Deep Equilibrium Architectures for Inverse Problems in Imaging

Davis Gilton, Gregory Ongie, and Rebecca Willett

Abstract—**Recent efforts on solving inverse problems in imaging via deep neural networks use architectures inspired by a fixed number of iterations of an optimization method. The number of iterations is typically quite small due to difficulties in training networks corresponding to more iterations; the resulting solvers cannot be run for more iterations at test time without incurring significant errors. This paper describes an alternative approach corresponding to an *infinite* number of iterations, yielding a consistent improvement in reconstruction accuracy above state-of-the-art alternatives and where the computational budget can be selected at test time to optimize context-dependent trade-offs between accuracy and computation. The proposed approach leverages ideas from Deep Equilibrium Models, where the fixed-point iteration is constructed to incorporate a known forward model and insights from classical optimization-based reconstruction methods.**

I. Introduction

A collection of recent efforts surveyed in [1] consider the problem of using training data to solve inverse problems in imaging. Specifically, imagine we observe a corrupted set of measurements y of an image $x^\star$ under a linear measurement operator A with some noise ε according to

$$y = Ax^\star + \varepsilon. \qquad (1)$$

Our task is to compute an estimate of $x^\star$ given measurements y and knowledge of A. This task is particularly challenging when the inverse problem is *ill-posed*, *i.e.,* when the system is underdetermined or ill-conditioned, in which case simple methods such as least squares estimation may not have a unique solution or may produce estimates that are highly sensitive to noise.

Decades of research has explored geometric models of image structure that can be used to regularize solutions to this inverse problem, including [2], [3], [4] and many others. More recent efforts have focused instead on using large collections of training images, $\{x_i^*\}_{i=1}^n$, to *learn* effective regularizers.

One particularly popular and effective approach involves augmenting standard iterative inverse problem solvers with learned deep networks. This approach, which we refer to as **deep unrolling (DU)**, is reviewed in §II-A. The basic idea is to build an architecture that mimics a small number of iterations

of an iterative algorithm. In practice, the number of iterations is quite small (typically 5-10) because of memory constraints and numerical precision issues arising in backpropagation. This paper sidesteps this key limitation of deep unrolling methods with a novel approach based on **deep equilibrium models (DEMs)** [5], which are designed for training arbitrarily deep networks. The result is a novel approach to training networks to solve inverse problems in imaging that **yields a consistent improvement in performance above state-of-the-art alternatives** and where **the computational budget can be selected at test time** to optimize context-dependent trade-offs between accuracy and computation. The key empirical findings, which are detailed in §VI-D, are illustrated in Fig. 1.

A. Contributions

This paper presents a novel approach to machine learning-based methods for solving linear inverse problems in imaging. Unlike most state-of-the-art methods, which are based on unrolling a small number of iterations of an iterative reconstruction scheme ("deep unrolling"), our method is based on deep equilibrium models that correspond to a potentially infinite number of iterations. This framework yields more accurate reconstructions than the current state-of-the-art across a range of inverse problems, and gives users the ability to navigate a trade-off between reconstruction computation time and accuracy during inference. In addition, our approach inherits provable convergence guarantees depending on the "base" algorithm used to select a fixed point equation for the deep equilibrium framework. Experimental results also indicate that our proposed initialization for Deep Equilibrium Models based on pre-training is superior to random initialization, and the proposed approach is more robust to noise than past methods. Overall, the proposed DEM approach is a unique bridge between conventional fixed-point methods in numerical analysis and learning-based techniques for inverse problems in imaging.

II. Relationship to Prior Work

A. Review of Deep Unrolling Methods

Deep unrolling methods for solving inverse problems in imaging consist of a fixed number of architecturally identical "blocks," which are often inspired by a particular optimization algorithm. These methods represent the current state-of-the-art in MRI reconstruction, with most top submissions to the fastMRI challenge [6] being some sort of unrolled net. Deep unrolling architectures have also been successfully applied to other inverse problems in imaging, such as low-dose CT [7], light-field photography [8], and emission tomography [9].

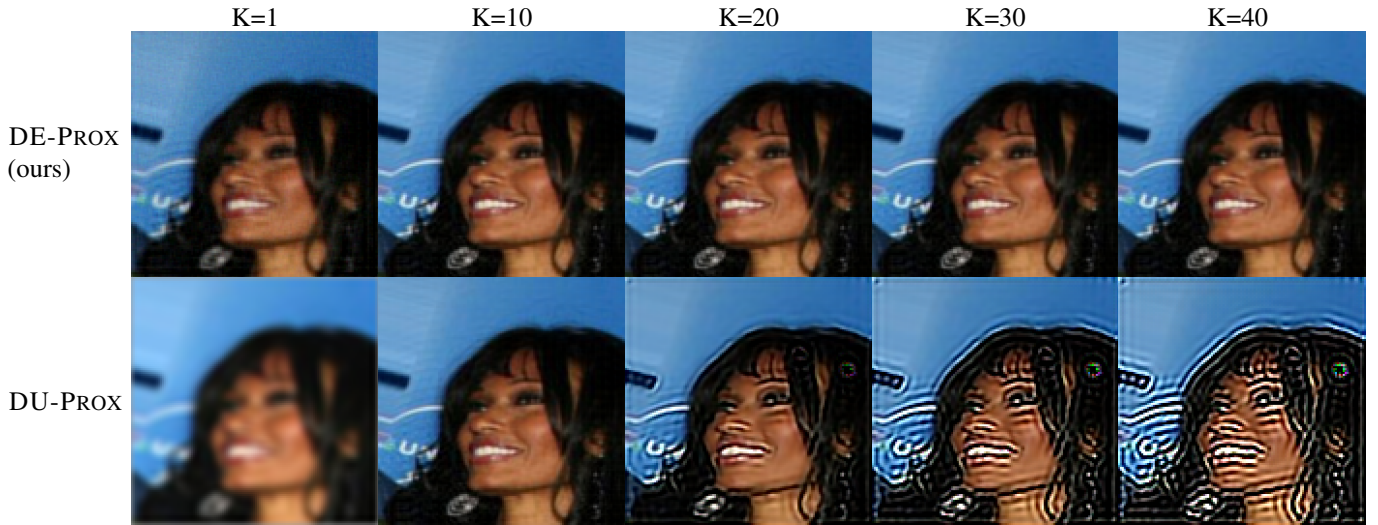| K=1 | K=10 | K=20 | K=30 | K=40 |

DE-PROX (ours)

DU-PROX

Fig. 1. Deep Unrolling (DU) methods are state-of-the-art deep networks for image reconstruction that unroll iterative optimization algorithms for a fixed number of iterations K. As illustrated above in an example with Deep Unrolled Proximal Gradient Descent (DU-PROX), these methods do not allow flexible operation at inference: unrolling for K iterations where K was not used at training results in severe artifacts. By utilizing Deep Equilibrium networks (DE-PROX above), our method trains inverse solvers to return good reconstructions *at convergence*, instead of at an arbitrary number of iterations, resulting in a flexible, higher-performing image reconstruction technique.
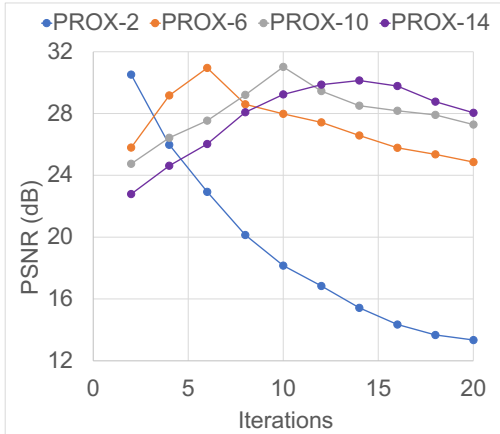

Fig. 2. Standard unrolled deep optimization networks typically require choosing some fixed number of iterates during training. Deviating from this fixed number at inference incurs a significant penalty in PSNR. The forward model here is 8x accelerated single-coil MRI reconstruction, and the unrolled algorithm is unrolled proximal gradient descent with K iterates, labeled PROX-K (Fig. 4). For further experimental details see §VI-D.

We describe here a specific deep unrolling method based on the gradient descent algorithm, although many other variants exist based on alternative optimization or fixed point iteration schemes [1]. Suppose we have a known regularization function r that could be applied to an image x; *e.g.,* in Tikhonov regularization, $r(x) = \frac{\lambda}{2}\|x\|_2^2$ for some scalar $\lambda > 0$. Then we could compute an image estimate $\widehat{x}$ by solving the optimization problem

$$\widehat{x} = \arg\min_x \frac{1}{2}\|y - Ax\|_2^2 + r(x). \quad (2)$$

If r is differentiable, this can be accomplished via gradient descent. That is, we start with an initial estimate $x^{(0)}$ such as $x^{(0)} = A^\top y$ and choose a step size $\eta > 0$, such that for iteration $k = 1, 2, 3, \ldots$, we set

$$x^{(k+1)} = x^{(k)} + \eta A^\top(y - Ax^{(k)}) - \eta\nabla r(x^{(k)}),$$

where ∇r is the gradient of the regularizer.

The basic idea behind deep unrolled methods is to fix some number of iterations K (typically K ranges from 5 to 10), declare that $x^{(K)}$ will be our estimate $\widehat{x}$, and model ∇r with a neural network, denoted $R_\theta(x)$, whose weights θ can be learned from training data. For example, we may define the unrolled gradient descent estimate to be $\widehat{x}^{(K)}(y; \theta) := x^{(K)}$ where $x^{(0)} = A^\top y$ and for $k = 0, \ldots, K - 1$ we have the recursive update

$$x^{(k+1)} = x^{(k)} + \eta A^\top(y - Ax^{(k)}) - \eta R_\theta(x^{(k)}). \quad (3)$$

Training attempts to minimize the cost function $\sum_{i=1}^n \|\widehat{x}^{(K)}(y_i; \theta) - x_i^*\|_2^2$ with respect to the network weights θ. This form of training is often called "end-to-end"; that is, we do not train the network R_θ that replaces ∇r in isolation, but rather on the quality of the resulting estimate $\widehat{x}^{(K)}$, which depends on the forward model A. Above we assume that all instances of R_θ have identical weights θ, although other works explore variants where the R_θ has iteration dependent weights [10].

The number of iterations in deep unrolling methods is kept small for two reasons. First, at deployment, these systems are optimized to compute image estimates quickly – a desirable property we wish to retain in developing new methods. Second, it is challenging to train deep unrolled networks for many iterations due to memory limitations of GPUs because the memory required to calculate the backpropagation updates scales linearly with the number of unrolled iterations. Recent work shows it is possible to train for a large number of iterates using a "checkpointing" technique which trades memory for computational cost at training time [11]. However, this approach

is only practical for a limited class of network architectures with fully invertible layers.

As one potential workaround, suppose we train a deep unrolled method for a small number of iterations K (e.g., $K = 5$), then extract the learned regularizer gradient R_θ and at inference time run the iterative scheme (3) until convergence (*i.e.,* for more iterations K than used in training). Our numerical results highlight how poorly this approach performs in practice (§VI-D). Choosing a sufficiently large number of iterations K (and hence the computational budget for inference) at training time is essential. As we illustrate in Fig. I, one cannot deviate from the choice of K used in training and expect good performance.

B. Review of Deep Equilibrium Models

In [5], the authors propose a method for training arbitrarily-deep networks given by the repeated application of a single layer. More precisely, consider an L-layer network with input y and weights θ. Letting $x^{(k)}$ denote the output of the kth hidden layer, we may write

$$x^{(k+1)} = f_\theta^{(k)}(x^{(k)}; y) \ \ \text{for} \ \ k = 0, \ldots, L-1$$

where k is the layer index and $f_\theta^{(k)}$ is a nonlinear transformation such as inner products followed by the application of a nonlinear activation function. Recent prior work explored forcing this transformation at each layer to be the same (i.e. *weight tying*), so that $f_\theta^{(k)} = f_\theta$ for all k and found evidence suggesting that such networks still yield competitive performance [12], [13]. Under weight tying, we have the recursion

$$x^{(k+1)} = f_\theta(x^{(k)}; y). \tag{4}$$

The limit of $x^{(K)}$ as $K \to \infty$, provided it exists, is a fixed point of the operator $f_\theta(\cdot, y)$. In [5] the authors learn network weights θ with constant memory costs using implicit differentiation, bypassing computation and numerical stability issues associated with related techniques on large-scale problems [14], [15]. This past work focused on sequence models and time-series tasks, assuming that each f_θ was a single layer of a neural network, and did not explore the image reconstruction task that is the focus of this paper.

Following the posting of a preprint of this paper, [16] proposed "fixed-point networks" using a strategy similar to ours, independently verifying the potential of this framework in image reconstruction. The key differences of the present work with [16] are: (1) we give convergence guarantees for three commonly-used unrolled algorithms (i.e., choices of f_θ) and identify specific parameter settings that ensure convergence, (2) we investigate acceleration schemes for the fixed-point iterations, and (3) we demonstrate the approach empirically for a different set of inverse problems, namely undersampled MRI and deblurring, whereas [16] focused on CT reconstruction.

C. Plug-and-Play and Regularization by Denoising Methods

Initiated by [17], a collection of methods based on the *plug-and-play (*PNP*)* framework have been proposed, allowing denoising algorithms to be used as priors for model-based image reconstruction (see, e.g., [18], [19]). The starting point of PNP is to write the reconstructed image as the minimizer of a cost function given by a sum of a data-fit term and a regularizer as in (2). Applying alternating directions method of multipliers (ADMM, [20]) to this minimization problem gives a collection of update equations, one of which has the form

$$\arg \min_x \frac{1}{2\sigma}\|z - x\|_2^2 + r(x),$$

where $r(x)$ is the regularizer and $\sigma > 0$ is a parameter; this update can be considered as a "denoising" of the image z. PNP methods replace this explicit optimization step with a "plugged-in" denoising method. Notably, some state-of-the-art denoisers (e.g., BM3D [4] and U-nets [21]) do not have an explicit r associated with them, but nevertheless empirically work well within the PNP framework. Convergence of PNP is studied in [22]. A related framework called *Regularization by Denoising (RED)* [23] is based on a similar philosophy as PNP, but instead considers an explicit regularizer of the form

$$r(x) = x^\top(x - \rho(x)),$$

where $\rho(x)$ corresponds to an image denoising function. Convergence of RED is studied in [24]. Recent work has found that both PNP and RED can be understood as part of a more general framework called *Consensus Equilibrium* [25].

Recent PNP and RED efforts focuses on using training data to *learn* denoisers [26], [27], [28], [29], [30]. In contrast to the unrolling methods described in §II-A, these methods are *not* trained end-to-end; rather, the denoising module is trained independent of the inverse problem at hand (*i.e.,* independent of the forward model A). As described by [1], decoupling the training of the learned component from A results in a reconstruction system that is flexible and does not need to be re-trained for each new A, but can require substantially more training samples to achieve the reconstruction accuracy of a method trained end-to-end for a specific A.

III. PROPOSED APPROACH

Our approach is to design an iteration map $f_\theta(\cdot; y)$ so that a fixed-point $x^{(\infty)}$ satisfying

$$x^{(\infty)} = f_\theta(x^{(\infty)}; y) \tag{5}$$

is a good estimate of the image $x^\star$ given its measurements y.

Here we describe choices of f_θ (and hence of the implicit infinite-depth neural network architecture) that explicitly account for the forward model A and generally for the inverse problem at hand. Specifically, we propose choosing f_θ based on different optimization algorithms applied to regularized least squares problem (2). This approach is similar to a DU approach (see II-A), but where the number of iterations is effectively infinite – a paradigm that has been beyond the reach of all previous deep unrolling architectures for solving inverse problems in imaging. Below we consider three specific choices of f_θ, but we note that many other options are possible.

A. *Deep Equilibrium Gradient Descent (*DE-GRAD*)*

Connecting the unrolled gradient descent iterations in (3) with the deep equilibrium model in (4), we let
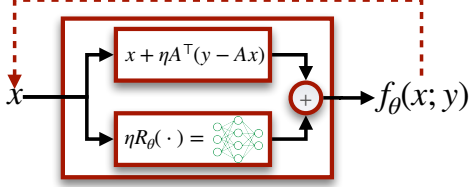
6)

es

m

Fig. 3. Deep Equilibrium Gradient Descent (DE-GRAD)

B. *Deep Equilibrium Proximal Gradient Descent (*DE-PROX*)*

Proximal gradient methods [31] use a *proximal operator* associated with a function h:

$$\text{prox}_h(x) = \arg\min_u \frac{1}{2}\|u - x\|_2^2 + h(u). \qquad (7)$$

Specifically, the proximal gradient descent algorithm applied to the optimization problem in (2) yields the iterates

$$x^{(k+1)} = \text{prox}_{\eta r}(x^{(k)} + \eta A^\top(y - Ax^{(k)})),$$

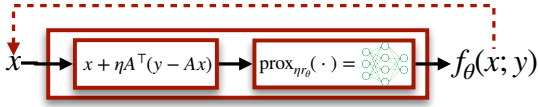where $\eta > 0$ is a step size. Similar to the deep unrolling

)

.

Fig. 4. Deep Equilibrium Proximal Gradient Descent (DE-PROX)

C. *Deep Equilibrium Alternating Directions Method of Multipliers (*DE-ADMM*)*

The Alternating Directions Method of Multipliers (ADMM, [20]) is an efficient first-order algorithm for large-scale constrained optimization problems. ADMM can be applied to the uncontrained optimization problem (2) by rewriting it as the equivalent constrained problem

$$\min_{x,z} \frac{1}{2}\|y - Ax\|_2^2 + r(z) \text{ subject to } z = x.$$

The augmented Lagrangian (in its "scaled form" – see [20]) associated with this problem is given by

$$L_\alpha(x, z, u) := \frac{1}{2}\|y - Ax\|_2^2 + r(z) + \frac{1}{2\alpha}\|z - x + u\|_2^2$$

where u is an additional auxiliary variable and $\alpha > 0$ is a user-defined parameter. The ADMM iterates are then

$$z^{(k+1)} = \arg\min_z L_\alpha(x^{(k)}, z, u^{(k)})$$
$$x^{(k+1)} = \arg\min_x L_\alpha(x, z^{(k+1)}, u^{(k)}) \qquad (9)$$
$$u^{(k+1)} = u^{(k)} + z^{(k+1)} - x^{(k+1)},$$

Here the z- and x-updates simplify as

$$z^{(k+1)} = \text{prox}_{\alpha r}(x^{(k)} - u^{(k)})$$
$$x^{(k+1)} = (I + \alpha A^\top A)^{-1}(\alpha A^\top y + z^{(k+1)} + u^{(k)}).$$

As in the DE-PROX approach, $\text{prox}_{\alpha r}(\cdot)$ can be replaced with a learned network, denoted R_θ. Making this replacement, and substituting $z^{(k+1)}$ directly into the expressions for $x^{(k+1)}$ and $u^{(k+1)}$ gives:

$$x^{(k+1)} = (I + \alpha A^\top A)^{-1}(\alpha A^\top y + R_\theta(x^{(k)} - u^{(k)}) + u^{(k)})$$
$$u^{(k+1)} = u^{(k)} + R_\theta(x^{(k)} - u^{(k)}) - x^{(k+1)}. \qquad (10)$$

Note that the updates for $x^{(k+1)}$ and $u^{(k+1)}$ depend only on the previous iterates $x^{(k)}$ and $u^{(k)}$. Therefore, the above updates can be interpreted as fixed-point iterations on the joint variable $q = (x, u)$, where the iteration map $f_\theta(q; y)$ is implicitly defined as the map that satisfies

$$q^{(k+1)} = f_\theta(q^{(k)}, y) \text{ with } q^{(k)} := (z^{(k)}, u^{(k)}) \qquad (11)$$
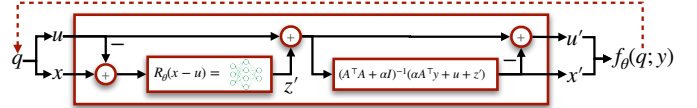

Fig. 5. Deep Equilibrium Alternating Direction Method of Multipliers (DE-ADMM)

IV. CALCULATING FORWARD PASSES AND GRADIENT UPDATES

Given a choice of iteration map $f_\theta(\cdot; y)$ defining a DEM, we confronted the following obstacles. (1) Forward calculation: given an observation y and network weights θ, we need to be able to compute a fixed point of $f_\theta(\cdot; y)$ *efficiently*. (2) Training: given a collection of training samples $\{x_i^\star\}_{i=1}^n$, we need to find the optimal network weights θ.

A. *Calculating Fixed-Points*

Both training and inference in a DEM require calculating a fixed point of the iteration map $f_\theta(\cdot; y)$ given some initial point y. The most straightforward approach is to use fixed-point iterations given in (4). Convergence of this scheme for specific f_θ designs is discussed in Section V.

However, fixed-point iterations may not converge quickly. By viewing unrolled deep networks as fixed-point iterations, we inherit the ability to accelerate inference with standard fixed-point accelerators. To our knowledge, this work is the first time iterative inversion methods incorporating deep networks have been accelerated using fixed-point accelerators.

a) Anderson Acceleration: Anderson acceleration [33][1] utilizes past iterates to identify promising directions to move during the iterations. This takes the form of identifying a vector $\alpha^{(k)} \in \mathbb{R}^m$ and setting, for $\beta > 0$

$$x^{(k+1)} = (1 - \beta) \sum_{i=0}^{m-1} \alpha_i^{(k)} x^{(k-i)} + \beta \sum_{i=0}^{m-1} \alpha_i^{(k)} f_\theta(x^{(k-i)}; y).$$

The vector $\alpha^{(k)}$ is the solution to the optimization problem:

$$\arg\min_\alpha ||G\alpha||_2^2, \quad \text{s.t.} \quad \mathbf{1}^\top \alpha = 1 \qquad (12)$$

where G is a matrix whose i^{th} column is the (vectorized) residual $f_\theta(x^{(k-i)}; y) - x^{(k-i)}$, with $i = 0, ..., m-1$. The optimization problem in (12) admits a least-squares solution, adding negligible computational overhead when m is small (*e.g.*, $m = 5$).

In Section VI-D we compare the performance and time characteristics of solving using Anderson acceleration with standard fixed-point iterations, as well as a technique which uses Broyden's method (a quasi-Newton algorithm) to find fixed points, as proposed in [5].

An important practical consideration is that accelerating fixed-point iterations arising from optimization algorithms with auxiliary variables (like ADMM) is non-trivial. Our implementation of DE-ADMM accelerates ADMM using the results of [35]. However, in general acceleration is not required to learn to solve inverse problems, and for other algorithms or settings standard fixed-point iterations may be attractive for their simplicity of implementation.

B. Gradient Calculation

In this section, we provide a brief overview of the training procedure used to train all networks in Section VI-D. We use stochastic gradient descent to find network parameters θ that (locally) minimize a cost function of the form $\frac{1}{n} \sum_{i=1}^n \ell(x^{(\infty)}(y_i; \theta), x_i^*)$ where $\ell(\cdot, \cdot)$ is a given loss function, x_i^* is the ith training image with paired measurements y_i, and $x^{(\infty)}(y_i; \theta)$ denotes the reconstructed image given as the fixed-point of $f_\theta(\cdot; y_i)$. For our image reconstruction experiments, we use the mean-squared error (MSE) loss:

$$\ell(x, x^\star) = \frac{1}{2}||x - x^\star||_2^2. \qquad (13)$$

However, the use of the MSE loss is not essential; any differentiable loss function is compatible with the approach presented below.

To simplify the calculations below, we consider gradients of the cost function with respect to a single training measurement/image pair, which we denote (y, x^*). Following [5], we leverage the fact that $x^{(\infty)} := x^{(\infty)}(y; \theta)$ is a fixed-point of $f_\theta(\cdot; y)$ to find the gradient of the loss with respect to the network parameters θ without backpropagating through an arbitrarily-large number of fixed-point iterations. We summarize this approach below.

[1]Anderson acceleration for Deep Equilibrium models was introduced in a NeurIPS tutorial by [34].

First, abbreviating $\ell(x^{(\infty)}, x^\star)$ by ℓ, then by the chain rule the gradient of ℓ with respect to the network parameters is given by

$$\frac{\partial \ell}{\partial \theta} = \frac{\partial x^{(\infty)}}{\partial \theta}^\top \frac{\partial \ell}{\partial x^{(\infty)}}. \qquad (14)$$

where $\frac{\partial x^{(\infty)}}{\partial \theta}$ is the Jacobian of $x^{(\infty)}$ with respect to θ, and $\frac{\partial \ell}{\partial x^{(\infty)}}$ is the gradient of ℓ with respect to its first argument evaluated at $x^{(\infty)}$. Since we assume ℓ is the MSE loss, the gradient $\frac{\partial \ell}{\partial x^{(\infty)}}$ is simply the residual between $x^\star$ and the equilibrium point: $\frac{\partial \ell}{\partial x^{(\infty)}} = x^{(\infty)} - x^\star$.

Now, in order to compute the Jacobian $\frac{\partial x^{(\infty)}}{\partial \theta}$ we start with the fixed point equation: $x^{(\infty)} = f_\theta(x^{(\infty)}; y)$. Differentiating both sides of this equation, and solving for $\frac{\partial x^{(\infty)}}{\partial \theta}$ gives

$$\frac{\partial x^{(\infty)}}{\partial \theta} = \left(I - \frac{\partial f_\theta(x; y)}{\partial x} \bigg|_{x=x^{(\infty)}} \right)^{-1} \frac{\partial f_\theta(x^{(\infty)}; y)}{\partial \theta}. \qquad (15)$$

Plugging this expression into (14) gives

$$\frac{\partial \ell}{\partial \theta} = \frac{\partial f_\theta(x^{(\infty)}; y)}{\partial \theta}^\top \left(I - \frac{\partial f_\theta(x; y)}{\partial x} \bigg|_{x=x^{(\infty)}} \right)^{-\top} (x^{(\infty)} - x^\star)$$

This converts the memory-intensive task of backpropagating through many iterations of $f_\theta(\cdot; y)$ to the problem of calculating an inverse Jacobian-vector product. To approximate the inverse Jacobian-vector product, first we define the vector $\beta^{(\infty)}$ by

$$\beta^{(\infty)} = \left(I - \frac{\partial f_\theta(x; y)}{\partial x} \bigg|_{x=x^{(\infty)}} \right)^{-\top} (x^{(\infty)} - x^\star).$$

Following [34], we note that $\beta = \beta^{(\infty)}$ is a fixed point of the equation

$$\beta = \left(\frac{\partial f_\theta(x; y)}{\partial x} \bigg|_{x=x^{(\infty)}} \right)^\top \beta + (x^{(\infty)} - x^\star), \qquad (16)$$

and the same machinery used to calculate the fixed point $x^{(\infty)}$ may be used to calculate $\beta^{(\infty)}$. For analysis purposes, we note that the limit of fixed-point iterations for solving (16) with initial iterate $\beta^{(0)} = 0$ is equivalent to the Neumann series:

$$\beta^{(\infty)} = \sum_{n=0}^\infty \left[\left(\frac{\partial f_\theta(x; y)}{\partial x} \bigg|_{x=x^{(\infty)}} \right)^\top \right]^n (x^{(\infty)} - x^\star). \qquad (17)$$

Convergence of the above Neumann series is discussed in Section V.

Conventional autodifferentiation tools permit quickly computing the vector-Jacobian products in (16) and (17). Once an accurate approximation to $\beta^{(\infty)}$ is calculated, the gradient in (14) is given by

$$\frac{\partial \ell}{\partial \theta} = \frac{\partial f(x^{(\infty)}; y)}{\partial \theta}^\top \beta^{(\infty)}. \qquad (18)$$

The gradient calculation process is summarized in the following steps, assuming a fixed point $x^{(\infty)}$ of $f_\theta(\cdot; y)$ is known:
1) Compute the residual $r = x^{(\infty)} - x^*$.
2) Compute an approximate fixed-point $\beta^{(\infty)}$ of the equation
$$\beta = \left(\frac{\partial f_\theta(x; y)}{\partial x} \big|_{x=x^{(\infty)}} \right)^\top \beta + r.$$
3) Compute $\frac{\partial \ell}{\partial \theta} = \frac{\partial f_\theta(x^{(\infty)}; y)}{\partial \theta}^\top \beta^{(\infty)}$.

V. CONVERGENCE THEORY

Here we study convergence of the proposed deep equilibrium models to a fixed-point at inference time, *i.e.,* given the iteration map $f_\theta(\cdot\,; y) : \mathbb{R}^d \to \mathbb{R}^d$ we give conditions that guarantee the convergence of the iterates $x^{(k+1)} = f_\theta(x^{(k)}; y)$ to a fixed-point $x^{(\infty)}$ as $k \to \infty$.

Classical fixed-point theory [36] ensures that the iterates converge to a unique fixed-point if the iteration map $f_\theta(\cdot\,; y)$ is *contractive, i.e.,* if there exists a constant $0 \le c < 1$ such that $\|f_\theta(x; y) - f_\theta(x'; y)\| \le c\|x - x'\|$ for all $x, x' \in \mathbb{R}^d$. Below we give conditions on the regularization network $R_\theta : \mathbb{R}^d \to \mathbb{R}^d$ (replacing the gradient or proximal mapping of a regularizer) used in the DE-GRAD, DE-PROX and DE-ADMM models that that ensure the resulting iteration map is contractive and thus the fixed-point iterations for these models converge.

In particular, following [27], we assume that the regularization network R_θ satisfies the following condition: there exists an $\epsilon > 0$ such that for all $x, x' \in \mathbb{R}^d$ we have

$$\|(R_\theta - I)(x) - (R_\theta - I)(x')\| \le \epsilon\|x - x'\| \qquad (19)$$

where $(R_\theta - I)(x) := R_\theta(x) - x$. In other words, we assume the map $R_\theta - I$ is ϵ-Lipschitz.

If we interpret R_θ as a denoising or de-artifacting network, then $R_\theta - I$ is the map that outputs the noise or artifacts present in a degraded image. In practice, often R_θ is implemented with a residual "skip-connection", such that $R_\theta = I + N_\theta$, where N_θ is, e.g., a deep U-net. Therefore, in this case, (19) is equivalent to assuming the trained network N_θ is ϵ-Lipschitz.

First, we have the following convergence result for DE-GRAD:

Theorem 1 (Convergence of DE-GRAD). *Assume that $R_\theta - I$ is ϵ-Lipschitz (19), and let $L = \lambda_{\max}(A^\top A)$ and $\mu = \lambda_{\min}(A^\top A)$, where $\lambda_{\max}(\cdot)$ and $\lambda_{\min}(\cdot)$ denote the maximum and minimum eigenvalue, respectively. If the step-size parameter $\eta > 0$ is such that $\eta < 1/(L+1)$, then the DE-GRAD iteration map $f_\theta(\cdot\,; y)$ defined in (6) satisfies*

$$\|f_\theta(x; y) - f_\theta(x'; y)\| \le \underbrace{(1 - \eta(1 + \mu) + \eta\epsilon)}_{=:\gamma}\|x - x'\|$$

for all $x, x' \in \mathbb{R}^d$. The coefficient γ is less than 1 if $\epsilon < 1 + \mu$, in which case the the iterates of DE-GRAD converge.

Proof. Let $f_\theta(x; y)$ be the iteration map for DE-GRAD. The Jacobian of $f_\theta(x; y)$ with respect to $x \in \mathbb{R}^d$, denoted by $\partial_x f_\theta(x; y)$, is given by

$$\partial_x f_\theta(x; y) = (I - \eta A^\top A) - \eta \partial_x R_\theta(x) \in \mathbb{R}^{d \times d}$$

where $\partial_x R_\theta(x) \in \mathbb{R}^{d \times d}$ is the Jacobian of $R_\theta : \mathbb{R}^d \to \mathbb{R}^d$ with respect to $x \in \mathbb{R}^d$. To prove $f_\theta(\cdot\,; y)$ is contractive it suffices to show $\|\partial_x f_\theta(x; y)\| < 1$ for all $x \in \mathbb{R}^d$ where $\|\cdot\|$ denotes the spectral norm. Towards this end, we have

$$\begin{aligned}
\|\partial_x f_\theta(x; y)\| &= \|(I - \eta A^\top A) - \eta \partial_x R_\theta(x)\| \\
&= \|\eta I + (1-\eta)I - \eta A^\top A - \eta \partial_x R_\theta(x)\| \\
&= \|(1-\eta)I - \eta A^\top A - \eta(\partial_x R_\theta(x) - I)\| \\
&\le \|(1-\eta)I - \eta A^\top A\| + \eta\|\partial_x R_\theta(x) - I\| \\
&\le \max_i |(1-\eta) - \eta\lambda_i| + \eta\epsilon \qquad (20)
\end{aligned}$$

where λ_i denotes the ith eigenvalue of $A^\top A$, and in the final inequality (20) we used our assumption that the map $(R_\theta - I)(x) := R_\theta(x) - x$ is ϵ-Lipschitz, and therefore the spectral norm of its Jacobian $\partial_x R_\theta(x) - I$ is bounded by ϵ.

Finally, by our assumption $\eta < \frac{1}{1+L}$ where $L := \max_i \lambda_i$, we have $\eta < \frac{1}{1+\lambda_i}$ for all i, which implies $(1 - \eta) - \eta\lambda_i > 0$ for all i. Therefore, the maximum in (20) is obtained at $\mu := \min_i \lambda_i$, which gives

$$\|\partial_x f_\theta(x; y)\| \le 1 - \eta(1 + \mu) + \eta\epsilon.$$

This shows f_θ is γ-Lipschitz with $\gamma = 1 - \eta(1 + \mu) + \eta\epsilon$, proving the claim. $\qquad\square$

Convergence of PNP approaches PNP-PROX and PNP-ADMM is studied in [27]. At inference time, the proposed DE-PROX and DE-ADMM methods are equivalent to the corresponding PNP method but with a retrained denoising network R_θ. Therefore, the convergence results in [27] apply directly to DE-PROX and DE-ADMM. To keep the paper self-contained, we restate these results below, specialized to the case of the quadratic data-fidelity term assumed in (2).

Theorem 2 (Convergence of DE-PROX). *Assume that $R_\theta - I$ is ϵ-Lipschitz (19), and let $L = \lambda_{\max}(A^\top A)$ and $\mu = \lambda_{\min}(A^\top A) > 0$, where $\lambda_{\max}(\cdot)$ and $\lambda_{\min}(\cdot)$ denote the maximum and minimum eigenvalue, respectively. Then the DE-PROX iteraion map $f_\theta(\cdot, y)$ defined in (8) is contractive if the step-size parameter η satisfies*

$$\frac{1}{\mu(1 + 1/\varepsilon)} < \eta < \frac{2}{L} - \frac{1}{L(1 + 1/\varepsilon)}.$$

Such an η exists if $\varepsilon < 2\mu/(L - \mu)$.

See Theorem 1 of [27].

Theorem 3 (Convergence of DE-ADMM). *Assume that $R_\theta - I$ is ϵ-Lipschitz (19), and let $L = \lambda_{\max}(A^\top A)$ and $\mu = \lambda_{\min}(A^\top A) > 0$, where $\lambda_{\max}(\cdot)$ and $\lambda_{\min}(\cdot)$ denote the maximum and minimum eigenvalue, respectively. Then the iteration map $f_\theta(\cdot\,; y)$ for DE-ADMM defined in (11) is contractive if the ADMM step-size parameter α parameter satisfies*

$$\frac{\varepsilon}{(1 + \varepsilon - 2\varepsilon^2)\mu} < \alpha.$$

See Corollary 1 of [27].

Unlike the convergence result for DE-GRAD given in Theorem 1, the convergence results for DE-PROX and DE-ADMM in Theorem 2 and Theorem 3 make the assumption that $\lambda_{min}(A^\top A) > 0$, i.e., A has a trivial nullspace. This condition is satisfied for certain inverse problems, such as denoising or deblurring, but violated in many others, including compressed sensing and undersampled MRI. However, in practice we observe that the iterates of DE-PROX and DE-ADMM still appear to converge even in situations where A has a nontrivial nullspace, indicating this assumption may be stronger than necessary.

Finally, an important practical concern when training deep equilibrium models is whether the fixed-point iterates used to compute gradients (as detailed in Section IV-B) will converge. Specifically, the gradient of the loss at the training pair (y, x^*)
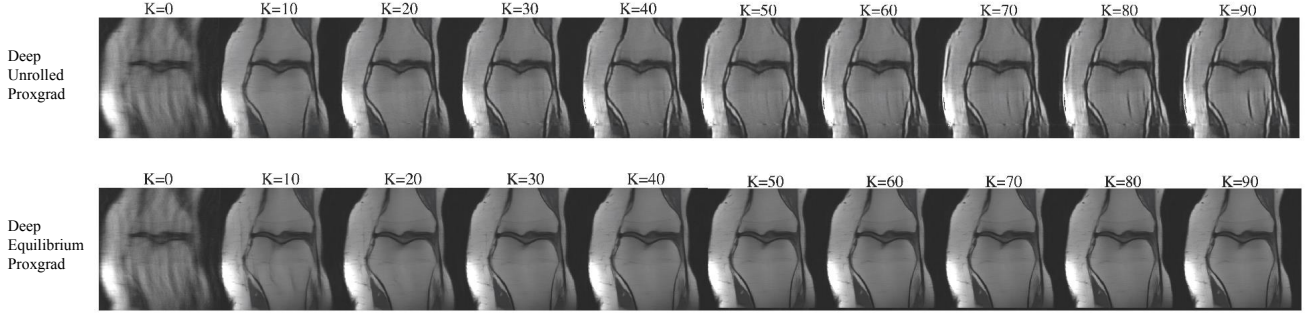
Fig. 6. While Deep Unrolled methods output high-quality reconstructions at the iteration they were trained for (here 10), running for more iterations results in severe artifacts. By contrast, Deep Equilibrium produces comparable images early on, while permitting improved reconstructions at further iterations if desired at inference time. Forward model here is $8\times$ MRI reconstruction. For further details see Section VI-A.

involves computing the truncated Neumann series in (17). This series converges if the Jacobian $\partial_x f_\theta(x; y)$ has spectral norm strictly less than 1 when evaluated at any $x \in \mathbb{R}^d$, which is true if and only if the iteration map $f_\theta(\cdot\,; y)$ is contractive. Therefore, the same conditions in Theorems 1-3 that ensure the iteration map is contractive also ensure that the Neumann series in (17) used to compute gradients converges.

VI. EXPERIMENTAL RESULTS

A. Comparison Methods and Inverse Problems

Our numerical experiments include comparisons with a variety of models and methods. Total-variation Regularized Least Squares (TV) is an important baseline that does not use any training data but rather leverages geometric models of image structure [3], [37], [38]. The PNP and RED methods are described in §II-C; we consider both the original ADMM variant of [17] PNP-ADMM and a proximal gradient PNP-PROX method as described in [27]. We utilize the ADMM formulation of RED. Deep Unrolled methods (DU) are described in §II-A; we consider DU using gradient descent, proximal gradient, and ADMM. The preconditioned Neumann network [39] does not have simple Deep Equilibrium or Plug-and-Play analogues, and is included as an alternative deep unrolled method.

All deep unrolled methods have *tied weights*, i.e. the network used at each iteration is the same. This is done to ensure that the number of parameters in the deep unrolled, deep equilibrium, and Plug-and-Play/RED methods are the same. Moreover, on modestly-sized datasets recent work has suggested that tied weights result in better reconstructions [40]. We do note that many recent winners of the 2020 fastMRI challenge have untied weights, suggesting that for larger datasets a higher number of parameters may be desirable [6].

We compare the above approaches across three inverse problems: Image deblurring (Deblur), compressed sensing (CS), and accelerated MRI reconstruction (MRI). In the Deblur setting we simulate blurry images using 9×9 pixel Gaussian blur kernel with variance 5 and consider two noise levels: Deblur (1) refers to the setting of additive white Gaussian noise with variance $\sigma = 0.01$ (high noise), and Deblur (2) refers to

the setting of additive white Gaussian noise with standard deviation $\sigma = 0.0001$ (low noise). In the CS setting, we take linear measurements of the image by forming inner products with random Gaussian vectors whose entries i.i.d. standard normals, and use an undersampling factor of $4\times$, *i.e.,* the corresponding forward model A is a random Gaussian matrix with $4\times$ fewer rows than columns. The measurements are then corrupted with additive white Gaussian noise with standard deviation $\sigma = 0.01$. In the MRI setting, we investigate recovery at $4\times$ and $8\times$ acceleration, where "$\rho\times$ acceleration" indicates an undersampling factor of ρ in k-space (not to be confused with the acceleration techniques used in finding fixed points). Our MRI experiments focus on the case of (virtual) single-coil MR data acquired on a Cartesian grid in k-space, where corresponding forward model A is a 2-D Fourier transform with subsampling in the phase encoding dimension. We additionally add complex white Gaussian noise to the undersampled k-space measurements with standard deviation $\sigma = 0.01$.

For the Deblur and CS problems, we utilize a subset of the Celebrity Faces with Attributes (CelebA) dataset [41], which consists of centered human faces. We train on a subset of 10000 of the training images. All images are resized to 128×128. For the MRI problem, we use a random subset of size 2000 of the fastMRI single-coil knee dataset [42] for training. We trim the ground truth MR images to a 320×320 pixel region-of-interest.

B. Architecture Specifics

For our learned network, we utilize a DnCNN architecture as in [27]. We also experimented with U-Nets, but found that DnCNN yielded superior performance for both our proposed deep equilibrium methods and the comparison methods. For both the CelebA and fastMRI datasets, we train six DnCNN denoisers with noise variances $\sigma^2 = 0.1, 0.05, 0.02, 0.01, 0.005, 0.001$ on the training split. Training follows the methodology of [27]. Specifically, to ensure contractivity of the learned component, we add spectral normalization to all layers, ensuring that each layer has a Lipschitz constant bounded above by 1. This normalization is enforced during pretraining as well as during the Deep Equilibrium training phrase.

TABLE I
MEAN PSNR AND SSIM OVER TEST SET; THE HIGHEST PSNRS AND SSIMS FOR EACH SETTING ARE IN BOLD.

		PLUG-N-PLAY (DNCNN DENOISER)		RED (DNCNN DENOISER)	DEEP UNROLLED METHODS TRAINED END-TO-END				DEEP EQUILIBRIUM (OURS)		
	TV	PROX	ADMM	ADMM	GRAD	PROX	ADMM	NEUMANN	GRAD	PROX	ADMM
DEBLUR (1)											
PSNR	26.79	29.77	29.95	29.78	32.23	31.64	31.45	32.39	**32.43**	31.87	32.30
SSIM	0.86	0.88	0.89	0.89	0.93	0.93	0.93	0.94	**0.94**	0.93	0.94
DEBLUR (2)											
PSNR	31.31	35.25	35.61	35.22	36.10	36.92	36.14	36.24	**37.99**	37.84	37.95
SSIM	0.90	0.96	0.96	0.96	0.97	0.97	0.97	0.97	**0.98**	0.97	0.98
CS (4X)											
PSNR	26.04	27.79	27.85	27.80	29.32	29.65	29.09	29.59	31.46	31.51	**31.64**
SSIM	0.83	0.87	0.87	0.86	0.88	0.89	0.88	0.89	0.93	0.93	**0.93**
MRI (8X)											
PSNR	26.64	28.45	28.39	29.89	31.13	30.97	31.82	31.64	**32.01**	31.02	31.38
SSIM	0.78	0.85	0.85	0.88	0.89	0.88	0.88	0.89	**0.89**	0.88	0.89
MRI (4X)											
PSNR	31.22	31.56	31.92	32.37	32.44	32.49	32.56	32.62	33.41	33.66	**33.72**
SSIM	0.88	0.89	0.89	0.90	0.91	0.91	0.91	0.92	0.91	0.92	**0.92**

During training, we utilize Anderson acceleration for both the forward and backward pass fixed-point iterations. In backward passes, the number of fixed-point iterations was limited to 50 due to memory constraints, but fixed-point iterations in forward passes were run until convergence was observed. We define this to be when the relative norm difference between iterations is less than 10^{-3}:

$$\frac{||x^{(k+1)} - x^{(k)}||_2}{||x^{(k)}||} < 10^{-3} \qquad (21)$$

Test time results were produced using the same method as the forward pass during training, but over the test set. We compare different fixed-point calculation methods for the forward and backward passes in Section VI-E.

Further details on settings, parameter choices, and data may be found in the supplementary materials and in our publicly-available code.[2]

C. Parameter Tuning and Pretraining

Each of the iterative optimization algorithms we test has its own set of hyperparameters to choose, *e.g.,* the step size η in DE-GRAD, plus any parameters used to calculate the initial estimate $x^{(0)}$. Tuning hyperparameters requires choosing a particular regularization network R_θ during tuning. We choose from a collection of R_θ that have been pretrained for Gaussian denoising at different noise levels. Pretraining can be done on the training dataset (*e.g.,* training on MRI images directly) or using an independent dataset (*e.g.,* the BSD500 image dataset [43]). We use the former approach in our experiments. We train a set of denoisers for each domain of our training set: a collection is trained for CelebA, and a separate collection for the fastMRI data.

To tune hyperparameters, we first choose parameters to optimize the performance of PNP on a validation set via a grid search over pretrained R_θ as well as algorithm-specific

hyperparameters (such as the η step-size parameter in gradient descent approaches). Then, we use the PNP hyperparameter settings as initial hyperparameter values when training Deep Equilibrium or Deep Unrolling methods.

Choosing η based on the theory presented in Section V is not always possible, since common forward models have nontrivial null space. In these cases, a good choice for η is one that ensures that a single iteration of the underlying optimization algorithm is contractive. For Deep Unrolled methods, contractivity is not strictly necessary, but Deep Equilibrium methods require contractivity to ensure a stable training process.

The η step size parameter in DU-GRAD and DU-PROX are optimized during training, but are initialized by the hyperparameter search described above. We found that optimizing over η for Deep Equilibrium approaches led to instability during training, and so freeze η during training for all Deep Equilibrium approaches.

D. Main Results

We present the main reconstruction accuracy comparison in Table I. Each entry for Deep Equilibrium (DE), Regularization by Denoising (RED), and Plug-and-Play (PNP) approaches is the result of running fixed-point iterations until the relative change between iterations is less than 10^{-3}. During training, all DE models were limited to a maximum 100 forward updates, but terminate iterations on the relative norm difference between iterations falling below 10^{-3}. The DU models are tested at the number of iterations for which they were trained and all parameters for TV reconstructions (including number of TV iterations) are cross-validated to maximize PSNR. Performance as a function of iteration is shown in Figs. 7(a), 7(b), and 7(c), with example reconstructions in Fig. 8. We provide an additional iteration-dependent comparison between DE-PROX and DU-PROX in Fig. 6. Further example reconstructions are available for qualitative evaluation in the supplementary materials.

[2]Available at: https://github.com/dgilton/deep_equilibrium_inverse

We observe our Deep Equilibrium-based approaches consistently outperform Deep Unrolled approaches across different choices of base algorithm (*i.e.,* GRAD,PROX,ADMM). Among choices of iterative reconstruction architectures for f_θ, there does not appear to be an obvious winner, suggesting the optimal choice may be problem- or setting-dependent.
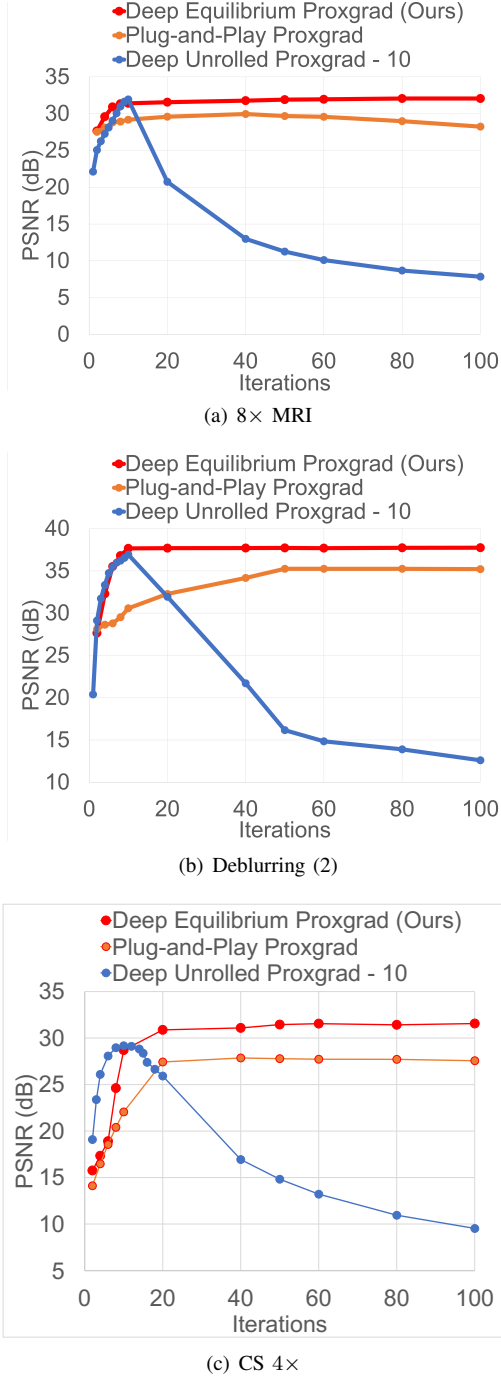

(a) 8× MRI

(b) Deblurring (2)

(c) CS 4×

Fig. 7. Iterations vs. reconstruction PSNR for DE-PROX and competing methods for (a) MRI reconstruction, (b) deblurring, and (c) compressed sensing. The deep unrolled ProxGrad was trained for 10 iterations. In all examples, deep unrolling is only effective at the number of iterations for which it is trained, whereas deep equilibrium achieves higher PSNR across a broad range of iterations, allowing a user to trade off computation time and accuracy.

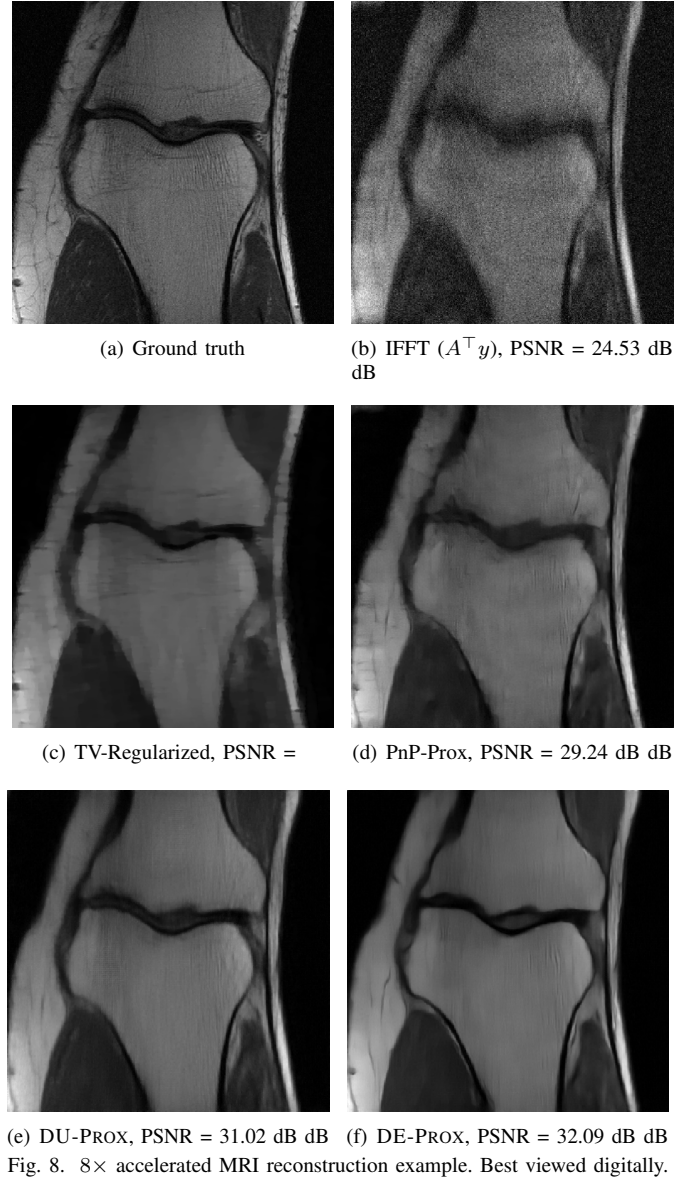As seen in Figs. 7(a), 7(b), and 7(c), the Deep Equilibrium

(a) Ground truth

(b) IFFT ($A^\top y$), PSNR = 24.53 dB dB

(c) TV-Regularized, PSNR =

(d) PnP-Prox, PSNR = 29.24 dB dB

(e) DU-PROX, PSNR = 31.02 dB dB   (f) DE-PROX, PSNR = 32.09 dB dB

Fig. 8. 8× accelerated MRI reconstruction example. Best viewed digitally.

approach generally outperforms Deep Unrolled solvers. The results shown in Figs. 7(a), 7(b), and 7(c) suggest that our approach requires *no more computation* than Deep Unrolled networks to achieve the same performance level and has an increasing advantage with further computation.

E. Effect of Acceleration

Here we explore the effect of using different fixed-point solvers during both the training and inference procedures. Leveraging acceleration can decrease computational costs during both training and inference and result in better empirical performance at inference. Table II compares Anderson accelerated Deep Equilibrium approaches with Deep Unrolling, Plug and Play, Deep Equilibrium utilizing Broyden's Method (as was used in [5]), and non-accelerated Deep Equilibrium. All results were determined using PyTorch 1.6.0 utilizing an Nvidia RTX 2080 Ti GPU. As mentioned previously, non-accelerated Deep Equilibrium at inference time has identical per-iteration cost and memory requirements as Plug and Play and Deep Unrolling.

	Total Time (s)	Time/Iteration (s)	PSNR (dB)
Plug & Play	1.24	0.025	31.56
DU-PROX	0.25	0.025	32.49
DE-PROX	1.22	0.025	31.86
DE-PROX (Anderson)	1.17	0.046	33.66
DE-PROX (Broyden)	1.85	0.039	33.61

TABLE II

MEAN COMPUTATION TIME REQUIRED TO REACH CONVERGENCE AND RESULTING MEAN RECONSTRUCTION PSNR IN $4\times$ ACCELERATED MRI RECONSTRUCTION WITH COMPLEX IMAGE SIZE 320×320, AS COMPUTED OVER THE TEST SET.

We observe that while finding the fixed-points using Broyden's method and Anderson acceleration requires more time per iterate, convergence occurs faster than the standard fixed-point iterates, so the net time spent at inference is less. Since the Broyden solution was slightly worse in terms of PSNR, Anderson acceleration was used for all other experiments.

For practical matters, the memory cost of each of the compared methods may also be an important factor to consider. At train time, Deep Unrolled methods require memory scaling linearly with the number of iterations used, while Deep Equilibrium methods require only the memory necessary to compute the gradient in (16), which is what permits training at convergence. Plug-and-Play methods are the least memory-intensive of all to train. At inference time, each method only needs to store at most a constant number of iterations, so all methods are cheap in terms of memory to evaluate.
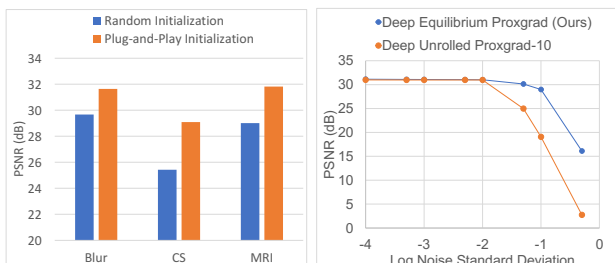

Fig. 9. (a) Comparison of learned DE-PROX reconstruction quality across three different inverse problems: Deblurring (Blur), compressed sensing (CS), and undersampled MRI reconstruction (MRI). In our experiments, initializing with a pretrained denoiser routinely offered as good or better reconstruction quality (in terms of PSNR) than a random initialization. (b) Noise sensitivity comparison between DU-PROX and DE-PROX. The forward model used is $8\times$ MRI reconstruction, and σ here corresponds to the level of Gaussian noise added to observations.

F. Effect of Pre-Training

Here we compare the effect of initializing the learned component R_θ in our deep equilibrium models with a pretrained denoiser versus initializing with random weights. We use the same hyperparameter tuning scheme described in Section VI-C.

We present our results on Deep Equilibrium Proximal Gradient Descent (DE-PROX) in Figure 9(a). We observe an improvement in reconstruction quality when utilizing our pretraining method compared to a random initialization. We also note that pretraining enables a simple choice of algorithm-specific hyperparameters, such as the initial internal step size for DE-PROX.

G. Noise Sensitivity

We observe empirically that the Deep Equilibrium approach to training achieves competitive reconstruction quality and increased flexibility with respect to allocating computation budget at inference time. Recent work in deep inversion has questioned these methods' robustness to noise and unexpected inputs [44], [45], [46].

To examine whether the Deep Equilibrium approach is brittle to simple changes in the noise distribution, we varied the level of Gaussian noise added to the observations at test time and observed the effect on reconstruction quality in a setting where DE-PROX and DU-PROX perform similarly. The results shown in Fig. 9(b) suggest that the Deep Equilibrium model DE-PROX is more robust to variation in the noise level than the analogous Deep Unrolled approach DU-PROX. The forward model used in Fig. 9(b) is $8\times$ MRI reconstruction. We note that we have also observed a similar resiliency for unrolled Gradient Descent and ADMM as well.

VII. CONCLUSIONS

This paper illustrates non-trivial quantitative benefits to using implicitly-defined infinite-depth networks for solving linear inverse problems in imaging. These empirical benefits complement convergence guarantees that are unavailable to widely-used deep unrolling methods. Other recent work has focused on such *implicit networks* akin to the deep equilibrium models considered here (e.g. [47]). Whether these models could lead to additional advances in image reconstruction remains an open question for future work. Furthermore, while the exposition in this work focused on *linear* inverse problems, nonlinear inverse problems may be solved with iterative approaches just as well. The conditions under which deep equilibrium methods proposed here may be used on such iterative approaches are an active area of investigation.

REFERENCES

[1] G. Ongie, A. Jalal, C. A. Metzler, R. G. Baraniuk, A. G. Dimakis, and R. Willett, "Deep learning techniques for inverse problems in imaging," *arXiv preprint arXiv:2005.06001*, 2020.

[2] A. N. Tikhonov, "On the stability of inverse problems," in *Dokl. Akad. Nauk SSSR*, vol. 39, 1943, pp. 195–198.

[3] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: nonlinear phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.

[4] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Transactions on image processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[5] S. Bai, J. Z. Kolter, and V. Koltun, "Deep equilibrium models," in *Advances in Neural Information Processing Systems*, 2019, pp. 690–701.

[6] M. J. Muckley, B. Riemenschneider, A. Radmanesh, S. Kim, G. Jeong, J. Ko, Y. Jun, H. Shin, D. Hwang, M. Mostapha *et al.*, "State-of-the-art machine learning mri reconstruction in 2020: Results of the second fastmri challenge," *arXiv preprint arXiv:2012.06318*, 2020.

[7] D. Wu, K. Kim, and Q. Li, "Computationally efficient deep neural network for computed tomography image reconstruction," *Medical physics*, vol. 46, no. 11, pp. 4763–4776, 2019.

[8] I. Y. Chun, Z. Huang, H. Lim, and J. Fessler, "Momentum-net: Fast and convergent iterative neural network for inverse problems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[9] A. Mehranian and A. J. Reader, "Model-based deep learning pet image reconstruction using forward-backward splitting expectation maximisation," *IEEE Transactions on Radiation and Plasma Medical Sciences*, 2020.

[10] J. Adler and O. Öktem, "Learned primal-dual reconstruction," *IEEE transactions on medical Imaging*, vol. 37, no. 6, pp. 1322–1332, 2018.

[11] M. Kellman, K. Zhang, E. Markley, J. Tamir, E. Bostan, M. Lustig, and L. Waller, "Memory-efficient learning for large-scale computational imaging," *IEEE Transactions on Computational Imaging*, vol. 6, pp. 1403–1414, 2020.

[12] R. Dabre and A. Fujita, "Recurrent stacking of layers for compact neural machine translation models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 6292–6299.

[13] S. Bai, J. Z. Kolter, and V. Koltun, "Trellis networks for sequence modeling," *arXiv preprint arXiv:1810.06682*, 2018.

[14] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," in *Advances in neural information processing systems*, 2018, pp. 6571–6583.

[15] E. Haber and L. Ruthotto, "Stable architectures for deep neural networks," *Inverse Problems*, vol. 34, no. 1, p. 014004, 2017.

[16] H. Heaton, S. W. Fung, A. Gibali, and W. Yin, "Feasibility-based fixed point networks," *arXiv preprint arXiv:2104.14090*, 2021.

[17] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, "Plug-and-play priors for model based reconstruction," in *2013 IEEE Global Conference on Signal and Information Processing*. IEEE, 2013, pp. 945–948.

[18] S. Sreehari, S. V. Venkatakrishnan, B. Wohlberg, G. T. Buzzard, L. F. Drummy, J. P. Simmons, and C. A. Bouman, "Plug-and-play priors for bright field electron tomography and sparse interpolation," *IEEE Transactions on Computational Imaging*, vol. 2, no. 4, pp. 408–423, 2016.

[19] U. S. Kamilov, H. Mansour, and B. Wohlberg, "A plug-and-play priors approach for solving nonlinear imaging inverse problems," *IEEE Signal Processing Letters*, vol. 24, no. 12, pp. 1872–1876, 2017.

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

[21] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[22] S. H. Chan, X. Wang, and O. A. Elgendy, "Plug-and-play admm for image restoration: Fixed-point convergence and applications," *IEEE Transactions on Computational Imaging*, vol. 3, no. 1, pp. 84–98, 2016.

[23] Y. Romano, M. Elad, and P. Milanfar, "The little engine that could: Regularization by denoising (red)," *SIAM Journal on Imaging Sciences*, vol. 10, no. 4, pp. 1804–1844, 2017.

[24] E. T. Reehorst and P. Schniter, "Regularization by denoising: Clarifications and new interpretations," *IEEE Transactions on Computational Imaging*, vol. 5, no. 1, pp. 52–67, 2018.

[25] G. T. Buzzard, S. H. Chan, S. Sreehari, and C. A. Bouman, "Plug-and-play unplugged: Optimization-free reconstruction using consensus equilibrium," *SIAM Journal on Imaging Sciences*, vol. 11, no. 3, pp. 2001–2020, 2018.

[26] T. Meinhardt, M. Moller, C. Hazirbas, and D. Cremers, "Learning proximal operators: Using denoising networks for regularizing inverse imaging problems," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1781–1790.

[27] E. Ryu, J. Liu, S. Wang, X. Chen, Z. Wang, and W. Yin, "Plug-and-play methods provably converge with properly trained denoisers," in *International Conference on Machine Learning*, 2019, pp. 5546–5557.

[28] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep cnn denoiser prior for image restoration," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3929–3938.

[29] T. Tirer and R. Giryes, "Super-resolution via image-adapted denoising cnns: Incorporating external and internal learning," *IEEE Signal Processing Letters*, vol. 26, no. 7, pp. 1080–1084, 2019.

[30] J. Liu, Y. Sun, C. Eldeniz, W. Gan, H. An, and U. S. Kamilov, "Rare: Image reconstruction using deep priors learned without groundtruth," *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 6, pp. 1088–1099, 2020.

[31] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[32] M. Mardani, Q. Sun, D. Donoho, V. Papyan, H. Monajemi, S. Vasanawala, and J. Pauly, "Neural proximal gradient descent for compressive imaging," in *Advances in Neural Information Processing Systems*, 2018, pp. 9573–9583.

[33] H. F. Walker and P. Ni, "Anderson acceleration for fixed-point iterations," *SIAM Journal on Numerical Analysis*, vol. 49, no. 4, pp. 1715–1735, 2011.

[34] Z. Kolter, D. Duvenaud, and M. Johnson, "Deep implicit layers - neural odes, deep equilibrium models, and beyond," 2020. [Online]. Available: http://implicit-layers-tutorial.org/

[35] J. Zhang, Y. Peng, W. Ouyang, and B. Deng, "Accelerating admm for efficient simulation and optimization," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 6, pp. 1–21, 2019.

[36] A. Granas and J. Dugundji, "Elementary fixed point theorems," in *Fixed Point Theory*. Springer, 2003, pp. 9–84.

[37] D. Strong and T. Chan, "Edge-preserving and scale-dependent properties of total variation regularization," *Inverse problems*, vol. 19, no. 6, p. S165, 2003.

[38] A. Beck and M. Teboulle, "Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems," *IEEE transactions on image processing*, vol. 18, no. 11, pp. 2419–2434, 2009.

[39] D. Gilton, G. Ongie, and R. Willett, "Neumann networks for linear inverse problems in imaging," *IEEE Transactions on Computational Imaging*, 2019.

[40] H. K. Aggarwal, M. P. Mani, and M. Jacob, "MoDL: Model-based deep learning architecture for inverse problems," *IEEE transactions on medical imaging*, vol. 38, no. 2, pp. 394–405, 2018.

[41] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[42] J. Zbontar, F. Knoll, A. Sriram, M. J. Muckley, M. Bruno, A. Defazio, M. Parente, K. J. Geras, J. Katsnelson, H. Chandarana, Z. Zhang, M. Drozdzal, A. Romero, M. Rabbat, P. Vincent, J. Pinkerton, D. Wang, N. Yakubova, E. Owens, C. L. Zitnick, M. P. Recht, D. K. Sodickson, and Y. W. Lui, "fastMRI: An open dataset and benchmarks for accelerated MRI," 2018.

[43] D. Martin, C. Fowlkes, D. Tal, and J. 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*, vol. 2. IEEE, 2001, pp. 416–423.

[44] V. Antun, F. Renna, C. Poon, B. Adcock, and A. C. Hansen, "On instabilities of deep learning in image reconstruction and the potential costs of ai," *Proceedings of the National Academy of Sciences*, 2020.

[45] A. Raj, Y. Bresler, and B. Li, "Improving robustness of deep-learning-based image reconstruction," *arXiv preprint arXiv:2002.11821*, 2020.

[46] M. Genzel, J. Macdonald, and M. März, "Solving inverse problems with deep neural networks–robustness included?" *arXiv preprint arXiv:2011.04268*, 2020.

[47] L. El Ghaoui, F. Gu, B. Travacca, and A. Askari, "Implicit deep learning," *arXiv preprint arXiv:1908.06315*, 2019.