

$$\hat{\mathbf{z}} = \underset{\mathbf{z} \in \mathbb{R}^k \,:\, \|\mathbf{z}\|_2 \le r}{\arg\min} \|\mathbf{y} - \mathbf{A}G(\mathbf{z})\|_2. \tag{12}$$

[3]Other piecewise linear activations can also be considered, but ReLU is of primary interest due to its widespread use in practice.

Then, for any $\delta \in (0, 1)$, if $m = \Omega(k \log \frac{Lr}{\delta})$ with a sufficiently large implied constant, then it holds with probability $1-e^{-\Omega(m)}$ that

$$\left\|\hat{\mathbf{x}} - \mathbf{x}^*\right\|_2 \leq 6 \min_{\mathbf{z}\in\mathbb{R}^k \,:\, \|\mathbf{z}\|_2 \leq r} \left\|G(\mathbf{z}) - \mathbf{x}^*\right\|_2 + 3\|\boldsymbol{\eta}\|_2 + 2\delta. \quad (13)$$

Theorem 2 (Upper Bound for ReLU Generative Models [17, Th. 1.1]): Let $G : \mathbb{R}^k \to \mathbb{R}^n$ be a neural network with ReLU activations, width w, and depth d, and let the measurement matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ have i.i.d. $\mathcal{N}(0, \frac{1}{m})$ entries. Suppose that, upon observing $\mathbf{y} = \mathbf{A}\mathbf{x}^* + \boldsymbol{\eta}$ for some noise vector $\boldsymbol{\eta}$, the decoder forms the estimate

$$\hat{\mathbf{x}} = G\big(\hat{\mathbf{z}}\big), \quad \text{where} \quad (14)$$

$$\hat{\mathbf{z}} = \arg\min_{\mathbf{z}\in\mathbb{R}^k} \|\mathbf{y} - \mathbf{A}G(\mathbf{z})\|_2. \quad (15)$$

Then, if $m = \Omega(kd \log w)$ with a sufficiently large implied constant, then it holds with probability $1 - e^{-\Omega(m)}$ that

$$\left\|\hat{\mathbf{x}} - \mathbf{x}^*\right\|_2 \leq 6 \min_{\mathbf{z}\in\mathbb{R}^k} \left\|G(\mathbf{z}) - \mathbf{x}^*\right\|_2 + 3\|\boldsymbol{\eta}\|_2. \quad (16)$$

We note that these results provide *non-uniform* recovery guarantees, holding with respect to the randomness in $\mathbf{A}$ for fixed $\mathbf{x}^*$. However, as noted in [89, Remark 1], if there is no representation error (i.e., $\mathbf{x}^* \in \text{Range}(G)$ and the first term in (13) or (16) is zero), the proofs in [17] also provide stronger *uniform* guarantees, establishing that a single matrix $\mathbf{A}$ works for all $\mathbf{x}^*$.

The first term in (13) (and (16)) amounts to being within a constant factor of the best approximation (thus measuring the *representation error*), and the second term captures the effect of noise. The 2δ term in (13) is more subtle, and captures the fact that more measurements are needed to accurately recover details of the signal at increasingly fine scales [17]. In contrast, no such term is present in (16).

The number of measurements above can be contrasted with the typical $O(s \log n)$ scaling for sparse priors. The most important distinction here is not the different logarithmic terms, but rather, the fact that *for accurate modeling, the required k (generative priors) may be much smaller than the required s (sparse priors)* due to G being more targeted to the task at hand.

Slightly more general statements are given in [17], in which the minimization problem in (12) or (15) is only solved to within ϵ, and 2ϵ is added to the right-hand side of (13) or (16). While gradient-based methods can be highly effective in practice [17], rigorously guaranteeing ϵ-optimality for small $\epsilon > 0$ may be very difficult due to the potentially complicated (e.g., highly non-convex) optimization landscape. In Section II-D, we summarize some results that overcome this limitation, at the expense of imposing stronger assumptions on G.

Overview of proofs: The proofs of both Theorems 1 and 2 are based on the *set-restricted eigenvalue condition* (S-REC), which formalizes the intuition that $\mathbf{A}\mathbf{x}_1$ and $\mathbf{A}\mathbf{x}_2$ should not be too close relative to the separation between two possible signals $\mathbf{x}_1$ and $\mathbf{x}_2$. For instance, if $\mathbf{A}\mathbf{x}_1 = \mathbf{A}\mathbf{x}_2$ then clearly the two cannot be distinguished. More generally, $\mathbf{x}_1 - \mathbf{x}_2$ should be *far from the nullspace* of $\mathbf{A}$.

Definition 1 (Set-Restricted Eigenvalue Condition (S-REC) [17, Definition 1]): Fix $\mathcal{S} \subseteq \mathbb{R}^n$, along with $\gamma > 0$ and $\delta \geq 0$. The matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ is said to satisfy the S-REC$(\mathcal{S}, \gamma, \delta)$ if, for all $\mathbf{x}_1$ and $\mathbf{x}_2$ in $\mathcal{S}$, it holds that $\|\mathbf{A}(\mathbf{x}_1 - \mathbf{x}_2)\|_2 \geq \gamma\|\mathbf{x}_1 - \mathbf{x}_2\|_2 - \delta$.

Notice that this definition bounds $\|\mathbf{A}(\mathbf{x}_1 - \mathbf{x}_2)\|_2$, whereas analogous definitions based on sparsity simply bound $\|\mathbf{A}\mathbf{x}\|_2$ for sparse $\mathbf{x}$ (e.g., see (8)). Intuitively, this is because $\|\mathbf{A}(\mathbf{x}_1 - \mathbf{x}_2)\|_2$ is the more directly relevant quantity, but $\|\mathbf{A}\mathbf{x}\|$ can be used for sparse signals since the difference of two sparse signals is still sparse (unlike for general generative priors).

It is shown in [17] that the S-REC$(\mathcal{S}, \gamma, \delta)$ with $\gamma = \frac{1}{2}$, coupled with a simpler property of the form $\|\mathbf{A}\mathbf{x}\| \leq 2\|\mathbf{x}\|$ (for some fixed $\mathbf{x}$), suffices to establish a recovery guarantee of the form (13) or (16), with the minimum being taken over $\mathcal{S}$. Since $\|\mathbf{A}\mathbf{x}\|_2 \leq 2\|\mathbf{x}\|_2$ holds with high probability by standard Gaussian concentration, it only remains to show that Gaussian matrices satisfy the S-REC with high probability.

When G satisfies the Lipschitz property (Theorem 1), the idea is to establish the desired behavior on a finite subset of $\mathcal{S} = \{G(\mathbf{z}) : \|\mathbf{z}\|_2 \leq r\}$, and then transfer this to the full set.[4] When working with a finite subset, one can study the norm-preserving properties of Gaussian matrices, as pioneered by Johnson and Lindenstrauss [71]. The rough intuition behind the scaling on m is that we need to cover $\mathcal{S}$ such that every signal in $\mathcal{S}$ is δ-close to some point, and by the Lipschitz property of G, this amounts to similarly covering $\{\mathbf{z} \in \mathbb{R}^k : \|\mathbf{z}\|_2 \leq r\}$ with closeness $\frac{\delta}{L}$. This is known to be possible with a set of size $\exp(O(k \log \frac{Lr}{\delta}))$, and the scaling on m arises as the log of this size.

For ReLU neural networks (Theorem 2), the idea is that since the ReLU activation function is piecewise linear, so is the overall function G (possibly with a huge number of pieces). Within a linear region, one can again appeal to standard norm-preserving properties of Gaussian matrices, and a union bound can then be applied over all pieces. A counting argument reveals that there are $w^{O(kd)}$ such pieces, and the bound on m arises as the log of this number.

C. Information-Theoretic Lower Bounds

To assess the degree of optimality of the upper bounds, it is useful to establish *information-theoretic lower bounds* (i.e., converse/impossibility results) stating that no estimation procedure can hope to improve beyond a certain limit, in terms of the estimation error and/or number of measurements. Results of this kind were independently established by Kamath et al. [73] and Liu and Scarlett [89].

The following theorem of [73] provides such a lower bound in the case of Lipschitz continuous generative priors, and serves as a counterpart to the upper bound in Theorem 1.

Theorem 3 (Lower Bound for Lipschitz Generative Models [73, Th. 1.1]): For any input/output sizes k and n, and positive constants L, r, and δ such that $\log \frac{Lr}{\delta} \geq 1$, there exists a generative model $G : \mathbb{R}^k \to \mathbb{R}^n$ such that the

[4] More precisely, to avoid a worsened logarithmic factor, [17] adopts a chaining argument that studies a *sequence* of finite sets corresponding to increasingly fine scales.

following holds: If there exists a random measurement matrix $\mathbf{A}$ and a decoder (with access to $\mathbf{A}$ and $\mathbf{y} = \mathbf{A}\mathbf{x}^*$) that is guaranteed to return $\hat{\mathbf{x}}$ satisfying

$$\left\|\hat{\mathbf{x}} - \mathbf{x}^*\right\|_2 \leq C \min_{\mathbf{z}\in\mathbb{R}^k \,:\, \|\mathbf{z}\|\leq r} \left\|G(\mathbf{z}) - \mathbf{x}^*\right\|_2 + \delta \tag{17}$$

with probability at least $\frac{3}{4}$ for some absolute constant C, then it must be the case that

$$m = \Omega\Big(\min\Big\{k \log \frac{Lr}{\delta}, n\Big\}\Big). \tag{18}$$

This result establishes that $O(k \log \frac{Lr}{\delta})$ is indeed the correct scaling (in the most interesting regime where this quantity is below $O(n)$), and that the additive dependence on δ in (13) is unavoidable, unlike the case of a sparse prior (see (6)). We note that this result holds for a "worst-case" generative model satisfying the assumptions of Theorem 1; it may very well be the case that further assumptions on G can decrease the required m.

Theorem 3 concerns the case that there is no noise (i.e., $\boldsymbol{\eta} = \mathbf{0}$), but crucially relies on considering signals with representation error in order to establish the hardness result. The opposite approach was taken in [89], in which it was assumed that there is no representation error, but that $\boldsymbol{\eta}$ is present in the form of i.i.d. Gaussian noise. An analog of Theorem 3 was given, though Theorem 3 has the advantage of holding for general combinations of (n, k, L, r, δ), whereas [89] requires n to be large enough such that $\log \frac{Lr}{\delta} = O(\log \frac{n}{k})$.

An advantage of the approach in [89], on the other hand, is that it also provides a lower bound establishing conditions under which Theorem 2 is near-optimal, i.e., handling the specific case of ReLU generative models, and characterizing the dependence on the network depth and width.

Before stating this lower bound for ReLU networks, it is useful to highlight what the upper bound in Theorem 2 gives in the case of Gaussian noise and no representation error. As stated in [89, Corollary 2], if we have

$$\mathbf{x}^* \in \text{Range}(G), \quad \text{and} \quad \boldsymbol{\eta} \sim N\Big(\mathbf{0}, \frac{\alpha}{m}\mathbf{I}_m\Big) \tag{19}$$

for some $\alpha > 0$, then there exists a measurement matrix[5] $\mathbf{A} \in \mathbb{R}^{m\times n}$ with squared Frobenius norm $\|\mathbf{A}\|_{\text{F}}^2 \leq n$ such that the mean squared error is upper bounded by

$$\mathbb{E}\Big[\left\|\hat{\mathbf{x}} - \mathbf{x}^*\right\|_2^2\Big] \leq O(\alpha). \tag{20}$$

Intuitively, this amounts to accurately reconstructing $\mathbf{x}$ with the amount of error matching the noise level.

The lower bound in this setting is more complicated than the case of Lipschitz generative models, so we provide an informal statement, and refer the reader to [89] for the details.

Theorem 4 (Lower Bound for ReLU Networks (Informal) [89, Th. 7]): Consider the case that $G : \mathbb{R}^k \to \mathbb{R}^n$ is a ReLU network with depth d and width w. Suppose that there exists a measurement matrix $\mathbf{A}$ with $\|\mathbf{A}\|_{\text{F}}^2 \leq n$ and a decoder such that when (19) holds, the resulting estimate $\hat{\mathbf{x}}$ is guaranteed to satisfy (20). Then, we have the following:

- There exists G with depth $d = 2$ and large width w such that it must be the case that $m = \Omega(k \log w)$.
- There exists G with width $w = O(n)$ and large depth d such that it must be the case that $m = \Omega(kd)$.
- There exists G with simultaneously large width and large depth such that it must be the case that $m = \Omega(kd \frac{\log w}{\log n})$.

Observe that the number of measurements matches the $O(kd \log w)$ upper bound to within a constant factor (first case) or an $O(\log n)$ factor (second and third cases). We note that certain cases are known where the logarithmic factor in the upper bound can be slightly reduced [99].

Overview of proofs: As is common in proving information-theoretic lower bounds, the high-level idea behind Theorems 3 and 4 is to establish that the relevant recovery guarantee implies being able to reliably distinguish certain well-separated signals. If there are many such signals, then reliably distinguishing them amounts to learning a certain amount of information, and since each measurement only provides a limited amount of information, a lower bound on the number of measurements follows.

In [73], the details are based on a reduction to communication complexity. A subset $\mathcal{X}_0$ of well-separated binary-valued signals is formed with $\log|\mathcal{X}_0| = \Omega(\min\{k \log \frac{Lr}{\delta}, n\})$, and $\mathbf{x}$ is restricted to be a weighted linear combination of several such signals plus a small Gaussian perturbation. A communication game is set up in which one party wishes to identify one of the binary-valued signals, and for which a lower bound on the number of bits transmitted is known for achieving constant-probability success. It is shown that transmitting a fine discretization of $\mathbf{y} = \mathbf{A}\mathbf{x}^* \in \mathbb{R}^m$ suffices for such success, from which a lower bound on m follows.

In [89], to prove Theorem 4 and a counterpart to Theorem 3, a different approach is taken. The idea is to construct a generative model G that produces *sparse signals*,[6] and then apply standard lower bounding techniques (e.g., based on Fano's inequality) that characterize the hardness of sparse recovery. By studying the Lipschitz constant and/or the depth and width of G, and combining these with the relevant lower bounds for sparse recovery, the desired results follow. An illustration of why neural networks can produce sparse signals is shown in Figure 2; the piecewise linear functions can readily be implemented using ReLU networks. An analog of Theorem 1 is obtained by forming a network that produces k-sparse signals (with input $\mathbf{z} \in \mathbb{R}^k$), whereas Theorem 2 is based on producing kk_0-sparse signals with $k_0 > 1$, using recursively-defined mappings that operate at k_0 different scales.

We refer the reader to [73], [89] for the full details of the above proof outlines.

D. Optimization Guarantees for Random Generative Priors

As we mentioned above, finding an optimal or near-optimal solution to problems such as (12) and (15) may not be possible with an efficient algorithm. Thus, there is substantial

[5]Equation (20) also holds when $\mathbf{A}$ is i.i.d. Gaussian according to Theorem 2, but it is convenient to work with fixed $\mathbf{A}$ in this part. As discussed following Theorem 2, the upper bound with fixed $\mathbf{A}$ crucially relies on having no representation error. In view of this, Theorem 4 as stated may appear to have a weakness of only lower bounding the number of measurements under a stricter uniform recovery guarantee. However, it is discussed in [89, Remark 1] that the proof readily provides a similar statement for non-uniform recovery.

[6]The ability of ReLU networks to produce sparse signals was also noted in [73], but no analog of Theorem 4 was sought.

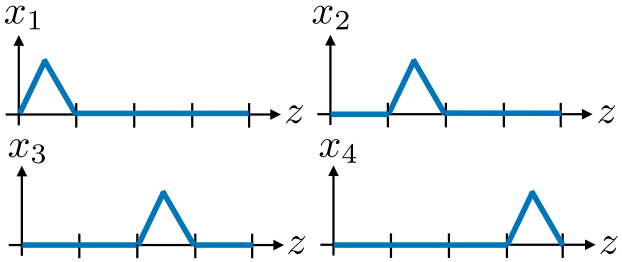

Fig. 2. Example function mapping $z \in \mathbb{R}$ to $\mathbf{x} = (x_1, x_2, x_3, x_4) \in \mathbb{R}^4$ such that the resulting signal is 1-sparse (or is the zero vector).

motivation to give recovery guarantees for specific tractable optimization procedures (which comes at the expense of stronger assumptions on G). In this subsection, we outline some examples of such guarantees.

We again consider $G : \mathbb{R}^k \to \mathbb{R}^n$ being a ReLU neural network, but now with two main additional assumptions, namely, (i) sufficient expansivity (i.e., increase in the number of nodes) from layer to layer, and (ii) random Gaussian network weights. Due to the second assumption, such networks would not produce meaningful signals in practice. However, as noted in [55], some trained networks do exhibit Gaussian-like statistics, and more importantly, understanding random networks is already highly challenging and serves as a good starting point towards increasingly more realistic scenarios.

We focus primarily on the results of Hand and Voroninski [55] and Huang et al. [62]. It was first shown in [55] that the optimization landscape in formulation (15) is favorable for gradient algorithms if the network architecture satisfies certain deterministic properties and if there are a sufficient number of random Gaussian measurements. Inspired by this landscape, [62] introduced a specific subgradient algorithm that provably converges. It was additionally established in [55] that under the above-mentioned assumptions of expansivity and random weights, the desired deterministic properties are satisfied with high probability. The assumptions made were then relaxed in various subsequent works [28], [34], [72], several of which we will discuss in Section II-E.

1) Model for G: We consider a generator $G : \mathbb{R}^k \to \mathbb{R}^n$ given by a d-layer fully connected neural network with ReLU activations and no bias terms. That is,

$$G(\mathbf{z}) = \text{relu}(\mathbf{W}_d \dots \text{relu}(\mathbf{W}_2\, \text{relu}(\mathbf{W}_1 \mathbf{z})) \dots), \tag{21}$$

where $\text{relu}(\cdot) = \max\{\cdot, 0\}$ applies entry-wise, $\mathbf{W}_i \in \mathbb{R}^{n_i \times n_{i-1}}$ for $i = 1, \dots, d$, and $n_0 = k$ and $n_d = n$.

2) Deterministic Conditions Used in the Analysis: Here we present two useful deterministic conditions on the generative model and measurement model. The results to follow will show that these deterministic conditions are sufficient for certain recovery guarantees, and are satisfied with high probability for i.i.d. Gaussian distributions on G and $\mathbf{A}$.

The first condition is the Weight Distribution Condition (WDC), which applies to individual weight matrices $\mathbf{W}_i$.

Definition 2 (Weight Distribution Condition (WDC) [55]): A matrix $\mathbf{W} \in \mathbb{R}^{\kappa \times \ell}$ satisfies the *Weight Distribution Condition* with constant ϵ if, for all non-zero $\mathbf{u}, \mathbf{v} \in \mathbb{R}^\ell$, it holds that

$$\left\| \sum_{i=1}^{\kappa} \mathbb{1}_{\mathbf{w}_i \cdot \mathbf{u} > 0} \mathbb{1}_{\mathbf{w}_i \cdot \mathbf{v} > 0} \cdot \mathbf{w}_i \mathbf{w}_i^T - \mathbf{Q}_{\mathbf{u},\mathbf{v}} \right\|_2 \le \epsilon,$$
$$\text{with } \mathbf{Q}_{\mathbf{u},\mathbf{v}} = \frac{\pi - \theta}{2\pi} \mathbf{I} + \frac{\sin\theta}{2\pi} \mathbf{M}_{\mathbf{u},\mathbf{v}}, \tag{22}$$

where $\mathbf{w}_i^T \in \mathbb{R}^\ell$ is the i-th row of $\mathbf{W}$; θ is the angle between $\mathbf{u}$ and $\mathbf{v}$; $\mathbf{M}_{\mathbf{u},\mathbf{v}} \in \mathbb{R}^{\ell \times \ell}$ is the matrix that maps $\frac{\mathbf{u}}{\|\mathbf{u}\|_2} \mapsto \frac{\mathbf{v}}{\|\mathbf{v}\|_2}$, $\frac{\mathbf{v}}{\|\mathbf{v}\|_2} \mapsto \frac{\mathbf{u}}{\|\mathbf{u}\|_2}$, and $\mathbf{t} \mapsto 0$ for all $\mathbf{t}$ orthogonal to $\text{span}(\{\mathbf{u}, \mathbf{v}\})$; and $\mathbb{1}_S$ is the indicator function on S.

This condition can be viewed as a generalization of an approximate isotropy condition; for example, if $\mathbf{u} = \mathbf{v}$, the condition states that $\sum_{i=1}^{\kappa} \mathbb{1}_{\mathbf{w}_i \cdot \mathbf{u} > 0} \mathbb{1}_{\mathbf{w}_i \cdot \mathbf{v} > 0} \cdot \mathbf{w}_i \mathbf{w}_i^T$ is close to $\frac{1}{2}\mathbf{I}$. The indicator functions in the summation arise from taking the derivative of the ReLU function.

The second condition is the Range Restricted Isometry Condition (RRIC), which applies to the pair $(G, \mathbf{A})$.

Definition 3 (Range Restricted Isometry Condition (RRIC) [55]): A matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ satisfies the *Range Restricted Isometry Condition* with respect to G with constant ϵ if, for all $\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \mathbf{z}_4 \in \mathbb{R}^k$, it holds that

$$\Big| \langle \mathbf{A}(G(\mathbf{z}_1) - G(\mathbf{z}_2)), \mathbf{A}(G(\mathbf{z}_3) - G(\mathbf{z}_4)) \rangle - \langle G(\mathbf{z}_1) - G(\mathbf{z}_2), G(\mathbf{z}_3) - G(\mathbf{z}_4) \rangle \Big| \le \epsilon \|G(\mathbf{z}_1) - G(\mathbf{z}_2)\|_2 \|G(\mathbf{z}_3) - G(\mathbf{z}_4)\|_2. \tag{23}$$

This condition states that $\mathbf{A}$ acts like an isometry when acting on pairs of secant directions (i.e., differences of two signals) with respect to the range of G.

3) Favorable Landscape for Compressive Sensing With Gradient Algorithm Under Deterministic Conditions: Under the deterministic conditions given above, it can be established that the loss landscape is favorable for optimization. Consider a signal given by $\mathbf{x}^* = G(\mathbf{z}^*)$ for some $\mathbf{z}^*$, and let the measurement vector be $\mathbf{y} = \mathbf{A}\mathbf{x}^* + \boldsymbol{\eta}$ with i.i.d. Gaussian $\boldsymbol{\eta}$. We are interested in the optimization problem

$$\min_{\mathbf{z}} f(\mathbf{z}), \quad f(\mathbf{z}) := \|\mathbf{A}G(\mathbf{z}) - \mathbf{y}\|_2^2. \tag{24}$$

The following result shows that under the WDC and RRIC, f does not have any spurious local minima outside of $\mathbf{z}$ and a negative multiple of $\mathbf{z}$. Here and subsequently, when we write $\text{poly}(d)$, we mean that the result holds true when this is replaced by d^c for a suitable constant $c > 0$ (possibly differing in each occurrence). In addition, we let $D_{\mathbf{v}} f(\mathbf{z})$ denote the directional derivative with direction $\mathbf{v} \in \mathbb{R}^k$, and let $\mathcal{B}(\mathbf{z}, r)$ denote the radius-r ball centered at $\mathbf{z}$.

Theorem 5 (Favorable Optimization Landscape [56, Th. 4]): Fix $\epsilon > 0$ such that $K_1 \text{poly}(d)\epsilon^{1/4} \le 1$, and let $d \ge 2$. Suppose that G is such that $\mathbf{W}_i$ satisfies the WDC with constant ϵ for all $i = 1, \dots, d$, and that $\mathbf{A}$ satisfies the RRIC with respect to G with constant ϵ. Then, for all non-zero $\mathbf{z}$ and $\mathbf{z}^*$, there exists $\mathbf{v}_{\mathbf{z},\mathbf{z}^*} \in \mathbb{R}^k$ such that the one-sided directional derivatives of f satisfy

$$D_{-\mathbf{v}_{\mathbf{z},\mathbf{z}^*}} f(\mathbf{z}) < -K_3 \frac{\sqrt{\epsilon}\, \text{poly}(d)}{2^d} \max\left\{ \|\mathbf{z}\|_2, \|\mathbf{z}^*\|_2 \right\}, \tag{25}$$
$$D_{\mathbf{t}} f(\mathbf{0}) < -\frac{1}{8\pi 2^d} \|\mathbf{z}^*\|_2,$$

$$\forall \mathbf{t} \neq 0, \mathbf{z} \notin \{\mathbf{0}\} \cup \mathcal{B}\Big(\mathbf{z}^*, K_2 \text{poly}(d)\epsilon^{1/4}\|\mathbf{z}^*\|_2\Big) \cup \mathcal{B}\Big(-\rho\mathbf{z}^*, K_2 \text{poly}(d)\epsilon^{1/4}\|\mathbf{z}^*\|_2\Big), \quad (26)$$

where $\rho = \rho_d$ is a positive number that converges to 1 as $d \to \infty$, and K_1, K_2, and K_3 are universal constants.

While the above expressions are somewhat technical, the simple idea is that except for points close to $\mathbf{z}^*$ and $-\rho\mathbf{z}^*$, we have a negative upper bound on the directional derivative, which precludes spurious minima. Moreover, the radius around $\mathbf{z}^*$ and $-\rho\mathbf{z}^*$ becomes arbitrarily small as ϵ decreases.

There is an explicit formula for $\mathbf{v}_{\mathbf{z},\mathbf{z}^*}$, given by

$$\mathbf{v}_{\mathbf{z},\mathbf{z}^*} = \begin{cases} \nabla f(\mathbf{z}) & \text{differentiable at } \mathbf{z}, \\ \lim_{\delta \downarrow 0} \nabla f\big(\mathbf{z} + \delta\mathbf{z}'\big) & \text{otherwise}, \end{cases} \quad (27)$$

where $\mathbf{z}'$ can be arbitrarily chosen such that G is differentiable at $\mathbf{z} + \delta\mathbf{z}'$ for sufficiently small δ. Such a $\mathbf{z}'$ exists by the piecewise linearity of G, and can be generated randomly with probability one.

Note that the dependence on 2^d in the bounds is an artifact of the underlying scaling of $f(\mathbf{z})$, and does not indicate a vanishingly small derivative. Roughly speaking, the ReLU activation functions zero out around half of its arguments. Hence, while $\mathbf{W}_i$ has spectral norm approximately one, the rows of $\mathbf{W}_i$ that are retained by the ReLU will have spectral norm approximately $\frac{1}{2}$. Thus, $f(\mathbf{z})$ itself is on the order of 2^{-d} under the RRIC and WDC for appropriately small ϵ.

4) Subgradient Algorithm and Convergence Guarantee Under Deterministic Conditions: Building on Theorem 5, [62] proposed a subgradient algorithm and showed that it has a rigorous convergence guarantee. Since the cost function $f(\mathbf{z})$ is continuous, piecewise quadratic, and not differentiable everywhere, the algorithm is defined with respect to a generalized gradient, called the Clarke subdifferential, generalized subdifferential, or generalized subgradient (e.g., see [27] for the definition).

The algorithm operates as follows given some initialization:

- Compute a vector in the subgradient of the objective at the current iterate;
- Update the current position using the subgradient and a fixed step size;
- If negating the current iterate reduces the value of the objective, then do so;
- Repeat until a stopping criterion is met.

Note that the third step is non-standard, and is motivated by the landscape properties stated in Theorem 5.

Theorem 6 (Optimization Guarantee [62, Th. 1]): Suppose that the WDC and RRIC hold with $\epsilon \leq \frac{C_1}{\text{poly}(d)}$, and the noise $\boldsymbol{\eta}$ satisfies $\|\boldsymbol{\eta}\|_2 \leq \frac{C_2\|\mathbf{z}^*\|_2}{\text{poly}(d)2^{d/2}}$. Consider the iterates $\{\mathbf{z}_t\}$ generated by the preceding algorithm with step size $\nu = C_3\frac{2^d}{\text{poly}(d)}$. There exists a number of iterations, denoted by τ and upper bounded by $\tau \leq \frac{C_4 f(\mathbf{z}_0)2^d}{\text{poly}(d)\epsilon\|\mathbf{z}^*\|_2}$ when the initialization is $\mathbf{z}_0$, such that

$$\big\|\mathbf{z}_\tau - \mathbf{z}^*\big\|_2 \leq C_5 \text{poly}(d)\sqrt{\epsilon}\|\mathbf{z}^*\|_2 + C_6 \text{poly}(d)2^{d/2}\|\boldsymbol{\eta}\|_2. \quad (28)$$

In addition, for all $t \geq \tau$, we have

$$\big\|\mathbf{z}_{t+1} - \mathbf{z}^*\big\|_2 \leq C^{t+1-\tau}\|\mathbf{z}_\tau - \mathbf{z}^*\|_2 + C_7 2^{d/2}\|\boldsymbol{\eta}\|_2, \quad (29)$$

and

$$\|G(\mathbf{z}_{t+1}) - G(\mathbf{z}^*)\|_2 \leq \tfrac{1.2}{2^{d/2}} C^{t+1-\tau}\|\mathbf{z}_\tau - \mathbf{z}^*\|_2 + 1.2C_7\|\boldsymbol{\eta}\|_2, \quad (30)$$

where $C = 1 - \frac{\nu}{2^d}\frac{7}{8} \in (0, 1)$. Here, $C_1, \ldots, C_7$ are universal positive constants.

In accordance with the above discussion on the dependence on 2^d, the initial value $f(\mathbf{z}_0)$ scales with respect to d as 2^{-d} under the WDC and RRIC. Hence, and in view of the assumption $\epsilon \leq \frac{C_1}{\text{poly}(d)}$, we find that Theorem 6 establishes that after a number of iterations that is polynomial in d, the modified subgradient algorithm converges linearly to $\mathbf{z}^*$, up to the noise level. Note that this convergence guarantee applies for an arbitrary initialization, though more iterations may be required for initializations that are large in norm. As with Theorem 5, while the landscape is non-convex, the theorem establishes under the WDC and RRIC that the non-convexity is mild and does not result in spurious minima, except $-\rho\mathbf{z}^*$ which the above algorithm avoids.

5) Random G Satisfies the Deterministic Conditions With High Probability: Finally, the following result establishes that the WDC (Definition 2) and RRIC (Definition 3) are satisfied with high probability provided that (i) G has i.i.d. Gaussian weights and is sufficiently expansive, and (ii) $\mathbf{A}$ has i.i.d. Gaussian entries and sufficiently many rows.

Proposition 1 (High-Probability Behavior of Random Models [56, Proposition 6]): Fix $0 < \epsilon < 1$. Assume that G follows the structure in (21) with $n_i \geq cn_{i-1}\log n_{i-1}$ for all $i = 1, \ldots, d$, and that $m > ckd \log \Pi_{i=1}^d n_i$. Moreover, assume that the entries of $\mathbf{W}_i$ are i.i.d. $\mathcal{N}(0, \frac{1}{n_i})$, and the entries of $\mathbf{A}$ are i.i.d. $\mathcal{N}(0, \frac{1}{m})$. Then, $\mathbf{W}_i$ satisfies the WDC with constant ϵ for all i and $\mathbf{A}$ satisfies the RRIC with respect to G with constant ϵ with probability at least $1 - O(\sum_{i=1}^d n_i e^{-\gamma n_{i-1}} - e^{-\gamma m})$. Here c and γ^{-1} are constants that depend polynomially on ϵ^{-1}.

Observe that the leading term in the number of measurements is kd, as is the case in Theorem 2. However, depending on the expansivity, the logarithmic term $\log \Pi_{i=1}^d n_i$ can be order-wise larger than $\log w$, and accordingly could potentially be improved.

The proof of Proposition 1 relies on tools from non-asymptotic random matrix theory [137]. Typically, establishing a matrix concentration result like the WDC with high probability would involve three steps: showing high probability concentration of the matrix applied to fixed vectors $(\mathbf{u}, \mathbf{v})$, bounding an appropriate Lipschitz constant, and taking a union bound over a net whose size depends on that Lipshitz constant. Because the matrix in the WDC is discontinuous with respect to $(\mathbf{u}, \mathbf{v})$, this approach must be modified. The authors of [56] show that the discontinuity can be smoothed to provide a semidefinite upper bound on the desired expression, and also smoothed to provide a semidefinite lower bound. Each of these can then be controlled by the standard approach mentioned above.

Along similar lines to the proof of Theorem 2, establishing the RRIC involves showing that the output of linear maps with

ReLU activations live in a union of linear spaces, and counting the number of such subspaces.

E. Further Developments

In this subsection, we provide several examples of follow-up theoretical results related to those outlined above. We keep this summary brief, and refer the reader to the references given for further details.

1) Statistical Guarantees: Some further developments related to the results in Section II-B (and to a lesser extent, Section II-C) are outlined as follows.

Mitigating representation error: While generative priors have clear benefits over conventional priors, they can suffer from the issue of *representation error*: If the signal $\mathbf{x}^*$ is not exactly in the range of G, then an optimization procedure such as (12) will always incur some amount of error no matter how many measurements m we take. In contrast, it is straightforward to devise sparsity-based solutions that are guaranteed to become arbitrarily accurate as m increases. To overcome this limitation, [37] proposed to model $\mathbf{x}^*$ as the sum of a generative component and a sparse component, and gave a theoretical guarantee that combines the features of Theorem 1 and analogous sparse recovery results.

With a similar motivation but a very different approach, various methods were proposed in [31], [32], [51], [63] based on optimizing intermediate layers in the neural network defining G, which helps to expand the range of the generator and mitigate representation error. Conditions were given under which the required number of measurements is provably smaller than Theorem 2, and improvements in the out-of-distribution robustness were observed experimentally.

Non-linear measurement models: While Theorem 1 concerns linear observation models, analogous guarantees have been provided for a variety of non-linear measurement models, including 1-bit observations [69], [85], [105], spiked matrix models [7], [28], phase retrieval [53], [84], principal component analysis [87], and general single-index models [83], [86], [88]. While these each come with their own challenges, the intuition behind their associated results is often similar to that discussed above for the linear model, with the $m = O(k \log \frac{Lr}{\delta})$ scaling typically remaining.

Robustness to outliers: Theorem 1 is primarily suited to well-behaved noise, such as Gaussian or sub-Gaussian. In contrast, heavy-tailed noise with large outliers can considerably worsen the performance, both in theory (e.g., due to the size of $\|\boldsymbol{\eta}\|_2$ in (13)) and in practice (e.g., since (12) is not a robust objective). Algorithms and theory addressing this challenge were given in [68], [139]. Briefly, using robust estimation techniques, one can attain an analog of Theorem 1 even when a constant fraction of the data is drawn from a heavy-tailed distribution that yields large outliers. See also [146] for a theoretical study of outlier detection using generative models.

General probabilistic priors: Theorems 1 and 2 treat G as a fixed function satisfying certain properties, without addressing the fact that even over the range of G, some signals may be more likely than others. This distinction is particularly important when it comes to generative models that fail to satisfy $k \ll n$ (e.g., invertible generative models with $k = n$ [5]). To address this, compressive sensing with general probabilistic priors was studied in [67]. Analogous to how covering properties play a key role in the proof of Theorem 1, it was shown that a *probabilistic* form of covering dictates the required number of measurements. Moreover, it was shown that in broad scenarios, using i.i.d. Gaussian measurements and letting $\hat{\mathbf{x}}$ be a random sample from the posterior of $\mathbf{x}^*$ is near-optimal for estimation.

2) Optimization Guarantees: Some further developments related to the results in Section II-D are outlined as follows.

Weakening the expansivity condition: In Proposition 1, the WDC and RRIC were established with high probability in the case of layer-wise expansivity, that is, $n_i \geq cn_{i-1} \log n_{i-1}$. This assumption was weakened in [34] to $n_i \geq cn_{i-1}$ by introducing the notion of pseudo-Lipschitzness and by placing nets over spheres in a suitably non-uniform manner.

Subsequently, it was shown in [72] and [28] that layer-wise expansivity is not necessary. Specifically, the recovery guarantee is possible even if some layers are contractive (i.e., they have fewer outputs than inputs), provided that all layers are sufficiently large relative to the input dimensionality k. This is shown in [72] in the case of a modified gradient algorithm, and [28] observed that the WDC of a given layer only needs to hold restricted to the range of previous layers.

Alternative architectures: The model (21) assumes that the architecture of the neural network G is fully-connected. In [90], a similar recovery guarantee was established in the case that G has a convolutional architecture.

Other inverse problems: Signal recovery guarantees with random generative priors have been established for a variety of inverse problems, including denoising, blind demodulation, phase retrieval, and spiked matrix models.

Various results on denoising can be found in [1], [55], [58], [78].[7] In particular, it is shown in [58] that solving the optimization problem $\min_{\mathbf{z}} \|\mathbf{y} - G(\mathbf{z})\|_2^2$ with a gradient method yields an optimal denoising rate of $O(\frac{k}{n})$ provided that the noise is sufficiently small and the generative model has Gaussian weights and is sufficiently expansive. An alternative approach that avoids the need for random weights and expansivity is given in [1], instead considering sparsity properties of the hidden layers (resulting from $\text{relu}(z) = 0$ for $z \leq 0$).

In the case of phase retrieval with random weights and expansivity, solving the optimization problem $\min_{\mathbf{z}} \|\mathbf{y} - |\mathbf{A}G(\mathbf{z})|\|_2^2$ allows for signal recovery with m being proportional to k (ignoring the n and d dependence) [53]. This dependence is information-theoretically optimal, and it is noteworthy that it is attained with an efficient algorithm under random generative priors. In contrast, for sparse priors, there is no known practical algorithm that achieves a recovery guarantee with a linear dependence on the sparsity s, even though doing so is known to be information-theoretically possible. See also [54] for simplified arguments in the case of phase retrieval without prior information (i.e., general signals in $\mathbb{R}^n$).

[7] These works also study the question of when $\mathbf{z}^*$ can be recovered from $G(\mathbf{z}^*)$ even in the absence of noise (e.g., see [78] for an NP-hardness result).

Random generative priors have also allowed for recovery results for the case of spiked matrix models [7], [28]. The number of measurements is again shown to be information-theoretically order-optimal using an efficient algorithm, unlike in the case of sparse models.

Other optimization algorithms: Analogous guarantees to Theorem 2 were given in [102], [119] for projected gradient descent, which alternates between gradient steps and projections onto the range of G. However, a notable limitation of such guarantees is that the projection step itself depends on the landscape of $G(\mathbf{z})$, and may accordingly be intractable. The results in Section II-D overcome this limitation at the expense assuming random weights (along with expansivity). Two further works gave guarantees that require neither random weights nor exact projections, but instead adopt further deterministic assumptions on G, roughly amounting to certain forms of smoothness. Specifically, [47] studied an algorithm based on Alternating Direction Method-of-Multipliers (ADMM), and [100] studied an algorithm based on Langevin dynamics. Algorithms based on Langevin dynamics have also been explored in several other works on inverse problems and beyond, e.g., see [67] for its use in posterior sampling with general probabilistic priors, and [106] for a general non-asymptotic analysis under non-convex objectives.

Another important class of algorithms uses *approximate message passing* (AMP), which is a powerful technique that has been utilized extensively in high-dimensional statistics [42]. Variants of AMP have successfully been devised with theoretical guarantees in several inverse problems with generative priors, including linear forward models [43], [93], spiked matrix recovery [7], and phase retrieval [6]. Similar to Section II-D, these results consider generative models with random weights and architectural assumptions such as expansivity. A key advantage of AMP is that its analysis is often powerful enough to attain precise constant factors, unlike typical analyses of gradient descent algorithms.

F. Ongoing Challenges

Compared to explicit priors such as sparsity and low rankness, the study of generative priors remains in its relatively early stages. In this subsection, we overview some of the ongoing challenges and open problems that may be considered in future work.

Generative model properties: While the Lipschitz constant and the depth/width are natural parameters to consider for the generative model, these are "global" properties that may not fully capture the precise structure imposed by typical generative priors. For instance, even if the global Lipschitz constant is huge, it may be that the function is mostly sufficiently smooth to ensure that few measurements suffice. In view of this, it would be of significant interest to identify additional properties that more precisely dictate the required number of measurements.

Structured measurements: Studies of compressive sensing with generative priors have predominantly focused on i.i.d. Gaussian measurement matrices. Non-Gaussian i.i.d. designs have also been considered [68], as well as certain classes of dependent measurements [99]. However, theory is still largely lacking for several kinds of measurements that are used in practice; for instance, in applications such as medical imaging, one is confined to using subsampled Fourier measurements due to the inherent design of the hardware. Very recently, a study of such settings with generative priors was initiated [15], and it was demonstrated that the required number of measurements can be characterized by a coherence parameter measuring the interplay between the range of the generative model and the measurement matrix.

Optimization guarantees with milder assumptions: Regarding the optimization results outlined in Section II-D and the related follow-up works, perhaps the most significant ongoing challenge is to expand the applicability of the theory beyond the case of random generative models, and more generally, to give analogous guarantees with as few restrictive assumptions on G, $\mathbf{A}$, and $\boldsymbol{\eta}$ as possible.

Constant factors: As exemplified in the results that we stated, most existing works on the theory of inverse problems with generative models have typically sought to characterize the scaling laws of the number of measurements, and not the finer question of precise constants. As discussed above, progress has been made in addressing this question using approximate message passing (AMP), but broadly speaking, there remains substantial room for progress in understanding the constant factors associated with bounds on the number of measurements with generative priors.

Role of training data: As we discussed earlier, the consideration of properties such as the Lipschitz constant essentially abstracts away the complicated details of how the generative model was trained. On the other hand, to attain a more complete picture of the entire learning and information processing pipeline, a refined theory might explicitly incorporate such aspects, e.g., explicitly quantifying notions such as representation and generalization, and unifying such considerations with the number of measurements in the inverse problem, the optimization algorithm used for decoding, and so on. While a completely holistic theory may be challenging, future research could potentially take gradual steps towards this.

Out-of-distribution performance: One of the main potential concerns of generative priors is that they may perform poorly under distribution shift, i.e., when the training data is not fully representative of the actual signal being recovered. Various works have started to address this limitation (see Section II-E), but overall, we believe that it remains under-explored relative to its importance.

III. Untrained Neural Network Priors

In this section, we consider untrained neural network priors, which, in contrast to the pre-trained generative priors considered above, work without any training data and solely based on the network architecture and the choice of optimization procedure for fitting the signal/image at inference time. For instance, one of the earliest such techniques called Deep Image Prior (DIP) [134] works by fitting a standard convolutional auto-encoder (the popular U-net [113]) to a single noisy image via gradient descent, and regularizing by early stopping.

Untrained networks have emerged as a highly successful alternative to data-driven methods, yielding excellent performance for a variety of problems, including denoising [57], [134] and compressive sensing [19], [64], [66], [135], [138], [147].

Despite being neural network based, this class of methods is conceptually related to sparsity-based methods, in that it is not data-driven and it relies on broader properties of signals/images (e.g., smoothness) rather than capturing the behavior of highly specific data distributions. Note that data is typically still used for hyperparameter tuning, similarly to sparsity-based methods.

On the other hand, untrained networks can provide significant improvements over sparsity-based methods; for example, they can give better image quality for accelerated magnetic resonance imaging [33]. While the precise reason for this is difficult to pinpoint, it may result from the architectures of untrained networks (e.g., incorporating operations such as convolution) being able to represent typical signals/images more effectively than sparsity-based priors (e.g., dictated by low total-variation norm).

In this section, we first discuss how signal recovery can be performed using untrained neural networks, and then overview the existing theory behind this approach.

A. Background

Consider the problem of reconstructing a signal $\mathbf{x}^* \in \mathbb{R}^n$ from noisy linear measurements, $\mathbf{y} = \mathbf{A}\mathbf{x}^* + \boldsymbol{\eta} \in \mathbb{R}^m$. The signal is often an image, in which case these equations correspond to its vectorized form.

We let $G : \mathbb{R}^p \to \mathbb{R}^n$ represent a neural network with p weights; in contrast to the previous section, here G is a function of the *network weights* $\mathbf{w} \in \mathbb{R}^p$ with a fixed input $\mathbf{z}$ (typically chosen at random and then fixed thereafter). This is the opposite of the previous section, where we treated $\mathbf{w}$ as fixed (pre-learned) and $\mathbf{z}$ as varying. The function $G(\mathbf{w})$ is our untrained neural network, and the fixed input $\mathbf{z}$ is considered part of the network.

The architecture of the network is critical, and is discussed in more detail later. For now, we note that a good choice for images is a simple five-layer convolutional network. We reconstruct an image by applying an optimization procedure (typically gradient descent) starting from a random initialization of the network weights, using the least-squares loss:

$$\mathcal{L}(\mathbf{w}) = \|\mathbf{y} - \mathbf{A}G(\mathbf{w})\|_2^2. \tag{31}$$

This optimization procedure, possibly with early-stopped iterations for regularization, yields the estimate $\hat{\mathbf{w}}$, from which we estimate the unknown signal as $\hat{\mathbf{x}} = G(\hat{\mathbf{w}})$.

This general approach is based on the empirical observation that untrained convolutional networks tend to fit a single natural image significantly faster than pure noise when optimized with gradient descent. However, for the method to work well, a good choice of architecture, optimization procedure, and regularization (e.g., early stopping) can be critical.

Deep image prior: Ulyanov et al. [134] first observed that using a standard convolutional auto-encoder (the popular U-net [113]) as a generator network, and regularizing with early stopping, enables excellent denoising performance. This method has been termed *deep image prior*.

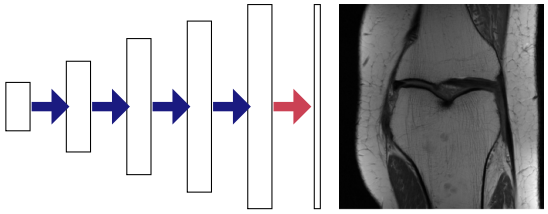

Fig. 3. A rough illustration of the deep decoder, a five-layer untrained convolutional neural network. The network performs 1x1 convolutions (i.e., linear combinations of channels) followed by convolutions with a fixed kernel to map one volume to another. Convolution with a fixed kernel often includes an upsampling operation, as displayed here.

Deep decoder: Many elements of auto-encoders turn out to be largely irrelevant to the strong performance of deep image prior. A more recent paper of Heckel and Hand [57] proposed a much simpler network architecture, termed the deep decoder. This network can be seen as retaining only the most relevant components of a convolutional autoencoder architecture to function as an image prior, and can be obtained from a standard convolutional autoencoder by removing the encoder, the skip connections, and perhaps most notably, the trainable convolutional filters of spatial extent larger than one.

B. Theory

Untrained neural networks enable provable denoising and compressive sensing. Here we discuss the associated recovery guarantees, along with intuition on when and why untrained neural networks enable accurate signal reconstruction.

1) Under-Parametrized Untrained Neural Networks: We say that an untrained neural network is *under-parametrized* if it has fewer parameters p (i.e., the dimension of $\mathbf{w}$) than its output dimension n, and *over-parametrized* otherwise. Untrained networks enable signal reconstruction in both regimes. We start with the under-parametrized regime, since it is conceptually simpler.

The deep decoder [57] is a neural network that transforms a random input volume[8] $\mathbf{B}_0 \in \mathbb{R}^{n_0 \times k_0}$ to an output image by applying convolutions with a fixed convolutional upsampling kernel, followed by weighted linear combinations of the channels, followed by an application of ReLU nonlinearities, and repeating these operations several times, e.g., five times for a five-layer network. See Figure 3 for a visualization. In the simpler case of only two layers, we have $\mathbf{B}_0 \in \mathbb{R}^{\frac{n}{2} \times k_0}$, and the deep decoder network is described as follows:

$$G(\mathbf{w}) = \text{relu}(\mathbf{U}_0 \mathbf{B}_0 \boldsymbol{W}_0)\mathbf{w}_1, \tag{32}$$

where $\mathbf{U}_0 \in \mathbb{R}^{n \times \frac{n}{2}}$ is a linear operator implementing a convolution with a fixed upsampling operator, and $\boldsymbol{W}_0 \in \mathbb{R}^{k_0 \times k_0}$ is a

[8]The input vector that we previously denoted by $\mathbf{z}$ corresponds to the vectorization of $\mathbf{B}_0$. Here it is convenient to work with a matrix-valued input.

parameter matrix forming linear combinations of the channels $\mathbf{U}_0\mathbf{B}_0$. Finally, we apply a ReLU non-linearity and again form linear combinations through multiplication with the parameter vector $\mathbf{w}_1 \in \mathbb{R}^{k_0}$, which yields the output image. The parameters of the network are the weights $\mathbf{w} = (\boldsymbol{W}_0, \mathbf{w}_1)$.

When the number of layers and the number of channels k_0 are not too large, this network is a concise image model, in that it can represent a natural image with much fewer network parameters than pixels. For example, [57, Fig. 1] shows that representing (or compressing) natural images with a deep decoder network that has 30 times fewer parameters than weights gives only a small loss in image quality. Moreover, for a given storage requirement, the image quality typically surpasses that of sparse wavelet representation, which is the basis for the JPEG2000 compression standard.

To summarize, the deep decoder can represent a natural image with very few parameters. At the same time, in the under-parametrized regime, it cannot represent random noise well; informally, an n-dimensional Gaussian noise vector requires roughly p parameters to represent a fraction of $\frac{p}{n}$ of its energy. For the two-layer deep decoder, this is formalized in the following proposition.

Proposition 2 (Lack of Noise Fitting With Under-parametrized Networks [57, Proposition 1]): Consider the two-layer deep decoder (32) with p parameters and arbitrary upsampling and input matrices. Let $\boldsymbol{\eta}$ be zero-mean Gaussian noise with identity covariance matrix. Then, with high probability,

$$\min_{\mathbf{w}} \|G(\mathbf{w}) - \boldsymbol{\eta}\|_2^2 \geq \|\boldsymbol{\eta}\|_2^2 \left(1 - c\frac{p \log n}{n}\right), \tag{33}$$

where c is a numerical constant.

Here and throughout this section, we state most results with the terminology "with high probability" used informally to avoid overly technical statements, but the precise forms can be found in the references given.

Proposition 2 reveals that when fitting an under-parametrized deep decoder to a noisy image (by minimizing the loss in (31)), we expect to fit only a small amount of noise, thus enabling denoising. The number of network parameters, p, trades off how well the network fits the underlying signal (larger is better) and how much noise it fits (smaller is better).

Beyond denoising, similar ideas can be applied to compressive sensing. Specifically, the proof of Proposition 2 establishes that any signal generated by an under-parametrized deep decoder lies in a union of low-dimensional subspaces. Hence, taking measurements with sufficiently many i.i.d. Gaussian measurements guarantees that only one such signal is consistent with the measurements.

2) Over-Parametrized Untrained Neural Networks: A sufficiently over-parametrized convolutional neural network can fit any single image perfectly, including noise. Thus, at first sight, it may seem surprising that over-parametrized untrained networks can enable accurate signal reconstruction. The reason reconstruction is still possible is that, when optimization is performed with gradient descent, the network fits a natural image significantly faster than it fits noise. This is illustrated in Figure 4, where gradient descent is applied to fit a clean image (Fig. 4(b)) and pure noise (Fig. 4(c)), by minimizing the least-squares loss (31) with $\mathbf{A} = \mathbf{I}$. After about 300 iterations the network fits the clean example image, but it requires around 3000 iterations to fit the noise. If we apply gradient descent with a noisy image (Fig. 4(a)), the network first fits the image part of the noisy image and only later the noise part. Thus, early stopping at about 300 iterations denoises the image.

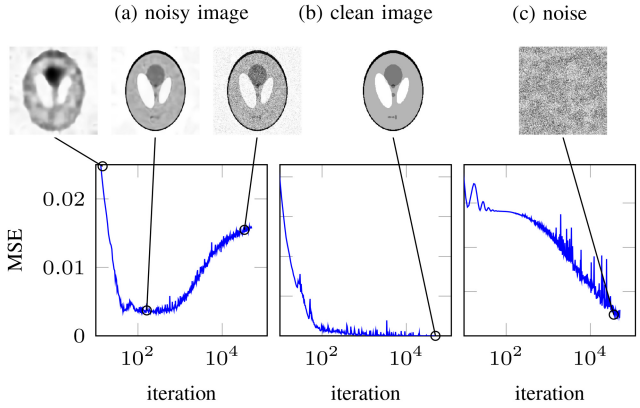

Fig. 4. Fitting an over-parametrized deep decoder network to (a) a noisy image, (b) a clean image, and (c) pure noise. Here, MSE denotes Mean Square Error of the network output with respect to the clean image in (a) and fitted images in (b) and (c). While the network can fit the noise due to over-parameterization, it fits natural images with significantly fewer iterations than noise. Hence, when fitting a noisy image, the image component is fitted faster than the noise component, which enables denoising via early stopping. The curves when using other common convolution networks (e.g., a convolutional generator network of a U-net) are very similar.

For denoising with an over-parametrized untrained network, regularization via early stopping is critical for performance, since with enough iterations the network fits the entire noisy image. The early stopping time plays an analogous role to the number of parameters for image reconstruction with an under-parametrized neural network: More iterations amounts to fitting the image part better, but also fitting more noise.

Empirically, untrained neural networks often perform best in the over-parametrized regime. Several variants of convolutional generator networks work well, including the deep decoder, a U-net, and a convolutional generator network [33], [57], [134].

Provable denoising with over-parametrized convolutional networks: Here we state a theoretical result formalizing the statement that convolutional generators optimized with gradient descent fit natural images faster than noise, and that fitting convolutional generators via early stopped gradient descent provably denoises "natural" images.

As a suitable model for natural images, we consider smooth signals. Specifically, a signal $\mathbf{x} \in \mathbb{R}^n$ is *q-smooth* if it can be represented as a linear combination of the q first trigonometric basis functions (illustrated in Figure 5). As motivation for this definition, [125, Fig. 4] shows that the power spectrum (i.e., the energy distribution by frequency) of a natural image decays rapidly from low frequencies to high frequencies.

We consider a randomly initialized network of the form (32). The result stated below relies on the insight that the behavior of large over-parametrized neural networks is dictated

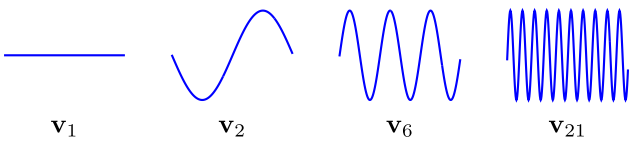

Fig. 5. The 1st, 2nd, 6th, and 21st trigonometric basis functions in dimension $n = 300$.

by the spectral properties of its Jacobian mapping at initialization.[9] The left-singular values of the expected Jacobian of this convolutional network at initialization are the trigonometric basis functions $\mathbf{v}_1, \ldots, \mathbf{v}_n$. Provided that the fixed convolutional filter of the convolution operation $\mathbf{U}_0$ in the deep decoder network is relatively narrow (which it is in practice), the associated singular values $\sigma_1 > \sigma_2 > \cdots > \sigma_n$ decay rapidly, so that large singular values are associated with low-frequency trigonometric basis functions and small singular values are associated with high-frequency basis functions.

The following result shows that for untrained convolutional networks, gradient descent fits the components of the noisy measurement $\mathbf{y} = \mathbf{x}^* + \boldsymbol{\eta}$ that align with the trigonometric basis functions at speeds determined by the associated singular values.

Theorem 7 (Denoising Guarantees With Over-Parametrized Networks [60, Th. 2]): Assume that $\mathbf{x}^*$ is a p-smooth signal, and let $\boldsymbol{\eta}$ be an arbitrary noise vector. Suppose that we fit a randomly initialized network of the form (32) via gradient descent with step size $\alpha \leq \frac{1}{\sigma_1^2}$ for t iterations to minimize the least-squares loss $\mathcal{L}(\mathbf{w}) = \|G(\mathbf{w}) - \mathbf{y}\|_2^2$ with $\mathbf{y} = \mathbf{x}^* + \boldsymbol{\eta}$. Suppose that the network is sufficiently wide, namely, $k_0 \geq \Omega(\frac{n}{\epsilon^4})$, for some $\epsilon > 0$. Then the estimate of the untrained network based on the t-th iterate $\mathbf{w}_t$ obeys, with high probability over the random initialization,

$$\left\|G(\mathbf{w}_t) - \mathbf{x}^*\right\|_2 \leq \left(1 - \alpha\sigma_p^2\right)^t \left\|\mathbf{x}^*\right\|_2 + \left(\sum_{i=1}^{n}\left(\left(1 - \alpha\sigma_i^2\right)^t - 1\right)^2 \boldsymbol{\eta}^T\mathbf{v}_i\right)^{1/2} + \epsilon, \quad (34)$$

where $\{\sigma_i\}_{i=1}^n$ are the singular values described above.

In this result, ϵ is an error term that becomes negligible if the network is sufficiently wide. The first term is the error for fitting the signal, and the second term corresponds to the noise fitted after t iterations. The signal-fitting term decreases to zero in the number of iterations, while the noise-fitting term increases in the number of iterations, up to the noise energy. Thus, there is a trade-off between signal-fitting and noise-fitting. After sufficiently many iterations, $(1 - \alpha\sigma_p^2)^t$ is small, and thus so is the signal fitting error. At the same time, after such a number of iterations, only the components of the noise that align with (roughly) the p-many lowest frequency trigonometric basis functions are fitted, provided that the singular values decay sufficiently fast.

A consequence of this result (see [60, Th. 1]) is that there is an optimal number of iterations such that for denoising a signal corrupted with Gaussian noise $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, the estimate based on early-stopped gradient descent obeys

$$\left\|G(\mathbf{w}_t) - \mathbf{x}^*\right\|_2 \leq O\left(\frac{p}{n}\right).$$

This ensures that only a fraction $\frac{p}{n}$ of the noise energy is fitted, and the rest of the noise that lies outside of the signal subspace spanned by the p-lowest frequency trigonometric basis functions is filtered out. That is, up to a constant factor, one attains optimal performance for denoising a p-smooth signal with Gaussian noise.

Provable compressive sensing with over-parametrized convolutional networks: Here we consider signal reconstruction from $m \ll n$ noiseless random Gaussian measurements with an untrained network, using gradient descent applied to the loss (31). For this setup, perhaps surprisingly, no regularization is necessary (in contrast to the denoising problem discussed above) since the network has an interesting self-regularization property.

The following result is a specialized version of that in [59, Th. 2], which considers general decay patterns of the singular values of the Jacobian. Sufficiently fast decay is needed for accurate reconstruction, and the following result focuses on geometric decay, which is motivated by the fast decay typically observed in practice.

Theorem 8 (Compressive Sensing Guarantees With Over-Parametrized Networks; Corollary of [59, Th. 2]): Let $\mathbf{A} \in \mathbb{R}^{m\times n}$ be an i.i.d. Gaussian random matrix, and suppose that we are given noiseless measurements $\mathbf{y} = \mathbf{A}\mathbf{x}^*$ of an $\frac{m}{3}$-smooth signal $\mathbf{x}^*$. Consider the two-layer neural network (31), with the convolutional kernel (of the convolution operator $\mathbf{U}_0$) chosen so that the singular values of the Jacobian of the network at initialization decay geometrically, i.e., $\sigma_i^2 = \gamma^i$ for some $\gamma \in (0, 1)$. Moreover, suppose the network is sufficiently wide, namely, the number of channels satisfies $k_0 \geq C\frac{m}{\xi^8}$ for some $\xi \in (0, 1)$ and a numerical constant C. Then, with high probability, the estimate $\mathbf{w}_\infty$ obtained by applying gradient descent to the loss (31) until convergence satisfies

$$\left\|G(\mathbf{w}_\infty) - \mathbf{x}^*\right\|_2^2 \leq O\left(\frac{\gamma^{m/3}}{1-\gamma}\|\mathbf{x}\|_2^2\right) + \xi^2\|\mathbf{x}^*\|_2^2. \quad (35)$$

This result guarantees the almost perfect recovery of an $\frac{m}{3}$-smooth signal from only m noiseless measurements, which is optimal up to a constant. Note that the guarantee is non-uniform, i.e., it holds with high probability over $\mathbf{A}$ for fixed $\mathbf{x}^*$. The mechanism underlying this result is that gradient descent fits the lowest-frequency components of the signal before the higher frequency component, similar to the denoising result stated in Theorem 7.

C. Discussion and Ongoing Challenges

Linear approximation and its limitations: The proofs of Theorems 7 and 8 rely on relating the dynamics of gradient descent applied to fitting an over-parametrized network to that of gradient descent of an associated linear network. This proof technique has been used in a variety of recent works [39], [65], [112], [128]. As such, the analysis readily extends to deeper neural networks.

[9]The Jacobian of the function $G : \mathbb{R}^p \to \mathbb{R}^n$ is the matrix $\mathbf{J} \in \mathbb{R}^{n\times p}$ whose (i, j)-th entry is equal to the derivative of the i-th output value with respect to the j-th input value.

However, the main shortcoming of the analysis is that it is constrained to networks operating in a regime where it behaves similar to an associated linear model (implicitly entering via the large-width assumption). This is a reasonable first-order approximation of what untrained networks actually do, but in practice untrained networks typically do not operate in the regime where they behave like associated linear models. What makes untrained networks work so well compared to linear models for denoising and signal reconstruction cannot be captured by this analysis, and therefore, an important avenue for future research is to develop a finer analysis for lower-width untrained networks.

Beyond convolutional networks: In this section, we focused on image reconstruction with untrained convolutional neural networks. We end this section by discussing architectures beyond convolutional networks, and signals beyond images, for which untrained neural networks can still serve as a powerful approach.

For example, coordinate-based neural representations for images, 3D shapes, and other signals have recently emerged as an alternative for traditional discrete representations such as sparse representations or convolutional neural networks. They have been employed for surface reconstruction [140], representing scenes and view synthesis [95], and for representing and working with images. Such networks are untrained neural networks, as they perform reconstruction without any training, in a similar fashion to convolutional networks.

A key component of many of the coordinate-based neural representations are sinusoidal mappings in the first layer [95], [126], [132]. These networks are closely related to convolutional untrained neural networks, since they can be shown to be equivalent to convolutional architectures if sufficiently wide.

Finally, untrained neural networks have also been used to reconstruct graph signals [110], as well as continuously-indexed objects through fitting probabilistic models [114], [150]. We expect that there is significant potential for further theoretical (and practical) developments in these directions.

IV. Unfolding Methods

Recent years have witnessed a surge of interest in algorithm unfolding (also known as unrolling) techniques to tackle various inverse problems arising in signal processing, image processing, and machine learning [96].

Unfolding methods map an iterative solver (algorithm) of an inverse problem onto a recurrent neural network structure. The different iterations of the iterative algorithm correspond to different layers of the neural network structure, with layer parameters corresponding to solver parameters. Instead of fixing the layer parameters, they are optimized in a data-driven manner using learning algorithms, such as empirical risk minimization via stochastic gradient descent, by leveraging a dataset consisting of input-output examples (i.e., training data). Compared to most standard neural network architectures, unfolding methods directly capture domain knowledge according to the iterative algorithm they are based on, and they often contain considerably fewer parameters. Empirically, unfolding methods have achieved state-of-the-art performance in a variety of applications of interest, e.g., being featured prominently in the fastMRI competition.[10]

We will focus our attention on how unfolding techniques apply to the classical sparse recovery problem, in view of the fact that – in addition to its myriad of applications – this is where much of the existing theory-oriented work has arisen. Moreover, sparse recovery is the problem for which algorithm unfolding was originally proposed in the pioneering work of Gregor and LeCun [50]. We refer the reader to recent review articles that overview how unfolding applies to numerous other inverse problems in various fields [96], [120], [121].

A. The Classical ISTA Algorithm

We consider the problem of recovering a sparse vector $\mathbf{x}^*$ given (noisy) linear measurements of the form $\mathbf{y} = \mathbf{A}\mathbf{x}^* + \boldsymbol{\eta}$, as outlined in Section I. A classical iterative algorithm to recover $\mathbf{x}^*$ is iterative shrinkage thresholding algorithm (ISTA) [35]. ISTA is closely connected to the Lasso method, whose optimization problem we repeat here for convenience:

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_1. \tag{36}$$

The ISTA algorithm is an instance of a more general class of techniques called *proximal gradient methods*, which roughly work by performing gradient steps on one term ($\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2$ for Lasso) and applying a so-called *proximal mapping* that encourages the other term to be small ($\lambda\|\mathbf{x}\|_1$ for Lasso).

More specifically, given an initialization $\mathbf{x}_0$, the ISTA algorithm produces the following iterates indexed by t:

$$\mathbf{x}_{t+1} = \Psi_{\lambda/\xi}\left(\mathbf{x}_t + \frac{1}{\xi} \cdot \mathbf{A}^T(\mathbf{y} - \mathbf{A}\mathbf{x}_t)\right), \tag{37}$$

where ξ is an upper bound on the largest eigenvalue of $\mathbf{A}^T\mathbf{A}$, and $\Psi_\theta(\mathbf{z})$ is the soft-thresholding function that is applied on each element of a vector argument as follows: $\Psi_\theta(x) = \mathsf{sign}(x) \cdot \max\{0, |x| - \theta\}$.

The most well-known unfolding method – Learned Iterative Shrinkage-Thresholding Algorithm (LISTA) [50] – leverages this approach to solve the sparse recovery problem using a neural network in a data-driven manner, and is described in the following subsection.

B. The Unfolding Principle: LISTA

The pioneering work of Gregor and LeCun [50] recognized that one can map the iterations of the ISTA algorithm to different layers of a neural network structure. Concretely, by letting $\mathbf{W}_1 = \frac{1}{\xi}\mathbf{A}^T$, $\mathbf{W}_2 = \mathbf{I} - \frac{1}{\xi}\mathbf{A}^T\mathbf{A}$, $\theta = \frac{\lambda}{\xi}$ in (37), we can write the τ iterations of the ISTA algorithm as follows:

$$\mathbf{x}_{t+1} = \Psi_\theta(\mathbf{W}_1\mathbf{y} + \mathbf{W}_2\mathbf{x}_t), \qquad t = 0, 1, \ldots, \tau - 1. \tag{38}$$

This gives rise to a τ-layer recurrent neural network structure where different layers correspond to different iterations of the ISTA algorithm; see Figure 6. The network non-linearity corresponds to the soft-threshold operator in lieu of the standard ReLU.

[10] https://fastmri.org/

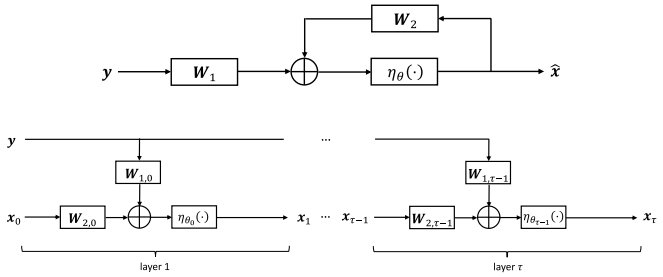

Fig. 6. (Top) Recurrent neural network structure defined using a feedback connection. (Bottom) Unrolled feed-forward neural network structure.

Moreover, by letting $\mathbf{W}_1$, $\mathbf{W}_2$, and θ be iteration-dependent (and accordingly denoted by $\mathbf{W}_{1,t}$, $\mathbf{W}_{2,t}$, and θ_t), we can write the iterations as follows:

$$\mathbf{x}_{t+1} = \Psi_{\theta_t}\big(\mathbf{W}_{1,t}\mathbf{y} + \mathbf{W}_{2,t}\mathbf{x}\big), \qquad t = 0, 1, \ldots, \tau - 1. \quad (39)$$

This gives rise to a τ-layer feed-forward neural network with side connections, with weight matrices $\mathbf{W}_{1,t}$ and $\mathbf{W}_{2,t}$, and non-linearity thresholds θ_t, as illustrated in Figure 6.

In [50], it was proposed to optimize the parameters of the resulting τ-layer feed-forward neural network in a data-driven manner. Specifically, given access to a dataset consisting of various (measurement, target) pairs corresponding to the linear model in (2), i.e., $\mathcal{S} = \{(\mathbf{y}_i', \mathbf{x}_i'), i = 1, \ldots, N\}$ with $\mathbf{y}_i' \in \mathbb{R}^m$ and $\mathbf{x}_i' \in \mathbb{R}^n$, one can consider the following empirical risk minimization problem:

$$\min_{\mathbf{W}_{1,t}, \mathbf{W}_{2,t}, \theta_t} \frac{1}{N} \sum_{i=1}^{N} \big\| \mathbf{x}_i' - \hat{\mathbf{x}}(\mathbf{y}_i') \big\|_2^2 \quad (40)$$

where $\hat{\mathbf{x}}(\mathbf{y}_i')$ is the τ-layer neural network output associated with the network input $\mathbf{y}_i'$.[11] This problem can then be solved using optimization techniques such as stochastic gradient descent. Note that the measurement matrix $\mathbf{A}$ does not need to be known to apply LISTA, though knowing it can be useful for forming other variations (to be described below).

A common variation is to tie the parameters across the various layers of the network, i.e., $\mathbf{W}_{1,t} = \mathbf{W}_1$, $\mathbf{W}_{2,t} = \mathbf{W}_2$, and $\theta_t = \theta$. This is particularly helpful when the training set is small.

The reformulation of τ ISTA iterations onto a τ-layer neural network with parameters that can be further tuned, as described in (38), (39), and (40), is referred to as LISTA. It is often referred to as a *model-based learning method*, because the network architecture is specifically defined according to a particular measurement model (the linear model), optimization procedure (Lasso), and iterative solver (ISTA). This idea can naturally be extended to many other settings, optimization problems, and solvers.

It has been shown empirically that, in comparison with ISTA, LISTA can deliver a more accurate sparse vector with significantly fewer layers/iterations (e.g., see [50]). The success of LISTA has spurred numerous applications of algorithm unrolling over the years, in imaging applications [10], [70], [116], [118], [127] and beyond [74], [109], [149] (see [96, Table I] for a longer list). More recently, significant interest has arisen in the theoretical foundations of unfolding algorithms.

[11] Not to be confused with regression and classification problems in which y often denotes the label to be predicted.

C. Theoretical Foundations of Unfolding Methods

Theoretical studies of unfolding methods broadly fall under the following two categories.

Optimization/Convergence Results: This class of results regards convergence properties, studying whether LISTA-type network architectures can produce an accurate solution faster compared to ISTA under idealized choices of weights.[12] This class of contributions also often demonstrates that one can simplify the classical LISTA approach of [50], e.g., by exploring certain relations/dependencies/couplings between the LISTA learnable parameters. Works giving results of this kind include [24], [25], [81].

Learning-Theoretic Oriented Results: Another class of contributions concentrates on learning-theoretic aspects, studying how the generalization error – corresponding to the difference between the expected error and the empirical error – behaves as a function of various quantities relating to the learning problem, including the number of training samples. Works giving results of this kind include [14], [26], [117], [124].

We proceed by highlighting some key results of both kinds.

1) Optimization/Convergence Results: In [24], Chen et al. showed that the LISTA learnable weight matrices asymptotically admit a partial weight coupling relationship given by

$$\mathbf{W}_{2,t} = \mathbf{I} - \mathbf{W}_{1,t}\mathbf{A}. \quad (41)$$

Accordingly, they simplified the LISTA structure as follows:

$$\mathbf{x}_{t+1} = \Psi_{\theta_t}\big(\mathbf{x}_t + \mathbf{W}_t^T(\mathbf{y} - \mathbf{A}\mathbf{x}_t)\big). \quad (42)$$

Note that this simplification requires knowledge of the matrix $\mathbf{A}$, which is not always known in practice. This simplified version of LISTA – which involves learning only a single weight matrix and threshold per layer – admits the following convergence guarantee.

Theorem 9 (Adapted From [24, Th. 2]): Assume that $\mathbf{x}^* \in \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\|_0 \leq s, \|\mathbf{x}\|_\infty \leq B\}$ and $\|\boldsymbol{\eta}\|_1 \leq \sigma$. Moreover, assume that $\mathbf{A}$ satisfies a coherence condition, and that s is sufficiently small as a function of the associated coherence parameter (see [24, Appendix B] for a formal statement). Then, there exists a sequence of parameters $\{\mathbf{W}_t, \theta_t\}$ such that the sequence of iterates in (42) with $\mathbf{x}_0 = \mathbf{0}$ satisfies

$$\big\| \mathbf{x}_t - \mathbf{x}^* \big\|_2 \leq sB \exp(-ct) + C\sigma, \quad (43)$$

where $c > 0$ and $C > 0$ are scalars that depend only on the linear operator $\mathbf{A}$ and signal sparsity s.

This result shows that a LISTA-like structure can produce a sequence of iterates that is linearly convergent for some sequence of parameters. In contrast, ISTA is generally sublinearly convergent until its iterates settle on a support [13]. Thus,

[12] It should be noted, however, that these results typically impose stronger assumptions on the signal, noise, and measurement matrix compared to classical ISTA theory.

this result provides theoretical evidence that a LISTA-like structure can outperform conventional methods.[13]

It should be noted that the sequences of parameters shown to exist in Theorem 9 do not necessarily correspond to the parameters learned using empirical risk minimization. Thus, results of this kind serve as a *justification for the architecture*, rather than a justification of the training procedure. Proving analogous results for empirical risk minimization would be of significant interest in future work.

Another variation considered in [24] is

$$\mathbf{x}_{t+1} = \Psi^{\text{ss}}_{p_t,\theta_t}\big(\mathbf{x}_t + \mathbf{W}_t^T(\mathbf{y} - \mathbf{A}\mathbf{x}_t)\big), \tag{44}$$

where one replaces the original soft-thresholding operator $\Psi_{\theta_t}(\cdot)$ with a thresholding operator with support selection $\Psi^{\text{ss}}_{p_t,\theta_t}(\cdot)$. This operator retains a proportion p_t of the entries as the "trusted support" at layer t, where p_t is a hyper-parameter that is manually tuned. Specifically, it is proposed to choose p_t proportional to t and capped to a maximal value:

$$p_t = \min\{pt, p_{\max}\}, \tag{45}$$

leaving only p and $p_{\max}$ to be tuned. This LISTA-like architecture with support selection can exhibit a convergence guarantee that is slightly better than that of Theorem 9, as stated in the following.

Theorem 10 (Adapted From [24, Th. 3]): Under the conditions of Theorem 9, there exists a sequence of parameters $\{\mathbf{W}_t, \theta_t\}$ such that the sequence of iterates in (44) with $\mathbf{x}_0 = \mathbf{0}$ and p_t in (45) satisfies

$$\|\mathbf{x}_t - \mathbf{x}\|_2 \leq sB\exp\Big(-\sum_{i=0}^{t-1} c_i'\Big) + C'\sigma, \tag{46}$$

where $c_i' \geq c$ $(\forall i)$ and $C' \leq C$, with (c, C) coming from Theorem 9. Moreover, under an additional assumption that the SNR is not too small [24, Assumption 2], we have the strict inequalities $c_i' > c$ for large enough i, and $C' < C$.

Recent works have also shown that the LISTA structure can be simplified further, without affecting (or even improving) the convergence rates; we proceed by outlining some examples.

Analytic LISTA (ALISTA): In the noiseless setting, it was shown in [81] that the LISTA structure in (44) can be simplified further to

$$\mathbf{x}_{t+1} = \Psi^{\text{ss}}_{p_t,\theta_t}\big(\mathbf{x}_t + \gamma_t\mathbf{W}^T(\mathbf{y} - \mathbf{A}\mathbf{x}_t)\big). \tag{47}$$

The matrix $\mathbf{W}$ – which is fixed across different layers – can be pre-computed by solving a data-free optimization problem (which depends only on $\mathbf{A}$), whereas the layer-wise threshold parameters $\{\theta_t\}_{t=0}^{\tau-1}$ and the layer-wise step-size parameters $\{\gamma_t\}_{t=0}^{\tau-1}$ are optimized using data. The parameters $\{p_t\}_{t=0}^{\tau-1}$ are again chosen according to (45). This scheme, known as analytic LISTA (ALISTA), has considerably fewer parameters to learn/train compared to the scheme in [24] or conventional LISTA [50]. Moreover, this simplified structure retains the linear convergence properties of the structure in [24]. Note that this variation requires knowledge of $\mathbf{A}$.

Hyper-LISTA: In [25], it was proposed to augment the ALISTA structure in (47) with an additional momentum term:

$$\mathbf{x}_{t+1} = \Psi^{\text{ss}}_{p_t,\theta_t}\big(\mathbf{x}_t + \gamma_t\mathbf{W}^T(\mathbf{y} - \mathbf{A}\mathbf{x}_t) + \beta_t(\mathbf{x}_t - \mathbf{x}_{t-1})\big). \tag{48}$$

This structure contains the learnable parameters p_t, θ_t, γ_t, and β_t, with $\mathbf{W}$ being pre-computed by solving a data-free optimization problem (again depending only on $\mathbf{A}$). With this momentum term, it was shown in [25] that the resulting network can exhibit a better linear convergence rate. They also prove that with instance-optimal parameters – where p_t, θ_t, γ_t, and β_t depend on $\mathbf{x}_t$ – the network exhibits super-linear convergence. Importantly, with such instance-optimal parameters, it is shown that the tuning procedure involves learning only three hyper-parameters. This ultra lightweight scheme, known as HyperLISTA, is therefore much simpler than the original LISTA or even ALISTA.

Other works: There have been various other works suggesting how to further improve unfolded ISTA networks and its variants [2], [79], [141]. For example, [2] studies strategies for LISTA that involve learning only step sizes – named Step-LISTA (SLISTA) – and that can outperform standard LISTA. Other earlier works studying the merit of unfolding methods include [97], [144], and cover distinct algorithms such as iterative hard thresholding (IHT).

2) Learning Results: We now overview learning-theoretic oriented results that further illuminate the merits of LISTA networks in comparison to standard neural networks. This class of emerging results expound how the expected (population) error deviates from the empirical error as a function of certain quantities relevant to the learning problem.

Suppose that we have access to a training set $\mathcal{S} = \{(\mathbf{y}_i', \mathbf{x}_i')\}_{i=1}^N$ containing a series of input-output i.i.d. samples of (measurement, target) pairs associated with the model in (2). This training set is used to learn the learnable parameters of the model-based network (e.g., LISTA) using a learning algorithm such as empirical risk minimization.

We define the population error and the empirical error associated with a certain model-based network h taken from a class of model-based networks $\mathcal{H}$ as

$$\mathcal{L}_{\text{P}}(h) = \mathbb{E}\big[\ell(h;(\mathbf{y},\mathbf{x}))\big], \quad \mathcal{L}_{\text{E}}(h) = \frac{1}{N}\sum_{i=1}^{N}\ell\big(h;(\mathbf{y}_i',\mathbf{x}_i')\big), \tag{49}$$

where $\ell(\cdot;\cdot)$ represents a per-sample loss function, taken here to be the ℓ_2-loss between the model-based network output for a given input and the associated ground truth. The generalization error is defined as follows:

$$\text{Gen}(h) = |\mathcal{L}_{\text{P}}(h) - \mathcal{L}_{\text{E}}(h)|, \tag{50}$$

which quantifies how much the expected error deviates from the empirical error for a certain model $h \in \mathcal{H}$.

The behaviour of the generalization error for a certain class of model-based networks is discussed by Behboodi et al. [14].

[13] It has been shown that ISTA can exhibit faster convergence rates provided that one can choose the Lasso regularization parameter λ adaptively over iterations [52], [143]. This idea is actually adopted in LISTA, because the parameters $\{\theta_t\}_{t\geq 1}$ correspond to a path of Lasso parameters $\{\lambda_t\}_{t\geq 1}$.

Their structure differs slightly from the classical LISTA structure and its variations discussed above. Specifically, in view of the fact that [14] assumes that the signal of interest is sparse in some orthogonal dictionary rather than being sparse itself (i.e., they write the vector of interest $\mathbf{x}$ in terms of a sparse vector $\mathbf{z}$ as $\mathbf{x} = \Phi\mathbf{z}$ for some orthogonal dictionary $\Phi \in \mathbb{R}^{n\times n}$), their network structure is composed of a LISTA-like multilayer encoder that converts the measurement vector onto a sparse vector, followed by a linear decoder that converts the sparse vector onto the vector of interest. For technical reasons, the final output may also be further scaled to have a bounded norm. Their network structure is also defined by various weight matrices – akin to LISTA – that depend on the forward operator, the dictionary, and other quantities, but the trainable parameters correspond only to the dictionary entries, and are tied across layers. (i.e., the dictionary parameterizes this class of model-based networks). Their approach therefore also requires knowledge of $\mathbf{A}$.

Theorem 11 (Adapted From [14, Th. 2]): The generalization error associated with the above-described τ-layer model-based network behaves as follows with high probability[14]:

$$\mathrm{Gen}(h) \leq O\left(\sqrt{\frac{mn\log\tau + n^2\log\tau}{N}}\right), \tag{51}$$

provided that $\mathbf{A}$ has a bounded spectral norm and $\|\mathbf{x}^*\|_2$ is bounded (see [14] for the precise conditions).

This result, whose proof relies on a Rademacher complexity analysis, suggests that model-based networks may exhibit better generalization capabilities than traditional neural networks, in line with empirical results [50]. Concretely, this generalization error bound for LISTA-like model-based networks depends on the number of layers only logarithmically, whereas generalization error bounds for traditional neural networks (albeit in classification settings) can scale exponentially in the number of layers [14], [117]. On the other hand, the dependence on m and n in (51) remains fairly strong, and it would be of interest to determine if it can be reduced, e.g., by exploiting sparsity (notice that the sparsity level s is absent in (51)).

Other works: Extensions of the above generalization results are covered in [14], [117], [124], involving different learnable parameters or different degrees of weight-sharing between different layers and a variety of network architectures. Notably, [124] offers a Rademacher complexity and local Rademacher complexity analysis of the generalization error and estimation error of model-based networks, respectively, showing that the soft-thresholding nonlinearity can play a key role in guaranteeing that model-based networks perform better than traditional neural networks. They also show that with a proper choice of parameters, the generalization error bound decays as a function of the number of layers. This result does not hold for standard ReLU networks, demonstrating the power of model-based networks. In [104], further guarantees were given for model-based networks, by deriving a bound on the number of training samples needed to ensure that the training loss decreases to zero as the number of training iterations increases.

[14]For example, with probability 0.99 or any other fixed value in (0, 1) (the precise value only affects the hidden constant in the $O(\cdot)$ notation).

D. Discussion and Ongoing Challenges

We conclude this section by discussing some limitations of the existing theory, and the associated ongoing challenges.

Convergence rates and training: As we already highlighted, results such as Theorems 9 and 10 demonstrate the existence of good weights for a given architecture, but it remains an important open challenge to theoretically determine how effective training procedures are in finding good weights, or whether they have provable limitations. Moreover, previous results in this line of works often impose somewhat restrictive assumptions (e.g., coherence properties of $\mathbf{A}$ and low sparsity) that it would be of interest to relax or remove.

Improved generalization analyses: Overall, the generalization properties of model-based networks deriving from unrolling techniques are still in their infancy. We highlighted some recent results showing that such objects can, in principle, generalize better than classical neural networks. However, a more complete picture may require a significantly better understanding of both model-based networks and standard networks. For instance, in over-parametrized settings, existing theory may suggest overfitting, but in practice the network may still generalize exceptionally well. It is of interest to develop new theoretical machinery that captures the interplay between key elements of the learning problem, including the influence of the optimization procedure. Some initial results exploring the interplay between algorithmic notions (e.g., convergence, stability, and sensitivity) and statistical notions (e.g., generalization) appear in [26] within the context of deep architectures with (unrolled) reasoning layers.

Beyond sparse recovery: We have focused on model-based networks for sparse recovery problems, deriving from a Lasso formulation and an associated ISTA solver. However, one can also derive model-based networks for numerous other inverse problems and information processing tasks. Thus, there remains considerable room for expanding the scope of the existing theory and algorithms, and understanding how model-based networks compare to classical methods or standard neural network architectures.

V. Other Topics on Deep Learning Methods in Inverse Problems

In this section, we briefly highlight some other topics that have been considered regarding deep learning methods in inverse problems (without seeking to be exhaustive), including certain areas where theory is largely or completely lacking.

Plug-and-play methods: While denoising is a seemingly relatively simple inverse problem, powerful strategies have been devised for using denoising as a building block for considerably more general inverse problems, e.g., [94], [111], [136]. As an example, the pioneering work [136] interpreted the iterative ADMM algorithm as alternating between an ℓ_2-regularized recovery problem and a denoising problem, and accordingly proposed to use a generic denoiser for the latter

(e.g., a pre-trained neural network based denoiser). The prior information on the signal is then encoded in the denoiser, and accordingly, this approach was termed *plug-and-play priors*. Related ideas have since been used in AMP algorithms [94] and *regularization by denoising* [111], among others.

A variety of theoretical guarantees, particularly optimization convergence guarantees, have been devised for these methods, e.g., see [23], [82], [92], [92], [108], [115], [129], [133] and the references therein. In particular, we highlight the recent work [82], which adopted a restricted eigenvalue condition (REC) analogous to the one used to prove Theorems 1 and 2. Specifically, the REC is defined with respect to the range of the denoiser (rather than the range of a generative model), and it is shown that this leads to accurate estimation of the underlying signal under suitable boundedness and Lipschitz assumptions on the residual function induced by the denoiser.

Instabilities in deep learning methods: In the machine learning literature, it is widely understood that neural networks for classification (and other tasks) can be highly sensitive to adversarial perturbations in the input [130]. A detailed theoretical and empirical study was recently given around analogous instability issues in inverse problems [49] (following a related empirical study in [4]); we proceed by highlighting the over-arching idea in this work.

The focus in [49] is on deep learning for the decoder, i.e., training a neural network to map $\mathbf{y} = \mathbf{A}\mathbf{x}$ to $\mathbf{x}$ (or similarly with noise). Suppose that such a network learns an accurate mapping for two signals $\mathbf{x}, \mathbf{x}'$ with outputs $\mathbf{y}, \mathbf{y}'$, and that $\mathbf{A}(\mathbf{x}-\mathbf{x}')$ is small compared to $\mathbf{x}-\mathbf{x}'$ itself (i.e., $\mathbf{x}-\mathbf{x}'$ is close to the *nullspace* or *kernel* of $\mathbf{A}$). This means that we have two (relatively) nearby $\mathbf{y}, \mathbf{y}'$ being mapped to two distant $\mathbf{x}, \mathbf{x}'$. Then, the network becomes *unstable* in the sense that the output is significantly different for two nearby inputs, resulting in sensitivity to adversarial noise. Perhaps more surprisingly, it is shown in [49] that even sensitivity to well-behaved *random* noise (e.g., Gaussian) can arise from this phenomenon, both in theory and practice.

In some cases, these difficulties could be circumvented by considering a sufficiently well-behaved measurement matrix (e.g., i.i.d. Gaussian). However, when one does not have the luxury of being able to design the measurements, the results of [49] point to the idea that learning methods should be *kernel-aware* in the sense of avoiding the above behavior for pairs of signals whose difference is close to the kernel of $\mathbf{A}$. Further details and discussions can be found in [49], and additional results regarding accuracy and stability can be found in [30].

Training, generalization, and out-of-distribution performance: As we highlighted in Sections II and IV, theoretical studies of data-driven deep learning methods for inverse problems still largely lack a good understanding of the precise role of training data, including the fundamental notion of generalization. Beyond the works on unfolding methods highlighted in Section IV, an example work on the generalization error in inverse problems is [3], with the generalization bounds depending on (i) a complexity measure of the signal space, and (ii) norms of the Jacobian matrices of both the network itself and the network composed with the forward model.

Moreover, even provably small generalization error on i.i.d. data may be insufficient in practical scenarios, where one often requires robustness to out-of-distribution samples. The above-mentioned works on instabilities [30], [49] study an important special case of such issues, and another example is mitigating representation error in the case of generative priors [32], [37], which we discussed in Section II-E.

Another limitation of the learning-based works that we have surveyed is that they are often based on the availability of "clean" training data, e.g., for learning a generative prior or tuning a neural network based decoder. To address this, various works have explored methods for training with only samples that are noisy (e.g., $\mathbf{x}+\mathbf{z}$ instead of $\mathbf{x}$) or compressed (e.g., $\mathbf{A}\mathbf{x}$ instead of $\mathbf{x}$) [18], [77], [131].

Overall, despite this initial progress, we believe that much more remains to be done around training and generalization, and that these issues will play a crucial role in future studies of data-driven methods.

Measurement matrix design: Beyond signal modeling and decoding techniques, deep learning methods have been proposed for designing the measurement matrix $\mathbf{A}$ in compressive sensing [98], [142]. However, these works have largely focused on algorithm design and empirical evaluation, rather than theory. While theoretical analyses for certain learning-based measurement designs do exist (e.g., [9]) with the possibility of specializing to scenarios involving neural networks, the theory of deep learning based measurement design currently appears to remain largely open.

Other decoding techniques: As we outlined in Section IV, there exist a variety of theoretical results for unfolding algorithms of interest. However, unfolding methods are just one of many classes of deep learning based decoders [101], varying according to the architecture, the degree of prior knowledge of $\mathbf{A}$, and so on. Accordingly, there remains considerable room for expanding the scope of theoretical studies in this domain.

Specialized inverse problems: Theoretical guarantees for deep learning based inverse problems have largely focused on the important special cases of denoising and compressive sensing, or problems closely related to these. Further theoretical studies on other specialized inverse problems (e.g., inpainting, super-resolution, etc.) could provide significant benefit to this continually developing research area. We also highlight the topic of *deep learning for coding and communication* [46], [75], [122], [123], which similarly poses a variety of specialized inverse problems whose study has largely relied on empirical evaluation.

VI. Conclusion

While studies of deep learning methods are typically driven by their excellent practical performance, they also pose a variety of unique and exciting theoretical questions. We have surveyed several prominent examples of the theory behind deep learning methods for inverse problems, and outlined a variety of ongoing challenges and open problems. Overall,

despite the rapid growth of this line of works, we believe that the topic remains in its early stages, with many of the most exciting developments still to come.

REFERENCES

[1] A. Aberdam, D. Simon, and M. Elad, "When and how can deep generative models be inverted?" 2020, *arXiv:2006.15555*.

[2] P. Ablin, T. Moreau, M. Massias, and A. Gramfort, "Learning step sizes for unfolded sparse coding," in *Proc. Conf. Neural Inf. Process. Syst.*, 2019, pp. 1–22.

[3] J. Amjad, Z. Lyu, and M. R. D. Rodrigues, "Deep learning model-aware regulatization with applications to inverse problems," *IEEE Trans. Signal Process.*, vol. 69, pp. 6371–6385, Nov. 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9609667

[4] V. Antun, F. Renna, C. Poon, B. Adcock, and A. C. Hansen, "On instabilities of deep learning in image reconstruction and the potential costs of AI," *Proc. Nat. Acad. Sci.*, vol. 117, no. 48, pp. 30088–30095, 2020.

[5] M. Asim, M. Daniels, O. Leong, A. Ahmed, and P. Hand, "Invertible generative models for inverse problems: Mitigating representation error and dataset bias," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 399–409.

[6] B. Aubin, B. Loureiro, A. Baker, F. Krzakala, and L. Zdeborová, "Precise asymptotics for phase retrieval and compressed sensing with random generative priors," in *Proc. NeurIPS Workshop Deep Learn. Inverse Problems*, 2019, pp. 1–6.

[7] B. Aubin, B. Loureiro, A. Maillard, F. Krzakala, and L. Zdeborová, "The spiked matrix model with generative priors," in *Proc. Conf. Neural Inf. Process. Syst.*, 2019, pp. 8366–8377.

[8] K. D. Ba, P. Indyk, E. Price, and D. P. Woodruff, "Lower bounds for sparse recovery," in *Proc. ACM-SIAM Symp. Discrete Algorithms*, 2010, pp. 1190–1197.

[9] L. Baldassarre, Y.-H. Li, J. Scarlett, B. Gözcü, I. Bogunovic, and V. Cevher, "Learning-based compressive subsampling," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 4, pp. 809–822, Jun. 2016.

[10] O. Bar-Shira et al., "Learned super resolution ultrasound for improved breast lesion characterization," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Interv.*, 2021, pp. 109–118.

[11] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, "Model-based compressive sensing," *IEEE Trans. Inf. Theory*, vol. 56, no. 4, pp. 1982–2001, Apr. 2010.

[12] R. G. Baraniuk and M. B. Wakin, "Random projections of smooth manifolds," *Found. Comput. Math.*, vol. 9, no. 1, pp. 51–77, 2009.

[13] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, 2009.

[14] A. Behboodi, H. Rauhut, and E. Schnoor, "Compressive sensing and neural networks from a statistical learning perspective," 2020, *arXiv:2010.15658*.

[15] A. Berk, S. Brugiapagli, B. Joshi, Y. Plan, M. Scott, and Ö. Yilmaz, "A coherence parameter characterizing generative compressed sensing with Fourier measurements," *IEEE J. Sel. Areas Inf. Theory*, early access, Nov. 8, 2022, doi: 10.1109/JSAIT.2022.3220196.

[16] P. J. Bickel, Y. Ritov, and A. B. Tsybakov, "Simultaneous analysis of Lasso and Dantzig selector," *Ann. Stat.*, vol. 37, no. 4, pp. 1705–1732, 2009.

[17] A. Bora, A. Jalal, E. Price, and A. G. Dimakis, "Compressed sensing using generative models," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 537–546.

[18] A. Bora, E. Price, and A. G. Dimakis, "AmbientGAN: Generative models from lossy measurements," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–22.

[19] E. Bostan, R. Heckel, M. Chen, M. Kellman, and L. Waller, "Deep phase decoder: Self-calibrating phase microscopy with an untrained deep neural network," *Optica*, vol. 7, no. 6, pp. 559–562, 2020.

[20] E. J. Candes and T. Tao, "Decoding by linear programming," *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.

[21] E. J. Candès and M. A. Davenport, "How well can we estimate a sparse vector?" *Appl. Comput. Harmonic Anal.*, vol. 34, no. 2, pp. 317–323, 2013.

[22] E. J. Candès, J. K. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Commun. Pure Appl. Math.*, vol. 59, no. 8, pp. 1207–1223, 2006.

[23] S. H. Chan, X. Wang, and O. A. Elgendy, "Plug-and-play ADMM for image restoration: Fixed-point convergence and applications," *IEEE Trans. Comput. Imag.*, vol. 3, no. 1, pp. 84–98, Mar. 2017.

[24] X. Chen, J. Liu, Z. Wang, and W. Yin, "Theoretical linear convergence of unfolded ISTA and its practical weights and thresholds," in *Proc. Conf. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 9079–9089.

[25] X. Chen, J. Liu, Z. Wang, and W. Yin, "Hyperparameter tuning is all you need for LISTA," in *Proc. Conf. Neural Inf. Process. Syst.*, 2021, pp. 11678–11689.

[26] X. Chen, Y. Zhang, C. Reisinger, and L. Song, "Understanding deep architectures with reasoning layer," in *Proc. Conf. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 1240–1252.

[27] C. Clason, "Nonsmooth analysis and optimization," 2017, *arXiv:1708.04180*.

[28] J. Cocola, "Signal recovery with non-expansive generative network priors," 2022, *arXiv:2204.13599*.

[29] A. Cohen, W. Dahmen, and R. DeVore, "Compressed sensing and best k-term approximation," *J. Amer. Math. Soc.*, vol. 22, no. 1, pp. 211–231, 2009.

[30] M. J. Colbrook, V. Antun, and A. C. Hansen, "The difficulty of computing stable and accurate neural networks: On the barriers of deep learning and Smale's 18th problem," *Proc. Nat. Acad. Sci.*, vol. 119, no. 12, 2022, Art. no. e2107151119.

[31] G. Daras, Y. Dagan, A. Dimakis, and C. Daskalakis, "Score-guided intermediate level optimization: Fast Langevin mixing for inverse problems," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 4722–4753.

[32] G. Daras, J. Dean, A. Jalal, and A. Dimakis, "Intermediate layer optimization for inverse problems using deep generative models," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 2421–2432.

[33] M. Z. Darestani and R. Heckel, "Accelerated MRI with un-trained neural networks," *IEEE Trans. Comput. Imag.*, vol. 7, pp. 724–733, Jul. 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9488215

[34] C. Daskalakis, D. Rohatgi, and E. Zampetakis, "Constant-expansion suffices for compressed sensing with generative priors," in *Proc. Conf. Neural Inf. Process. Syst.*, 2020, pp. 13917–13926.

[35] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Commun. Pure Appl. Math.*, vol. 57, no. 11, pp. 1413–1457, 2004.

[36] M. Davenport, M. Duarte, Y. C. Eldar, and G. Kutyniok, "Introduction to compressed sensing," in *Compressed Sensing: Theory and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2011.

[37] M. Dhar, A. Grover, and S. Ermon, "Modeling sparse deviations for compressed sensing using generative models," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1222–1231.

[38] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, Feb. 2016.

[39] S. S. Du, X. Zhai, B. Poczós, and A. Singh, "Gradient descent provably optimizes over-parameterized neural networks," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–19.

[40] Y. C. Eldar and M. Mishali, "Robust recovery of signals from a structured union of subspaces," *IEEE Trans. Inf. Theory*, vol. 55, no. 11, pp. 5302–5316, Nov. 2009.

[41] Y. C. Eldar, *Sampling Theory: Beyond Bandlimited Systems*. Cambridge, U.K.: Cambridge Univ. Press, 2015.

[42] O. Y. Feng, R. Venkataramanan, C. Rush, and R. J. Samworth, "A unifying tutorial on approximate message passing," *Found. Trends Mach. Learn.*, vol. 15, no. 4, pp. 335–536, 2022.

[43] A. K. Fletcher, S. Rangan, and P. Schniter, "Inference in deep networks in high dimensions," in *Proc. IEEE Int. Symp. Inf. Theory*, 2018, pp. 1884–1888.

[44] S. Foucart and H. Rauhut, *A Mathematical Introduction to Compressive Sensing*. New York, NY, USA: Birkhäuser, 2013.

[45] A. C. Gilbert, H. Q. Ngo, E. Porat, A. Rudra, and M. J. Strauss, "ℓ_2/ℓ_2-foreach sparse recovery with low risk," in *Proc. Int. Colloq. Automata Lang. Program.*, 2013, pp. 461–472.

[46] D. Gündüz, P. de Kerret, N. D. Sidiropoulos, D. Gesbert, C. R. Murthy, and M. van der Schaar, "Machine learning in the air," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2184–2199, Oct. 2019.

[47] F. L. Gómez, A. Eftekhari, and V. Cevher, "Fast and provable ADMM for learning with generative priors," in *Proc. Conf. Neural Inf. Process. Syst.*, 2019, pp. 12004–12016.

[48] I. Goodfellow et al., "Generative adversarial nets," in *Proc. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[49] N. M. Gottschling, V. Antun, B. Adcock, and A. C. Hansen, "The troublesome kernel—On hallucinations, no free lunches and the accuracy-stability trade-off in inverse problems," 2020, *arXiv:2001.01258*.

[50] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 399–406.

[51] S. Gunn, J. Cocola, and P. Hand, "Regularized training of intermediate layers for generative models for inverse problems," 2022, *arXiv:2203.04382*.

[52] E. T. Hale, W. Yin, and Y. Zhang, "Fixed-point continuation for ℓ_1-minimization: Methodology and convergence," *SIAM J. Optim.*, vol. 19, no. 3, pp. 1107–1130, 2008.

[53] P. Hand, O. Leong, and V. Voroninski, "Phase retrieval under a generative prior," in *Proc. Conf. Neural Inf. Process. Syst.*, 2018, pp. 9154–9164.

[54] P. Hand, O. Leong, and V. Voroninski, "Optimal sample complexity of subgradient descent for amplitude flow via non-Lipschitz matrix concentration," *Commun. Math. Sci.*, vol. 19, no. 7, pp. 2035–2047, 2021.

[55] P. Hand and V. Voroninski, "Global guarantees for enforcing deep generative priors by empirical risk," in *Proc. Conf. Learn. Theory*, 2018, pp. 970–978.

[56] P. Hand and V. Voroninski, "Global guarantees for enforcing deep generative priors by empirical risk," *IEEE Trans. Inf. Theory*, vol. 66, no. 1, pp. 401–418, Jan. 2020.

[57] R. Heckel and P. Hand, "Deep decoder: Concise image representations from untrained non-convolutional networks," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–14.

[58] R. Heckel, W. Huang, P. Hand, and V. Voroninski, "Rate-optimal denoising with deep neural networks," *Inf. Inf. J. IMA*, vol. 10, no. 4, pp. 1251–1285, 2021.

[59] R. Heckel and M. Soltanolkotabi, "Compressive sensing with untrained neural networks: Gradient descent finds the smoothest approximation," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1–10.

[60] R. Heckel and M. Soltanolkotabi, "Denoising and regularization via exploiting the structural bias of convolutional generators," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–31.

[61] C. Hegde and R. G. Baraniuk, "Signal recovery on incoherent manifolds," *IEEE Trans. Inf. Theory*, vol. 58, no. 12, pp. 7204–7214, Dec. 2012.

[62] W. Huang, P. Hand, R. Heckel, and V. Voroninski, "A provably convergent scheme for compressive sensing under random generative priors," *J. Fourier Anal. Appl.*, vol. 27, no. 2, pp. 1–34, 2021.

[63] S. A. Hussein, T. Tirer, and R. Giryes, "Image-adaptive GAN based reconstruction," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, 2020, pp. 3121–3129.

[64] R. Hyder and M. S. Asif, "Generative models for low-dimensional video representation and reconstruction," *IEEE Trans. Signal Process.*, vol. 68, pp. 1688–1701, Feb. 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9018176

[65] A. Jacot, F. Gabriel, and C. Hongler, "Neural tangent kernel: Convergence and generalization in neural networks," in *Proc. Conf. Neural Inf. Process. Syst.*, 2018, pp. 8580–8589.

[66] G. Jagatap and C. Hegde, "Algorithmic guarantees for inverse imaging with untrained network priors," in *Proc. Conf. Neural Inf. Process. Syst.*, 2019, pp. 14832–14842.

[67] A. Jalal, S. Karmalkar, A. Dimakis, and E. Price, "Instance-optimal compressed sensing via posterior sampling," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 4709–4720.

[68] A. Jalal, L. Liu, A. G. Dimakis, and C. Caramanis, "Robust compressed sensing using generative models," in *Proc. Conf. Neural Inf. Process. Syst.*, 2020, pp. 1–15.

[69] Y. Jiao, D. Li, M. Liu, X. Lu, and Y. Yang, "Just least squares: Binary compressive sampling with low generative intrinsic dimension," 2021, *arXiv:2111.14486*.

[70] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser, "Deep convolutional neural network for inverse problems in imaging," *IEEE Trans. Image Process.*, vol. 26, pp. 4509–4522, 2017.

[71] W. B. Johnson and J. Lindenstrauss, "Extensions of Lipschitz mappings into a Hilbert space," *Contemp. Math.*, vol. 26, p. 28, Jan. 1984.

[72] B. Joshi, X. Li, Y. Plan, and Ö. Yilmaz, "PLUGIn: A simple algorithm for inverting generative models with recovery guarantees," in *Proc. Conf. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 24719–24729.

[73] A. Kamath, E. Price, and S. Karmalkar, "On the power of compressed sensing with generative models," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 5101–5109.

[74] S. Khobahi, N. Shlezinger, M. Soltanalian, and Y. C. Eldar, "LoRD-Net: Unfolded deep detection network with low-resolution receivers," *IEEE Trans. Signal Process.*, vol. 69, pp. 5651–5664, Oct. 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9557819

[75] H. Kim, S. Oh, and P. Viswanath, "Physical layer communication via deep learning," *IEEE J. Sel. Areas Inf. Theory*, vol. 1, no. 1, pp. 5–18, May 2020.

[76] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*.

[77] J. Lehtinen et al., "Noise2Noise: Learning image restoration without clean data," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 2971–2980.

[78] Q. Lei, A. Jalal, I. S. Dhillon, and A. G. Dimakis, "Inverting deep generative models, one layer at a time," in *Proc. Conf. Neural Inf. Process. Syst.*, 2019, pp. 13910–13919.

[79] Z. Li, K. Wu, Y. Guo, and C. Zhang, "Learned ISTA with error-based thresholding for adaptive sparse coding," 2021, *arXiv:2112.10985*.

[80] Z. C. Lipton and S. Tripathi, "Precise recovery of latent vectors from generative adversarial networks," 2017, *arXiv:1702.04782*.

[81] J. Liu, X. Chen, Z. Wang, and W. Yin, "ALISTA: Analytic weights are as good as learned weights in LISTA," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–33.

[82] J. Liu, M. S. Asif, B. Wohlberg, and U. Kamilov, "Recovery analysis for plug-and-play priors using the restricted eigenvalue condition," in *Proc. Conf. Neural Inf. Process. Syst.*, 2021, pp. 5921–5933.

[83] J. Liu and Z. Liu, "Non-iterative recovery from nonlinear observations using generative models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 233–243.

[84] Z. Liu, S. Ghosh, and J. Scarlett, "Towards sample-optimal compressive phase retrieval with sparse and generative priors," in *Proc. Conf. Neural Inf. Process. Syst.*, 2021, pp. 17656–17668.

[85] Z. Liu, S. Gomes, A. Tiwari, and J. Scarlett, "Sample complexity bounds for 1-bit compressive sensing and binary stable embeddings with generative priors," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 6216–6225.

[86] Z. Liu and J. Han, "Projected gradient descent algorithms for solving nonlinear inverse problems with generative priors," in *Proc. Int. Joint Conf. Artif. Intell.*, 2022, pp. 3271–3277.

[87] Z. Liu, J. Liu, S. Ghosh, J. Han, and J. Scarlett, "Generative principal component analysis," in *Proc. Int. Conf. Learn. Represent.*, 2022, pp. 1–25.

[88] Z. Liu and J. Scarlett, "The generalized Lasso with nonlinear observations and generative priors," in *Proc. Conf. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 19125–19136.

[89] Z. Liu and J. Scarlett, "Information-theoretic lower bounds for compressive sensing with generative models," *IEEE J. Sel. Areas Inf. Theory*, vol. 1, no. 1, pp. 292–303, May 2020.

[90] F. Ma, U. Ayaz, and S. Karaman, "Invertibility of convolutional generative networks from partial measurements," in *Proc. Conf. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 9651–9660.

[91] M. T. McCann, K. H. Jin, and M. Unser, "Convolutional neural networks for inverse problems in imaging: A review," *IEEE Signal Process. Mag.*, vol. 34, no. 6, pp. 85–95, Nov. 2017.

[92] T. Meinhardt, M. Moller, C. Hazirbas, and D. Cremers, "Learning proximal operators: Using denoising networks for regularizing inverse imaging problems," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 1781–1790.

[93] C. A. Metzler, A. Mousavi, and R. G. Baraniuk, "Learned D-AMP: Principled neural network based compressive image recovery," in *Proc. Conf. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1770–1781.

[94] C. A. Metzler, A. Maleki, and R. G. Baraniuk, "From denoising to compressed sensing," *IEEE Trans. Inf. Theory*, vol. 62, no. 9, pp. 5117–5144, Sep. 2016.

[95] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 405–421.

[96] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," *IEEE Signal Process. Mag.*, vol. 38, no. 2, pp. 18–44, Mar. 2021.

[97] T. Moreau and J. Bruna, "Understanding trainable sparse coding with matrix factorization," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–13.

[98] S. Mulleti, H. Zhang, and Y. C. Eldar, "Learning to sample: Data-driven sampling and reconstruction of FRI signals," 2021, *arXiv: 2106.14500*.

[99] A. Naderi and Y. Plan, "Sparsity-free compressed sensing with applications to generative priors," *IEEE J. Sel. Areas Inf. Theory*, early access, Nov. 9, 2022, doi: 10.1109/JSAIT.2022.3219807.

[100] T. V. Nguyen, G. Jagatap, and C. Hegde, "Provable compressed sensing with generative priors via Langevin dynamics," 2021, *arXiv:2102.12643*.

[101] G. Ongie, A. Jalal, C. A. Metzler, R. G. Baraniuk, A. G. Dimakis, and R. Willett, "Deep learning techniques for inverse problems in imaging," *IEEE J. Sel. Areas Inf. Theory*, vol. 1, no. 1, pp. 39–56, May 2020.

[102] P. Peng, S. Jalali, and X. Yuan, "Solving inverse problems via auto-encoders," *IEEE J. Sel. Areas Inf. Theory*, vol. 1, no. 1, pp. 312–323, May 2020.

[103] E. Price and D. P. Woodruff, "$(1+\epsilon)$-approximate sparse recovery," in *Proc. IEEE Symp. Found. Comput. Sci.*, 2011, pp. 295–304.

[104] W. Pu, Y. C. Eldar, and M. R. D. Rodrigues, "Optimization guarantees for ISTA and ADMM based unfolded networks," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2022, pp. 8687–8691.

[105] S. Qiu, X. Wei, and Z. Yang, "Robust one-bit recovery via ReLU generative networks: Near-optimal statistical rate and global landscape analysis," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 7857–7866.

[106] M. Raginsky, A. Rakhlin, and M. Telgarsky, "Non-convex learning via stochastic gradient Langevin dynamics: A nonasymptotic analysis," in *Proc. Conf. Learn. Theory*, 2017, pp. 1674–1703.

[107] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM Rev.*, vol. 52, no. 3, pp. 471–501, 2010.

[108] E. T. Reehorst and P. Schniter, "Regularization by denoising: Clarifications and new interpretations," *IEEE Trans. Comput. Imag.*, vol. 5, no. 1, pp. 52–67, Mar. 2019.

[109] G. Revach, N. Shlezinger, X. Ni, A. L. Escoriza, R. J. G. van Sloun, and Y. C. Eldar, "KalmanNet: Neural network aided Kalman filtering for partially known dynamics," *IEEE Trans. Signal Process.*, vol. 70, pp. 1532–1547, Jan. 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9733186

[110] S. Rey, S. Segarra, R. Heckel, and A. G. Marques, "Untrained graph neural networks for denoising," 2021, *arXiv:2109.11700*.

[111] Y. Romano, M. Elad, and P. Milanfar, "The little engine that could: Regularization by denoising (RED)," *SIAM J. Imag. Sci.*, vol. 10, no. 4, pp. 1804–1844, 2017.

[112] B. Ronen, D. W. Jacobs, Y. Kasten, and S. Kritchman, "The convergence rate of neural networks for learned functions of different frequencies," in *Proc. Conf. Neural Inf. Process. Syst.*, 2019, pp. 4763–4772.

[113] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention*. Cham, Switzerland: Springer, 2015, pp. 234–241.

[114] D. Rosenbaum et al., "Inferring a continuous distribution of atom coordinates from cryo-EM images using VAEs," 2021, *arXiv:2106.14108*.

[115] E. Ryu, J. Liu, S. Wang, X. Chen, Z. Wang, and W. Yin, "Plug-and-play methods provably converge with properly trained denoisers," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 5546–5557.

[116] Y. B. Sahel, J. P. Bryan, B. Cleary, S. L. Farhi, and Y. C. Eldar, "Deep unrolled recovery in sparse biological imaging," *IEEE Signal Process. Mag.*, vol. 39, no. 2, pp. 45–57, Mar. 2022.

[117] E. Schnoor, A. Behboodi, and H. Rauhut, "Generalization error bounds for iterative recovery algorithms unfolded as neural networks," 2021, *arXiv:2112.04364*.

[118] C. J. Schuler, M. Hirsch, S. Harmeling, and B. Schölkopf, "Learning to deblur," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 7, pp. 1439–1451, Jul. 2016.

[119] V. Shah and C. Hegde, "Solving linear inverse problems using GAN priors: An algorithm with provable guarantees," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2018, pp. 4609–4613.

[120] N. Shlezinger, J. Whang, Y. C. Eldar, and A. G. Dimakis, "Model-based deep learning," 2021, *arXiv:2012.08405*.

[121] N. Shlezinger, Y. C. Eldar, and S. P. Boyd, "Model-based deep learning: On the intersection of deep learning and optimization," 2022, *arXiv:2205.02640*.

[122] N. Shlezinger, N. Farsad, Y. C. Eldar, and A. J. Goldsmith, "ViterbiNet: A deep learning based Viterbi algorithm for symbol detection," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3319–3331, May 2020.

[123] N. Shlezinger, N. Farsad, Y. C. Eldar, and A. J. Goldsmith, "Model-based machine learning for communications," 2021, *arXiv:2101.04726*.

[124] A. Shultzman, E. Azar, M. R. D. Rodrigues, and Y. C. Eldar, "Generalization and estimation error bounds for model-based neural networks," in *Proc. Int. Conf. Learn. Represent.*, 2023, pp. 1–11.

[125] E. P. Simoncelli and B. A. Olshausen, "Natural image statistics and neural representation," *Annu. Rev. Neurosci.*, vol. 24, no. 1, pp. 1193–1216, 2001.

[126] V. Sitzmann, J. N. P. Martel, A. W. Bergman, D. B. Lindell, and G. Wetzstein, "Implicit neural representations with periodic activation functions," in *Proc. Conf. Neural Inf. Process. Syst.*, 2020, pp. 7462–7473.

[127] O. Solomon et al., "Deep unfolded robust PCA with application to clutter suppression in ultrasound," *IEEE Trans. Med. Imag.*, vol. 39, no. 4, pp. 1051–1063, Apr. 2020.

[128] M. Soltanolkotabi, A. Javanmard, and J. D. Lee, "Theoretical insights into the optimization landscape of over-parameterized shallow neural networks," *IEEE Trans. Inf. Theory*, vol. 65, no. 2, pp. 742–769, Feb. 2019.

[129] Y. Sun, Z. Wu, X. Xu, B. Wohlberg, and U. S. Kamilov, "Scalable plug-and-play ADMM with convergence guarantees," *IEEE Trans. Comput. Imag.*, vol. 7, pp. 849–863, Jul. 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9473005

[130] C. Szegedy et al., "Intriguing properties of neural networks," in *Proc. Int. Conf. Learn. Represent.*, 2014, pp. 1–10.

[131] J. Tachella, D. Chen, and M. Davies, "Unsupervised learning from incomplete measurements for inverse problems," in *Proc. Conf. Neural Inf. Process. Syst.*, 2022, pp. 1–13.

[132] M. Tancik et al., "Fourier features let networks learn high frequency functions in low dimensional domains," in *Proc. Conf. Neural Inf. Process. Syst.*, 2020, pp. 1–11.

[133] T. Tirer and R. Giryes, "Image restoration by iterative denoising and backward projections," *IEEE Trans. Image Process.*, vol. 28, pp. 1220–1234, 2019.

[134] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 9446–9454.

[135] D. Van Veen, A. Jalal, M. Soltanolkotabi, E. Price, S. Vishwanath, and A. G. Dimakis, "Compressed sensing with deep image prior and learned regularization," 2018, *arXiv:1806.06438*.

[136] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, "Plug-and-play priors for model based reconstruction," in *Proc. IEEE Global Conf. Signal Inf. Process.*, 2013, pp. 945–948.

[137] R. Vershynin, "Introduction to the non-asymptotic analysis of random matrices," in *Compressed Sensing: Theory and Applications*, Y. Eldar and G. Kutyniok, Eds. Cambridge, U.K.: Cambridge Univ. Press, 2012.

[138] F. Wang et al., "Phase imaging with an untrained neural network," *Light Sci. Appl.*, vol. 9, no. 1, pp. 1–7, 2020.

[139] X. Wei, Z. Yang, and Z. Wang, "On the statistical rate of nonlinear recovery in generative models with heavy-tailed data," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6697–6706.

[140] F. Williams, T. Schneider, C. Silva, D. Zorin, J. Bruna, and D. Panozzo, "Deep geometric prior for surface reconstruction," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10122–10131.

[141] K. Wu, Y. Guo, Z. Li, and C. Zhang, "Sparse coding with gated learned ISTA," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–27.

[142] S. Wu et al., "Learning a compressed sensing measurement matrix via gradient unrolling," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6828–6839.

[143] L. Xiao and T. Zhang, "A proximal-gradient homotopy method for the sparse least-squares problem," *SIAM J. Optim.*, vol. 23, no. 2, pp. 1062–1091, 2013.

[144] B. Xin, Y. Wang, W. Gao, D. Wipf, and B. Wang, "Maximal sparsity with deep networks?" in *Proc. Conf. Neural Inf. Process. Syst.*, 2016, pp. 4340–4348.

[145] R. Yeh, C. Chen, T. Y. Lim, M. Hasegawa-Johnson, and M. N. Do, "Semantic image inpainting with perceptual and contextual losses," 2016, *arXiv:1607.07539*.

[146] J. Yi, A. D. Le, T. Wang, X. Wu, and W. Xu, "Outlier detection using generative models with theoretical performance guarantees," 2018, *arXiv:1810.11335*.

[147] J. Yoo, K. H. Jin, H. Gupta, J. Yerly, M. Stuber, and M. Unser, "Time-dependent deep image prior for dynamic MRI," *IEEE Trans. Med. Imag.*, vol. 40, no. 12, pp. 3337–3348, Dec. 2021.

[148] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola. "Dive into deep learning." 2022. [Online]. Available: https://d2l.ai

[149] L. Zhang, G. Wang, and G. B. Giannakis, "Real-time power system state estimation and forecasting via deep unrolled neural networks," *IEEE Trans. Signal Process.*, vol. 67, no. 15, pp. 4069–4077, Aug. 2019.

[150] E. D. Zhong, T. Bepler, J. H. Davis, and B. Berger, "Reconstructing continuous distributions of 3D protein structure from cryo-EM images," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–20.