Optimization Guarantees of Unfolded ISTA and ADMM Networks With Smooth Soft-Thresholding

Shaik Basheeruddin Shah, *Student Member, IEEE,* Pradyumna Pradhan, Wei Pu, *Member, IEEE,* Ramunaidu Randhi, Miguel R. D. Rodrigues, *Fellow, IEEE*, Yonina C. Eldar, *Fellow, IEEE*

***Abstract*—Solving linear inverse problems plays a crucial role in numerous applications. Algorithm unfolding based, model-aware data-driven approaches have gained significant attention for effectively addressing these problems. Learned iterative soft-thresholding algorithm (LISTA) and alternating direction method of multipliers compressive sensing network (ADMM-CSNet) are two widely used such approaches, based on ISTA and ADMM algorithms, respectively. In this work, we study optimization guarantees, i.e., achieving near-zero training loss with the increase in the number of learning epochs, for finite-layer unfolded networks such as LISTA and ADMM-CSNet with smooth soft-thresholding in an over-parameterized (OP) regime. We achieve this by leveraging a modified version of the Polyak-Łojasiewicz, denoted PL*, condition. Satisfying the PL* condition within a specific region of the loss landscape ensures the existence of a global minimum and exponential convergence from initialization using gradient descent based methods. Hence, we provide conditions, in terms of the network width and the number of training samples, on these unfolded networks for the PL* condition to hold, by deriving the Hessian spectral norm. Additionally, we show that the threshold on the number of training samples increases with the increase in the network width. Furthermore, we compare the threshold on training samples of unfolded networks with that of a standard fully-connected feed-forward network (FFNN) with smooth soft-thresholding non-linearity. We prove that unfolded networks have a higher threshold value than FFNN. Consequently, one can expect a better expected error for unfolded networks than FFNN.**

***Index Terms*—Optimization Guarantees, Algorithm Unfolding, LISTA, ADMM-CSNet, Polyak-Łojasiewicz condition**

I. Introduction

LINEAR inverse problems are fundamental in many engineering and science applications [1], [2], where the aim is to recover a vector of interest or target vector from an observation vector. Existing approaches to address these problems can be categorized into two types; model-based and data-driven. Model-based approaches use mathematical formulations that represent knowledge of the underlying model, which connects observation and target information. These approaches are computationally efficient and require accurate model knowledge for good performance [3], [4]. In data-driven approaches, a machine learning (ML) model, e.g., a neural network, with a training dataset, i.e., a supervised setting, is generally considered. Initially, the model is trained by minimizing a certain loss function. Then, the trained model is used on unseen test data. Unlike model-based methods, data-driven approaches do not require underlying model knowledge. However, they require a large amount of data and computational resources while training [3], [4].

By utilizing both domains' knowledge, i.e., the mathematical formulation of the model and ML ability, a new approach, called model-aware data-driven, has been introduced [5], [6]. This approach involves the construction of a neural network architecture based on an iterative algorithm, which solves the optimization problem associated with the given model. This process is called algorithm unrolling or unfolding [6], [7]. It has been observed that the performance, in terms of accurate recovery of the target vector and training data requirements, of model-aware data-driven networks is better when compared with existing techniques [5], [8]. Recently, algorithm unrolling has been used in many applications [8]–[17]. Specifically, learned iterative soft-thresholding algorithm (LISTA) and alternating direction method of multipliers compressive sensing network (ADMM-CSNet) are two popular unfolded networks that have been used in many applications such as image compressive sensing [8], image deblurring [13], image super-resolution [14], super-resolution microscopy [15], clutter suppression in ultrasound [16], power system state estimation [17], and many more.

Nevertheless, theoretical studies supporting these unfolded networks remain to be established. There exist a few theoretical studies that address the challenges of generalization [18]–[20] and convergence rate [21]–[23] in unfolded networks. For instance, in [18], the authors showed that unfolded networks exhibit higher generalization capability compared with standard ReLU networks by deriving an upper bound on the generalization and estimation errors. In [21]–[23] the authors examined the LISTA network convergence to the ground truth as the number of layers increases i.e., layer-wise convergence (which is analogous to iteration-wise convergence in the ISTA algorithm). However, in [21]–[23], the network weights are not learned but are calculated analytically by solving a data-free optimization problem. In this work, we study guarantees to achieve near-zero training loss with an increase in the number of learning epochs, i.e., *optimization guarantees*, by using gradient descent (GD) for both LISTA and ADMM-CSNet with smooth activation in an over-parameterized regime. Note

Part of this work has been accepted for presentation at IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 2022.
S. B. Shah and Y. C. Eldar are with the Weizmann Institute of Science, Rehovot, Israel. P. Pradyumna and R. Ramu Naidu are with the Department of Humanities and Sciences, Indian Institute of Petroleum and Energy, India. W. Pu is with the University of Electronic Science and Technology, China. M. Rodrigues is with the Department of Electronic and Electrical Engineering, University College London, UK.

This work was supported both by The Alan Turing Institute and Weizmann – UK Making Connections Programme (Ref. 129589). The fourth author is thankful to the National Board for Higher Mathematics (NBHM), Govt. of India, for its support (02011/26/2023/NBHM(R.P)/R&D II/5867).

This article has been accepted for publication in IEEE Transactions on Signal Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSP.2024.3412981

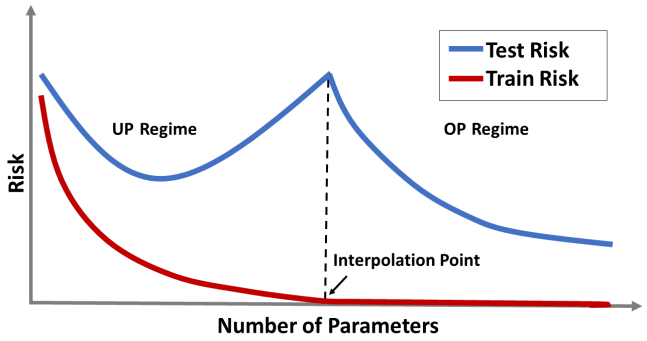

Fig. 1: Double descent risk curve.

that, our work differs from [21]–[23], as we focus on the convergence of training loss with the increase in the number of epochs by fixing the number of layers in the network.

In classical ML theory, we aim to minimize the expected/test risk by finding a balance between under-fitting and over-fitting, i.e., achieving the bottom of the classical U-shaped test risk curve [24]. Modern ML results establish that large models that try to fit train data exactly, i.e., interpolate, *often* show high test accuracy even in the presence of noise [25]–[30]. Recently, ML practitioners proposed a way to numerically justify the relationship between classical and modern ML practices. They achieved this by proposing a performance curve called the double-descent test risk curve [25], [26], [28], [29], which is depicted in Fig. 1. This curve shows that increasing the model capacity (e.g., model parameters) until interpolation results in the classical U-shaped risk curve; further increasing it beyond the interpolation point reduces the test risk. Thus, understanding the conditions – as a function of the training data – that allow perfect data fitting is crucial.

Neural networks can be generally categorized into under-parameterized (UP) and over-parameterized (OP), based on the number of trainable parameters and the number of training data samples. If the number of trainable parameters is less than the number of training samples, then the network is referred to as an UP model, otherwise, it is referred to as an OP model. The loss landscape of both UP and OP models is generally non-convex. However, OP networks satisfy *essential non-convexity* [31]. Particularly, the loss landscape of an OP model has a non-isolated manifold of global minima with non-convexity around any small neighborhood of a global minimum. Despite being highly non-convex, GD based methods work well for training OP networks [32]–[35]. Recently, in [31], [36], the authors provided a theoretical justification for this. Specifically, they proved that the loss landscape, corresponding to the squared loss function, of a typical smooth OP model satisfies a modified version of the Polyak-Łojasiewicz condition, denoted PL*, on most of the parameter space. Indeed, a necessary (but not sufficient) condition to satisfy PL* is that the model should be in the OP regime. Satisfying PL* on a region in the parameter space guarantees the existence of a global minimum in that region, and exponential convergence to the global minimum from a Gaussian initialization using simple GD, with an increase in the number of learning epochs.

Motivated by the aforementioned PL*-based mathematical framework of OP networks, in this paper, we analyze optimization guarantees of finite-layer OP based unfolded ISTA and ADMM networks. As the analysis of PL* depends on the double derivative of the model [31], we consider a smooth version of the soft-thresholding as an activation function. The major contributions of the paper are summarized as follows:

- As the linear inverse problem aims to recover a vector, we initially extend the gradient-based optimization analysis of the OP model with a scalar output, proposed in [31], to a vector output. In the process, we prove that a necessary condition to satisfy PL* is $P \gg mT$, where P denotes the number of parameters, m is the dimension of the model output vector, and T denotes the number of training samples.
- In [31], [36], the authors provided a condition on the width of a fully-connected feed-forward neural network (FFNN) with scalar output to satisfy the PL* condition by utilizing the Hessian spectral norm of the network. Motivated by this work, we derive the Hessian spectral norm of finite-layer LISTA and ADMM-CSNet with smoothed soft-thresholding non-linearity. We show that the norm is on the order of $\tilde{\Omega}\left(1/\sqrt{m}\right)$, where m denotes the width of the network which is equal to the target vector dimension.
- By employing the Hessian spectral norm, we derive necessary conditions on both m and T to satisfy the PL* condition for both LISTA and ADMM-CSNet. Moreover, we demonstrate that the threshold on T, which denotes the maximum number of training samples that a network can memorize, increases as the network width increases.
- We compare the threshold on the number of training samples of LISTA and ADMM-CSNet with that of FFNN, solving a given linear inverse problem. Our findings show that LISTA/ADMM-CSNet exhibits a higher threshold value than FFNN. We demonstrate this by proving that the upper bound on the minimum eigenvalue of the tangent kernel matrix at initialization is high for LISTA/ADMM-CSNet compared to FFNN. This implies that, with fixed network parameters, the unfolded network is capable of memorizing a larger number of training samples compared to FFNN. Therefore, we expect to obtain a better expected error (which is upper bounded by the sum of generalization and training error [37]) for unfolded networks than FFNN.
- We numerically evaluate the parameter efficiency of unfolded networks in comparison to FFNNs. In particular, we demonstrate that FFNNs require a higher number of parameters to achieve near-zero empirical training loss compared to LISTA/ADMM-CSNet for given T.

To be specific, the contributions in [38] are as follows: We provided a closed-form expression for the Hessian spectral norm of unfolded networks with its mathematical derivation omitted. Additionally, we provided the bounds on the number of training samples for both LISTA and ADMM-CSNet to achieve near-zero training loss using the gradient descent (GD) approach in an over-parameterized (OP) regime. Furthermore,

we justified the same with a few simulation results.

The paper is organized as follows. Section II reviews LISTA and ADMM-CSNet, and formulates the problem. Section III extends the PL*-based optimization guarantees of an OP model with scalar output to a model with multiple outputs. Section IV begins by deriving the Hessian spectral norm of the unfolded networks. Then, it provides conditions on the network width and on the number of training samples to satisfy the PL* condition. It also establishes a comparative analysis of the threshold for the number of training samples among LISTA, ADMM-CSNet, and FFNN. Section V introduces experimental results and Section VI concludes the paper.

The following notations are used throughout the paper. The set of real numbers is denoted by $\mathbb{R}$. We use bold lowercase letters, e.g., $\mathbf{y}$, for vectors, capital letters, e.g., W, for matrices, and bold capital letters, e.g., $\mathbf{H}$, for tensors. Symbols $||\mathbf{z}||_1$, $||\mathbf{z}||$, and $||\mathbf{z}||_\infty$ denote the l_1-norm, l_2-norm, and l_∞-norm of vector $\mathbf{z}$, respectively. The spectral norm and Frobenius norm of a matrix W are written as $||W||$ and $||W||_F$, respectively. We use $[L]$ to denote the set $\{1, 2, \dots, L\}$, where L is a natural number. The first-order derivative or gradient of a function $L(\mathbf{w})$ w.r.t. $\mathbf{w}$ is written as $\nabla_{\mathbf{w}} L(\mathbf{w})$. The asymptotic upper bound and lower bound on a quantity are described using $O(\cdot)$ and $\Omega(\cdot)$, respectively. Notations $\tilde{O}(\cdot)$ and $\tilde{\Omega}(\cdot)$ are used to suppress the logarithmic terms in $O(\cdot)$ and $\Omega(\cdot)$, respectively. For example, $O\left(\frac{1}{m}\ln(m)\right)$ is written as $\tilde{O}\left(\frac{1}{m}\right)$. Symbols $\gg$ and $\ll$ mean "much greater than" and "much lesser than", respectively. Consider a matrix G with $G_{i,j} = \sum_k A_{i,j,k} v_k$, where $A_{i,j,k}$ is a component in tensor $\mathbf{A} \in \mathbb{R}^{m_1 \times m_2 \times m_3}$. The spectral norm of G is bounded as

$$\|G\| \leq \|\mathbf{A}\|_{2,2,1} \|\mathbf{v}\|_\infty. \tag{1}$$

Here $\|\mathbf{A}\|_{2,2,1}$ is the $(2,2,1)$-norm of the tensor $\mathbf{A}$, defined as

$$\|\mathbf{A}\|_{2,2,1} = \sup_{\|\mathbf{r}\|=\|\mathbf{s}\|=1} \sum_{k=1}^{m_3} \left| \sum_{i=1}^{m_1} \sum_{j=1}^{m_2} A_{i,j,k} r_i s_j \right|, \tag{2}$$

where $\mathbf{r} \in \mathbb{R}^{m_1 \times 1}$ and $\mathbf{s} \in \mathbb{R}^{m_2 \times 1}$.

II. Problem Formulation

A. LISTA and ADMM-CSNet

Consider the following linear inverse problem

$$\mathbf{y} = A\mathbf{x} + \mathbf{e}. \tag{3}$$

Here $\mathbf{y} \in \mathbb{R}^{n \times 1}$ is the observation vector, $\mathbf{x} \in \mathbb{R}^{m \times 1}$ is the target vector, $A \in \mathbb{R}^{n \times m}$ is the forward linear operator matrix with $m > n$, and $\mathbf{e}$ is noise with $\|\mathbf{e}\|_2 < \epsilon$, where the constant $\epsilon > 0$. Our aim is to recover $\mathbf{x}$ from a given $\mathbf{y}$.

In model-based approaches, an optimization problem is formulated using prior knowledge about the target vector and is usually solved using an iterative algorithm. For instance, by assuming $\mathbf{x}$ is a k-sparse vector [39], the least absolute shrinkage and selection operator (LASSO) problem is formulated as

$$\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{y} - A\mathbf{x}\|^2 + \gamma \|\mathbf{x}\|_1, \tag{4}$$

where γ is a regularization parameter. We consider k as a constant value throughout our analysis. Iterative algorithms, such as ISTA and ADMM [40], are generally used to solve the LASSO problem. The update of $\mathbf{x}$ at the l^{th} iteration in ISTA is [41]

$$\mathbf{x}^l = S_{\gamma\tau}\left\{\left(\mathbf{I} - \tau A^T A\right)\mathbf{x}^{l-1} + \tau A^T \mathbf{y}\right\}, \tag{5}$$

where $\mathbf{x}^0$ is a bounded input initialization, τ controls the iteration step size, and $S_\lambda(\cdot)$ is the soft-thresholding operator applied element-wise on a vector argument $S_\lambda(x) = \text{sign}(x)\max\left(|x| - \lambda, 0\right)$. The l^{th} iteration in ADMM is [42]

$$\begin{aligned} \mathbf{x}^l &= \left(A^T A + \rho \mathbf{I}\right)^{-1}\left(A^T \mathbf{y} + \rho\left(\mathbf{z}^{l-1} - \mathbf{u}^{l-1}\right)\right), \\ \mathbf{z}^l &= S_{\frac{\gamma}{\rho}}\left(\mathbf{x}^l + \mathbf{u}^{l-1}\right), \\ \mathbf{u}^l &= \mathbf{u}^{l-1} + \left(\mathbf{x}^l - \mathbf{z}^l\right), \end{aligned} \tag{6}$$

where $\mathbf{x}^0$, $\mathbf{z}^0$, and $\mathbf{u}^0$, are bounded input initializations to the network and $\rho > 0$ is a penalty parameter. Model-based approaches are in general sensitive to inaccurate knowledge of the underlying model [3], [4]. In turn, data-driven approaches use an ML model to recover the target vector. These approaches generally necessitate a substantial volume of data and computational resources for training [3], [4].

A model-aware data-driven approach can be developed using algorithm unfolding or unrolling [6]. In unfolding, a neural network is constructed by mapping each iteration in the iterative algorithm (such as (5) or (6)) to a network layer. Hence, an iterative algorithm with L-iterations leads to an L-layer cascaded deep neural network. The network is then trained by using the available dataset containing a series of pairs $\{\mathbf{y}_i, \mathbf{x}_i\}, i \in [T]$. For example, the update of $\mathbf{x}$ at the l^{th} iteration in ISTA, given in (5), is rewritten as

$$\mathbf{x}^l = S_\lambda\left\{W_2^l \mathbf{x}^{l-1} + W_1^l \mathbf{y}\right\}, \tag{7}$$

where $\lambda = \gamma\tau$, $W_1^l = \tau A^T$, and $W_2^l = \mathbf{I} - \tau A^T A$. By considering W_1^l, W_2^l, and λ as network learnable parameters, one can map the above l^{th} iteration to an l^{th} layer in the network as shown in Fig. 2. The corresponding unfolded network is called learned ISTA (LISTA) [5].

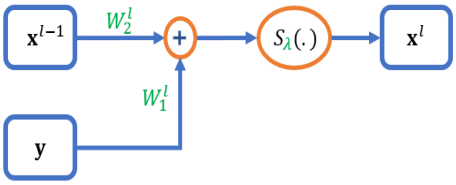

Fig. 2: l^{th} layer of the unfolded ISTA network.

Similarly, by considering $W_1^l = \left(A^T A + \rho \mathbf{I}\right)^{-1} A^T$, $W_2^l = \left(A^T A + \rho \mathbf{I}\right)^{-1} \rho$, and $\lambda = \frac{\gamma}{\rho}$ as learnable parameters, (6) is rewritten as

$$\begin{aligned} \mathbf{x}^l &= W_1^l \mathbf{y} + W_2^l\left(\mathbf{z}^{l-1} - \mathbf{u}^{l-1}\right), \\ \mathbf{z}^l &= S_\lambda\left(\mathbf{x}^l + \mathbf{u}^{l-1}\right), \\ \mathbf{u}^l &= \mathbf{u}^{l-1} + \left(\mathbf{x}^l - \mathbf{z}^l\right). \end{aligned} \tag{8}$$

The above l^{th} iteration in ADMM can be mapped to an l^{th} layer in a network as shown in Fig. 3, leading to ADMM-CSNet [8]. From a network point of view, the inputs of l^{th}

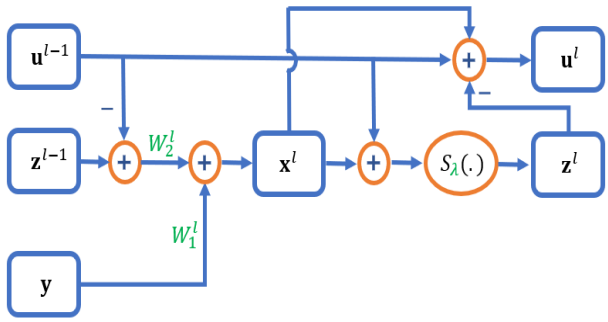

Fig. 3: l^{th} layer of the unfolded ADMM network.

layer are $\mathbf{x}^{l-1}$ and $\mathbf{y}$ for LISTA, and $\mathbf{z}^{l-1}$, $\mathbf{u}^{l-1}$ and $\mathbf{y}$ for ADMM-CSNet.

It has been observed that the performance of LISTA and ADMM-CSNet is better in comparison with ISTA, ADMM, and traditional networks, in many applications [5], [8]. For instance, to achieve good performance the number of layers required in an unrolled network is generally much smaller than the number of iterations required by the iterative solver [5]. In addition, an unrolled network works effectively even if the linear operator matrix, A, is not known exactly. An unrolled network typically requires less data for training compared to standard deep neural networks [3] to achieve a certain level of performance on unseen data. Due to these advantages, LISTA and ADMM-CSNet have been used in many applications [8], [13]–[17]. That said, the theoretical foundations supporting these networks remain to be established. While there have been some studies focusing on the generalization [18]–[20] and convergence rate [21]–[23] of unfolded networks, a comprehensive study of the optimization guarantees is lacking. Here, we analyze the conditions on finite L-layer LISTA and ADMM-CSNet to achieve near-zero training loss with the increase in the number of epochs.

B. Problem Formulation

We consider the following questions: Under what conditions does the training loss in LISTA and ADMM-CSNet converge to zero as the number of epochs tends to infinity using GD? Additionally, how do these conditions differ for FFNNs?

For the analysis, we consider the following training setting: Let $\mathbf{x} = F(\mathbf{w}, \lambda; \mathbf{y})$ be an L-layer unfolded model, where $\mathbf{y} \in \mathbb{R}^{n\times 1}$ is the model input vector, $\mathbf{x} \in \mathbb{R}^{m\times 1}$ is the model output, and $\mathbf{w} \in \mathbb{R}^{P\times 1}$ and λ are the learnable parameters. To simplify the analysis, λ is assumed to be constant, henceforth, we write $F(\mathbf{w}, \lambda; \mathbf{y})$ as $F(\mathbf{w}; \mathbf{y})$. This implies that $\mathbf{w}_{P\times 1} = \text{Vec}\left([\mathbf{W}]_{L\times m\times(m+n)}\right)$ is the only learnable (untied) parameter vector, where

$$\mathbf{W} = \left[W^1 \; W^2 \; \dots \; W^L\right], \tag{9}$$

and $\left[W^l\right]_{m\times(m+n)} = \left[W_1^l \;\; W_2^l\right]$ is the parameter matrix corresponding to the l^{th}-layer. Alternatively, we can write

$$\mathbf{W} = \left[[\mathbf{W}_1]_{L\times m\times n} \;\; [\mathbf{W}_2]_{L\times m\times m}\right], \tag{10}$$

$\mathbf{W}_1 = \left[W_1^1 \; \dots \; W_1^L\right]$ and $\mathbf{W}_2 = \left[W_2^1 \; \dots \; W_2^L\right]$.

Consider the training dataset $\{\mathbf{y}_i, \mathbf{x}_i\}_{i=1}^T$. An optimal parameter vector $\mathbf{w}^*$, such that $F(\mathbf{w}^*; \mathbf{y}_i) \approx \mathbf{x}_i$, $\forall i \in [T]$, is found by minimizing an empirical loss function $L(\mathbf{w})$, defined as

$$L(\mathbf{w}) = \sum_{i=1}^{T} l(\mathbf{f}_i, \mathbf{x}_i), \tag{11}$$

where $l(\cdot)$ is the loss function, $\mathbf{f}_i = (\mathcal{F}(\mathbf{w}))_i = F(\mathbf{w}, \mathbf{y}_i)$, $\mathcal{F}(\cdot) : \mathbb{R}^{P\times 1} \to \mathbb{R}^{m\times T}$, and $(\mathcal{F}(\mathbf{w}))_i$ is the i^{th} column in $\mathcal{F}(\mathbf{w})$. We consider the squared loss, hence

$$L(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{T} \|\mathbf{f}_i - \mathbf{x}_i\|^2 = \frac{1}{2}\|\mathcal{F}(\mathbf{w}) - X\|_F^2, \tag{12}$$

where $X = [\mathbf{x}_1, \dots, \mathbf{x}_T]$. We choose GD as the optimization algorithm for minimizing $L(\mathbf{w})$, so that, the updating rule is

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \nabla_{\mathbf{w}} L(\mathbf{w})$$

where η is the learning rate. In this study, the training data $\{\mathbf{y}_i, \mathbf{x}_i\}_{i=1}^T$ is generated with model priors k and A following standard compressed sensing (CS) theory. Post-data generation, the proposed theoretical or numerical analysis remains independent of both prior values. This is because unrolled models, as described in equations (13) and (14), remain independent of both A and k.

Our aim is to derive conditions on LISTA and ADMM-CSNet such that $L(\mathbf{w})$ converges to zero with an increase in the number of epochs using GD, i.e., $\lim_{t\to\infty} L(\mathbf{w}_t) = 0$. In addition, we compare these conditions with those of FFNN, where we obtain the conditions for FFNN by extending the analysis given in [31]. Specifically, in Section IV-C, we derive a bound on the number of training samples to achieve near zero training loss for unfolded networks. We show that this threshold is lower for FFNN compared to unfolded networks.

III. Revisiting PL*-Based Optimization Guarantees

In [31] the authors proposed PL*-based optimization theory for a model with a scalar output. Motivated by this, in this section, we extend this theory to a multi-output model, as we aim to recover a vector in a linear inverse problem.

Consider an ML model, not necessarily an unfolded network, $\mathbf{x} = F(\mathbf{w}; \mathbf{y})$, with the training setup mentioned in Section II-B, where $\mathbf{y} \in \mathbb{R}^{n\times 1}$, $\mathbf{x} \in \mathbb{R}^{m\times 1}$, and $\mathbf{w} \in \mathbb{R}^{P\times 1}$. Furthermore, assume that the model is $L_{\mathcal{F}}$-Lipschitz continuous and $\beta_{\mathcal{F}}$-smooth. A function $\mathcal{F}(\cdot) : \mathbb{R}^P \to \mathbb{R}^{m\times T}$ is $L_{\mathcal{F}}$-Lipschitz continuous if

$$\|\mathcal{F}(\mathbf{w}_1) - \mathcal{F}(\mathbf{w}_2)\|_F \le L_{\mathcal{F}}\|\mathbf{w}_1 - \mathbf{w}_2\|, \; \forall \mathbf{w}_1, \mathbf{w}_2 \in \mathbb{R}^P,$$

and is $\beta_{\mathcal{F}}$-smooth if the gradient of the function is $\beta_{\mathcal{F}}$-Lipschitz, i.e.,

$$\|\nabla_{\mathbf{w}}\mathcal{F}(\mathbf{w_1}) - \nabla_{\mathbf{w}}\mathcal{F}(\mathbf{w_2})\|_F \le \beta_{\mathcal{F}}\|\mathbf{w_1} - \mathbf{w_2}\|,$$

for all $\mathbf{w}_1, \; \mathbf{w}_2 \in \mathbb{R}^P$. The Hessian spectral norm of $\mathcal{F}(\cdot)$ is defined as

$$\|\mathbf{H}_{\mathcal{F}}(\mathbf{w})\| = \max_{i\in[T]} \|\mathbf{H}_{\mathcal{F}_i}(\mathbf{w})\|,$$

where $\mathbf{H}_{\mathcal{F}} \in \mathbb{R}^{T\times m\times P\times P}$ is a tensor with $(\mathbf{H}_{\mathcal{F}})_{i,j,k,l} = \frac{\partial^2(\mathcal{F}(\mathbf{w}))_{j,i}}{\partial w_k \partial w_l}$ and $\mathbf{H}_{\mathcal{F}_i} = \frac{\partial^2(\mathcal{F}(\mathbf{w}))_i}{\partial \mathbf{w}^2}$. As stated earlier, the loss

landscape of the OP model typically satisfies PL* on most of the parameter space. Formally, the PL* condition is defined as follows [43], [44]:

Definition 1. *Consider a set $C \subset \mathbb{R}^{P\times 1}$ and $\mu > 0$. Then, a non-negative function $L(\mathbf{w})$ satisfies μ-PL* condition on C if $\|\nabla_{\mathbf{w}} L(\mathbf{w})\|^2 \geq \mu L(\mathbf{w}),\ \forall \mathbf{w} \in C$.*

Definition 2. *The tangent kernel matrix, $[K(\mathbf{w})]_{mT\times mT}$, of the function $\mathcal{F}(\mathbf{w})$, is a block matrix with $(i,j)^{th}$ block defined as*

$$(K(\mathbf{w}))_{i,j} = [\nabla_{\mathbf{w}} \mathbf{f}_i]_{m\times P} [\nabla_{\mathbf{w}} \mathbf{f}_j]^T_{P\times m},\ i \in [T] \text{ and } j \in [T],$$

where $\mathcal{F}(\cdot) : \mathbb{R}^{P\times 1} \to \mathbb{R}^{m\times T}$, $\mathbf{f}_i = (\mathcal{F}(\mathbf{w}))_i$, and $(\mathcal{F}(\mathbf{w}))_i$ is the i^{th} column in $\mathcal{F}(\mathbf{w})$.

From the above definitions, we have the following lemma, which is called μ-uniform conditioning [31] of a multi-output model $\mathcal{F}(\mathbf{w})$:

Lemma 1. *$\mathcal{F}(\mathbf{w})$ satisfies μ-PL* on set C if the minimum eigenvalue of the tangent kernel matrix, $K(\mathbf{w})$, is greater than or equal to μ, i.e., $\lambda_{min}(K(\mathbf{w})) \geq \mu,\ \forall \mathbf{w} \in C$.*

Proof. From (12), we have

$$\begin{aligned}\|\nabla_{\mathbf{w}} L(\mathbf{w})\|^2 &= \left[\hat{\mathbf{f}} - \hat{\mathbf{x}}\right]^T \left[\nabla_{\mathbf{w}} \hat{\mathbf{f}}\right]_{mT\times P} \left[\nabla_{\mathbf{w}} \hat{\mathbf{f}}\right]^T_{P\times mT} \left[\hat{\mathbf{f}} - \hat{\mathbf{x}}\right] \\ &= \left[\hat{\mathbf{f}} - \hat{\mathbf{x}}\right]^T [K(\mathbf{w})]_{mT\times mT} \left[\hat{\mathbf{f}} - \hat{\mathbf{x}}\right],\end{aligned}$$

where $\hat{\mathbf{f}} = \text{Vec}(\mathcal{F}(\mathbf{w}))$ and $\hat{\mathbf{x}} = \text{Vec}(X)$. The above equation can be lower-bounded as

$$\|\nabla_{\mathbf{w}} L(\mathbf{w})\|^2 \geq \lambda_{\min}(K(\mathbf{w})) \|\hat{\mathbf{f}} - \hat{\mathbf{x}}\|_2^2 \geq \mu L(\mathbf{w}),$$

completing the proof. □

Observe that $K(\mathbf{w})$ is a positive semi-definite matrix. Thus, a necessary condition to satisfy the PL* condition (that is, a necessary condition to obtain a full rank $K(\mathbf{w})$), for a multi-output model is $P \gg mT$. For a scalar output model, the equivalent condition is $P \gg T$ [31]. Note that if $P \ll T$, i.e., an UP model with a scalar output, then $\lambda_{\min}(K(\mathbf{w})) = 0$, implies that an UP model does not satisfy the PL* condition.

Practically, computing $\lambda_{\min}(K(\mathbf{w}))$ for every $\mathbf{w} \in C$, to verify the PL* condition, is not feasible. One can overcome this by using the Hessian spectral norm of the model $\|\mathbf{H}_{\mathcal{F}}(\mathbf{w})\|$ [31]:

Theorem 1. *Let $\mathbf{w}_0 \in \mathbb{R}^{P\times 1}$ be the parameter initialization of an $L_{\mathcal{F}}$-Lipschitz and $\beta_{\mathcal{F}}$-smooth model $\mathcal{F}(\mathbf{w})$, and $B(\mathbf{w}_0, R) = \{\mathbf{w} |\ \|\mathbf{w} - \mathbf{w}_0\| \leq R\}$ be a ball with radius $R > 0$. Assume that $K(\mathbf{w}_0)$ is well conditioned, i.e., $\lambda_{min}(K(\mathbf{w}_0)) = \lambda_0$ for some $\lambda_0 > 0$. If $\|\mathbf{H}_{\mathcal{F}}(\mathbf{w})\| \leq \frac{\lambda_0 - \mu}{2L_{\mathcal{F}}\sqrt{TR}}$ for all $\mathbf{w} \in B(\mathbf{w}_0, R)$, then the model satisfies μ-uniform conditioning in $B(\mathbf{w}_0, R)$; this also implies that $L(\mathbf{w})$ satisfies μ-PL* in the ball $B(\mathbf{w}_0, R)$.*

The intuition behind the above theorem is that small $\|\mathbf{H}_{\mathcal{F}}(\mathbf{w})\|$ leads to a small change in the tangent kernel. Precisely, if the tangent kernel is well conditioned at the initialization, then a small $\|\mathbf{H}_{\mathcal{F}}(\mathbf{w})\|$ in $B(\mathbf{w}_0, R)$ guarantees that the tangent kernel is well conditioned within $B(\mathbf{w}_0, R)$. The following theorem states that satisfying PL* guarantees the existence of a global minimum and exponential convergence to the global minimum from $\mathbf{w}_0$ using GD:

Theorem 2. *Consider a model $\mathcal{F}(\mathbf{w})$ that is $L_{\mathcal{F}}$-Lipschitz continuous and $\beta_{\mathcal{F}}$-smooth. If the square loss function $L(\mathbf{w})$ satisfies the μ-PL* condition in $B(\mathbf{w}_0, R)$ with $R = \frac{2L_{\mathcal{F}}\|\mathcal{F}(\mathbf{w}_0) - X\|_F}{\mu} = O\left(\frac{1}{\mu}\right)$, then we have the following:*

- *There exist a global minimum, $\mathbf{w}^*$, in $B(\mathbf{w}_0, R)$ such that $\mathcal{F}(\mathbf{w}^*) = X$.*
- *GD with step size $\eta \leq \frac{1}{L_{\mathcal{F}}^2 + \beta_{\mathcal{F}}\|\mathcal{F}(\mathbf{w}_0) - X\|_F}$ converges to a global minimum at an exponential convergence rate, specifically, $L(\mathbf{w}_t) \leq (1 - \eta\mu)^t L(\mathbf{w}_0)$.*

The proofs of Theorems 1 and 2 are similar to the proofs of Theorems 2 and 6, respectively, in [31]. However, as linear inverse problems deal with vector recovery, the proofs rely on Frobenius norms instead of Euclidean norms.

IV. Optimization Guarantees

We now analyze the optimization guarantees of both LISTA and ADMM-CSNet by considering them in the OP regime. Hence, the aim is further simplified to study under what conditions LISTA and ADMM-CSNet satisfy the PL* condition. As mentioned in Theorem 1, one can verify the PL* condition using the Hessian spectral norm of the network. Thus, in this section, we first compute the Hessian spectral norm of both LISTA and ADMM-CSNet. The mathematical analysis performed here is motivated by [36], where the authors derived the Hessian spectral norm of an FFNN with a scalar output. Then, we provide conditions on both the network width and the number of training samples to satisfy the PL* condition. Subsequently, we compare among unfolded networks and FFNN to evaluate the threshold on the number of training samples.

A. Assumptions

For the analysis, we consider certain assumptions on the unfolded ISTA and ADMM networks. The inputs of the networks are bounded, i.e., there exist some constants C_x, C_u, C_z, and C_y such that $|x_i^0| \leq C_x$, $|u_i^0| \leq C_u$, $|z_i^0| \leq C_z$, $\forall i \in [m]$, and $|y_i| \leq C_y,\ \forall i \in [n]$. As the computation of the Hessian spectral norm involves a second-order derivative, we approximate the soft-thresholding activation function, $S_\lambda(\cdot)$, in the unfolded network with the double-differentiable/smooth soft-thresholding activation function, $\sigma_\lambda(\cdot)$, formulated using soft-plus, where $\sigma_\lambda(x) = \log\left(1 + e^{x-\lambda}\right) - \log\left(1 + e^{-x-\lambda}\right)$. Fig. 4 depicts $S_\lambda(x)$ and $\sigma_\lambda(x)$ for $\lambda = 5$. Observe that $\sigma_\lambda(x)$ approximates well the shape of $S_\lambda(x)$. There are several works in the literature that approximate the soft-thresholding function with a smooth version of it [45]–[51]. The analysis proposed in this work can be extended as is to other smooth approximations. Since λ is assumed to be a constant (refer to Section II-B), henceforth, we write $\sigma_\lambda(\cdot)$ as $\sigma(\cdot)$. It is well known that $\sigma(\cdot)$ is L_σ-Lipschitz continuous and β_σ-smooth.

Let $\mathbf{W}_0, \mathbf{W}_{10}, \mathbf{W}_{20}, W_{10}^l$ and W_{20}^l denote the initialization of $\mathbf{W}, \mathbf{W}_1, \mathbf{W}_2$, W_1^l and W_2^l, respectively. We use identical independent random Gaussian initialization for each parameter

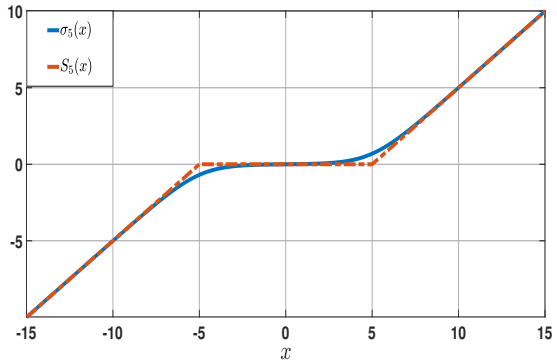

Fig. 4: Soft-threshold function, $S_\lambda(x)$, and its smooth approximation, $\sigma_\lambda(x)$ (formulated using the soft-plus function), with $\lambda = 5$.

with mean 0 and variance 1, i.e., $\left(W_{10}^l\right)_{i,j} \sim \mathcal{N}(0,1)$ and $\left(W_{20}^l\right)_{i,j} \sim \mathcal{N}(0,1)$, for all i, j, and $l \in [L]$. This guarantees well conditioning of the tangent kernel at initialization [31], [32]. The Gaussian initialization imposes certain bounds, with high probability, on the spectral norm of the weight matrices. In particular, we have the following:

Lemma 2. *If $\left(W_{10}^l\right)_{i,j} \sim \mathcal{N}(0,1)$ and $\left(W_{20}^l\right)_{i,j} \sim \mathcal{N}(0,1)$, $\forall l \in [L]$, i.e., independent and identical (i.i.d.) Gaussian initialized, then with probability at least $1 - 2\exp\left(-\frac{m}{2}\right)$ we have $\left\|W_{10}^l\right\| \le c_{10}\sqrt{n} = O(\sqrt{n})$ and $\left\|W_{20}^l\right\| \le c_{20}\sqrt{m} = O(\sqrt{m})$, $\forall l \in [L]$, where $c_{10} = 1 + 2\sqrt{m}/\sqrt{n}$ and $c_{20} = 3$.*

Proof. Any matrix $U \in \mathbb{R}^{m_1 \times m_2}$ with i.i.d. Gaussian initialized satisfies the following inequality with probability at least $1 - 2\exp\left(-\frac{t^2}{2}\right)$, where $t \ge 0$, [52]: $\|U\| \le \sqrt{m_1} + \sqrt{m_2} + t$. Using this fact and considering $t = \sqrt{m}$, we get $\|W_{10}^l\| = O(\sqrt{n})$ and $\|W_{20}^l\| = O(\sqrt{m})$. □

The following lemma shows that the spectral norm of the weight matrices within a finite radius ball is of the same order as at initialization.

Lemma 3. *If $\mathbf{W}_{10}$ and $\mathbf{W}_{20}$ are initialized as stated in Lemma 2, then for any $\mathbf{W}_1 \in B(\mathbf{W}_{10}, R_1)$ and $\mathbf{W}_2 \in B(\mathbf{W}_{20}, R_2)$, where R_1 and R_2 are positive scalars, we have $\left\|W_1^l\right\| = O(\sqrt{n})$ and $\left\|W_2^l\right\| = O(\sqrt{m})$, $\forall l \in [L]$.*

Proof. From triangular inequality, we have

$$\left\|W_1^l\right\| \le \left\|W_{10}^l\right\| + \left\|W_1^l - W_{10}^l\right\|_F \le c_{10}\sqrt{n} + R_1 = O(\sqrt{n}),$$
$$\left\|W_2^l\right\| \le \left\|W_{20}^l\right\| + \left\|W_2^l - W_{20}^l\right\|_F \le c_{20}\sqrt{m} + R_2 = O(\sqrt{m}),$$

completing the proof. □

As the width of the network can be very high (dimension of the target vector), to obtain the constant asymptotic behavior, the learnable parameters W_1^l and W_2^l are normalized by $\frac{1}{\sqrt{n}}$ and $\frac{1}{\sqrt{m}}$, respectively, and the output of the model is normalized by $\frac{1}{\sqrt{m}}$. This way of normalization is called neural tangent kernel (NTK) parameterization [53], [54]. With these assumptions, the output of a finite L-layer LISTA network is

$$\mathbf{f} = \frac{1}{\sqrt{m}}\mathbf{x}^L, \tag{13}$$

where

$$\mathbf{x}^l = \sigma(\tilde{\mathbf{x}}^l) = \sigma\left(\frac{W_1^l}{\sqrt{n}}\mathbf{y} + \frac{W_2^l}{\sqrt{m}}\mathbf{x}^{l-1}\right) \in \mathbb{R}^{m\times 1},\ l \in [L].$$

Likewise, the output of a finite L-layer ADMM-CSNet is

$$\mathbf{f} = \frac{1}{\sqrt{m}}\mathbf{z}^L, \tag{14}$$

where

$$\mathbf{z}^l = \sigma\left(\tilde{\mathbf{z}}^l\right) = \sigma\left(\mathbf{x}^l + \mathbf{u}^{l-1}\right),$$
$$\mathbf{x}^l = \frac{1}{\sqrt{n}}W_1^l\mathbf{y} + \frac{1}{\sqrt{m}}W_2^l\left(\mathbf{z}^{l-1} - \mathbf{u}^{l-1}\right),$$
$$\mathbf{u}^l = \mathbf{u}^{l-1} + \left(\mathbf{x}^l - \mathbf{z}^l\right),\ l \in [L].$$

To maintain uniformity in notation, hereafter, we denote the output of the network as $\mathbf{f} = \frac{1}{\sqrt{m}}\mathbf{g}^L$, where $\mathbf{g}^l = \mathbf{x}^l$ for LISTA and $\mathbf{g}^l = \mathbf{z}^l$ for ADMM-CSNet.

B. Hessian Spectral Norm

For better understanding, we first compute the Hessian spectral norm of one layer, i.e., $L = 1$, unfolded network.

1) Analysis of 1-Layer Unfolded Network: The Hessian matrix of a 1-layer LISTA or ADMM-CSNet for a given training sample i is[1]

$$[\mathbf{H}_{\mathcal{F}_i}] = [\mathbf{H}]_{m\times P\times P} = \begin{bmatrix} H_1 & H_2 & \cdots & H_m \end{bmatrix}, \tag{15}$$

where $[H_s]_{P\times P} = \frac{\partial^2 f_s}{\partial \mathbf{w}^2}$, $\mathbf{w} = \mathrm{Vec}(W^1) = \mathrm{Vec}\left([W_1^1, W_2^1]\right)$, f_s denotes the s^{th} component in the network output vector $\mathbf{f}$, i.e., $f_s = \frac{1}{\sqrt{m}}\mathbf{v}_s^T\mathbf{g}^1$, and $\mathbf{v}_s$ is a vector with s^{th} element set to be 1 and others to be 0. The Hessian spectral norm given in (15) can be bounded as $\max_{s\in[m]}\{\|H_s\|\} \le \|\mathbf{H}\| \le \sum_s \|H_s\|$. By leveraging the chain rule, we have

$$H_s = \frac{\partial f_s}{\partial \mathbf{g}^1}\frac{\partial^2 \mathbf{g}^1}{\partial \mathbf{w}^2}. \tag{16}$$

We bound H_s, as given below, by using the inequality given in (1),

$$\|H_s\| \le \left\|\frac{\partial f_s}{\partial \mathbf{g}^1}\right\|_\infty \left\|\frac{\partial^2 \mathbf{g}^1}{\partial \mathbf{w}^2}\right\|_{2,2,1}. \tag{17}$$

From (13) or (14), we get

$$\left\|\frac{\partial f_s}{\partial \mathbf{g}^1}\right\|_\infty = \left\|\frac{1}{\sqrt{m}}\mathbf{v}_s^T\right\|_\infty = O\left(\frac{1}{\sqrt{m}}\right). \tag{18}$$

In addition,

$$\left\|\frac{\partial^2 \mathbf{g}^1}{(\partial \mathbf{w})^2}\right\|_{2,2,1} = \left\|\begin{bmatrix} \partial^2\mathbf{g}^1/\left(\partial W_1^1\right)^2 & \partial^2\mathbf{g}^1/\partial W_1^1\partial W_2^1 \\ \partial^2\mathbf{g}^1/\partial W_2^1\partial W_1^1 & \partial^2\mathbf{g}^1/\left(\partial W_2^1\right)^2 \end{bmatrix}\right\|_{2,2,1}$$
$$\le \left\|\frac{\partial^2 \mathbf{g}^1}{\left(\partial W_1^1\right)^2}\right\|_{2,2,1} + 2\left\|\frac{\partial^2 \mathbf{g}^1}{\partial W_1^1\partial W_2^1}\right\|_{2,2,1} + \left\|\frac{\partial^2 \mathbf{g}^1}{\left(\partial W_2^1\right)^2}\right\|_{2,2,1}. \tag{19}$$

[1]Note that, to simplify the notation, we denoted $\mathbf{H}_{\mathcal{F}_i}$ as $\mathbf{H}$.

We now compute the $(2,2,1)$-norms in the above equation for both LISTA and ADMM-CSNet. To begin with, for LISTA, we have the following second-order partial derivatives of layer-wise output, $\mathbf{g}^1$, w.r.t. parameters:

$$\left(\frac{\partial^2 \mathbf{g}^1}{\left(\partial W_1^1\right)^2}\right)_{i,jj',kk'} = \frac{\partial^2 \mathbf{x}_i^1}{\partial (W_1^1)_{jj'} \partial (W_1^1)_{kk'}} = \frac{1}{n}\sigma''\left(\tilde{\mathbf{x}}_i^1\right)\mathbf{y}_{j'}\mathbf{y}_{k'}\mathbb{I}_{i=k=j},$$

$$\left(\frac{\partial^2 \mathbf{g}^1}{\left(\partial W_2^1\right)^2}\right)_{i,jj',kk'} = \frac{1}{m}\sigma''\left(\tilde{\mathbf{x}}_i^1\right)\mathbf{x}_{j'}^0\mathbf{x}_{k'}^0\mathbb{I}_{i=k=j},$$

$$\left(\frac{\partial^2 \mathbf{g}^1}{\partial W_2^1 \partial W_1^1}\right)_{i,jj',kk'} = \frac{1}{\sqrt{mn}}\sigma''\left(\tilde{\mathbf{x}}_i^1\right)\mathbf{x}_{j'}^0\mathbf{y}_{k'}\mathbb{I}_{i=k=j},$$

where $\mathbb{I}_{\{\}}$ denotes the indicator function. By utilizing the definition of $(2,2,1)$-norm given in (2), bounds on inputs of the network, and smoothness of the activation function, the $(2,2,1)$-norms of the above quantities are obtained as shown below:

$$\left\|\frac{\partial^2 \mathbf{g}^1}{\left(\partial W_1^1\right)^2}\right\|_{2,2,1} = \sup_{\|V_1\|_F=\|V_2\|_F=1}\frac{1}{n}\sum_{i=1}^{m}\left|\sigma''\left(\tilde{\mathbf{x}}_i^1\right)(V_1\mathbf{y})_i\,(V_2\mathbf{y})_i\right| \le \sup_{\|V_1\|_F=\|V_2\|_F=1}\frac{1}{2n}\beta_\sigma\left(\|V_1\mathbf{y}\|^2+\|V_2\mathbf{y}\|^2\right) \le \frac{1}{2n}\beta_\sigma\left(\|\mathbf{y}\|^2+\|\mathbf{y}\|^2\right) \le \beta_\sigma C_y^2 = O(1)$$

$$\left\|\frac{\partial^2 \mathbf{g}^1}{\left(\partial W_2^1\right)^2}\right\|_{2,2,1} = \sup_{\|V_1\|_F=\|V_2\|_F=1}\frac{1}{m}\sum_{i=1}^{m}\left|\sigma''\left(\tilde{\mathbf{x}}_i^1\right)\left(V_1\mathbf{x}^0\right)_i\left(V_2\mathbf{x}^0\right)_i\right| \le \frac{1}{2m}\beta_\sigma\left(\left\|\mathbf{x}^0\right\|^2+\left\|\mathbf{x}^0\right\|^2\right) \le \beta_\sigma {C_x}^2 = O(1)$$

$$\left\|\frac{\partial^2 \mathbf{g}^1}{\partial W_2^1 \partial W_1^1}\right\|_{2,2,1} = \sup_{\|V_1\|_F=\|V_2\|_F=1}\frac{1}{\sqrt{mn}}\sum_{i=1}^{m}\left|\sigma''\left(\tilde{\mathbf{x}}_i^l\right)\left(V_1\mathbf{x}_i^0\right)_i\left(V_2\mathbf{y}\right)_i\right| \le \frac{1}{2\sqrt{mn}}\beta_\sigma\left(\left\|\mathbf{x}^0\right\|^2+\|\mathbf{y}\|^2\right) \le \sqrt{\frac{m}{4n}}\beta_\sigma C_x^2+\sqrt{\frac{n}{4m}}\beta_\sigma C_y^2 = O(1).$$

Substituting the above bounds in (19) implies $\left\|\frac{\partial^2 \mathbf{g}^1}{(\partial W^1)^2}\right\|_{2,2,1} = O(1)$.

Similarly, for ADMM-CSNet, the equivalent second-order partial derivatives are

$$\left(\frac{\partial^2 \mathbf{g}^1}{\left(\partial W_1^1\right)^2}\right)_{i,jj',kk'} = \frac{1}{n}\sigma''\left(\tilde{\mathbf{z}}_i^1\right)\mathbf{y}_{j'}\mathbf{y}_{k'}\mathbb{I}_{i=k=j},$$

$$\left(\frac{\partial^2 \mathbf{g}^1}{\left(\partial W_2^1\right)^2}\right)_{i,jj',kk'} = \frac{1}{m}\sigma''\left(\tilde{\mathbf{z}}_i^1\right)(\mathbf{z}^0-\mathbf{u}^0)_{j'}(\mathbf{z}^0-\mathbf{u}^0)_{k'}\mathbb{I}_{i=k=j},$$

$$\left(\frac{\partial^2 \mathbf{g}^1}{\partial W_2^1 \partial W_1^1}\right)_{i,jj',kk'} = \frac{1}{\sqrt{mn}}\sigma''\left(\tilde{\mathbf{z}}_i^1\right)(\mathbf{z}^0-\mathbf{u}^0)_{j'}\mathbf{y}_{k'}\mathbb{I}_{i=k=j}.$$

The corresponding $(2,2,1)$-norm bounds are

$$\left\|\frac{\partial^2 \mathbf{g}^1}{\left(\partial W_1^1\right)^2}\right\|_{2,2,1} \le \frac{1}{2n}\beta_\sigma\left(\|\mathbf{y}\|^2+\|\mathbf{y}\|^2\right) \le \beta_\sigma C_y^2 = O(1),$$

$$\left\|\frac{\partial^2 \mathbf{g}^1}{\left(\partial W_2^1\right)^2}\right\|_{2,2,1} \le \frac{1}{2m}\beta_\sigma\left(2mC_z^2+2mC_u^2\right) = O(1),$$

$$\left\|\frac{\partial^2 \mathbf{g}^1}{\partial W_1^1 \partial W_2^1}\right\|_{2,2,1} \le \beta_\sigma\sqrt{\frac{m}{4n}}\left(C_y^2+(C_z+C_u)^2\right) = O(1).$$

Using these bounds, we get $\left\|\frac{\partial^2 \mathbf{g}^1}{(\partial W^1)^2}\right\|_{2,2,1} = O(1)$. From the above analysis, we conclude that the $(2,2,1)$-norm of the tensor, $\frac{\partial^2 \mathbf{g}^1}{(\partial W^1)^2}$, is of the order of $O(1)$ and the ∞-norm of the vector, $\frac{\partial f_s}{\partial \mathbf{g}^1}$, is of the order of $O\left(\frac{1}{\sqrt{m}}\right)$. This implies,

$$\|H_s\| = O\left(\frac{1}{\sqrt{m}}\right) \text{ and } \|\mathbf{H}\| = \Omega\left(\frac{1}{\sqrt{m}}\right) = O\left(\sqrt{m}\right). \tag{20}$$

Therefore, the Hessian spectral norm of a 1-layer LISTA or ADMM-CSNet depends on the width (dimension of the target vector) of the network. We now generalize the above analysis for an L-layer unfolded network.

2) Analysis of L-Layer Unfolded Network: The Hessian matrix of an L-layer unfolded ISTA or ADMM network for a given i^{th} training sample is written as

$$[\mathbf{H}]_{m\times P\times P} = \begin{bmatrix} H_1 & H_2 & \cdots & H_m \end{bmatrix}, \tag{21}$$

where H_s for $s \in [m]$ is

$$[H_s]_{P\times P} = \begin{bmatrix} H_s^{1,1} & H_s^{1,2} & \cdots & H_s^{1,L} \\ H_s^{2,1} & H_s^{2,2} & \cdots & H_s^{2,L} \\ \vdots & \vdots & \ddots & \vdots \\ H_s^{L,1} & H_s^{L,2} & \cdots & H_s^{L,L} \end{bmatrix}, \tag{22}$$

$\left[H_s^{l_1,l_2}\right]_{P_1\times P_1} = \frac{\partial^2 f_s}{\partial \mathbf{w}^{l_1}\partial \mathbf{w}^{l_2}}$, where $P_1 = m^2+mn$, $l_1 \in [L]$, $l_2 \in [L]$, $\mathbf{w}^l = \text{Vec}(W^l) = \text{Vec}\left([W_1^l\ W_2^l]\right)$ denotes the weights of l^{th}-layer, and $f_s = \frac{1}{\sqrt{m}}\mathbf{v}_s^T\mathbf{g}^L$. From (21) and (22), the spectral norm of $\mathbf{H}$, $\|\mathbf{H}\|$, is bounded by its block-wise spectral norm, $\|H_s\|$, as stated in the following theorem:

Theorem 3. *The Hessian spectral norm, $\|\mathbf{H}\|$, of an L-layer unfolded ISTA (ADMM) network, defined as in* (13) *(*(14)*), is bounded as $\max_{s\in[m]}\{\|H_s\|\} \le \|\mathbf{H}\| \le \sum_{s\in[m]}\|H_s\|$, where*

$$\|H_s\| \le \sum_{l_1,l_2}\left\|H_s^{l_1,l_2}\right\| \le \sum_{l_1,l_2} C_1\mathcal{Q}_{2,2,1}(f_s)\,\mathcal{Q}_\infty(f_s) \le C\mathcal{Q}_{2,2,1}(f_s)\,\mathcal{Q}_\infty(f_s). \tag{23}$$

The constant C_1 depends on L and L_σ, $C = L^2C_1$,

$$\mathcal{Q}_\infty(f_s) = \max_{1\le l\le L}\left\{\left\|\frac{\partial f_s}{\partial \mathbf{g}^l}\right\|_\infty\right\}, \text{ and} \tag{24}$$

$$\mathcal{Q}_{2,2,1}(f_s) = \max_{1\le l_1\le l_2<l_3\le L}\left\{\left\|\frac{\partial^2 \mathbf{g}^{l_1}}{\left(\partial \mathbf{w}^{l_1}\right)^2}\right\|_{2,2,1}, \left\|\frac{\partial \mathbf{g}^{l_1}}{\partial \mathbf{w}^{l_1}}\right\|\left\|\frac{\partial^2 \mathbf{g}^{l_2}}{\partial \mathbf{g}^{(l_2-1)}\partial \mathbf{w}^{l_2}}\right\|_{2,2,1}, \left\|\frac{\partial \mathbf{g}^{l_1}}{\partial \mathbf{w}^{l_1}}\right\|\left\|\frac{\partial \mathbf{g}^{l_2}}{\partial \mathbf{w}^{l_2}}\right\|\left\|\frac{\partial^2 \mathbf{g}^{l_3}}{\left(\partial \mathbf{g}^{l_3-1}\right)^2}\right\|_{2,2,1}\right\}. \tag{25}$$

Proof of the above theorem is given in the Appendix. Similar to 1-layer case, the bound on $\|\mathbf{H}\|$ depends on the ∞-norms of $\frac{\partial f_s}{\partial \mathbf{g}^l}$, $l \in [L]$ and $(2,2,1)$-norms of layer-wise derivatives (basically these are order 3 tensors). We now aim to derive

the bounds on the quantities $\mathcal{Q}_{2,2,1}(f_s)$ and $\mathcal{Q}_{\infty}(f_s)$ for both unfolded ISTA and ADMM networks.

Similar to Lemma 2 and 3, the Gaussian initialization of the weight matrices imposes a bound on the hidden layer output of the unfolded network, which is stated in the following lemma:

Lemma 4. *If* $\left(W_{10}^l\right)_{i,j} \sim \mathcal{N}(0,1)$ *and* $\left(W_{20}^l\right)_{i,j} \sim \mathcal{N}(0,1)$, $\forall l \in [L]$, *then for any* $\mathbf{W}_1 \in B(\mathbf{W}_{10}, R_1)$ *and* $\mathbf{W}_2 \in B(\mathbf{W}_{20}, R_2)$, *we have* $\left\|\mathbf{x}^l\right\| \leq c^l_{\mathrm{ISTA};\mathbf{x}}$ *for LISTA, and* $\left\|\mathbf{z}^l\right\| \leq c^l_{\mathrm{ADMM};\mathbf{z}}$ *and* $\left\|\mathbf{u}^l\right\| \leq c^l_{\mathrm{ADMM};\mathbf{u}}$ *for ADMM-CSNet. The updating rules are*

$$\begin{aligned} c^l_{\mathrm{ISTA};\mathbf{x}} &= L_\sigma\left(c_{10}+\frac{R_1}{\sqrt{n}}\right)\sqrt{n}C_y + L_\sigma\left(c_{20}+\frac{R_2}{\sqrt{m}}\right)c^{l-1}_{\mathrm{ISTA};\mathbf{x}} + \sigma(0) \\ &= O\left(\sqrt{m}\right) \\ c^l_{\mathrm{ADMM};\mathbf{z}} &= L_\sigma\left(c_{10}+\frac{R_1}{\sqrt{n}}\right)\sqrt{n}C_y + L_\sigma\left(c_{20}+\frac{R_2}{\sqrt{m}}\right)c^{l-1}_{\mathrm{ADMM};\mathbf{z}} \\ &\quad + L_\sigma\left(1+c_{20}+\frac{R_2}{\sqrt{m}}\right)c^{l-1}_{\mathrm{ADMM};\mathbf{u}} + \sigma(0) = O\left(\sqrt{m}\right), \\ c^l_{\mathrm{ADMM};\mathbf{u}} &= \left(c_{10}+\frac{R_1}{\sqrt{n}}\right)\sqrt{n}C_y + \left(c_{20}+\frac{R_2}{\sqrt{m}}\right)c^{l-1}_{\mathrm{ADMM};\mathbf{z}} \\ &\quad + \left(c_{20}+\frac{R_2}{\sqrt{m}}+1\right)c^{l-1}_{\mathrm{ADMM};\mathbf{u}} + c^l_{\mathrm{ADMM};\mathbf{z}} = O\left(\sqrt{m}\right), \end{aligned}$$

where $c^0_{\mathrm{ISTA};\mathbf{x}} = \sqrt{m}C_x$, $c^0_{\mathrm{ADMM};\mathbf{z}} = \sqrt{m}C_z$, $c^0_{\mathrm{ADMM};\mathbf{u}} = \sqrt{m}C_u$, $|x^0_i| \leq C_x$, $|u^0_i| \leq C_u$, *and* $|z^0_i| \leq C_z$, $\forall i \in [m]$.

Refer to the Appendix for proof of the above lemma. The three updating rules in Lemma 4 are of the order of $\sqrt{m}$ and $\sqrt{n}$ w.r.t. m and n, respectively. However, as the width of the unfolded network is controlled by m, we consider the bounds on $\mathcal{Q}_{2,2,1}(f_s)$ and $\mathcal{Q}_{\infty}(f_s)$ w.r.t. m in this work.

The following theorem gives the bound on $\|\mathbf{H}\|$ by deriving the bounds on the quantities $\mathcal{Q}_{2,2,1}(f_s)$ and $\mathcal{Q}_{\infty}(f_s)$. The proof of Theorem 4 basically uses the bounds on the weight matrices (Lemma 2 and Lemma 3), bound on the hidden layer output (Lemma 4), and properties of the activation function (L_σ-Lipschitz continuous and β_σ-smooth).

Theorem 4. *Consider an* L*-layer unfolded ISTA or ADMM network,* $\mathbf{F}(\mathbf{W})$*, with random i.i.d. Gaussian initialization* $\mathbf{W}_0$*. Then, the quantities* $\mathcal{Q}_{2,2,1}(f_s)$ *and* $\mathcal{Q}_{\infty}(f_s)$ *satisfy the following equality w.r.t.* m*, over initialization, at any point* $\mathbf{W} \in B(\mathbf{W}_0, R)$*, for some fixed* $R > 0$*:*

$$\mathcal{Q}_{2,2,1}(f_s) = O(1) \text{ and } \mathcal{Q}_{\infty}(f_s) = \tilde{O}\left(\frac{1}{\sqrt{m}}\right), \tag{26}$$

with probabilities 1 *and* $1 - me^{-c\ln^2(m)}$ *for some constant* $c > 0$*, respectively. This implies*

$$\|H_s\| \leq \sum_{l_1,l_2}\left\|H_s^{l_1,l_2}\right\| = \tilde{O}\left(\frac{1}{\sqrt{m}}\right) \tag{27}$$

and the Hessian spectral norm satisfies

$$\|\mathbf{H}\| = \tilde{\Omega}\left(\frac{1}{\sqrt{m}}\right) = \tilde{O}\left(\sqrt{m}\right). \tag{28}$$

The proof of Theorem 4 is motivated by [36]. Readers are directed to the supplementary material [55], which provides the complete proof.

In summary, from both 1-layer and L-layer analyses, we claim that the Hessian spectral norm bound of an unfolded network is proportional to the square root of the width of the network.

Note that the aforementioned analysis assumed λ to be a fixed constant value. Nonetheless, the analysis can be readily extended to accommodate a learnable λ. It can be verified that the Hessian spectral norm remains within the same order even when λ is treated as a learnable parameter.

C. Conditions on Unfolded Networks to Satisfy PL*

From Theorem 1, the Hessian spectral norm of a model should hold the following condition to satisfy μ-uniform conditioning in a ball $B(\mathbf{w}_0, R)$: $\|\mathbf{H}_{\mathcal{F}}(\mathbf{w})\| \leq \frac{\lambda_0-\mu}{2L_{\mathcal{F}}\sqrt{T}R}$, $\forall \mathbf{w} \in B(\mathbf{w}_0, R)$. Since $\|\mathbf{H}_{\mathcal{F}}(\mathbf{w})\| = \max_{i\in[T]}\|\mathbf{H}_{\mathcal{F}_i}(\mathbf{w})\|$, the above condition can be further simplified as

$$\|\mathbf{H}_{\mathcal{F}_i}(\mathbf{w})\| \leq \frac{\lambda_0-\mu}{2L_{\mathcal{F}}\sqrt{T}R}, \ \forall i \in [T] \text{ and } \mathbf{w} \in B(\mathbf{w}_0, R). \tag{29}$$

Substituting the Hessian spectral norm bound of LISTA and ADMM-CSNet, stated in Theorem 4, in (29) provides a constraint on the network width such that the square loss function satisfies the μ-PL* condition in $B(\mathbf{w}_0, R)$:

$$m = \tilde{\Omega}\left(\frac{TR^2}{(\lambda_0-\mu)^2}\right), \text{ where } \mu \in (0, \lambda_0). \tag{30}$$

Therefore, from Theorem 2, we claim that for a given fixed T one should consider the width of the unfolded network as given in (30) to achieve near-zero training loss. However, the m (target vector dimension) value is generally fixed for a given linear inverse problem. Hence, we provide the constraint on T instead of m. Substituting the $\|\mathbf{H}_{\mathcal{F}_i}(\mathbf{w})\|$ bound in (29) also provides a threshold on T, which is summarized in the following theorem:

Theorem 5. *Consider a finite* L*-layer unfolded network as given in* (13) *or* (14) *with* m *as the network width. Assume that the model is well-conditioned at initialization, i.e.,* $\lambda_{min}(K_{Unfolded}(\mathbf{w}_0)) = \lambda_{0,Unfolded}$*, for some* $\lambda_{0,Unfolded} > 0$*. Then, the loss landscape corresponding to the square loss function satisfies the* μ*-PL** *condition in a ball* $B(\mathbf{w}_0, R)$*, if the number of training samples,* $T_{Unfolded}$*, satisfies the following condition:*

$$T_{Unfolded} = \tilde{O}\left(\frac{m(\lambda_{0,Unfolded}-\mu)^2}{R^2}\right), \ \mu \in (0, \lambda_{0,Unfolded}). \tag{31}$$

Thus, while addressing a linear inverse problem using unfolded networks, one should consider the number of training samples as given in (31), to obtain zero training loss as the number of GD epochs increases to infinity. Observe that the threshold on T increases with the increase in the network width. We attribute this to the fact that a high network width is associated with more trainable parameters in the network, which provides the ability to handle/memorize more training samples. Conversely, a smaller network width leads to fewer trainable parameters, thereby impacting the network's performance in handling training samples.

Comparison with FFNN: In [31], the authors computed the Hessian spectral norm of an FFNN with a scalar output, which is of the order of $\tilde{O}\left(\frac{1}{\sqrt{m}}\right)$. Following the analysis procedure of an m-output model given in Section IV-B, one can obtain the Hessian spectral norm of an FFNN with m-output and smoothed soft-thresholding non-linearity as given below:

$$\|\mathbf{H}\| = \tilde{\Omega}\left(\frac{1}{\sqrt{m}}\right) = \tilde{O}\left(\sqrt{m}\right). \tag{32}$$

This implies that the bound on the number of training samples, T_{FFNN}, for an m-output FFNN to satisfy the μ-PL* is

$$T_{\text{FFNN}} = \tilde{O}\left(\frac{m(\lambda_{0,\text{FFNN}} - \mu)^2}{R^2}\right), \ \mu \in (0, \lambda_{0,\text{FFNN}}) \tag{33}$$

Note that m is a fixed value in both (31) and (33), R is of the order of $O\left(\frac{1}{\mu}\right)$ (refer to Theorem 2), and μ depends on $\lambda_0 = \lambda_{\min}\left(K\left(\mathbf{w}_0\right)\right)$. Therefore, from (31) and (33), the parameter that governs the number of training samples of a network is the minimum eigenvalue of the tangent kernel matrix at initialization. Hence, we compare both T_{Unfolded} and T_{FFNN} by deriving the upper bounds on $\lambda_{0,\text{Unfolded}}$ and $\lambda_{0,\text{FFNN}}$. Specifically, in the following theorem, we show that the upper bound of $\lambda_{0,\text{Unfolded}}$ is higher compared to $\lambda_{0,\text{FFNN}}$.

Theorem 6. *Consider an L-layered FFNN, defined as*

$$\mathbf{f}_{FFNN} = \frac{1}{\sqrt{m}}\mathbf{x}^L, \mathbf{x}^l = \sigma\left(\frac{W^l}{\sqrt{m}}\mathbf{x}^{l-1}\right) \in \mathbb{R}^m, \ l \in [L], \tag{34}$$

with $\mathbf{x}^0 = \sqrt{\frac{m}{n}}\mathbf{y} \in \mathbb{R}^n$, $W^1 \in \mathbb{R}^{m\times n}$, *and* $W^l \in \mathbb{R}^{m\times m} \ \forall l \in [L] - \{1\}$. *Also, consider the unfolded network defined in* (13) *or* (14). *Then, the upper bound on the minimum eigenvalue of the tangent kernel matrix at initialization for unfolded network,* $UB_{Unfolded}$ *(either* UB_{LISTA} *or* $UB_{ADMM\text{-}CSNet}$*), is greater than that of FFNN,* UB_{FFNN}*, i.e.,* $UB_{Unfolded} > UB_{FFNN}$.

Proof of the above theorem is given in the Appendix. To better understand Theorem 6, substitute $L = 2$ in equations (38), (39), and (40). This leads to

$$\text{UB}_{\text{FFNN}} = \hat{L}^4\hat{y}\left[\|W_0^1\|^2 + \|\mathbf{v}_s^T W_0^2\|^2\right],$$

$$\text{UB}_{\text{LISTA}} = \hat{L}^4\hat{y}\left[\|W_{10}^1\|^2 + \|\mathbf{v}_s^T W_{20}^2\|^2\right] + \hat{L}^2\hat{y} + \hat{L}^4\hat{x}\left[\|W_{20}^1\|^2 + \|\mathbf{v}_s^T W_{20}^2\|^2\right] + 2\hat{L}^4\sqrt{\hat{x}\hat{y}}\|W_{10}^1\|\|W_{20}^1\|,$$

and

$$\text{UB}_{\text{ADMM-CSNet}} = \hat{L}^4\hat{y}\left[\|W_{10}^1\|^2 + \|\mathbf{v}_s^T W_{20}^2\|^2\right] + \hat{L}^2\hat{y} + \frac{\|\mathbf{u}^{(1)}\|^2}{m} + \hat{L}^4\hat{a}^{(0)}\left[\|W_{20}^1\|^2 + \|\mathbf{v}_s^T W_{20}^2\|^2\right] + 2\hat{L}\|\tilde{\mathbf{z}}^{(l)}\|\|\mathbf{u}^{(1)}\| + \hat{L}^4\|\mathbf{u}^{(0)}\|^2 + \hat{L}^4\left[2\sqrt{\hat{y}\hat{a}^{(0)}}\|W_{10}^1\|\|W_{20}^1\| + 2\sqrt{\hat{a}^{(0)}}\|W_{20}^1\|\|\mathbf{u}^{(0)}\| + 2\sqrt{\hat{y}}\|W_{10}^1\|\|\mathbf{u}^{(0)}\|\right].$$

Since the dimension of W_1^1 (W_2^2) of unfolded networks is the same as W^1 (W^2) of FFNN, we conclude that $\text{UB}_{\text{Unfolded}} > \text{UB}_{\text{FFNN}}$ for $L = 2$. One can verify that this relation holds for any L value using the generalized expressions given in (38), (39), and (40).

Figures 5 (a) and 5 (b) depict the variation of $10\log_{10}\left(\lambda_{\min}\left(K(\mathbf{w}_0)\right)\right)$ w.r.t. L (here we considered $T = 10$, $m = 100$, $n = 20$, and $k = 2$) and P (here we vary m, n, and k values by fixing $T = 10$, $L = 6$ for unfolded, and $L = 8$ for FFNN), respectively, for LISTA, ADMM-CSNet, and FFNN. From these figures, we see that $\lambda_{0,\text{Unfolded}} > \lambda_{0,\text{FFNN}}$. Consequently, from Theorem 6, (31), and (33), we also claim that the upper bound of T_{Unfolded} is higher compared to T_{FFNN}. As a result, $T_{\text{Unfolded}} > T_{\text{FFNN}}$ whenever $\lambda_{0,\text{Unfolded}} > \lambda_{0,\text{FFNN}}$. Moreover, from the aforementioned equations, it is evident that $\text{UB}_{\text{ADMM-CSNet}}$ exceeds UB_{LISTA}. Consequently, it is reasonable to anticipate that $\lambda_{0,\text{ADMM-CSNet}}$ will surpass $\lambda_{0,\text{LISTA}}$. This inference is substantiated by the data depicted in figures 5 (a) and 5 (b). This implies that the upper bound on $T_{\text{ADMM-CSNet}}$ exceeds the upper bound on T_{LISTA}. Through simulations, we show that $T_{\text{ADMM-CSNet}} > T_{\text{LISTA}} > T_{\text{FFNN}}$ in the following section. Since the threshold on T — guaranteeing memorization — is higher for unfolded networks than FFNN, we should obtain a better expected error, which is upper bounded by the sum of generalization and training error [37], for unfolded networks than FFNN for a given T value such that $T_{\text{FFNN}} < T \leq T_{\text{Unfolded}}$; in such scenarios, unfolded networks exhibit zero training error and a smaller generalization error [18].

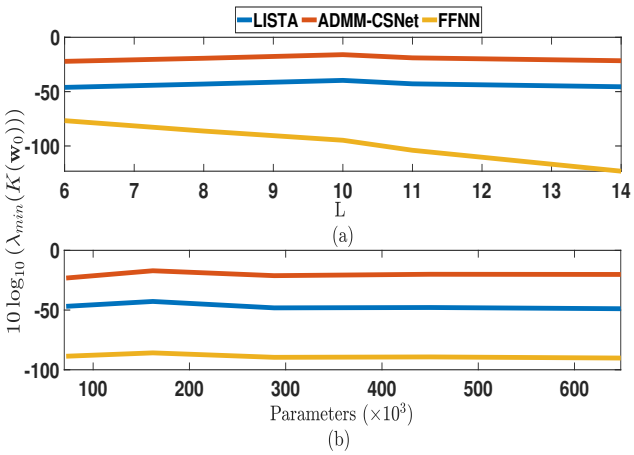

Fig. 5: Variation of the minimum eigenvalue of tangent kernel matrix at initialization: (a) With respect to the number of layers. (b) With respect to the network learnable parameters.

V. Numerical Experiments

We perform the following simulations to support the proposed theory. For all the simulations in this section, we fix the following for LISTA, ADMM-CSNet, and FFNN: 1. Parameters are i.i.d. Guassian initialized with zero mean and unit variance, i.e., $\mathcal{N}(0, 1)$. 2. Networks are trained with the aim of minimizing the square loss function (12) using stochastic GD. Note that the theoretical analysis proposed in this work is for GD, however, to address the computation and storage issues, we considered stochastic GD for the numerical analysis. 3. Modified soft-plus activation function (refer to IV-A) with $\lambda = 1$ is used as the non-linear activation function. 4. A batch size of $\frac{T}{5}$ is considered. 5. All the simulations are repeated for 10 trials.

This article has been accepted for publication in IEEE Transactions on Signal Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSP.2024.3412981

Threshold on T: From (31), the choice of T plays a vital role in achieving near-zero training loss. To illustrate this, consider two linear inverse models: $\mathbf{y}_1 = A_1\mathbf{x}_1 + \mathbf{e}_1$ and $\mathbf{y}_2 = A_2\mathbf{x}_2 + \mathbf{e}_2$, where $\mathbf{y}_1 \in \mathbb{R}^{20\times 1}$, $\mathbf{x}_1 \in \mathbb{R}^{100\times 1}$, $A_1 \in \mathbb{R}^{20\times 100}$, $\|\mathbf{x}_1\|_0 = 2$, $\mathbf{y}_2 \in \mathbb{R}^{200\times 1}$, $\mathbf{x}_2 \in \mathbb{R}^{1000\times 1}$, $A_2 \in \mathbb{R}^{200\times 1000}$, and $\|\mathbf{x}_2\|_0 = 10$. Generate synthetic data using a random linear operator matrix, which follows the i.i.d. uniform distribution, and then normalize it to ensure $\|A_1\|_F = \|A_2\|_F = 10$. Both models are subjected to Gaussian noise ($\mathbf{e}_1$ and $\mathbf{e}_2$) with a signal-to-noise ratio (SNR) of 10 dB. Here, we generated the data by following standard CS theory for model priors such as the sparsity of the target vector (k) and linear operator matrix (A). Once the data is generated, the following numerical analysis is independent of the prior values.

Construct an L-layer LISTA and ADMM-CSNet with $L = 11$. Here, we train LISTA for 30K epochs and ADMM-CSNet for 40K epochs. For the first model, we choose 0.12 and 0.09 as learning rates for LISTA and ADMM-CSNet, respectively. For the second model, we choose 1.2 for LISTA and 0.9 for ADMM-CSNet. Figures 6 and 7 depict the variation of mean square loss/error (MSE) w.r.t. T for both LISTA and ADMM-CSNet, respectively. For a fixed m there exists a threshold (by considering a specific MSE value) on T such that choosing a T value that is less than this threshold leads to near-zero training loss. Notably, this threshold is high for ADMM-CSNet compared to LISTA. Moreover, observe that this threshold increases as the network width grows.

For comparison, construct an L-layer FFNN, to recover $\mathbf{x}_1$ and $\mathbf{x}_2$, that has the same number of parameters as that of unfolded, hence, we choose $L = 14$. Here, we train the network for $40K$ epochs with a learning rate of 0.04 for the first model and 0.3 for the second model. Fig. 8 shows the variation of MSE w.r.t. T. From Fig. 8, we conclude that the threshold for FFNN is lower compared to LISTA and ADMM-CSNet.

To assess the generalization capacity of our proposed theory, we consider the aforementioned first model with a different setup. Specifically, we employ the modified sigmoid linear unit (SiLU), defined as $\sigma_\lambda(x) = \frac{x-\lambda}{1+e^{x-\lambda}} - \frac{x-\lambda}{1+e^{-x-\lambda}}$, as a smooth approximation to soft-thresholding activation. In Fig. 9, we illustrate the variation of MSE w.r.t. T for both LISTA and ADMM-CSNet. LISTA is trained for 30K epochs, while ADMM-CSNet is trained for 40K epochs. ADMM-CSNet employs a learning rate of 0.95, whereas LISTA utilizes different learning rates $(17, 11, 1, 0.9)$ corresponding to different T values $(10-30, 50, 70-90, >100)$, respectively, to achieve near-zero training loss. Observe that the threshold on the number of training samples is still high for ADMM-CSNet compared to LISTA. This justifies the generalization ability of the proposed theory. Additionally, we noted that the training error of FFNN fails to converge under this configuration, suggesting its inability to effectively memorize the provided training data.

Comparison Between Unfolded and Standard Networks: We compare LISTA and ADMM-CSNet with FFNN in terms of parameter efficiency. To demonstrate this, consider the first linear inverse model given in the above simulation. Then,

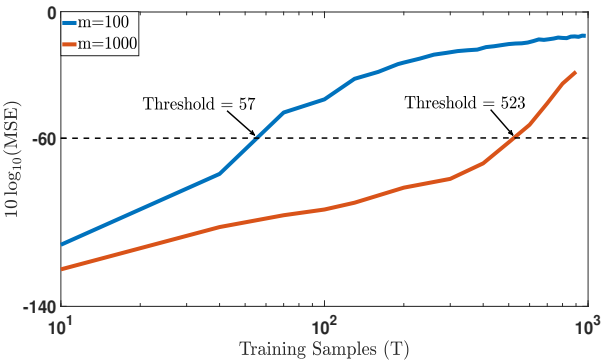

Fig. 6: Training loss vs T for LISTA.

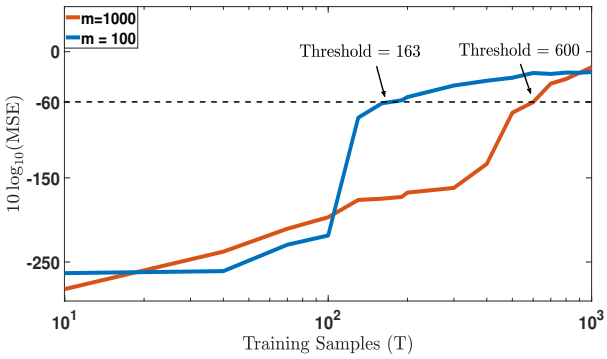

Fig. 7: Training loss vs T for ADMM-CSNet.

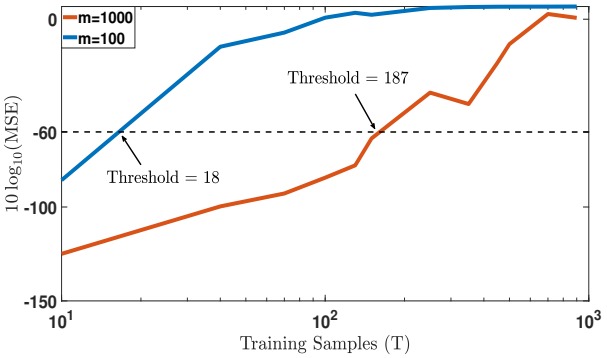

Fig. 8: Training loss vs T for FFNN.

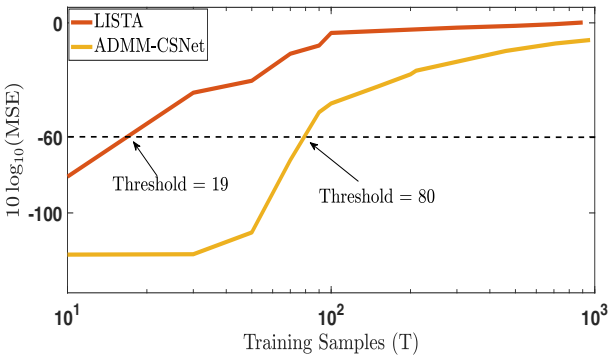

Fig. 9: Training loss vs T for both LISTA and ADMM-CSNet by considering the modified SiLU as an activation function.

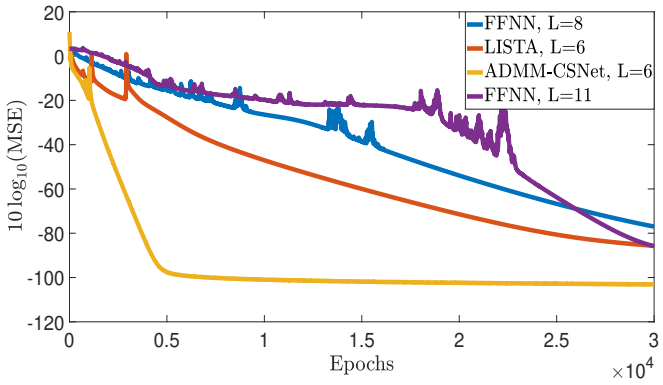

Fig. 10: Comparison between LISTA, ADMM-CSNet, and FFNN in terms of the required number of parameters, P, for training loss convergence.

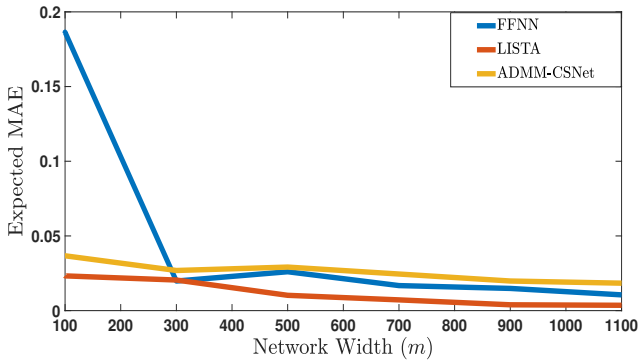

Fig. 11: Variation of the expected MAE w.r.t. m for both LISTA and ADMM-CSNet.

construct LISTA, ADMM-CSNet, and FFNN with a fixed number of parameters and consider $T = 30$. Also, consider the same learning rates that are associated with the first model in the above simulation for LISTA, ADMM-CSNet, and FFNN. Here we choose $L = 6$ for both LISTA and ADMM-CSNet, and $L = 8$ for FFNN, resulting in a total of $72K$ parameters. As shown in Fig. 10, the convergence of training loss to zero is better for LISTA and ADMM-CSNet compared to FFNN. Fig. 10 also shows the training loss convergence of FFNN with $L = 11$. Now, FFNN has $102K$ learnable parameters, and its performance is comparable to LISTA for higher epoch values. Therefore, to achieve a better training loss FFNN requires more trainable parameters.

Generalization: In this simulation, we show that zero-training error leads to better generalization. To demonstrate this, consider LISTA/ADMM-CSNet/FFNN with a fixed T and observe the variation of the expected mean absolute error (MAE) w.r.t. m. If the generalization performance is better, then it is anticipated that the expected MAE reduces as the m increases. Because an increase in m improves the possibility of getting near-zero training loss for a fixed T. In Fig. 11, we present the results for LISTA, ADMM-CSNet, and FFNN with $T = 100$. Notably, the expected MAE diminishes as m increases, i.e., as the number of parameters grows. Further, it is observed that for this choice of T, the training error is near-zero for m values exceeding approximately 300 for FFNN, and approximately 250 for both LISTA and ADMM-CSNet. This finding underscores the importance of zero-training error in generalization.

However, it is important to note that the generalization results presented here are preliminary and require a rigorous analysis for more robust conclusions; because considering a smaller value of T may not yield satisfactory generalization performance. Thus, it is important to find a lower bound on T to optimize both the training process and overall generalization capability.

VI. Conclusion

In this work, we provided optimization guarantees for finite-layer LISTA and ADMM-CSNet with smooth nonlinear activation. We begin by deriving the Hessian spectral norm of these unfolded networks. Based on this, we provided conditions on both the network width and the number of training samples, such that the empirical training loss converges to zero as the number of learning epochs increases using the GD approach. Additionally, we showed that LISTA and ADMM-CSNet outperform the standard FFNN in terms of threshold on the number of training samples and parameter efficiency. We provided simulations to support the theoretical findings.

The work presented in this paper is an initial step to understand the theory behind the performance of unfolded networks. While considering certain assumptions, our work raises intriguing questions for future research. For instance, we approximated the soft-threshold activation function by devising a doubly differentiable function using soft-plus. However, it is important to analyze the optimization guarantees without relying on any such approximations. A promising avenue for investigation is the Gram-matrix-based analysis proposed in [32]. Additionally, we assumed a constant value for λ in $\sigma_\lambda(\cdot)$. It is interesting to explore the impact of treating λ as a learnable parameter. Furthermore, analyzing the changes in the analysis for other loss functions presents an intriguing avenue for further research.

Appendix

Proof of ***Theorem*** *3:* The Hessian block $H_s^{l_1,l_2}$ can be decomposed as given in (35), using the following chain rule:

$$\frac{\partial f_s}{\partial \mathbf{w}^l} = \frac{\partial \mathbf{g}^l}{\partial \mathbf{w}^l}\left(\prod_{l'=l+1}^{L}\frac{\partial \mathbf{g}^l}{\partial \mathbf{g}^{l'-1}}\right)\frac{\partial f_s}{\partial \mathbf{g}^L}.$$

$$H_s^{l_1,l_2} = \frac{\partial^2 \mathbf{g}^{l_1}}{\left(\partial \mathbf{w}^{l_1}\right)^2}\frac{\partial f_s}{\partial \mathbf{g}^{l_1}}\mathbb{I}_{l_1=l_2} + \left(\frac{\partial \mathbf{g}^{l_1}}{\partial \mathbf{w}^{l_1}}\prod_{l'=l_1+1}^{l_2-1}\frac{\partial \mathbf{g}^{l'}}{\partial \mathbf{g}^{l'-1}}\right)\frac{\partial^2 \mathbf{g}^{l_2}}{\partial \mathbf{w}^{l_2}\partial \mathbf{g}^{l_2-1}}\left(\frac{\partial f_s}{\partial \mathbf{g}^{l_2}}\right) + \sum_{l=l_2+1}^{L}\left(\frac{\partial \mathbf{g}^{l_1}}{\partial \mathbf{w}^{l_1}}\prod_{l'=l_1+1}^{l-1}\frac{\partial \mathbf{g}^{l'}}{\partial \mathbf{g}^{l'-1}}\right)\frac{\partial^2 \mathbf{g}^{l'}}{\left(\partial \mathbf{g}^{l'-1}\right)^2}\left(\frac{\partial \mathbf{g}^{l_2}}{\partial \mathbf{w}^{l_2}}\prod_{l'=l_2+1}^{l}\frac{\partial \mathbf{g}^{l'}}{\partial \mathbf{g}^{l'-1}}\right)\left(\frac{\partial f_s}{\partial \mathbf{g}^l}\right). \quad (35)$$

From (35), the spectral norm of $H_s^{l_1,l_2}$ can be bounded as

$$\begin{aligned}\left\|H_s^{l_1,l_2}\right\|_2 \leq & \left\|\frac{\partial^2 \mathbf{g}^{l_1}}{\left(\partial \mathbf{w}^{(l_1)}\right)^2}\right\|_{2,2,1}\left\|\frac{\partial f_s}{\partial \mathbf{g}^{l_1}}\right\|_{\infty}+L_\sigma^{l_2-l_1-1}\left\|\frac{\partial \mathbf{g}^{l_1}}{\partial \mathbf{w}^{l_1}}\right\|_F \\ & \left\|\frac{\partial^2 \mathbf{g}^{l_2}}{\partial \mathbf{w}^{l_2}\partial \mathbf{g}^{l_2-1}}\right\|_{2,2,1}\left\|\frac{\partial f_s}{\partial \mathbf{g}^{l_2}}\right\|_{\infty}+\sum_{l=l_2+1}^{L} L_\sigma^{2l-l_1-l_2}\left\|\frac{\partial \mathbf{g}^{l_1}}{\partial \mathbf{w}^{l_1}}\right\|_F \\ & \left\|\frac{\partial^2 \mathbf{g}^{l}}{\left(\partial \mathbf{g}^{l'-1}\right)^2}\right\|_{2,2,1}\left\|\frac{\partial \mathbf{g}^{l_2}}{\partial \mathbf{w}^{l_2}}\right\|_F\left\|\frac{\partial f_s}{\partial \mathbf{g}^{l}}\right\|_{\infty}.\end{aligned} \tag{36}$$

Note that (36) uses the fact that $\left\|\frac{\partial \mathbf{g}^{l'}}{\partial \mathbf{g}^{l'-1}}\right\|_F \leq L_\sigma$. By using the notations given in (24) and (25), we get

$$\left\|H_s^{l_1,l_2}\right\| \leq C_1 \mathcal{Q}_{2,2,1}\left(f_s\right) \mathcal{Q}_{\infty}\left(f_s\right),$$

where C_1 is a constant depend on L and L_σ. □

*Proof of **Lemma 4**:* For $l=0$, $\|\mathbf{x}^0\| \leq \sqrt{m}\|\mathbf{x}^0\|_\infty \leq \sqrt{m}C_x$, $\|\mathbf{z}^0\| \leq \sqrt{m}\|\mathbf{z}^0\|_\infty \leq \sqrt{m}C_z$, and $\|\mathbf{u}^0\| \leq \sqrt{m}\|\mathbf{u}^0\|_\infty \leq \sqrt{m}C_u$. Whereas for $l=1,2,\ldots,L$, we have

$$\begin{aligned}\left\|\mathbf{x}^l\right\| &= \left\|\sigma\left(\frac{W_1^l}{\sqrt{n}}\mathbf{y}+\frac{W_2^l}{\sqrt{m}}\mathbf{x}^{l-1}\right)\right\| \\ &\leq L_\sigma\left\|\frac{W_1^l}{\sqrt{n}}\right\|\|\mathbf{y}\|+L_\sigma\left\|\frac{W_2^l}{\sqrt{m}}\right\|\left\|\mathbf{x}^{l-1}\right\|+\sigma(0) \\ &\leq L_\sigma\left(c_{10}+\frac{R_1}{\sqrt{n}}\right)\sqrt{n}C_{\mathbf{y}}+L_\sigma\left(c_{20}+\frac{R_2}{\sqrt{m}}\right)c_{\mathrm{ISTA};x}^{l-1}+\sigma(0) \\ &= c_{\mathrm{ISTA};\mathbf{x}}^{l}.\end{aligned}$$

Here, we used Lemma 3 and L_σ-Lipschitz continuous of the activation function $\sigma(\cdot)$. Similarly,

$$\begin{aligned}&\left\|\mathbf{z}^l\right\| = \left\|\sigma\left(\frac{1}{\sqrt{n}}W_1^l\mathbf{y}+\frac{1}{\sqrt{m}}W_2^l\left(\mathbf{z}^{l-1}-\mathbf{u}^{l-1}\right)+\mathbf{u}^{l-1}\right)\right\| \\ &\leq L_\sigma\frac{1}{\sqrt{n}}\left\|W_1^l\right\|\|\mathbf{y}\|+L_\sigma\frac{1}{\sqrt{m}}\left\|W_2^l\right\|\left\|\mathbf{z}^{l-1}\right\|+\frac{1}{\sqrt{m}}L_\sigma\left\|W_2^l\right\|\left\|\mathbf{u}^{l-1}\right\| \\ &\quad+L_\sigma\left\|\mathbf{u}^{l-1}\right\|+\sigma(0) \\ &\leq L_\sigma\left(c_{10}+\frac{R_1}{\sqrt{n}}\right)\sqrt{n}C_y+L_\sigma\left(c_{20}+\frac{R_2}{\sqrt{m}}\right)c_{\mathrm{ADMM};\mathbf{z}}^{l-1} \\ &\quad+L_\sigma\left(1+c_{20}+\frac{R_2}{\sqrt{m}}\right)c_{\mathrm{ADMM};\mathbf{u}}^{l-1}+\sigma(0) \\ &= c_{\mathrm{ADMM};\mathbf{z}}^{l}\end{aligned}$$

and

$$\begin{aligned}&\left\|\mathbf{u}^l\right\| = \left\|\mathbf{u}^{l-1}+\left(\frac{1}{\sqrt{n}}W_1^l\mathbf{y}+\frac{1}{\sqrt{m}}W_2^l\left(\mathbf{z}^{l-1}-\mathbf{u}^{l-1}\right)-\mathbf{z}^l\right)\right\| \\ &\leq \left\|\mathbf{u}^{l-1}\right\|+\left\|\frac{1}{\sqrt{n}}W_1^l\mathbf{y}\right\|+\left\|\frac{1}{\sqrt{m}}W_2^l\mathbf{z}^{l-1}\right\|+\left\|\frac{1}{\sqrt{m}}W_2^l\mathbf{u}^{l-1}\right\|+\left\|\mathbf{z}^l\right\| \\ &\leq \left(c_{10}+\frac{R_1}{\sqrt{n}}\right)\sqrt{n}C_y+\left(c_{20}+\frac{R_2}{\sqrt{m}}\right)c_{\mathrm{ADMM};\mathbf{z}}^{l-1} \\ &\quad+\left(c_{20}+\frac{R_2}{\sqrt{m}}+1\right)c_{\mathrm{ADMM};\mathbf{u}}^{l-1}+c_{\mathrm{ADMM};\mathbf{z}}^{l} \\ &= c_{\mathrm{ADMM};\mathbf{u}}^{l},\end{aligned}$$

completing the proof. □

*Proof of **Theorem 6:*** Consider the real symmetric NTK matrix $[K(\mathbf{w}_0)]_{mT\times mT}$. Utilizing the Rayleigh quotient of $K(\mathbf{w}_0)$, we can write the following for any $\mathbf{x}$ such that $\|\mathbf{x}\|_2=1$:

$$\lambda_{\min}\left(K\left(\mathbf{w}_0\right)\right) \leq \mathbf{x}^\top K\left(\mathbf{w}_0\right)\mathbf{x} \leq \lambda_{\max}(K\left(\mathbf{w}_0\right)).$$

Let $\mathbf{x}$ be a vector having all zeros except the s^{th} component to be 1. Thus $\lambda_{\min}(K\left(\mathbf{w}_0\right)) \leq [K\left(\mathbf{w}_0\right)]_{\mathrm{s,s}}$, for any $s \in [mT]$. Assume $s=1$, this implies,

$$\lambda_{\min}\left(K\left(\mathbf{w}_0\right)\right) \leq \left\langle \nabla_{\mathbf{w}_0}\mathbf{f}_1, \nabla_{\mathbf{w}_0}\mathbf{f}_1\right\rangle, \tag{37}$$

where $\mathbf{f}_1$ is the 1^{st} component in the the model output vector $\mathbf{f}$ corresponding to the first training sample. We now aim to compute $\left\langle \nabla_{\mathbf{w}_0}\mathbf{f}_1, \nabla_{\mathbf{w}_0}\mathbf{f}_1\right\rangle$ for FFNN, LISTA, and ADMM-CSNet.

Consider a one-layer FFNN, then from (34), the s^{th} component of $\mathbf{f}_{\text{FFNN}}$ is, $\mathbf{f}_s = \frac{1}{\sqrt{m}}\sigma\left(\frac{1}{\sqrt{n}}W_0^1(s,:)\mathbf{y}\right)$, where $W_0^1(s,:)$ represents the s^{th} row of W_0^1. This implies,

$$\left\langle \nabla_{W_0^1}\mathbf{f}_s, \nabla_{W_0^1}\mathbf{f}_s\right\rangle = \left[\frac{\sigma'(\tilde{\mathbf{x}}_s^1)}{\sqrt{mn}}\right]^2\|\mathbf{y}\|^2 \leq \hat{L}^2\hat{y},$$

where $\hat{L}=\frac{L_\sigma}{\sqrt{m}}$, and $\hat{y}=\frac{\|\mathbf{y}\|^2}{n}$. Similarly, for a 2-layered FFNN, we have

$$\begin{aligned}\left\langle \nabla_{\mathbf{w}_0}\mathbf{f}_s, \nabla_{\mathbf{w}_0}\mathbf{f}_s\right\rangle &= \left\langle \nabla_{W_0^1}\mathbf{f}_s, \nabla_{W_0^1}\mathbf{f}_s\right\rangle+\left\langle \nabla_{W_0^2}\mathbf{f}_s, \nabla_{W_0^2}\mathbf{f}_s\right\rangle \\ &\leq (\hat{L}^2)^2\hat{y}\left[\left\|W_0^1\right\|^2+\left\|W_0^2(s,:)\right\|^2\right].\end{aligned}$$

Generalizing the above equations, one can derive the upper bound on $\lambda_{0,\text{FFNN}}$ for an L-layer FFNN as

$$\begin{aligned}\lambda_{0,\text{FFNN}} &\leq \text{UB}_{\text{FFNN}} \\ &= \hat{L}^{2L}\hat{y}\left[\sum_{i=1}^{L-1}\|\mathbf{v}_s^T W_0^L\|^2\prod_{j=1,j\neq i}^{L-1}\|W_0^j\|^2+\prod_{j=1}^{L-1}\|W_0^j\|^2\right].\end{aligned} \tag{38}$$

Likewise, consider $L=1$, then from (13), the s^{th} component of $\mathbf{f}_{\text{LISTA}}$ is

$$\mathbf{f}_s = \frac{1}{\sqrt{m}}\sigma\left(\frac{1}{\sqrt{n}}W_{10}^1(s,:)\mathbf{y}+\frac{1}{\sqrt{m}}W_{20}^1(s,:)\mathbf{x}\right).$$

This implies,

$$\begin{aligned}\left\langle \nabla_{\mathbf{w}_0}\mathbf{f}_s, \nabla_{\mathbf{w}_0}\mathbf{f}_s\right\rangle &= \left\langle \nabla_{W_{10}^1}\mathbf{f}_s, \nabla_{W_{10}^1}\mathbf{f}_s\right\rangle+\left\langle \nabla_{W_{20}^1}\mathbf{f}_s, \nabla_{W_{20}^1}\mathbf{f}_s\right\rangle \\ &\leq \hat{L}^2\left[\hat{y}+\hat{x}\right],\end{aligned}$$

where $\hat{x}=\frac{\|\mathbf{x}\|^2}{m}$. If $L=2$, then the s^{th} component of $\mathbf{f}_{\text{LISTA}}$ is

$$\begin{aligned}\left\langle \nabla_{\mathbf{w}_0}\mathbf{f}_s, \nabla_{\mathbf{w}_0}\mathbf{f}_s\right\rangle &= \left\langle \nabla_{W_{10}^2}\mathbf{f}_s, \nabla_{W_{10}^2}\mathbf{f}_s\right\rangle+\left\langle \nabla_{W_{20}^2}\mathbf{f}_s, \nabla_{W_{20}^2}\mathbf{f}_s\right\rangle \\ &\quad+\left\langle \nabla_{W_{10}^1}\mathbf{f}_s, \nabla_{W_{10}^1}\mathbf{f}_s\right\rangle+\left\langle \nabla_{W_{20}^1}\mathbf{f}_s, \nabla_{W_{20}^1}\mathbf{f}_s\right\rangle \\ &\leq \hat{L}^2\left[\hat{y}+\hat{L}^2\|\tilde{\mathbf{x}}^{(1)}\|^2\right]+\hat{L}^4\left[\hat{y}+\hat{x}\right]\left\|\mathbf{v}_s^\top W_{20}^2\right\|^2.\end{aligned}$$

By extending the above equations, we obtain the upper bound on $\lambda_{0,\text{LISTA}}$ for an L-layer LISTA as

$$\begin{aligned}&\lambda_{0,\text{LISTA}} \leq \text{UB}_{\text{LISTA}} = \hat{L}^2\left(\hat{y}+\hat{x}\right), \text{ for } L=1 \\ &\lambda_{0,\text{LISTA}} \leq \text{UB}_{\text{LISTA}} = \hat{L}^{2L}\left(\hat{y}+\hat{x}\right)\|\mathbf{v}_s^T W_{20}^L\|^2\prod_{l=2}^{L-1}\|W_{20}^l\|^2 \\ &+\sum_{k=2}^{L-1}\hat{L}^{2L-2k+2}\left[\hat{y}+\hat{L}^2\left\|\tilde{\mathbf{x}}^{(k-1)}\right\|^2\right]\|\mathbf{v}_s^T W_{20}^L\|^2\prod_{l=k+1}^{L-1}\|W_{20}^l\|^2 \\ &+\hat{L}^2\left[\hat{y}+\hat{L}^2\|\tilde{x}^{(L-1)}\|^2\right], \text{ for } L>1,\end{aligned} \tag{39}$$

where $\hat{L} = \frac{L_\sigma}{\sqrt{m}}$, $\hat{y} = \frac{\|\mathbf{y}\|^2}{n}$, and $\hat{x} = \frac{\|\mathbf{x}\|^2}{m}$. Repeating the same analysis, one can derive the upper bound on $\lambda_{0,\text{ADMM-CSNet}}$ of an L-layer ADMM-CSNet as

$$\begin{aligned}\lambda_{0,\text{ADMM-CSNet}} &\leq \text{UB}_{\text{ADMM-CSNet}} = \hat{L}^2 \left[\hat{y} + \hat{a}^{(L-1)}\right] \\ &+ \sum_{k=1}^{L-1} \hat{L}^{2L-2k+2} \left[\hat{y} + \hat{a}^{(k-1)}\right] \|\mathbf{v}_s^T W_{20}^L\|^2 \prod_{l=k+1}^{L-1} \|W_{20}^l\|^2,\end{aligned} \tag{40}$$

where $\hat{a}^{(l)} = \frac{\|\mathbf{z}^{(l)} - \mathbf{u}^{(l)}\|^2}{m},\ \forall l \in [L-1] \cup \{0\}$. □

REFERENCES

[1] Y. C. Eldar and G. Kutyniok, *Compressed sensing: theory and applications*. Cambridge University Press, 2012.

[2] D. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[3] N. Shlezinger, J. Whang, Y. C. Eldar, and A. G. Dimakis, "Model-based deep learning," *Proc. IEEE*, vol. 111, no. 5, pp. 465–499, 2023.

[4] N. Shlezinger, J. Whang, Y. C. Eldar, and A. G. Dimakis, "Model-Based Deep Learning: Key Approaches and Design Guidelines," in *Proc. IEEE Data Sci. Learn. Workshop (DSLW)*, pp. 1–6, 2021.

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

[6] 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, 2021.

[7] N. Shlezinger, Y. C. Eldar, and S. P. Boyd, "Model-based deep learning: On the intersection of deep learning and optimization," *IEEE Access*, vol. 10, pp. 115384–115398, 2022.

[8] Y. Yang, J. Sun, H. Li, and Z. Xu, "ADMM-CSNet: A Deep Learning Approach for Image Compressive Sensing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 3, pp. 521–538, 2020.

[9] M. Zhu, T.-H. Chang, and M. Hong, "Learning to Beamform in Heterogeneous Massive MIMO Networks," *IEEE Trans. Wireless Commun.*, vol. 22, no. 7, pp. 4901–4915, 2023.

[10] Q. Hu, Y. Cai, Q. Shi, K. Xu, G. Yu, and Z. Ding, "Iterative Algorithm Induced Deep-Unfolding Neural Networks: Precoding Design for Multiuser MIMO Systems," *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 1394–1410, 2021.

[11] M.-W. Un, M. Shao, W.-K. Ma, and P. C. Ching, "Deep MIMO Detection Using ADMM Unfolding," in *Proc. IEEE Data Sci. Workshop*, pp. 333–337, 2019.

[12] A. Balatsoukas-Stimming and C. Studer, "Deep Unfolding for Communications Systems: A Survey and Some New Directions," in *Proc. IEEE Int. Workshop Signal Process. Syst. (SiPS)*, pp. 266–271, 2019.

[13] Y. Li, M. Tofighi, J. Geng, V. Monga, and Y. C. Eldar, "Efficient and Interpretable Deep Blind Image Deblurring Via Algorithm Unrolling," *IEEE Trans. Med. Imag.*, vol. 6, pp. 666–681, 2020.

[14] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang, "Deep Networks for Image Super-Resolution With Sparse Prior," in *Proc. IEEE Int. Conf. Comput. Vis.*, December 2015.

[15] G. Dardikman-Yoffe and Y. C. Eldar, "Learned SPARCOM: unfolded deep super-resolution microscopy," *Opt. Express*, vol. 28, pp. 27736–27763, Sep 2020.

[16] O. Solomon, R. Cohen, Y. Zhang, Y. Yang, Q. He, J. Luo, R. J. G. van Sloun, and Y. C. Eldar, "Deep Unfolded Robust PCA With Application to Clutter Suppression in Ultrasound," *IEEE Trans. Med. Imag.*, vol. 39, no. 4, pp. 1051–1063, 2020.

[17] 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, 2019.

[18] 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.

[19] E. Schnoor, A. Behboodi, and H. Rauhut, "Generalization Error Bounds for Iterative Recovery Algorithms Unfolded as Neural Networks," *arXiv.2112.04364*, 2022.

[20] A. Behboodi, H. Rauhut, and E. Schnoor, "Compressive Sensing and Neural Networks from a Statistical Learning Perspective," *arXiv.2010.15658*, 2021.

[21] 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.

[22] X. Chen, J. Liu, Z. Wang, and W. Yin, "Theoretical Linear Convergence of Unfolded ISTA and Its Practical Weights and Thresholds," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, Curran Associates, Inc., 2018.

[23] X. Chen, J. Liu, Z. Wang, and W. Yin, "Hyperparameter Tuning is All You Need for LISTA," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 11678–11689, Curran Associates, Inc., 2021.

[24] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*, vol. 2. Springer, 2009.

[25] M. Belkin, D. Hsu, S. Ma, and S. Mandal, "Reconciling modern machine-learning practice and the classical bias–variance trade-off," *Proc. Nat. Acad. Sci.*, vol. 116, no. 32, pp. 15849–15854, 2019.

[26] M. Belkin, "Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation," *Acta Numerica*, vol. 30, p. 203–248, 2021.

[27] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding Deep Learning (Still) Requires Rethinking Generalization," *Commun. ACM*, vol. 64, p. 107–115, feb 2021.

[28] P. Nakkiran, G. Kaplun, Y. Bansal, T. Yang, B. Barak, and I. Sutskever, "Deep Double Descent: Where Bigger Models and More Data Hurt," in *Proc. Int. Conf. Learn. Represent.*, 2020.

[29] S. Spigler, M. Geiger, S. d'Ascoli, L. Sagun, G. Biroli, and M. Wyart, "A jamming transition from under- to over-parametrization affects generalization in deep learning," *J. Phys. A*, vol. 52, p. 474001, oct 2019.

[30] M. Belkin, S. Ma, and S. Mandal, "To Understand Deep Learning We Need to Understand Kernel Learning," in *Proc. Int. Conf. Mach. Learn.*, vol. 80, pp. 541–549, PMLR, 10–15 Jul 2018.

[31] C. Liu, L. Zhu, and M. Belkin, "Loss landscapes and optimization in over-parameterized non-linear systems and neural networks," *Appl. Comput. Harmon. Anal.*, vol. 59, pp. 85–116, 2022.

[32] S. S. Du, X. Zhai, B. Poczos, and A. Singh, "Gradient Descent Provably Optimizes Over-parameterized Neural Networks," in *Proc. Int. Conf. Learn. Represent.*, 2019.

[33] S. Du, J. Lee, H. Li, L. Wang, and X. Zhai, "Gradient Descent Finds Global Minima of Deep Neural Networks," in *Int. Conf. Mach. Learn.*, vol. 97, pp. 1675–1685, PMLR, 09–15 Jun 2019.

[34] Z. Allen-Zhu, Y. Li, and Z. Song, "A convergence theory for deep learning via over-parameterization," in *Proc. Int. Conf. Mach. Learn.*, pp. 242–252, PMLR, 2019.

[35] D. Zou, Y. Cao, D. Zhou, and Q. Gu, "Stochastic Gradient Descent Optimizes Over-parameterized Deep ReLU Networks," *CoRR*, vol. abs/1811.08888, 2018.

[36] C. Liu, L. Zhu, and M. Belkin, "On the linearity of large non-linear models: when and why the tangent kernel is constant," *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 15954–15964, 2020.

[37] D. Jakubovitz, R. Giryes, and M. R. Rodrigues, "Generalization error in deep learning," in *Compressed Sensing and Its Applications: Third International MATHEON Conference 2017*, pp. 153–193, Springer, 2019.

[38] 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.*, pp. 8687–8691, 2022.

[39] R. Tibshirani, "Regression Shrinkage and Selection via the Lasso," *J. Roy. Statist Soc. Ser. B (Methodol.)*, vol. 58, no. 1, pp. 267–288, 1996.

[40] N. Parikh and S. Boyd, "Proximal Algorithms," *Found. Trends Optim.*, vol. 1, no. 3, pp. 127–239, 2014.

[41] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, vol. 57, no. 11, pp. 1413–1457, 2004.

[42] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*. 2011.

[43] B. T. Polyak, "Gradient methods for minimizing functionals," *Ž. Vyčisl. Mat. Mat. Fiz.*, vol. 3, no. 4, pp. 643–653, 1963.

[44] S. Lojasiewicz, "A topological property of real analytic subsets," *Coll. du CNRS, Les équations aux dérivées partielles*, vol. 117, no. 87-89, p. 2, 1963.

[45] Y. Ben Sahel, J. P. Bryan, B. Cleary, S. L. Farhi, and Y. C. Eldar, "Deep Unrolled Recovery in Sparse Biological Imaging: Achieving fast, accurate results," *IEEE Signal Process. Mag.*, vol. 39, no. 2, pp. 45–57, 2022.

[46] A. M. Atto, D. Pastor, and G. Mercier, "Smooth sigmoid wavelet shrinkage for non-parametric estimation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, pp. 3265–3268, 2008.

[47] X.-P. Zhang, "Thresholding neural network for adaptive noise reduction," *IEEE Trans. Neural Netw.*, vol. 12, no. 3, pp. 567–584, 2001.
[48] X.-P. Zhang, "Space-scale adaptive noise reduction in images based on thresholding neural network," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 3, pp. 1889–1892 vol.3, 2001.
[49] H. Pan, D. Badawi, and A. E. Cetin, "Fast Walsh-Hadamard Transform and Smooth-Thresholding Based Binary Layers in Deep Neural Networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 4650–4659, June 2021.
[50] J. Youn, S. Ravindran, R. Wu, J. Li, and R. van Sloun, "Circular Convolutional Learned ISTA for Automotive Radar DOA Estimation," in *Proc. 19th Eur. Radar Conf. (EuRAD)*, pp. 273–276, 2022.
[51] K. Kavukcuoglu, P. Sermanet, Y.-l. Boureau, K. Gregor, M. Mathieu, and Y. Cun, "Learning Convolutional Feature Hierarchies for Visual Recognition," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 23, Curran Associates, Inc., 2010.
[52] R. Vershynin, "Introduction to the non-asymptotic analysis of random matrices," *arXiv.1011.3027*, 2010.
[53] A. Jacot, F. Gabriel, and C. Hongler, "Neural Tangent Kernel: Convergence and Generalization in Neural Networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018.
[54] J. Lee, L. Xiao, S. Schoenholz, Y. Bahri, R. Novak, J. Sohl-Dickstein, and J. Pennington, "Wide Neural Networks of Any Depth Evolve as Linear Models Under Gradient Descent," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, Curran Associates, Inc., 2019.
[55] S. B. Shah, P. Pradhan, W. Pu, R. Ramunaidu, M. R. D. Rodrigues, and Y. C. Eldar, "Supporting Material: Optimization Guarantees of Unfolded ISTA and ADMM Networks With Smooth Soft-Thresholding," *2023*.

Shaik Basheeruddin Shah (Student Member, IEEE) received the Bachelor of Engineering degree from the Electronics and Communication Engineering Department, Vasireddy Venkatadri Institute of Technology (VVIT), India, in 2013, the Master of Technology degree from the Computational Engineering Department, Rajiv Gandhi University of Knowledge and Technologies (RGUKT), India, in 2015, and the Ph.D. degree from the Department of Electrical Engineering, Shiv Nadar University, NCR-India, in 2021. He is currently a Postdoctoral Researcher with the Faculty of Mathematics and Computer Science, Weizmann Institute of Science, Israel. His research interests include signal representation, sampling theory, compressive sensing, and model-based learning.

Pradyumna Pradhan (Student Member, IEEE) received his integrated M.Sc. degree in mathematics from the National Institute of Technology Rourkela, India, in 2020. He is currently a Ph.D. student at the Indian Institute of Petroleum and Energy, Visakhapatnam, India. His research interests include compressed sensing, deep learning, and model-aware data-driven techniques for sparse signal recovery.

Wei Pu (Member, IEEE) received the B.S. and Ph.D. degrees in electronic engineering from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2012 and 2018, respectively. From 2017 to 2018, he was a Visiting Student with the Department of Electrical Engineering, Columbia University, New York, NY, USA. From 2019 to 2022, he was a Research Fellow with University College London (UCL), London, U.K. He is currently a Professor with UESTC. His research interests include synthetic aperture radar, sparse signal processing, and deep learning. He was a recipient of the Newton International Fellowship from the Royal Society, U.K.

Ramu Naidu Randhi (Member, IEEE) received his M.Sc. degree in mathematics from the University of Hyderabad, India, in 2008, the Ph.D. degree in mathematics at Indian Institute of Technology Hyderabad, India, in 2015. He is currently an Associate Professor at the Indian Institute of Petroleum and Energy, Visakhapatnam, India. His research interests include compressed sensing, deep learning, model-based AI, machine learning, sparse optimization theory, and frame theory.

Miguel R. D. Rodrigues (Fellow, IEEE) received the Licenciatura degree in electrical and computer engineering from the University of Porto, Porto, Portugal, and the Ph.D. degree in electronic and electrical engineering from the University College London (UCL), London, U.K. He is currently a Professor of Information Theory and Processing, UCL, and a Turing Fellow with the Alan Turing Institute - the UK National Institute of Data Science and Artificial Intelligence. His research interests include the general areas of information theory, information processing, and machine learning. His work has led to more than 200 articles in leading journals and conferences in the field, a book on Information-Theoretic Methods in Data Science (Cambridge University Press), and the IEEE Communications and Information Theory Societies Joint Paper Award 2011. He is an Associate Editor for the IEEE Transactions on Information Theory, and the IEEE Open Journal of the Communications Society. He was an Associate Editor for the IEEE Communications Letters, and the Lead Guest Editor of the special issue on "Information-Theoretic Methods in Data Acquisition, Analysis, and Processing" of the IEEE Journal on Selected Topics in Signal Processing. He was the Co-Chair of the Technical Programme Committee of the IEEE Information Theory Workshop 2016, Cambridge, U.K. He is a member of the IEEE Signal Processing Society Technical Committee on "Signal Processing Theory and Methods", and the EURASIP SAT on Signal and Data Analytics for Machine Learning (SiG-DML).

Yonina C. Eldar (Fellow, IEEE) received the B.Sc. degree in physics, in 1995, and the B.Sc. degree in electrical engineering, in 1996, both from TelAviv University (TAU), Tel-Aviv, Israel, and the Ph.D. degree in electrical engineering and computer science, in 2002, from Massachusetts Institute of Technology (MIT), Cambridge, MA, USA. She is currently a Professor with the Department of Mathematics and Computer Science, Weizmann Institute of Science, Rehovot, Israel, where she holds the Dorothy and Patrick Gorman Professorial Chair and heads the Center for Biomedical Engineering. She was previously a Professor in the Department of Electrical Engineering with the Technion, where she held the Edwards Chair in Engineering. She is a member of the Israel Academy of Sciences and Humanities and of the Academia Europaea (elected 2023), a EURASIP Fellow, a Fellow of the Asia-Pacific Artificial Intelligence Association, and a Fellow of the 8400 Health Network. Her research interests include statistical signal processing, sampling theory and compressed sensing, learning and optimization methods, and their applications to biology, medical imaging and optics.