Deep Learning Model-Aware Regulatization with Applications to Inverse Problems

Jaweria Amjad, Zhaoyan Lyu, and Miguel R. D. Rodrigues, *Senior Member, IEEE*

***Abstract*—There are various inverse problems – including reconstruction problems arising in medical imaging — where one is often aware of the forward operator that maps variables of interest to the observations. It is therefore natural to ask whether such knowledge of the forward operator can be exploited in deep learning approaches increasingly used to solve inverse problems.**

In this paper, we provide one such way via an analysis of the generalisation error of deep learning approaches to inverse problems. In particular, by building on the algorithmic robustness framework, we offer a generalisation error bound that encapsulates key ingredients associated with the learning problem such as the complexity of the data space, the size of the training set, the Jacobian of the deep neural network and the Jacobian of the composition of the forward operator with the neural network. We then propose a 'plug-and-play' regulariser that leverages the knowledge of the forward map to improve the generalization of the network. We likewise also use a new method allowing us to tightly upper bound the Jacobians of the relevant operators that is much more computationally efficient than existing ones. We demonstrate the efficacy of our model-aware regularised deep learning algorithms against other state-of-the-art approaches on inverse problems involving various sub-sampling operators such as those used in classical compressed sensing tasks, image super-resolution problems and accelerated Magnetic Resonance Imaging (MRI) setups.

***Index Terms*—Deep Learning, Generalization Error, Jacobian, Inverse Problems, Regularization, Robustness**

I. Introduction

In various signal and image processing challenges arising in practice – including medical imaging, remote sensing, and many more – one often desires to recover a number of latent variables from physical measurements. This class of problems – generally known as *inverse problems* – can often be modelled as follows:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n} \tag{1}$$

where $\mathbf{y} \in \mathcal{Y} \subset \mathbb{R}^q$ represents a q-dimensional vector containing the physical measurements, $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^p$ represents a p-dimensional vector containing the variables of interest, and $\mathbf{n} \in \mathcal{N} \subseteq \mathbb{R}^q$ is a bounded perturbation modelling measurement noise. The forward operator modelling the relationship between physical measurements and variables of interests is in turn modelled (in the absence of noise) using a matrix $\mathbf{A} \in \mathbb{R}^{q \times p}$. This forward operator satisfies certain regularity conditions [1] whereby $\forall \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$

$$\|\mathbf{A}\mathbf{x}_1 - \mathbf{A}\mathbf{x}_2\|_2 \leq \Lambda_a \|\mathbf{x}_1 - \mathbf{x}_2\|_2 \tag{2}$$

where Λ_a represents the maximum singular value of the forward map $\mathbf{A}$.

Two broad classes of approaches have been adopted to solve inverse problems: (i) *model-based* methods and (ii) *data-driven* methods. Model-based methods exploit knowledge of the forward operator and/or the signal/noise model in order to recover the variables of interest from the measurements [1]. For example, well-known inverse problem recovery algorithms often leverage knowledge of data priors capturing stochastic [2] or geometric structure [3].On the other hand, data-driven methods do not leverage explicitly the knowledge of the underlying physical and data models; instead, such methods rely on the availability of various data pairs $(\mathbf{x}, \mathbf{y})$ in order to learn how to invert the forward operator associated with the inverse problem [4]. The challenge relates to the fact that these approaches – specially deep learning ones – typically require the availability of various training examples that are not always available in a number of applications such as medical image analysis. This inevitably hinders the applicability of data-driven approaches to inverse problems arising in various scientific and engineering use-cases.

In this paper, our overarching goal is to understand using first-principles how to use knowledge readily available in various inverse problems in order to improve the performance of deep learning based data-driven methods. We approach this challenge by offering new generalization guarantees that capture how the generalization ability is affected by various key quantities associated with the learning problem. Such interplay then immediately leads to an entirely new model-aware regularization strategy acting as a proxy to import knowledge about the underlying physical model onto the deep learning process.

Concretely, our contributions can be summarized as follows:

- We present generalization error bounds for Deep Neural Networks (DNN) based inverse problem solvers. Notably, such bounds depend on various quantities including the Jacobian matrix of the neural network along with the Jacobian matrix of the composition of the neural network with the inverse problem forward map.
- We then propose new regularization strategies that are capable of using knowledge about the inverse problem

J. Amjad, Z. Lyu and M. R. D. Rodrigues are with the Department of Electronic and Electrical Engineering, Univeristy College London, London WC1E 6BT, U.K. (e-mail: jaweria.amjad@ucl.ac.uk; z.lyu.17@ucl.ac.uk; m.rodrigues@ucl.ac.uk)

The work of J. Amjad was supported by Commonwealth Scholarship Commission in UK. The work of Z. Lyu was supported by China Scholarship Commission. This work was also supported by EPSRC grant EP/P02243X/1. *(Corresponding author: Jaweria Amjad.)*

[1] Note that such forward operators encountered in various applications of interest including Magnetic resonance Imaging (MRI), Computed Tomography (CT) etc obey some form of regularity constraint such as given in (2).

model during the neural network learning process via the control of the spectral and Frobenius norms of such Jacobian matrices.

- We also showcase computationally efficient methods to estimate the spectral and Frobenius norms of the aforementioned Jacobian matrices in order to accelerate the neural network learning process.
- Finally, we demonstrate the empirical performance of our algorithms on various inverse problems with different degrees of ill-posednesss. These include the reconstruction of high-dimensional data from low-dimensional noisy measurements where the forward model is either a compressive random Gaussian matrix, a decimation operator used to generate low resolution images from the corresponding high-resolution version, or a subsampling matrix usually employed in accelarated Magnetic Resonance Imaging applications.

The remainder of the paper is organized as follows: After presenting an overview of the related work in Section II, we introduce our system setup in Section III. We then present generalization bounds applicable to neural network based inverse problem solvers in Section IV, leading up to model-aware regularizers in Section V. Section VI offers various experimental results showcasing our model-aware deep learning approach can lead to substantial gains in relation to model-agnostic ones. Finally, concluding remarks are drawn in Section VII. All the proofs are relegated to the appendices.

Notation: We use lower case boldface characters to denote vectors, upper case boldface characters to represent matrices and sets are represented by calligraphic font. For example $\mathbf{x}$ is a vector, $\mathbf{X}$ is a matrix and $\mathcal{X}$ is a set. $\mathcal{N}_{\mathcal{X}}(\delta, \ell_2)$ represents the covering number of a metric space $(\mathcal{X}, \ell_2)$ using balls of radius δ.

II. Related Work

Our work connects to various directions in the literature.

A. Model-based techniques for inverse problems

The main challenge in solving (ill-posed) inverse problems relates to the fact that – without any prior assumption – it is not possible to recover the variables of interest from the observations (even when the forward model is perfectly known). Classical model-based approaches address this challenge via the formulation of optimization problems that include two terms in the objective: (1) a data fidelity term and (2) a data regularization one. The fidelity term encourages the solution to be consistent with the observations whereas the regularization one encourages solutions that conform to a certain postulated data prior. There are a large number of model-based approaches in the literature: Popular variational methods use a regularizer that promotes smoothness of the solutions [5], [6] whereas sparsity-driven methods use regularizers that promote sparsity of the solutions in some transform domain [7], [8], [9]. In addition to the challenging task of determining a suitable data prior, these traditional approaches tend to require relatively complex solvers inevitably restricting their applicability.

B. Data-driven techniques for inverse problems

The recent years have witnessed a surge of interest in data-driven approaches – with a focus on deep learning ones – to solve inverse problems [10]. In particular, inspired by the success of deep learning in classification tasks, such approaches typically "solve" an inverse problem by using a neural network that has learnt how to map the model output to the model input based on a number of input-output examples [4]. Such approaches have been applied to a large number of inverse problems such as image denoising [11], [12], image super-resolution [13], MRI reconstruction [14], [15], CT reconstruction [16], and many more. However, these data-driven approaches typically require rich enough datasets – which are not always available in various domains such as medical imaging – in order to learn how to solve the inverse problem [17].

C. Model-aware data driven approaches

In view of the fact that the underlying physical model is known in various scenarios, there has been an increased interest in model-aware data-driven approaches to inverse problems. Some approaches leverage knowledge of the forward model to provide a rough estimate of the inverse problem solution (e.g. using some form of pseudo-inverse of the forward operator) that is then further processed using a neural network [18], [19], [20].

Another approach that is becoming increasingly popular relies on algorithm unfolding or unrolling [21], [22], [23]. By starting with a typical optimization based formulation to tackle the underlying inverse problem – where knowledge of the physical model is explicitly used – unfolding then maps iterative solvers onto a neural network architecture whose parameters can be further tuned in a data-driven manner.

Finally there is also a new suite of techniques that leverage the knowledge of forward operator as follows: the reconstruction of the desired data vector given the measurements vector is carried out using a (regularized) optimization problem using the underlying model; however, the regularizer within such an optimization problem is itself learnt directly from a set of data examples. One such recent (unsupervised) approach relies on the use of adversarially learnt data dependent regularizers [24]. Another suite of techniques uses instead data representations learnt directly from data in any underlying model based optimization problem. For example, in [25], the authors propose to learn the underlying low dimensional manifold of the latent signal of interest using a generative adversarial network (GAN) allowing them to constrain in any optimization problem the reconstruction of the original data from the data measurements to conform to such learnt manifold. While this method yields powerful representations, its training hinges upon the acquisition of a sufficient amount of training data for it to generalize well enough to the test data. A similar approach which employs the structure of a GAN as an implicit regularizer was proposed in [26]. The work shows that a hand crafted network architecture inherently favours solutions that look like natural images – hence can serve as a suitable prior in image restoration tasks. Finally there are approaches where a learned denoising autoencoder is treated as a regularization step in an iterative reconstruction method [27], [28].

Our work departs from these contributions in the sense that – whereas we also use a deep network to solve an inverse problem – we leverage knowledge of the underlying forward operator model via appropriate regularization strategies deriving from a principled generalization error analysis. The proposed approach gives rise to a prior which is tailored to a particular inverse problem.

D. Other related work

There is also a considerable volume of literature offering analysis of the generalization ability of deep neural networks demonstrating that the generalization error of highly parametrized models can be bounded in terms of certain parameter norms [29], [30]. However, the majority of these bounds are applicable to classification problems rather than regression based ones[2].

The fact that enforcing Lipschitz regularity in deep neural networks endows them with several desirable properties is well recognized [32]-[43]. Several works in literature have demonstrated a link between improved generalization performance and constrained gradient norms of DNN classifiers [38], [37]. For example, a small Lipschitz constant has also been shown to result in better generalization error guarantees [44], [30]. However, many of the existing techniques constrain only the Lipschitz constants of the layer-wise affine transformations in the network [32]-[36]. These approaches do not take into account the non-linearities in the network and thus under-utilize the Lipschitz capacity of the network by biasing it to learn simplistic functions [39].

In this work, motivated by our analysis, we propose to constrain the spectral norm of the input-output network Jacobian matrix which serves as a tight upper bound on the Lipschitz constant of the relevant mapping. We then offer an algorithm to efficiently estimate it without significantly increasing the computational overhead. The computation of the Lipschitz constant has been shown to be infeasible in [35]. Therefore, we propose to instead penalize a tight upper bound approximation – the spectral norm of the Jacobian matrix on the available training samples. To the best of our knowledge, our algorithm is the most efficient method to achieve this.

III. Setup

We consider the linear observation model in eq. (1), with the following additional assumptions: the input space $\mathcal{X} \subseteq \mathbb{R}^p$ is compact with respect to the ℓ_2 metric; the noise space $\mathcal{N} = \{\mathbf{n} : \|\mathbf{n}\|_2 \leq \eta\} \subseteq \mathbb{R}^q$ is also compact with respect to the ℓ_2 metric; and the output space – which is defined as $\mathcal{Y} = \{\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n} : \mathbf{x} \in \mathcal{X}, \mathbf{n} \in \mathcal{N}\} \subseteq \mathbb{R}^q$ – can also be shown to be compact with respect to the ℓ_2 metric. Finally, we also define the sample space $\mathcal{D} = \{\mathbf{s} = (\mathbf{x}, \mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}) : \mathbf{x} \in \mathcal{X}, \mathbf{n} \in \mathcal{N}\}$ that is compact with respect to the ℓ_2 metric.

Our approach to solve this problem is based on the standard supervised learning paradigm. We assume access to a training set $\mathcal{S} = \{\mathbf{s}_i = (\mathbf{x}_i, \mathbf{y}_i = \mathbf{A}\mathbf{x}_i + \mathbf{n}_i)\}_{i \leq m}$ consisting of m data points drawn independently and identically distributed (IID) from the sample space $\mathcal{D}$ according to the unknown data distribution μ, consistent with the forward model in (1).

We use such a training set to learn a hypothesis $f_{\mathcal{S}} : \mathcal{Y} \to \mathcal{X}$ mapping the measurement variables to variables of interest. We then use such a hypothesis to map new measurement variables $\mathbf{y} \in \mathcal{Y}$ to the variables of interest $\mathbf{x} \in \mathcal{X}$ that were not necessarily originally present in the training set.

We restrict our attention to mappings based on feed-forward neural networks. Such a feed forward neural network can be represented as a composition of d layer-wise mappings delivering an estimate of the variable of interest given the measurement variable as follows:

$$f_{\mathcal{S}}(\mathbf{y}) = \left(f_{\theta_d} \circ \dots f_{\theta_1}\right)(\mathbf{y}; \Theta) \tag{3}$$

where $f_{\mathcal{S}}(\cdot)$ represents the feed-forward neural network, $f_{\theta_i}(\cdot)$ represents the i-th layerwise mapping parameterized by θ_i, and $\Theta = \{\theta_1, \dots \theta_d\}$ is the set of tunable parameters in the neural network. The parameters of the feed-forward neural network are typically tuned based on the available training set using a learning algorithm such as stochastic gradient descent [45].

One is typically interested in the performance of the learnt neural network not only on the training data but also on (previously unseen) testing data. Therefore, it is useful to quantify the generalization error associated with the learnt neural network given by:

$$GE(f_{\mathcal{S}}) = |l_{\text{exp}}(f_{\mathcal{S}}) - l_{\text{emp}}(f_{\mathcal{S}})| \tag{4}$$

where $l_{\text{exp}}(f_{\mathcal{S}}) = \mathbb{E}_{\mathbf{s} \sim \mu}[l(f_{\mathcal{S}}, \mathbf{s})]$ represents the expected error, $l_{\text{emp}}(f_{\mathcal{S}}) = \frac{1}{m} \sum_i l(f_{\mathcal{S}}, \mathbf{s}_i)$ represents the empirical error, and the loss function $l : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}_0^+$ — which measures the discrepancy between the neural network prediction and the ground truth — is taken to be the ℓ_2 distance given by:

$$l(f_{\mathcal{S}}, \mathbf{s}) = \|f_{\mathcal{S}}(\mathbf{y}) - \mathbf{x}\|_2 \tag{5}$$

Our ensuing analysis offers bounds to the generalization error in (4) of deep feed-forward neural networks based inverse problems solvers as a function of a number of relevant quantities. These quantities include the covering number of the sample space $\mathcal{D}$, the size of the training set $\mathcal{S}$, and properties of the network encapsulated in its input-output Jacobian matrix given by:

$$\mathbf{J}(\mathbf{y}) = \begin{bmatrix} \frac{\partial f_{\mathcal{S}}(\mathbf{y})_1}{\partial \mathbf{y}_1} & \cdots & \frac{\partial f_{\mathcal{S}}(\mathbf{y})_1}{\partial \mathbf{y}_q} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_{\mathcal{S}}(\mathbf{y})_p}{\partial \mathbf{y}_1} & \cdots & \frac{\partial f_{\mathcal{S}}(\mathbf{y})_p}{\partial \mathbf{y}_q} \end{bmatrix}$$

The bounds also depend on quantities associated with the linear model in eq. (1) such as the forward operator and the noise bound. Our analysis will therefore also inform how to import knowledge about the forward-operator associated with the inverse problem onto the learning procedure in order to improve the generalization error.

[2] Exceptions include [31], but their results suffered from an exponential dependence on network depth.

IV. Analysis: Generalization Error Bounds

Our analysis builds upon the *algorithmic robustness* framework in [46].

Definition 1. A learning algorithm is said to be $(K, \epsilon(\mathcal{S}))$-robust if the sample space $\mathcal{D}$ can be partitioned into K disjoint sets $\mathcal{K}_k$, $k = 1, \ldots, K$, such that for all $\mathbf{s}_i = (\mathbf{x}_i, \mathbf{y}_i = \mathbf{A}\mathbf{x}_i + \mathbf{n}_i) \in \mathcal{S}$ and all $\mathbf{s} = (\mathbf{x}, \mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}) \in \mathcal{D}$

$$\mathbf{s}_i, \mathbf{s} \in \mathcal{K}_k \implies |l(f_{\mathcal{S}}, \mathbf{s}_i) - l(f_{\mathcal{S}}, \mathbf{s})| \leq \epsilon(\mathcal{S}) \tag{6}$$

This notion has already been used to analyse the performance of deep neural networks in [38], [36], [41]. However, such analyses applicable to classification tasks do not carry over immediately to inverse problems based tasks where – in addition to exploiting knowledge about the forward operator associated with the inverse problem for the computation of $\epsilon(\mathcal{S})$ and K – there are some technical complications that may arise due the fact that the loss functions are typically unbounded [3]. We begin addressing these challenges by offering a simple result that showcases how the distance between the neural network estimates of the variables of interest depends on the distance between the variables of interest themselves and, importantly, the Jacobian of the network, the Jacobian of the composition of the network with the forward model associated with the inverse problem, and the noise power associated with the inverse problem.

Theorem 1. Consider a neural network $f_{\mathcal{S}}(\cdot) : \mathcal{Y} \to \mathcal{X}$ based solver of the inverse problem in (1), learnt using a training set $\mathcal{S}$. Then, for any $\mathbf{s}_1 = (\mathbf{x}_1, \mathbf{y}_1 = \mathbf{A}\mathbf{x}_1 + \mathbf{n}_1),\ \mathbf{s}_2 = (\mathbf{x}_2, \mathbf{y}_2 = \mathbf{A}\mathbf{x}_2 + \mathbf{n}_2) \in \mathcal{D}$, it follows that

$$\|f_{\mathcal{S}}(\mathbf{y}_2) - f_{\mathcal{S}}(\mathbf{y}_1)\|_2 \leq \Lambda_{f \circ a}\|\mathbf{x}_2 - \mathbf{x}_1\|_2 + 2\eta\Lambda_f$$

where $\Lambda_{f \circ a}$ and Λ_f are upper bounds to the Lipschitz constants of the neural network and the composition of the neural network and the forward operator respectively, given by:

$$\begin{aligned} \Lambda_{f \circ a} &= \sup_{\mathbf{y} \in conv(\mathcal{Y})} \|\mathbf{J}(\mathbf{y})\,\mathbf{A}\|_2 \\ \Lambda_f &= \sup_{\mathbf{y} \in conv(\mathcal{Y})} \|\mathbf{J}(\mathbf{y})\,\|_2 \end{aligned} \tag{7}$$

Proof. See Appendix. □

We now state another theorem – building upon Theorem 1 – articulating about the robustness of a deep neural network based solver of an inverse problem.

Theorem 2. A neural network trained to solve the inverse problem in (1) based on a training set $\mathcal{S}$ is $(K, \epsilon(\mathcal{S}))$-robust such that for any $\delta > 0$,

$$\begin{aligned} K &\leq \mathcal{N}_{\mathcal{X}}(\delta, \ell_2) \\ \epsilon(\mathcal{S}) &\leq 2(1 + \Lambda_{f \circ a})\delta + 2\Lambda_f\eta \end{aligned}$$

where $\mathcal{N}_{\mathcal{X}}(\delta, \ell_2)$ is the covering number of $\mathcal{X}$.

Proof. See Appendix. □

We now state our main result relating to the generalization error of a deep neural network trained to solve an inverse problem.

Theorem 3. A neural network trained to solve the inverse problem in (1) based on a training set $\mathcal{S}$ consisting of m i.i.d. training samples obeys with probability $1 - \zeta$, for any $\zeta > 0$, the GE bound given by:

$$\begin{aligned} GE(f_{\mathcal{S}}) \leq & 2(1 + \Lambda_{f \circ a})\delta + 2\Lambda_f\eta \\ & + M\sqrt{\frac{2\mathcal{N}_{\mathcal{X}}(\delta, \ell_2)\log(2) + 2\log{(1/\zeta)}}{m}} \end{aligned}$$

for $\max_{\mathbf{s}} |l(f_{\mathcal{S}}, \mathbf{s})| \leq M < \infty$ and any $\delta > 0$.

Proof. See Appendix. □

One can derive various insights from this theorem that are applicable to any differentiable feed forward neural network along with any linear forward map : (1) first, in line with traditional bounds [29], [47], the generalization error depends on the size of training set $\mathcal{S}$; (2) second, in line with more recent bounds [38], [36], [31], the generalization error also depends on the complexity of the data space $\mathcal{D}$; [4] (3) third, although the ℓ_2-loss is unbounded in nature, on a compact sample space, the DNN is able to predict samples such that the loss is finite and therefore the GE is provably bounded; 4)Finally, Theorem 3 also reveals that the operator norm of the Jacobian of the network and the composite map also play a critical role: the lower the value of these norms, the lower the generalization error. More importantly, the proposed generalization bound is also non-vacuous in the network parameters because as opposed to the product of the norms of layer-wise weight matrices appearing in other generalization error bounds such as [36], [44], [30], the norm of the network Jacobian matrix does not seem to exhibit exponential dependence on network depth. This is in sharp contrast with existing generalization bounds that typically contain a term that deteriorates exponentially with depth.

It should also be noted that for the linear inverse problems in imaging, the input space $\mathcal{X}$ can be assumed to be a C_M regular k-dimensional manifold [48]. The constant C_M varies for different manifolds and represents their "intrinsic" properties. This is a reasonable assumption for the visual data and has previously been used to represent such input spaces. The covering number for such manifolds can be bounded via $\left(\frac{2C_M}{\delta}\right)^k$ [48], [41].

V. Model-Aware Jacobian Regularization

Our approach to leverage knowledge about the inverse problem model onto the learning process involves regularization. In particular, Theorem 3 suggests that penalizing the spectral norm of the Jacobian of the neural network and the spectral norm of the Jacobian of the composition of the neural network with the inverse problem forward operator, which – as shown above – also serve as an upper bound to the Lipschitz constants

[3] Existing work applies to uniformly bounded loss function (e.g. [46], [38]).

[4] The complexity of the sample space – which can be captured via its covering number – is often small in view of the fact that in various applications data lies on a manifold with small intrinsic dimension [38].

of these mappings, should improve the generalization ability of a neural network based inverse problem solver.

The use of Lipschitz regularization to improve the generalization ability of deep neural networks has already been recognized by various works [36]-[38]. However, the fact that introducing Lipschitz regularity in the end-to-end mapping involving the composition of the neural network and the inverse problem forward map may also control generalization does not appear to have been acknowledged in previous works. We therefore propose two model-aware regularization strategies:

Model-Aware Spectral Norm Based Regularization: Our first regularization strategy directly penalizes the operator norm of the Jacobians for the neural network and for the composition of the neural network and the forward map.Training in a minibatch stochastic gradient setup, where the optimization is carried out over minibatches $\mathcal{B} = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_{|\mathcal{B}|}\}$, leads to the following objective:

$$\frac{1}{|\mathcal{B}|}\sum_{i=1}^{|\mathcal{B}|} l(f_{\mathcal{S}}, \mathbf{s}_i)+\beta_1 \max_{\mathbf{s}=(\mathbf{x},\mathbf{y})\in\mathcal{B}} \|\mathbf{J}(\mathbf{y})\mathbf{A}\|_2+\beta_2 \max_{\mathbf{s}=(\mathbf{x},\mathbf{y})\in\mathcal{B}} \|\mathbf{J}(\mathbf{y})\|_2 \tag{8}$$

where β_1, β_2 are hyper-parameters. Note that $\beta_2 = 0$ in a noise free setting.

Model-Aware Frobenius Norm Based Regularization: Our second regularization strategy stems from the fact that the Frobenius norm upper bounds the Spectral norm. Regularisation strategies that punish the Frobenius norm of the network Jacobian have been associated with significant improvement in robustness of DNN classifiers [49], [38], [43]. Therefore, our cost function in (8) directly gives rise to the following objective function:

$$\frac{1}{|\mathcal{B}|}\sum_{i=1}^{|\mathcal{B}|} l(f_{\mathcal{S}}, \mathbf{s}_i)+\beta_1 \max_{\mathbf{s}=(\mathbf{x},\mathbf{y})\in\mathcal{B}} \|\mathbf{J}(\mathbf{y})\mathbf{A}\|_F+\beta_2 \max_{\mathbf{s}=(\mathbf{x},\mathbf{y})\in\mathcal{B}} \|\mathbf{J}(\mathbf{y})\|_F \tag{9}$$

We, however propose to regularize the following upper bound on (9) given by:

$$\frac{1}{|\mathcal{B}|}\sum_{i=1}^{|\mathcal{B}|} l(f_{\mathcal{S}}, \mathbf{s}_i) + \beta_1 \sum_{i=1}^{|\mathcal{B}|} \|\mathbf{J}(\mathbf{y}_i)\mathbf{A}\|_F^2 + \beta_2 \sum_{i=1}^{|\mathcal{B}|} \|\mathbf{J}(\mathbf{y}_i)\|_F^2 \tag{10}$$

This is mainly because the sum of square of the Frobenius norm results in simpler gradient computation. Additionally the regularization terms in (10) can be approximated in a computationally efficient setting as explained in the sequel.

A. Efficient Computation of the Norms of the Jacobian Based Regularizers

The challenge associated with the use of the training objectives in (8) and (10) relates to the computation of the Spectral norm and Frobenius norm of both $\mathbf{J}$ and $\mathbf{JA}$ because computing and storing the Jacobian matrix of deep neural networks incurs a huge cost. There are already computationally efficient algorithms to approximate the Frobenius and spectral norm of the Jacobian [43], [39]. Here, for completeness, we illustrate how to re-purpose these algorithms within our setup; we also illustrate that these algorithms lead to efficient approximation.

Algorithm 1: Estimation of the $\|\mathbf{JA}\|_F^2$

```
Input: Mini-batch $\mathcal{B}$,number of projections $n$.
Output: Square of the Frobenius norm of the Jacobian $\mathcal{A}_F$.
$\mathcal{A}_F \leftarrow 0$
for $(\mathbf{y}, \mathbf{x}) \in \mathcal{B}$ do
    $i \leftarrow 0$
    while $i < n$ do
        Initialize $\{\mathbf{z}\} \sim \mathcal{N}(0, \mathbf{I})$
        $\mathbf{z} \leftarrow \mathbf{z}/||\mathbf{z}||$
        $\mathcal{A}_F \leftarrow \mathcal{A}_F + p\|vjp(f(\mathbf{y}), \mathbf{y}, \mathbf{z}) \cdot \mathbf{A}\|_2^2/(n|\mathcal{B}|)$
```

Algorithm 2: Estimation of the spectral norm of $\mathbf{JA}$

```
Input: Mini-batch $\mathcal{B}$, number of power iterations $n$.
Output: Maximum singular value, $\sigma$, of the matrix $\mathbf{JA}$.
for $(\mathbf{y}, \mathbf{x}) \in \mathcal{B}$ do
    Initialize $\{\mathbf{u}\} \in \mathbb{R}^p$
    $i \leftarrow 0$
    while $i < n$ do
        $\mathbf{v} \leftarrow \mathbf{A}^T vjp(f(\mathbf{y}), \mathbf{y}, \mathbf{u})$
        $\mathbf{u} \leftarrow jvp(f(\mathbf{y}), \mathbf{y}, \mathbf{Av})$
        $i \leftarrow i + 1.$
    $\sigma \leftarrow \|\mathbf{u}\|_2/\|\mathbf{v}\|_2$
```

1) Frobenius Norm Regularization of $\mathbf{JA}$*:* The random projection based method proposed in [43] used to approximate the square of the Frobenius norm of the network Jacobian matrix $\mathbf{J}$ can be immediately extended to approximate the square of the Frobenius norm of the $\mathbf{JA}$ as shown in Algorithm 1. The technique leverages the reverse mode automatic differentiation to compute vector Jacobian product – the $vjp(\cdot,\cdot,\cdot)$ – of random vector sampled from the unit sphere of dimension $p-1$ with the network Jacobian. It has been shown in [43] that the proposed technique converges to the true value as $\mathcal{O}(n^{-1/2})$ where n is the number of random projections used for the estimation of the Frobenius norm. The algorithm when used for regularization, has also been shown to result in only an inconsequential overhead in compute requirements [43].

2) Spectral Norm Regularization of $\mathbf{JA}$*:* In turn, the method in [39] used to approximate the spectral norm of the network Jacobian $\mathbf{J}$ can also be immediately re-purposed to approximate the spectral norm of $\mathbf{JA}$ as shown in Algorithm 2. The procedure leverages the power method [50] to approximate the spectral norm of the Jacobian based regularization terms in (8). It starts by choosing (randomly) an initial (nonzero) approximation of the left singular vector $\mathbf{u}$ in $\mathbb{R}^p$ associated with the highest singular value of the matrix $\mathbf{JA}$. It then leverages the automatic differentiation to iteratively compute the Jacobian vector product and vector Jacobian product as follows:

$$\mathbf{v} \leftarrow \mathbf{A}^T \left[\frac{df(\mathbf{y})}{d\mathbf{y}}\right]^T \mathbf{u}, \qquad \mathbf{u} \leftarrow \left[\frac{df(\mathbf{y})}{d\mathbf{y}}\right] \mathbf{Av}$$

The spectral norm σ is then equal to $\|\mathbf{u}\|_2/\|\mathbf{v}\|_2$.

The algorithm exploits the reverse and forward mode automatic differentiation to compute the vector Jacobian product $vjp(\cdot,\cdot,\cdot)$, and the Jacobian vector products $jvp(\cdot,\cdot,\cdot)$ respectively. All major deep learning frameworks offer support

Algorithm 3: Computation of the jvp.

```
Input: Mini-batch $\mathcal{B}$, model outputs $f(\mathbf{y})$, vector $\mathbf{Av}$.
Output: $\mathbf{JAv}$
Initialize a dummy tensor $\mathbf{d}$.
$\mathbf{g} \leftarrow vjp(f(\mathbf{y}), \mathbf{y}, \mathbf{d})$
$\mathbf{u} \leftarrow vjp(\mathbf{g}, \mathbf{p}, \mathbf{Av})$
return $\mathbf{u}$
```

for the computation of reverse mode vector Jacobian product. The forward mode Jacobian vector product can easily be computed via the reverse mode automatic differentiation using the method described in Algorithm 3 [51].

Note again that the merit of Algorithms 1 and 2 lies in computing the Frobenius and spectral norms of Jacobians without explicitly computing the Jacobians themselves that is prohibitive in high-dimensional settings.

3) Algorithm Accuracy and Complexity: We now study the efficacy offered by Algorithm 2 via a simple experiment involving the reconstruction of MNIST data from a noisy versions.

We generate the noisy MNIST data by passing the clean data through the linear model in (1) with the forward operator set to be equal to an identity one. We also further contaminate the MNIST data with a noise sampled uniformly from a ℓ_2-sphere of radius 0.3. We then reconstruct the data from the noisy version using two neural networks, a 4-layer fully connected neural network and a 5-layer convolutional neural network. These networks are trained using ADAM optimizer for 300 epochs using the ℓ_2 loss function in (5).

Our experiments have two main goals:

1) First, we want to test that Algorithm 2 indeed results in a faithful estimate of the spectral norm of the network Jacobian. To this end, we compare the output of the Algorithm 2 with the spectral norm computed using power method applied to a Jacobian matrix explicitly computed using Tensorflow. It can be seen in Fig. 1 that for equal number of power iterations $(n = 3)$ the results obtained using both methods are almost identical.
2) Second, we want to quantify the computational benefit afforded to us by Algorithm 2 – owing to its implicit matrix vector products computation – in contrast to estimating the spectral norm via explicit matrix vector products. In particular, Table I compares computation and memory requirements of the algorithm against alternatives associated with the training of both the fully connected and the convolutional neural networks. It can also be seen that our algorithm provides considerable gains in relation to the alternatives. It should be noted the time and memory requirement for the calculation of Jacobian of the DnCNN – which is a convolutional neural network – is higher than those of the FC neural network. This is because all the major machine learning libraries compute the gradients backwards from the output to the input using the chain rule of derivatives – leading to an increase in the compute and memory requirements for CNNs that have large layerwise activation sizes.

TABLE I: Time and memory requirements for training a 4-layer fully connected NN and 5-layer CNN [12] on the full training set of MNIST with a batch size of 100 and $p = q = 784$.

	4-layer FC NN		5-layer DnCNN	
	time	memory	time	memory
Vanilla	29m	595Mb	1h8m	1057Mb
Alg. 2 $(n = 1)$	47.5m	659Mb	3h,42m	1825Mb
Alg. 2 $(n = 2)$	1h,1m	787Mb	5h,4m	2849Mb
Alg. 2 $(n = 3)$	1h,13m	787Mb	6h,31m	4897Mb
Alg. 2 $(n = 4)$	1h,18m	787Mb	9h,42m	4897Mb
tf batch J $(n = 3)$	63h,7m	4659Mb	$--$	$\approx$ 160Gb

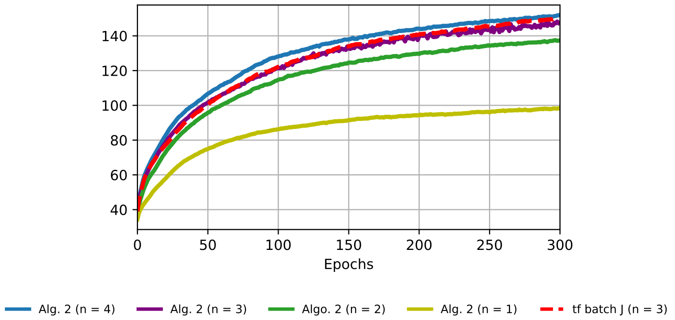


Fig. 1: Maximum singular values of the batch Jacobians for a 4-layer fully connected network with $p = q = 784$.

In summary, both for fully connected and convolutional neural networks, our experiments suggest that regularizing the network using Algorithm 2, offers considerable computational gains in comparison to direct computation of the spectral norm. In fact, the explicit computation of the network Jacobian would be practically impossible even for a modestly sized network. For example, for convolutional neural networks, even a minibatch Jacobian of 10 samples occupies 16GB of memory making it infeasible to approximate any norm. In contrast, with Algorithm 2 both jvp and vjp can be computed approximately in linear time using most major deep learning frameworks.

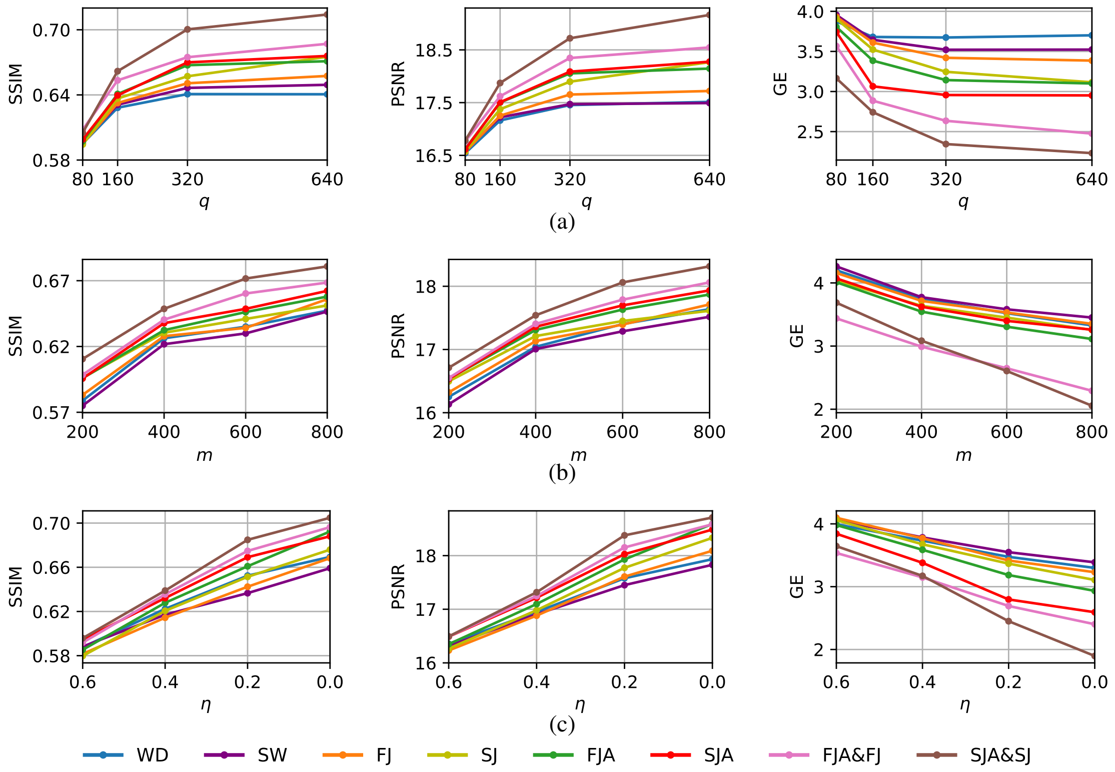
VI. Experiments

We now conduct a series of experiments in order to assess the efficacy of our proposed model-aware deep learning regularization strategy on range of popular inverse problems. These include (a) the reconstruction of images from low-dimensional Gaussian measurements (b) the generation of high-resolution images from a low-resolution version and (c) the reconstruction of MRI images from k-space sub-sampled measurements. These various inverse problems involve different measurement operators, exhibiting different condition numbers, enabling us to verify the merit of our proposed regularizers under various settings.

A. Image Reconstruction in the Presence of Gaussian Measurements

1) Experimental procedure: Our first set of experiments involves the reconstruction of images from noisy compressive Gaussian measurements. In particular, we consider our linear model in (1) where $\mathbf{A}$ is a (wide) random Gaussian matrix[5] with

[5] Forward maps generated using these rules fulfil the restricted isometry property with high probability [52].



(a)

(b)

(c)

Fig. 2: Reconstruction of MNIST images given Gaussian compressive measurements using a fully connected neural network. (2a) (Left) SSIM versus number of Gaussian measurements, (Centre) PSNR versus number of Gaussian measurements, (Right) GE versus number of Gaussian measurements for various regularization strategies such that $\eta = 0.3$ and $m = 500$. (2b) (Left) SSIM versus number of training examples, (Centre) PSNR versus number of training examples, (Right) GE versus number of training examples for various regularization strategies such that $\eta = 0.3$ and $q = 160$. (2c) (Left) SSIM versus noise level, (Centre) PSNR versus noise level, (Right) GE versus noise level for various regularization strategies such that $m = 500$ and $q = 160$.

i.i.d. entries sampled from a Gaussian distribution with mean zero and variance $1/q$ and the noise is sampled uniformly from a sphere of radius η [6]. We consider 28×28 greyscale images of handwritten digits taken from the MNIST dataset [53]. We construct a dataset $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{n}$ whereby the q-dimensional measurement vector $\mathbf{y}_i$ is obtained from the p-dimensional vector $\mathbf{x}_i$ – which is derived by converting a 28×28 greyscale image onto a 784 dimensional vector – via the linear model in (1). We also scale the pixel values in the images to the range $[0, 1]$ prior to the application of the linear operator.

For the reconstruction of the original images from the noisy compressive measurements, we consider a 4-layer fully connected neural network consisting of an input layer of width equal to the measurement size – q, followed by three layers, each containing neurons equal to the dimension of the ground truth – p. All the layers except the last one have an associated Rectified Linear Unit (ReLU) activation function.

The reconstruction network is trained using the ADAM optimizer for 600 epochs using various regularization strategies. These strategies include: (a) model aware Spectral norm regularization of Jacobian in (8) which is denoted by SJA&SJ ($\beta_1, \beta_2 > 0$) or only SJA ($\beta_1, \beta_2 = 0$); (b) model-aware Frobenius norm regularization in (10) which is denoted by FJA&FA ($\beta_1, \beta_2 > 0$) or only FJA ($\beta_1, \beta_2 = 0$); and (c) model agnostic regularization approaches such as weight decay (WD), spectral norm regularization of weights (SW) [32], Spectral norm regularization of Jacobian (SJ) and Frobenius norm regularization of Jacobian (FJ) [43]. Note that comparing our regularization strategies with WD, SW, SJ and FJ will allow us to assess the benefits of model-aware regularization since WD, SW, SJ and FJ do not take into account the presence of the linear operator. The regularization parameters appearing in the various strategies (including β_1 and β_2 for our regularizers) are always fine-tuned using a grid search method.

To assess the efficacy of the proposed regularizers on inverse problems with different levels of ill-posedness and corruption, we conduct various experimental studies. Specifically, we look at the performance of networks trained under different regularized loss functions when q is varied such that it takes values in the set $\{80, 160, 320, 640\}$ for $m = 500$ and η fixed at 0.3. Likewise, we also observe the performance of different regularizers when the noise level η is gradually increased from 0 to 0.6 while keeping m and q fixed at 500 and 160 respectively. Finally we also gauge how different regularizers behave under the training sets of size $200, 400, 600$ and 800

[6] We generate bounded noise to validate our theory. However, our experiments (not included in the paper) showcase that the proposed regularization strategies show gains even when the noise is not strictly bounded.

while keeping q fixed at 160 and $\eta = 0.3$.

The reconstruction performance of the various regularization schemes is compared in terms of the generalization gap determined using the generalization error in eq. (4) and other quality metrics such as Structural Similarity Index (SSIM) and Peak Signal to Noise Ratio (PSNR).

2) Results: Fig. 2 presents a performance comparison of networks regularized with our model aware Jacobian regularizers and the baseline techniques for various training scenarios.

In Fig. 2a, we plot the test set SSIM, PSNR and GE of the reconstructed MNIST images versus the number of measurements q. It can be seen that our proposed model-aware strategies lead to performance gains in comparison with existing ones, where the gains are more pronounced with the increase in the number of measurements. This shows that – owing to the explicit exploitation of the forward map – model aware regularizers are better able to leverage the additional measurements. The generalization error between the training and the test set for different measurement sizes also shows a similar trend with model aware regularizers consistently outbeating the competing baseline techniques.

In Fig. 2b, we plot the test set SSIM, PSNR and GE of the reconstructed MNIST images versus number of training examples. Here again, the regularizers that incorporate the knowledge of the forward map outperform the regularization techniques that do not. This result also reinforces the hypothesis that even in situations where we may only have small number of training examples, model-aware regularization can result in a better generalization performance.

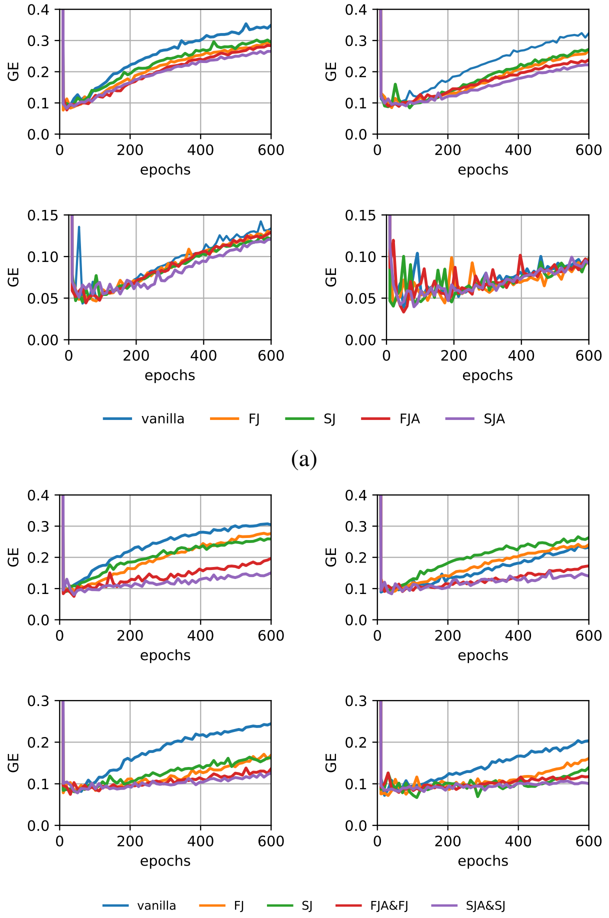
Finally, in Fig. 2c, we study the effect of different regularizers in the presence of different levels of noise. For measurements with high noise levels, the model agnostic and model aware regularizers show similar performance. This is because in low SNR conditions the effect of noise may dominate the effect of the forward operator. In contrast, for measurements with low levels of noise, model aware regularizers show superior performance to the existing model agnostic ones. The GE plot for these experiments again shows that the proposed regularizers results in superior generalization behaviour when the noise levels are low.

It should be noted that SJA&SJA consistently outperforms FJA&FJ. This is because Frobenius norm regularization minimizes the sum of square of all the elements in the matrix – not taking into account the correlation between the rows of the Jacobian – and thus is more restrictive than the Spectral norm regularization.

These results support our analysis that model induced regularizers improve the performance of neural networks over model agnostic regularization translating into better reconstructions.

Algorithm 4: Estimation of the regularization coefficient β for Jacobian regularizer.

```
Input: magnitude $r$ of the regularization term and $l$ of the
       loss over Mini-batch $\mathcal{B}$, scaling factor $\gamma$
Output: Value of the regularization coefficient
$\alpha \leftarrow$ floor(log($^l/_r$)) ;   // $l$ is the unregularized
  empirical loss $^1/_{|\mathcal{B}|}\sum_i l(f_\mathcal{S}, \mathbf{s}_i)$
$\beta \leftarrow {}^{10^\alpha}/_\gamma$ ; // The values of $10, 20$ and $30$ were
  tested for $\gamma$. 20 usually gave the best
  results.
```



(a)

(b)

Fig. 3: Generalization error Vs number of epoch plots for the SR problem using different regularization strategies. (3a) GE plots for EDSR (left) and WDSR (right) when $\eta = 0$. (Top) $^p/_q = 4$ (Bottom) $^p/_q = 2$ SR task. (3b) GE plots for EDSR (left) and WDSR (right) when $\eta = 3$. (Top) $^p/_q = 4$ (Bottom) $^p/_q = 2$.

B. Image Super-resolution

1) Experimental procedure: We now study the performance of our regularizers on the classical super resolution (SR) problem involving the recovery of high resolution images from their low resolution versions. The SR problem can be mathematically formulated via the linear model in (1) where $\mathbf{n}$ represents the measurement noise and the forward operator $\mathbf{A}$ can be defined as the product of a blur matrix $\mathbf{H} \in \mathbb{R}^{p \times p}$ and a subsampling matrix $\mathbf{L} \in \mathbb{R}^{p \times q}$. The point spread function (PSF) of the matrix $\mathbf{H}$ can be uniform, Gaussian or bicubic and is assumed to be known in advance [56]. In our experiments we sample the noise uniformly from a sphere of radius η and assume the PSF to be a 5×5 Gaussian kernel. For

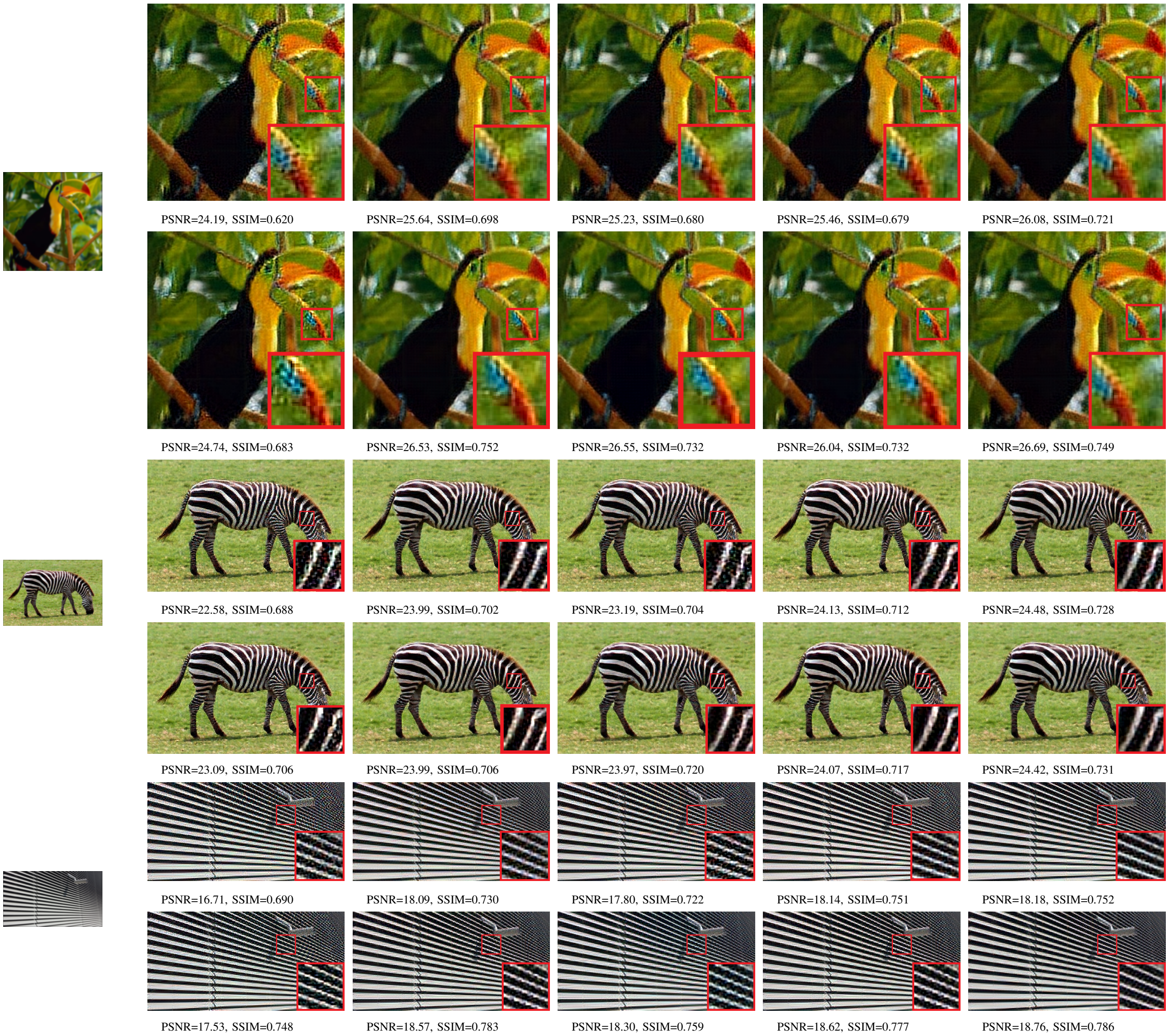


Fig. 4: Sample reconstruction results for the image SR task under a noiseless setting ($\eta = 0$). For each sample, the top row contains HR images recovered using EDSR [54] and the bottom row contains reconstructions for the WDSR [55]. (Left to Right) vanilla, FJ, SJ, FJA and SJA.

our training procedure, we sample images from the BSD300 database [57]. The dataset $\mathcal{S} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^m$ is generated by obtaining $128 \times 128 \times 3$ cutouts from these images; vectorizing them; and then obtaining the q-dimensional measurement $\mathbf{y}_i$ via the linear model in (1). The exact value of q depends on the subsampling ratio $^p/_q$. We test our regularizers for subsampling ratios of 2 and 4. The training set size, m, is fixed to 500 samples, while the regularization parameters are tuned on a validation set of size 3500. We also apply an adaptive policy taking into account feedback from training in order to tune the regularization parameters – as opposed to keeping them fixed – using the approach summarized in Algorithm 4 [7].

To reconstruct the high resolution (HR) images from the low resolution (LR) measurements, we train two state-of-the-art ResNet architectures – the Enhanced Deep Residual Networks (EDSR) [54] and the Wide Activation Residual Networks (WDSR) [55]. These architectures have specially been designed for solving the SR problem leading up to exceptional performance on various datasets and SR challenges. We train these networks using the ADAM optimizer for 600 epochs. In these set of experiments, we compare the performance of the proposed model aware regularizers in eqs. (8) and (10) against their model agnostic counterparts; we

[7] Our empirical results show that using such an adaptive technique results in better validation performance and lesser training time. Since this technique takes into account the training performance to compute the regularization coefficients at each step and does not rely on a hit and trial method to find the 'best' hyperparamter, it results in lesser overall training time for the model.

TABLE II: Reconstruction performance of EDSR and WDSR on various test datasets.

			$p/q = 2$						$p/q = 4$					
			Set5		Set14		Urban100		Set5		Set14		Urban100	
			PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
EDSR	$\eta = 0$	Vanilla	32.63	0.902	29.57	0.849	27.44	0.873	25.87	0.701	24.05	0.655	21.43	0.629
		FJ	32.98	0.913	29.96	0.868	27.92	0.888	26.61	0.731	24.81	0.680	22.28	0.663
		SJ	33.32	0.916	30.33	0.861	28.18	0.885	26.33	0.723	24.54	0.671	22.00	0.650
		FJA	33.05	0.913	30.01	0.862	27.91	0.884	26.65	0.729	24.84	0.675	22.29	0.659
		SJA	**33.60**	**0.922**	**30.66**	**0.876**	**28.68**	**0.910**	**26.90**	**0.745**	**24.97**	**0.685**	**22.37**	**0.665**
	$\eta = 3$	Vanilla	28.60	0.833	26.01	0.744	23.04	0.716	24.75	0.654	23.09	0.575	20.42	0.522
		FJ	28.68	0.822	26.23	0.738	23.35	0.720	24.78	0.656	23.12	0.574	20.47	0.522
		SJ	28.72	0.823	26.14	0.737	23.18	0.709	24.98	0.677	23.26	0.592	20.68	0.542
		FJA+FJ	29.34	0.848	26.73	0.759	23.98	0.751	25.38	0.680	23.59	0.599	20.80	0.547
		SJA+SJ	**29.50**	**0.849**	**26.95**	**0.768**	**24.00**	**0.753**	**25.50**	**0.692**	**23.61**	**0.603**	**20.83**	**0.551**
WDSR	$\eta = 0$	Vanilla	33.76	0.921	30.17	0.869	28.49	0.893	26.19	0.739	24.08	0.670	21.52	0.648
		FJ	34.20	0.925	30.57	0.877	28.76	0.900	26.80	0.757	24.75	0.685	22.11	0.663
		SJ	34.05	0.928	30.69	0.880	28.88	0.902	26.77	0.752	24.71	0.686	22.10	0.662
		FJA	34.25	**0.929**	30.66	0.881	28.94	0.906	27.00	**0.765**	24.85	0.689	**22.22**	**0.669**
		SJA	**34.55**	0.927	**30.79**	**0.881**	**29.01**	**0.905**	**27.06**	0.762	**24.89**	**0.690**	22.20	0.665
	$\eta = 3$	Vanilla	28.63	0.827	26.20	0.751	23.27	0.730	24.95	0.666	23.23	0.588	20.53	0.532
		FJ	29.15	0.841	26.56	0.759	23.85	0.749	25.34	0.699	23.35	0.598	20.74	0.549
		SJ	29.31	0.842	26.64	0.760	23.88	0.752	25.36	0.704	23.39	0.604	20.80	0.558
		FJA+FJ	29.73	0.853	26.87	0.762	24.15	0.757	25.52	0.708	23.60	0.607	20.91	0.563
		SJA+SJ	**29.91**	**0.858**	**27.19**	**0.858**	**24.37**	**0.769**	**25.54**	**0.711**	**23.65**	**0.612**	**20.93**	**0.565**

also demonstrate with the help of generalization error curves how incorporating the knowledge of the forward operator also induces generalization gains.

In order to evaluate the impact of incorporating the knowledge of the forward operator in the proposed regularizers, we test the performance of our trained networks on various publicly available datasets such as Set5, Set14 and Urban 100 dataset. The reconstruction performance of the various schemes is compared in terms of visualizations and quality metrics such as GE, Structural Similarity Index (SSIM) and Peak Signal to Noise Ratio (PSNR).

2) Results: In order to investigate the improvement in the generalization behaviour induced by the proposed model aware regularizers, we have plotted the GE between the training and validation set – computed via eq. (4) for solving the SR tasks in Fig. 3. It can be seen in Figs 3a and 3b that for the subsampling ratio of $p/q = 4$, the regularization methods which incorporate the knowledge of the forward operator outperform the regularization techniques that do not. Our results for the noise free SR problems when $p/q = 2$, – presented in Fig 3a do not exhibit significant gaps in the generalization performance. This is expected since for $p/q = 2$, SR is a comparatively easier recovery problem and therefore exploiting the knowledge of the forward model may not provide much benefit. However, in the presence of noise, regularizing the networks shows improved generalization performance even for $p/q = 2$, as shown in Fig. 3b. These results validate our theory that model aware regularization techniques induce performance gains by reducing the effect of overfitting – resulting in a better generalization behaviour.

Fig. 4 presents a visual comparison of the outputs achieved using model aware Jacobian regularizers and the baseline techniques on various datasets. It can be seen that for both EDSR and WDSR, our propsed regularized leads to perceptual gains in contrast to the standard (vanilla) training. Although the model agnostic regularization techniques also result in improved visualizations, a close inspection of the recovered images reveals that the model aware regularizers are able to recover finer image details. It should be noted that the reconstruction results are achieved with only 500 training samples.

Finally, in Table II, we demonstrate the effectiveness of the proposed regularizers on various out of sample datasets. On the $p/q = 2$ SR task – in comparison to vanilla training for both the EDSR and WDSR – the model aware regularization techniques result in a gain of up to 1.24 dB and 0.04 in terms of the PSNR and SSIM respectively. The proposed model aware regularizers also show improvement over their model agnostic counterparts. A similar trend can be observed in the $p/q = 4$ SR task where model aware regularizers achieve a performance gain of upto 0.97dB and 0.036 in terms of PSNR and SSIM respectively . The performance improvement over the vanilla training in the WDSR network are slightly less pronounced than EDSR but still noticeable. This is because WDSR is a more competetive network than EDSR.

These results support our analysis that model induced regularizers improve the performance of neural networks over model agnostic regularization translating into better reconstructions.

C. k-Space Subsampled Measurements

1) Experimental procedure: Our second set of experiments involves the reconstruction of MRI images from sub-sampled Fourier measurements. Our linear model is such that the linear operator in (1) is given by $\mathbf{A} = \mathbf{F}^{-1}\mathbf{M}\mathbf{F}$ where $\mathbf{F}$ is a 2D Fourier transform matrix, $\mathbf{F}^{-1}$ is the 2D inverse Fourier transform matrix, and the mask $\mathbf{M}$ is diagonal matrix containing binary entries on its diagonal where the fraction of non-zero entries signify the subsampling ratio s. [8] Our linear model is also such that the noise for each sample in (1) is sampled uniformly from an ℓ_2 sphere of radius η.

We construct our dataset by retrospectively under-sampling the Fourier transform of the ground truth images, obtained from the NYU fastMRI's knee database [58]. The subsampling is achieved by the Cartesian 1D and 2D random sampling masks in k-space, retaining only 25% and 20% of the total Fourier samples, respectively. We also normalize the images to the range $[0, 1]$ before applying the forward transform and adding noise with level $\eta = 5$. The training was achieved using a set of 500 samples and a minibatch of size 5.

We consider the state-of-the-art UNet architecture [59] under different regularization strategies (including SJA&SJ and FJA&FJ) to reconstruct the original images from the noisy under-sampled Fourier measurements. A schematic of the network architecture is shown in Fig. 5. This network is trained using the ADAM optimizer for 300 epochs using the different regularization strategies. However, we only apply the regularization in 10% of the steps per epoch in order to speed up the optimization. We also use Algorithm 4 to tune the hyperparameters β_1 and β_2.

We also consider for comparison purposes competing techniques such as (a) wavelet sparsity regularized reconstruction [3]; (b) adversarial regularizers [24]; and (c) postprocessing via UNet method [18]. For a fair comparison, the UNet architecture and training routines are kept the same for our work and the postprocessing method. Note also that unlike [18], we train the post processing UNet on ℓ_2 loss function. For the Adversarial regularization method, we modify the official implementation of the adversarial regularizer, present on Github [24], provided by the authors of the publication to suit the forward model used in this work. The batch size and other hyperparameters such as the step size and the choice of the adversarial regularizer network were kept the same as in the original implementation. Both the postprocessing and the adversarial regularization method involve a 'preprocessing' step. That is, both techniques obtain an initial course estimate of the signal of interest by applying a classical regularized reconstruction method, $\mathbf{A}^{\dagger}(\cdot)$ to the measurement $\mathbf{y}$. For our experiments, we use the output of the Wavelet sparsity regularized reconstruction method as this initial estimate.

We compare once again the reconstruction performance of the various approaches in terms of GE, visualizations and quality metrics such as Structural Similarity Index (SSIM) and Peak Signal to Noise Ratio (PSNR).

2) Results: Table III compares the performance of the different reconstruction approaches. The proposed regularizers consistently outbeat all the other methods in terms of PSNR and SSIM. The performance gains are more pronounced for 1D sampling mask in view of the fact that these introduce aliasing artifacts that appear to be better dealt with with our approaches in relation to competing ones.

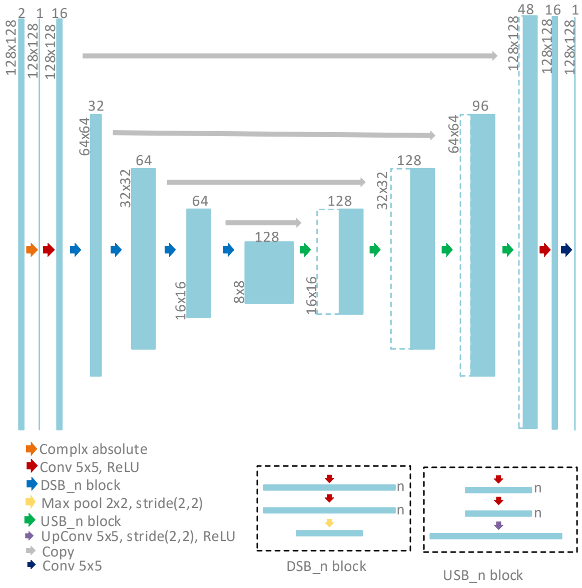


Fig. 5: UNet Architecture.

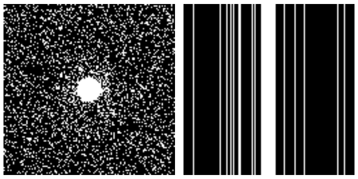


Fig. 6: k-space acquisition masks (Left): Random 2D 5-fold subsampling mask with the centre fully sampled.(Right) Random 1D 4-fold subsampling mask.

In Fig. 7, we plot and compare the generalization error performance of the competing deep learning approaches including reconstruction using an unregularized UNet architecture, UNet post-processing method [18] and UNet regularized with our proposed model aware regularization approaches [9]. The GE curves clearly reinforce our theoretical results and demonstrate that our regularizers results in an improved generalization behaviour. The networks regularized with SJA&SJ behave significantly better than the other unregualized networks for both 2D and 1D mask. However, UNet with FJA&FJ does not show significant reduction in generalization error in comparison to the vanilla UNet and UNet postprocessor for the 1D mask. This may be because the 1D subsampling mask induces the aliasing artifacts and therefore is potentially a challenging recovery problem.

Figs. 8 and 9 in turn offer a visual comparison of the quality of the reconstructed images for the different approaches. It can be seen that our proposed regularization approaches appear to lead to better reconstruction quality in relation to the competing methods. Since the Jacobian regularization method can be directly used with any deep learning based reconstruction method, we also include reconstruction results when a postprocessing

[8] These types of forward mappings are particularly important for applications such as accelerated MRI reconstruction where the field of view is scanned by obtaining sparse measurements in k-space domain leading to reduced MRI acquisition periods.

[9] The adversarial regularizer method [24] employs a regularized iterative gradient descent algorithm to recover the ground truth.

TABLE III: Comparison of different reconstruction metrics using the different approaches.

	2D mask ($s = 0.2$)		1D mask ($s = 0.25$)	
	PSNR	SSIM	PSNR	SSIM
Wavelet sparsity reg	28.49	0.72	24.56	0.50
Adversarial Regularizer [24]	29.89	0.77	25.44	0.54
UNet as post-processor [18]	30.01	0.79	28.36	0.74
UNet w FJA&FJ	30.80	0.80	28.96	0.75
UNet w SJA&SJ	30.89	0.81	29.30	0.78

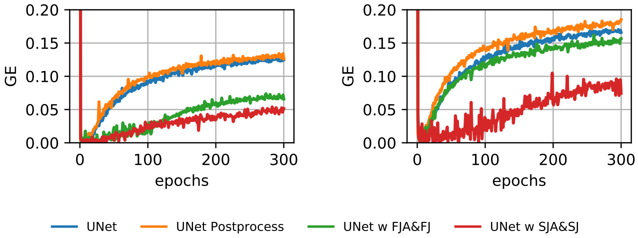


Fig. 7: Generalization error plots for the MRI reconstruction.(left) 2D acquisition mask ($s = 0.2$) (right) 1D acquisition mask ($s = 0.25$).

UNet is regularized via SJA&SJ regularizer. It can be seen that perceptually the reconstruction achieved through this method outperforms all the other techniques. However there is no improvement in terms of PSNR and SSIM over the UNet with SJA&SJ (without the preprocessing). A close inspection of the reconstructed images reveals that the proposed method introduces less artifacts than the other reconstructions.

VII. Conclusion

This paper – leveraging knowledge of underlying physical models – proposes a new deep learning approach to solve inverse problems. The crux of the approach – stemming directly from a rigorous generalization error analysis – is a new neural network learning procedure involving the use of cost functions in capturing knowledge of the underlying inverse problem model via appropriate regularization. This regularizer, owing to its plug-and-play nature can be integrated into any deep learning based solver of inverse problems without extra hassle. Empirical results on a variety of problems have shown that our proposed regularization approach can outperform considerably standard model agnostic regularizers and reconstruction schemes specialized for inverse problems. This work adds to recent ones by showing there is much value incorporating model knowledge onto data-driven approaches.

References

[1] Heinz Werner Engl, Martin Hanke, and Andreas Neubauer. *Regularization of inverse problems*, volume 375. Springer Science & Business Media, 1996.

[2] Jennifer L Mueller and Samuli Siltanen. *Linear and nonlinear inverse problems with practical applications*, volume 10. Siam, 2012.

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

[4] Alice Lucas, Michael Iliadis, Rafael Molina, and Aggelos K Katsaggelos. Using deep neural networks for inverse problems in imaging: beyond analytical methods. *IEEE Signal Processing Magazine*, 35(1):20–36, 2018.

[5] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992.

[6] Andrey N Tikonov and Vasily Y Arsenin. Solutions of ill-posed problems. *New York: Winston*, 1977.

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

[8] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[9] Gaofeng Wang, Jun Zhang, and Guang-Wen Pan. Solution of inverse problems in image processing by wavelet expansion. *IEEE Transactions on Image Processing*, 4(5):579–593, 1995.

[10] Simon Arridge, Peter Maass, Ozan Öktem, and Carola-Bibiane Schönlieb. Solving inverse problems using data-driven models. *Acta Numerica*, 28:1–174, 2019.

[11] Xiaojiao Mao, Chunhua Shen, and Yu-Bin Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In *Advances in neural information processing systems*, pages 2802–2810, 2016.

[12] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017.

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

[14] Chang Min Hyun, Hwa Pyung Kim, Sung Min Lee, Sungchul Lee, and Jin Keun Seo. Deep learning for undersampled mri reconstruction. *Physics in Medicine & Biology*, 63(13):135007, 2018.

[15] Alexander Selvikvåg Lundervold and Arvid Lundervold. An overview of deep learning in medical imaging focusing on mri. *Zeitschrift für Medizinische Physik*, 29(2):102–127, 2019.

[16] Junyoung Park, Donghwi Hwang, Kyeong Yun Kim, Seung Kwan Kang, Yu Kyeong Kim, and Jae Sung Lee. Computed tomography super-resolution using deep convolutional neural network. *Physics in Medicine & Biology*, 63(14):145011, 2018.

[17] Sarah B Scruggs, Karol Watson, Andrew I Su, Henning Hermjakob, John R Yates III, Merry L Lindsey, and Peipei Ping. Harnessing the heart of big data. *Circulation research*, 116(7):1115–1119, 2015.

[18] Kyong Hwan Jin, Michael T McCann, Emmanuel Froustey, and Michael Unser. Deep convolutional neural network for inverse problems in imaging. *IEEE Transactions on Image Processing*, 26(9):4509–4522, 2017.

[19] Eunhee Kang, Junhong Min, and Jong Chul Ye. A deep convolutional neural network using directional wavelets for low-dose x-ray ct reconstruction. *Medical physics*, 44(10):e360–e375, 2017.

[20] Yoseob Han and Jong Chul Ye. Framing u-net via deep convolutional framelets: Application to sparse-view ct. *IEEE transactions on medical imaging*, 37(6):1418–1429, 2018.

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

[22] Vishal Monga, Yuelong Li, and Yonina C Eldar. Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. *arXiv preprint arXiv:1912.10557*, 2019.

[23] Jonas Adler and Ozan Öktem. Learned primal-dual reconstruction. *IEEE transactions on medical imaging*, 37(6):1322–1332, 2018.

[24] Sebastian Lunz, Ozan Öktem, and Carola-Bibiane Schönlieb. Adversarial regularizers in inverse problems. In *Advances in Neural Information Processing Systems*, pages 8507–8516, 2018. https://github.com/lunz-s/DeepAdverserialRegulariser/.

[25] Ashish Bora, Ajil Jalal, Eric Price, and Alexandros G Dimakis. Compressed sensing using generative models. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 537–546. JMLR. org, 2017.

[26] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9446–9454, 2018.

[27] Yaniv Romano, Michael Elad, and Peyman Milanfar. The little engine that could: Regularization by denoising (red). *SIAM Journal on Imaging Sciences*, 10(4):1804–1844, 2017.

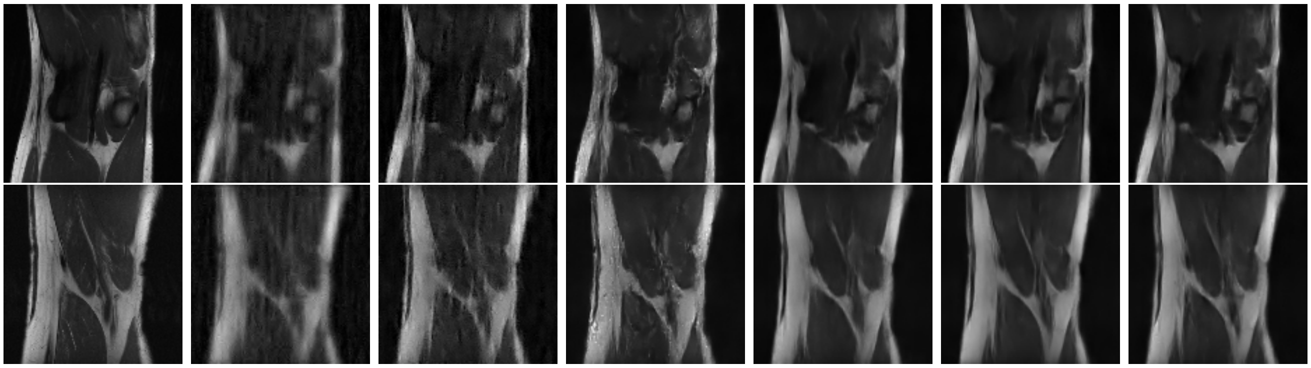


Fig. 8: Sample results from the reconstruction from k-space subsampled measurements acquired using a 2D acquisition mask in Fig. 6. (Left to Right) ground truth, reconstruction using Wavelet sparsity regularization, Adversarial Regularizer [24], postprocessing using UNet [18], UNet with FJA&FJ and UNet with SJA&SJ, postprocessing using UNet [18] regularized with SJA&SJ.

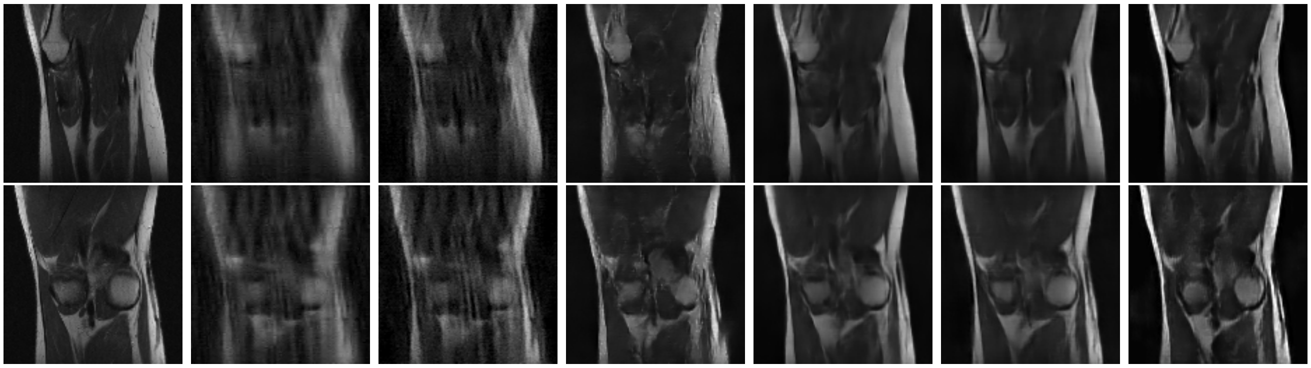


Fig. 9: Sample results from the reconstruction from k-space subsampled measurements acquired using a 1D acquisition mask in Fig. 6. (Left to Right) ground truth, reconstruction using Wavelet sparsity regularization, Adversarial Regularizer [24], postprocessing using UNet [18], UNet with FJA&FJ and UNet with SJA&SJ, postprocessing using UNet [18] regularized with SJA&SJ.

[28] Singanallur V Venkatakrishnan, Charles A Bouman, and Brendt Wohlberg. Plug-and-play priors for model based reconstruction. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 945–948. IEEE, 2013.

[29] Behnam Neyshabur, Zhiyuan Li, Srinadh Bhojanapalli, Yann LeCun, and Nathan Srebro. Towards understanding the role of over-parametrization in generalization of neural networks. *arXiv preprint arXiv:1805.12076*, 2018.

[30] Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*, pages 6240–6249, 2017.

[31] Jaweria Amjad, Jure Sokolić, and Miguel RD Rodrigues. On deep learning for inverse problems. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 1895–1899. IEEE, 2018.

[32] Yuichi Yoshida and Takeru Miyato. Spectral norm regularization for improving the generalizability of deep learning. *arXiv preprint arXiv:1705.10941*, 2017.

[33] Henry Gouk, Eibe Frank, Bernhard Pfahringer, and Michael Cree. Regularisation of neural networks by enforcing lipschitz continuity. *arXiv preprint arXiv:1804.04368*, 2018.

[34] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

[35] Aladin Virmaux and Kevin Scaman. Lipschitz regularity of deep neural networks: analysis and efficient estimation. In *Advances in Neural Information Processing Systems*, pages 3835–3844, 2018.

[36] Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. *arXiv preprint arXiv:1704.08847*, 2017.

[37] Harris Drucker and Yann Le Cun. Improving generalization performance using double backpropagation. *IEEE Transactions on Neural Networks*, 3(6):991–997, 1992.

[38] Jure Sokolic, Raja Giryes, Guillermo Sapiro, and Miguel RD Rodrigues. Robust large margin deep neural networks. *IEEE Transactions on Signal Processing*, 2017.

[39] Cem Anil, James Lucas, and Roger Grosse. Sorting out lipschitz function approximation. In *International Conference on Machine Learning*, pages 291–301, 2019.

[40] Christian Etmann. A closer look at double backpropagation. *arXiv preprint arXiv:1906.06637*, 2019.

[41] Kui Jia, Shuai Li, Yuxin Wen, Tongliang Liu, and Dacheng Tao. Orthogonal deep neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[42] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pages 5767–5777, 2017.

[43] Judy Hoffman, Daniel A Roberts, and Sho Yaida. Robust learning with jacobian regularization. *arXiv preprint arXiv:1908.02729*, 2019.

[44] Behnam Neyshabur. Implicit regularization in deep learning. *arXiv preprint arXiv:1709.01953*, 2017.

[45] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[46] Huan Xu and Shie Mannor. Robustness and generalization. *Machine learning*, 86(3):391–423, 2012.

[47] Colin Wei and Tengyu Ma. Data-dependent sample complexity of deep neural networks via lipschitz augmentation. *arXiv preprint arXiv:1905.03684*, 2019.

[48] Nakul Verma. Distance preserving embeddings for general n-dimensional manifolds. *The Journal of Machine Learning Research*, 14(1):2415–2448, 2013.

[49] Roman Novak, Yasaman Bahri, Daniel A Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. Sensitivity and generalization in neural networks: an empirical study. *arXiv preprint arXiv:1802.08760*, 2018.

[50] RV Mises and Hilda Pollaczek-Geiringer. Praktische verfahren der

gleichungsauflösung. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 9(2):152–164, 1929.

[51] Jamie Townsend. A new trick for calculating Jacobian vector products. https://j-towns.github.io/2017/06/12/A-new-trick.html, 2017. Accessed: 2020-01-17.

[52] Graeme Pope. Compressive sensing: A summary of reconstruction algorithms. Master's thesis, ETH, Swiss Federal Institute of Technology Zurich, Department of Computer ..., 2009.

[53] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/, 2010.

[54] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017.

[55] Jiahui Yu, Yuchen Fan, Jianchao Yang, Ning Xu, Zhaowen Wang, Xinchao Wang, and Thomas Huang. Wide activation for efficient and accurate image super-resolution. *arXiv preprint arXiv:1808.08718*, 2018.

[56] Tomer Peleg and Michael Elad. A statistical prediction model based on sparse representations for single image super-resolution. *IEEE transactions on image processing*, 23(6):2569–2582, 2014.

[57] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 416–423. IEEE, 2001.

[58] Jure Zbontar, Florian Knoll, Anuroop Sriram, Tullie Murrell, Zhengnan Huang, Matthew J. Muckley, Aaron Defazio, Ruben Stern, Patricia Johnson, Mary Bruno, Marc Parente, Krzysztof J. Geras, Joe Katsnelson, Hersh Chandarana, Zizhao Zhang, Michal Drozdzal, Adriana Romero, Michael Rabbat, Pascal Vincent, Nafissa Yakubova, James Pinkerton, Duo Wang, Erich Owens, C. Lawrence Zitnick, Michael P. Recht, Daniel K. Sodickson, and Yvonne W. Lui. fastmri: An open dataset and benchmarks for accelerated mri, 2018.

[59] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[60] Jean Gallier. Notes on convex sets, polytopes, polyhedra, combinatorial topology, voronoi diagrams and delaunay triangulations, 2008.

[61] B Bolzano. Functionenlehre, edited by k. *Rychlik. Royal Bohemian Academy of Sciences, Prague*, 1930.

[62] Maurice Fréchet. *Généralisation d'un théorème de Weierstrass*. gauthier-Villars, 1904.

[63] Andrew Tonge. Equivalence constants for matrix norms: a problem of goldberg. *Linear Algebra and its Applications*, 306(1-3):1–13, 2000.

[64] Andrew D Lewis. A top nine list: Most popular induced matrix norms. *Queen's University, Kingston, Ontario, Tech. Rep*, 2010.

APPENDIX

Proof of Theorem 1. We first note that the line between $\mathbf{y}_1 = \mathbf{A}\mathbf{x}_1 + \mathbf{n}_1$ and $\mathbf{y}_2 = \mathbf{A}\mathbf{x}_2 + \mathbf{n}_2$ is given by $\bar{\theta}\mathbf{y}_1 + \theta\mathbf{y}_2$ where $\theta \in (0,1)$ and $\bar{\theta} = 1 - \theta$. Let us now define a function $h(\theta)$ as follows:

$$h(\theta) = f_{\mathcal{S}}(\bar{\theta}\mathbf{y}_1 + \theta\mathbf{y}_2) = f_{\mathcal{S}}\left(\mathbf{A}\left(\bar{\theta}\mathbf{x}_1 + \theta\mathbf{x}_2\right) + \bar{\theta}\mathbf{n}_1 + \theta\mathbf{n}_2\right)$$

By the generalized fundamental theorem of calculus, it can be shown that:

$$f_{\mathcal{S}}(\mathbf{y}_2) - f_{\mathcal{S}}(\mathbf{y}_1) = \int_0^1 \frac{dh(\theta)}{d\theta} d\theta$$

where

$$\frac{d}{d\theta}(h(\theta)) = \mathbf{J}\left(\bar{\theta}\mathbf{y}_1 + \theta\mathbf{y}_2\right)\left[\mathbf{A}\left(\mathbf{x}_2 - \mathbf{x}_1\right) + \left(\mathbf{n}_2 - \mathbf{n}_1\right)\right]$$

Then, from the sub-multiplicative property of matrix norms, it is immediate to show that:

$$\begin{aligned}
&\|f_{\mathcal{S}}(\mathbf{y}_2) - f_{\mathcal{S}}(\mathbf{y}_1)\|_2 \\
&= \left\| \int_0^1 \mathbf{J}\left(\bar{\theta}\mathbf{y}_1 + \theta\mathbf{y}_2\right)\left[\mathbf{A}\left(\mathbf{x}_2 - \mathbf{x}_1\right) + \left(\mathbf{n}_2 - \mathbf{n}_1\right)\right] d\theta \right\|_2 \\
&\leq \left\| \int_0^1 \mathbf{J}\left(\bar{\theta}\mathbf{y}_1 + \theta\mathbf{y}_2\right) \mathbf{A}(\mathbf{x}_2 - \mathbf{x}_1) d\theta \right\|_2 \\
&\qquad + \left\| \int_0^1 \mathbf{J}\left(\bar{\theta}\mathbf{y}_1 + \theta\mathbf{y}_2\right)(\mathbf{n}_2 - \mathbf{n}_1) d\theta \right\|_2 \\
&\leq \left\| \int_0^1 \mathbf{J}\left(\bar{\theta}\mathbf{y}_1 + \theta\mathbf{y}_2\right) \mathbf{A} d\theta \right\|_2 \|\mathbf{x}_2 - \mathbf{x}_1\|_2 \\
&\qquad + \left\| \int_0^1 \mathbf{J}\left(\bar{\theta}\mathbf{y}_1 + \theta\mathbf{y}_2\right) d\theta \right\|_2 \|\mathbf{n}_2 - \mathbf{n}_1\|_2
\end{aligned}$$

It is also possible to show that:

$$\begin{aligned}
\left\| \int_0^1 \mathbf{J}\left(\bar{\theta}\mathbf{y}_1 + \theta\mathbf{y}_2\right) \mathbf{A} d\theta \right\|_2 &\overset{(a)}{\leq} \int_0^1 \|\mathbf{J}\left(\bar{\theta}\mathbf{y}_1 + \theta\mathbf{y}_2\right)\mathbf{A}\|_2 d\theta \\
&\leq \sup_{\substack{\mathbf{y}_1, \mathbf{y}_2 \in \mathcal{Y} \\ \theta \in [0,1]}} \|\mathbf{J}\left(\bar{\theta}\mathbf{y}_1 + \theta\mathbf{y}_2\right)\mathbf{A}\|_2
\end{aligned}$$

Therefore, given that $\bar{\theta}\mathbf{y}_1 + \theta\mathbf{y}_2$ is in convex-hull of $\mathcal{Y}$ for $\theta \in [0,1]$, it follows immediately that:

$$\begin{aligned}
\|f_{\mathcal{S}}(\mathbf{y}_2) - f_{\mathcal{S}}(\mathbf{y}_1)\|_2 &\leq \sup_{\mathbf{y} \in conv(\mathcal{Y})} \|\mathbf{J}(\mathbf{y})\mathbf{A}\|_2 \|\mathbf{x}_2 - \mathbf{x}_1\|_2 \\
&\qquad + \sup_{\mathbf{y} \in conv(\mathcal{Y}})\|\mathbf{J}(\mathbf{y})\|_2 \|\mathbf{n}_2 - \mathbf{n}_1\|_2 \\
&\leq \sup_{\mathbf{y} \in conv(\mathcal{Y})} \|\mathbf{J}(\mathbf{y})\mathbf{A}\|_2 \|\mathbf{x}_2 - \mathbf{x}_1\|_2 \\
&\qquad + 2\eta \sup_{\mathbf{y} \in conv(\mathcal{Y})} \|\mathbf{J}(\mathbf{y})\|_2 \qquad (11)
\end{aligned}$$

where $conv(\mathcal{Y})$ represents the convex hull of $\mathcal{Y}$. □

Proof of Theorem 2. We can construct a finite δ-cover $\mathcal{X}' = \{\mathbf{x}'_i, i = 1, \ldots, K\}$ of the compact space $\mathcal{X}$ with $K \leq \mathcal{N}_{\mathcal{X}}(\delta, \ell_2)$. We can therefore also construct a finite cover $\mathcal{D}' = \{\mathbf{s}'_i = (\mathbf{x}'_i, \mathbf{A}\mathbf{x}'_i), \mathbf{x}'_i \in \mathcal{X}', i = 1, \ldots K\}$ of the space $\mathcal{D} = \{\mathbf{s} = (\mathbf{x}, \mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}) : \mathbf{x} \in \mathcal{X}, \mathbf{n} \in \mathcal{N}\}$.

This implies that the sample space $\mathcal{D}$ can be partitioned into K disjoint subsets $\mathcal{K}_i, i = 1, \ldots, K$ where $\mathcal{K}_i$ corresponds to the Voronoi region of $\mathbf{s}'_i = (\mathbf{x}'_i, \mathbf{y}'_i = \mathbf{A}\mathbf{x}'_i), \mathbf{x}'_i \in \mathcal{X}'$. Consequently, for a point $\mathbf{s} = (\mathbf{x}, \mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n})$ taken from the subset $\mathcal{K}_i$ we can guarantee:

$$\|\mathbf{x}'_i - \mathbf{x}\|_2 \leq \delta \qquad (12)$$

Let us now choose $\mathbf{s}_1 = (\mathbf{x}_1, \mathbf{y}_1 = \mathbf{A}\mathbf{x}_1 + \mathbf{n}_1) \in \mathcal{S}$ and $\mathbf{s}_2 = (\mathbf{x}_2, \mathbf{y}_2 = \mathbf{A}\mathbf{x}_2 + \mathbf{n}_2) \in \mathcal{D}$ from a particular subset in our partitioned $\mathcal{D}$. Then,

$$\begin{aligned}
&|l(f_{\mathcal{S}}, \mathbf{s}_2) - l(f_{\mathcal{S}}, \mathbf{s}_1)| \\
&= |\|\mathbf{x}_2 - f_{\mathcal{S}}(\mathbf{y}_2)\|_2 - \|\mathbf{x}_1 - f_{\mathcal{S}}(\mathbf{y}_1)\|_2| \\
&\overset{(a)}{\leq} \|\mathbf{x}_2 - \mathbf{x}_1\|_2 + \|f_{\mathcal{S}}(\mathbf{y}_2) - f_{\mathcal{S}}(\mathbf{y}_1)\|_2 \\
&\overset{(b)}{\leq} \|\mathbf{x}_2 - \mathbf{x}_1\|_2 + \Lambda_{f \circ a}\|\mathbf{x}_2 - \mathbf{x}_1\|_2 + \Lambda_f \|\mathbf{n}_2 - \mathbf{n}_1\|_2 \\
&\overset{(c)}{\leq} 2(1 + \Lambda_{f \circ a})\delta + 2\Lambda_f \eta \qquad (13)
\end{aligned}$$

where the inequality (a) is due to reverse triangle inequality and Minkowski-inequality, (b) holds because of Theorem 1. Finally (c) holds due to (12) and because η upper bounds the ℓ_2 norm of noise.

We have therefore shown that we can partition the set $\mathcal{D}$ onto K non-overlapping subsets so that if a training sample $\mathbf{s}_1 \in \mathcal{S}$ and another sample $\mathbf{s}_2 \in \mathcal{D}$ belong to the same subset then (13) holds [10]. □

Proof of Theorem 3. We first establish a simple Lemma.

Lemma 1. The Lipschitz constant of a differentiable function f on a compact set $\mathcal{Z}$ is bounded.

Proof. Let $f : \mathbb{R}^p \to \mathbb{R}^q$ be a differentiable function, defined on a compact set $\mathcal{Z} \subseteq \mathbb{R}^p$. Let also $g(\theta) = f(\mathbf{z}' + \theta(\mathbf{z}'' - \mathbf{z}'))$, for some $\theta \in [0, 1]$, so that $g(0) = f(\mathbf{z}')$ and $g(1) = f(\mathbf{z}'')$ where $\mathbf{z}', \mathbf{z}''$ are any two fixed points taken for $\mathcal{Z}$. Then, by the fundamental theorem of calculus, we have

$$f(\mathbf{z}') - f(\mathbf{z}'') = \int_0^1 \mathbf{J}(\mathbf{z}' + \theta(\mathbf{z}'' - \mathbf{z}'))d\theta(\mathbf{z}' - \mathbf{z}'')$$

where $\mathbf{J}(\mathbf{z})$ is the Jacobian matrix of f at $\mathbf{z}$.

From the multiplicative property of norms, we also have that

$$\|f(\mathbf{z}') - f(\mathbf{z}'')\| \le \left\| \int_0^1 \mathbf{J}(\mathbf{z}' + \theta(\mathbf{z}'' - \mathbf{z}'))d\theta \right\|_2 \|\mathbf{z}' - \mathbf{z}''\|_2$$

Next, by the triangle inequality for integrals, it can be shown that

$$\begin{aligned}\left\| \int_0^1 \mathbf{J}(\mathbf{z}' + \theta(\mathbf{z}'' - \mathbf{z}'))d\theta \right\|_2 &\le \sup_{\substack{\mathbf{z}', \mathbf{z}'' \in \mathcal{Z} \\ \theta \in [0,1]}} \|\mathbf{J}(\mathbf{z}' + \theta(\mathbf{z}'' - \mathbf{z}'))\|_2 \\ &\le \sup_{\mathbf{z} \in conv(\mathcal{Z})} \|\mathbf{J}(\mathbf{z})\|_2\end{aligned}$$

where $conv(\mathcal{Z})$ represents the convex hull of the compact set $\mathcal{Z}$. Note that the Carathéodory's theorem of convex hulls can be used to prove that the convex hull of compact set in a finite dimensional space $\mathbb{R}^p$ is also compact [60].

Next, for a continuous function f defined on a compact set, there exists a finite λ_0 such that [61], [62].

$$\left| \frac{\partial}{dz_j}(f(\mathbf{z})_i) \right| \le \lambda_0 \tag{14}$$

where $\frac{\partial}{dz_j}(f(\mathbf{z})_i)$ is the element at row (i, j)-th element of the Jacobian matrix $\mathbf{J}$. This, then leads to the following

$$\sup_{\mathbf{z} \in conv(\mathcal{Z})} \|\mathbf{J}(\mathbf{z})\|_2 \overset{(a)}{\le} \sup_{\mathbf{z} \in conv(\mathcal{Z})} c\|\mathbf{J}(\mathbf{z})\|_\infty \overset{(b)}{\le} cp\lambda_0$$

where (a) is due to the equivalence of matrix norms and c is a constant dependent on the dimensions of the Jacobian matrix [63]. Finally the last inequality follows from the definition of the $\|.\|_\infty$ matrix norm [64]. □

We are now in a position to prove the Theorem. In particular, it can be shown that the GE of a $(K, \epsilon(\mathcal{S}))$-robust deep neural network, with probability greater than $1 - \zeta$, obeys [46]

$$\begin{aligned}GE &\le \epsilon(\mathcal{S}) \\ &\quad + \max_{\mathbf{s}} |l(f_{\mathcal{S}}, \mathbf{s})| \sqrt{\frac{2K\log(2) + 2\log(1/\zeta)}{m}}\end{aligned} \tag{15}$$

We can immediately use the robustness result in Theorem 2 to determine two quantities in this generalization error bound: $\epsilon(\mathcal{S})$ and K. However – in contrast with existing results that assume that the loss function is uniformly bounded so that $\max_{\mathbf{s}} |l(f_{\mathcal{S}}, \mathbf{s})| \le M < \infty$ (e.g. see [46]) – the loss function associated with our inverse problem is not necessarily bounded. However, it is still possible to show that $\max_{\mathbf{s}} |l(f_{\mathcal{S}}, \mathbf{s})|$ is finite.

In particular, let us observe that $\forall\ \mathbf{s} = (\mathbf{x}, \mathbf{y}), \mathbf{s}' = (\mathbf{x}', \mathbf{y}') \in \mathcal{D}$

$$\begin{aligned}|l(f_{\mathcal{S}}, \mathbf{s}) - l(f_{\mathcal{S}}, \mathbf{s}')| &= \big| \|\mathbf{x} - f_{\mathcal{S}}(\mathbf{y})\|_2 - \|\mathbf{x}' - f_{\mathcal{S}}(\mathbf{y}')\|_2 \big| \\ &\le \|\mathbf{x} - \mathbf{x}'\|_2 + \|f_{\mathcal{S}}(\mathbf{y}) - f_{\mathcal{S}}(\mathbf{y}')\|_2 \\ &\overset{(a)}{\le} \|\mathbf{x} - \mathbf{x}'\|_2 + \Lambda_f \|\mathbf{y} - \mathbf{y}'\|_2 \\ &\overset{(b)}{\le} (1 + \Lambda_f)\|\mathbf{s} - \mathbf{s}'\|_2\end{aligned}$$

where (a) is due to Corollary 2 in [38] and (b) holds because the metric on $\mathcal{D}$ upper bounds the metrics on constituent metric spaces $\mathcal{X}$ and $\mathcal{Y}$.

Let us also observe that the Lipschitz constant of the loss function is finite because – via Lemma 1 – the Lipschitz constant of the neural network Λ_f is also finite.

This immediately implies that the loss function is Lipschitz continuous hence continuous, and – by the Extreme Value theorem [62] – that it is also bounded on $\mathcal{D}$, so that $\max_{\mathbf{s}} |l(f_{\mathcal{S}}, \mathbf{s})| \le M < \infty$.

The Theorem follows immediately from Theorem 2. □

[10] Note that this bounding technique produces a slightly different bound than an alternative one where we would bound the second term $\|f_{\mathcal{S}}(\mathbf{y}_2) - f_{\mathcal{S}}(\mathbf{y}_1)\|_2$ on the right hand side of 13(a) by $\Lambda_f\|\mathbf{y}_2 - \mathbf{y}_1\|$ (instead of $\Lambda_{foa}\|\mathbf{x}_2 - \mathbf{x}_1\|$ which is possible via Theorem 1). However, the proposed bounding technique results in a tighter characterization of $\epsilon(\mathcal{S})$ since $\Lambda_{foa} = \sup_{\mathbf{y} \in conv(\mathcal{Y})} \|\mathbf{J}(\mathbf{y})\,\mathbf{A}\|_2 \le \sup_{\mathbf{y} \in conv(\mathcal{Y})} \|\mathbf{J}(\mathbf{y})\|_2 \|\mathbf{A}\|_2 = \Lambda_f \Lambda_a$.