Memory-efficient model-based deep learning with convergence and robustness guarantees

Aniket Pramanik [Student Member, IEEE],
Department of Electrical and Computer Engineering at the University of Iowa, Iowa City, IA, 52242, USA

M. Bridget Zimmerman,
Department of Biostatistics at the University of Iowa, Iowa City, IA, 52242, USA

Mathews Jacob [Fellow, IEEE]
Department of Electrical and Computer Engineering at the University of Iowa, Iowa City, IA, 52242, USA

Abstract

Computational imaging has been revolutionized by compressed sensing algorithms, which offer guaranteed uniqueness, convergence, and stability properties. Model-based deep learning methods that combine imaging physics with learned regularization priors have emerged as more powerful alternatives for image recovery. The main focus of this paper is to introduce a memory efficient model-based algorithm with similar theoretical guarantees as CS methods. The proposed iterative algorithm alternates between a gradient descent involving the score function and a conjugate gradient algorithm to encourage data consistency. The score function is modeled as a monotone convolutional neural network. Our analysis shows that the monotone constraint is necessary and sufficient to enforce the uniqueness of the fixed point in arbitrary inverse problems. In addition, it also guarantees the convergence to a fixed point, which is robust to input perturbations. We introduce two implementations of the proposed MOL framework, which differ in the way the monotone property is imposed. The first approach enforces a strict monotone constraint, while the second one relies on an approximation. The guarantees are not valid for the second approach in the strict sense. However, our empirical studies show that the convergence and robustness of both approaches are comparable, while the less constrained approximate implementation offers better performance. The proposed deep equilibrium formulation is significantly more memory efficient than unrolled methods, which allows us to apply it to 3D or 2D+time problems that current unrolled algorithms cannot handle.

Keywords

Model-based deep learning; Monotone operator learning; Deep equilibrium models

---

aniket-pramanik@uiowa.edu .

I. INTRODUCTION

The recovery of images from a few noisy measurements is a common problem in several imaging modalities, including MRI [1], CT [2], PET [3], and microscopy [4]. In the undersampled setting, multiple images can give a similar fit to the measured data, making the recovery ill-posed. Compressive sensing (CS) algorithms pose the recovery as a convex optimization problem, where a strongly convex prior is added to the data-consistency term to regularize the recovery [5]. The main benefit of convex priors is in the uniqueness of the solutions. In particular, the strong convexity of the priors guarantees that the overall cost function in (3) is strongly convex, even when **A** operator has a large null space. Another desirable property of convex priors in CS is the robustness of the solution to input perturbations.

In recent years, several flavors of model-based deep learning algorithms, which combine imaging physics with learned priors, were introduced to significantly improve the performance compared to CS algorithms. For example, plug and play (PnP) methods use denoiser modules to replace the proximal mapping steps in CS algorithms [6]–[11], and the algorithms are run until convergence. While earlier approaches chose off-the-shelf denoisers such as BM3D [12], recent methods use pre-trained convolutional neural network (CNN) modules [9], [10]. The pre-trained CNN modules that learn the image prior are agnostic to the forward model, which enables their use in arbitrary inverse problems. These methods come with convergence and uniqueness guarantees when the forward model is full-rank or the data term is strongly convex [13]. When the data term is not strongly convex, weaker convergence guarantees are available [9], but uniqueness is not guaranteed. Another category of approaches relies on unrolled optimization; these algorithms unroll finite number of iterative optimization steps in CS algorithms to obtain a deep network, which is composed of CNN blocks and optimization blocks to enforce data consistency; the resulting deep network is trained in an end-to-end fashion [14]–[17]. A key difference between unrolled methods and PnP methods is that the CNN block is trained end-to-end, assuming a specific forward model; such model-based methods typically offer better performance than PnP methods that are agnostic to the forward model [14]–[20]. Unlike PnP approaches that run the algorithm until convergence, the number of iterations in unrolled methods are restricted by the memory of the GPU devices during training; this often limits the applicability of unrolled algorithms to large-scale multi-dimensional problems. Several strategies were introduced to overcome the memory limitations of unrolled methods. For an unrolled network with N iterations and shared CNN modules across iterations, the computational complexity and memory demand of backpropagation are $\mathcal{O}(N)$ and $\mathcal{O}(N)$, respectively. The forward steps can be recomputed during backpropagation, which reduces the memory demand to $\mathcal{O}(1)$, while the computational complexity increases to $\mathcal{O}\left(N^2\right)$. Forward checkpointing [21] saves the variables for every K layers during forward propagation, which reduces the computational demand to $\mathcal{O}(NK)$, while the memory demand is $\mathcal{O}(N/K)$. Reverse recalculation has been proposed to reduce the memory demand to $\mathcal{O}(1)$ and computational complexity to $\mathcal{O}(N)$ [22]. However, the approach in [22] requires multiple iterations to invert each CNN block, resulting in high computational complexity in practical applications.

Gilton et al. recently extended the deep equilibrium (DEQ) model [23] to significantly improve the memory demand [24] of unrolled methods. Unlike unrolled methods, DEQ schemes run the iterations until convergence, similar to PnP algorithms. This property allows one to perform forward and backward propagation using fixed-point iteration involving a single physical layer, which reduces the memory demand to $\mathcal{O}(1)$, while the computational complexity is $\mathcal{O}(N)$; this offers better tradeoffs than the alternatives discussed above [21], [22]. The runtime of DEQ methods that are iterated until convergence are variable compared to unrolled methods, which use a finite number of iterations. In addition, the convergence of the iterative algorithm is crucial for the accuracy of backpropagation steps in DEQ, unlike in unrolled methods. Convergence guarantees were introduced in [10], [24] for the alternating direction method of multipliers (ADMM), proximal gradient (PG), and forward-backward DEQ algorithms. The convergence guarantees rely on restrictive conditions on the CNN denoising blocks, which are dependent on the forward models. Unfortunately, when the minimum singular value of the forward operator is small (e.g., highly accelerated parallel MRI) or zero (e.g., super-resolution), the CNN denoiser needs to be close to an identity operator for the iterations to converge. Another challenge associated with DEQ methods is the way the non-expansive constraints on the network are imposed. Most methods [10], [24] use spectral normalization of each layer of the network. Our experiments in Fig. 3 show that spectral normalization often translates to networks with lower performance. Another theoretical problem associated with current DEQ methods is the potential non-uniqueness of the fixed point, which can also affect the stability/robustness of the algorithm in the presence of input perturbations. We note that the stability of deep image reconstruction networks is a debated topic. While deep networks are reported to be more fragile to input perturbations than are conventional algorithms [25], some of the recent works have presented a more optimistic view [26], [27].

The main goal of this work is to introduce a model-based DEQ algorithm that shares the desirable properties of convex CS algorithms, including guaranteed uniqueness of the fixed point solutions, convergence, and robustness to input perturbations. By enabling the training of the CNN modules in an end-to-end fashion, the proposed algorithm can match the performance of unrolled approaches while being significantly more memory efficient. The main contributions of this paper are:

- We introduce a forward-backward DEQ algorithm (14) involving a learned network $\mathcal{F}$. Existing algorithms [8], [16], [28] such as MoDL and RED are special cases of this algorithm when the damping parameter $\alpha = 1$.

- We show that constraining the CNN module as an $m > 0$ monotone operator is necessary and sufficient to guarantee the uniqueness of the fixed point of the algorithm. Because the monotone constraint is central to our approach, we term the proposed scheme as the monotone operator learning (MOL) algorithm.

- We show that an m-monotone operator $\mathcal{F}$ can be realized as a residual CNN: $\mathcal{F} = \mathcal{I} - \mathcal{H}_\theta$ where the Lipschitz constant of the denoiser module $\mathcal{H}_\theta$ is $L[\mathcal{H}_\theta] = 1 - m$. We also determine the range of values of α and $L[\mathcal{H}_\theta]$ for which the algorithm converges; the analysis and the experiments in Fig. 3 show that the

direct application of the MoDL and RED ($\alpha = 1$) algorithms to the DEQ setting will diverge unless a highly constrained CNN ($L[\mathcal{H}_\theta] < 0.24$) is used, which restricts performance. By contrast, the use of a smaller α translates to higher $L[\mathcal{H}_\theta]$ and hence improved performance.

- We theoretically analyze the worst-case sensitivity of the the resulting DEQ scheme. Our analysis shows that the norm of the perturbations in the reconstructed images are linearly proportional to the norm of the measurement perturbations, with the proportionality dependent on $1/m$.

- We introduce two implementations of the proposed MOL algorithm. The first approach uses spectral normalization to enforce the monotone constraint in the strict sense. We also introduce an approximate implementation, where we replace $L[\mathcal{H}_\theta]$ by an approximation $l[\mathcal{H}_\theta]$. While the second approach does not satisfy the monotone constraint in the strict sense, our experiments in Fig. 3 shows that the resulting algorithm converges, while Fig. 6 shows that the robustness of both schemes to adversarial and Gaussian noise are comparable. We note that spectral normalization based estimate for Lipschitz constant is very conservative; our experiments in Fig. 4 show that the second approach offers improved performance over the exact approach.

- We experimentally compare the performance against unrolled algorithms that use similar-sized CNNs in two-dimensional MR imaging problems. Our results show that the performance of the MOL scheme is comparable to that of unrolled algorithms. In addition, the MOL scheme is associated with a ten-fold reduction in memory compared to the unrolling algorithms with ten unrolls. The significant gain in memory demand allows us to extend our algorithm to the 3D or 2D+time setting, where it offers improved performance over unrolled 2D approaches. The experimental results in Fig. 6 and 7 show the improved robustness of the proposed scheme compared to existing unrolled algorithms [16], [19] and UNET [29]. The recorded run-times in Table I show that MOL has higher computational complexity (≈ 2.5 times) compared to unrolling algorithms due to more iterations, when compared with fixed number of unrolls in the latter. Our experiments in Fig. 6 show that the increased computational complexity translate to an improvement in robustness performance over unrolled algorithms, when Lipschitz regularization is applied on the networks.

II.   BACKGROUND

We consider recovery of an image $\mathbf{x}$ from its noisy undersampled measurements $\mathbf{b}$, specified by

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

where $\mathbf{A}$ is a linear operator and $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ is an additive white Gaussian noise. The measurement model provides a conditional probability $p(\mathbf{b} \mid \mathbf{x}) = \mathcal{N}(\mathbf{Ax}, \sigma^2\mathbf{I})$. The maximum a posteriori (MAP) estimation of $\mathbf{x}$ from the measurements $\mathbf{b}$ poses the recovery as

$$\mathbf{x}_{\text{MAP}} = \underset{\mathbf{x}}{\text{argmax}} \log p(\mathbf{x} \mid \mathbf{b}).$$ (2)

Using Bayes' rule, $p(\mathbf{x} \mid \mathbf{b}) \propto p(\mathbf{b} \mid \mathbf{x})p(\mathbf{x})$ and from the monotonicity of the log function, one obtains

$$\mathbf{x}_{\text{MAP}} = \underset{\mathbf{x}}{\text{argmin}} C(\mathbf{x}),$$

where

$$C(\mathbf{x}) = \underbrace{\frac{\lambda}{2} \parallel \mathbf{A}\mathbf{x} - \mathbf{b} \parallel_2^2}_{\mathscr{D}(\mathbf{x}) = -\log p(\mathbf{b} \mid \mathbf{x})} + \underbrace{\phi(\mathbf{x})}_{-\log p(\mathbf{x})}.$$ (3)

Here, $\lambda = \frac{1}{\sigma^2}$. The first term $\mathscr{D}(\mathbf{x}) = -\log p(\mathbf{b} \mid \mathbf{x})$ is the data-consistency term, while the second term is the log prior. Compressed sensing algorithms use convex prior distributions (e.g., $\phi(\mathbf{x}) = \parallel \mathbf{x} \parallel_{\ell_1}$) to result in a strongly convex cost function with unique minimum.

We note that the minimum of (3) satisfies the fixed-point relation:

$$\underbrace{\lambda \mathbf{A}^H(\mathbf{A}\mathbf{x} - \mathbf{b})}_{\mathscr{G}(\mathbf{x})} + \underbrace{\nabla_{\mathbf{x}}\phi(\mathbf{x})}_{\mathscr{F}(\mathbf{x})} = 0,$$ (4)

where $\mathbf{A}^H$ is the Hermitian operator of $\mathbf{A}$. We note that the first term $\mathscr{G}(\mathbf{x})$ is the noise in the measurements, translated to the image domain. When the above fixed-point relation holds, $\mathscr{F}(\mathbf{x})$ is essentially a noise estimator, often referred to as the score function. $\mathscr{F}$ points to the maximum of the prior $p(\mathbf{x})$.

Several algorithms that converge to the fixed point of (4) have been introduced in the CS setting [30]–[32]. For example, forward-backward algorithms rewrite (4) as $(\mathscr{I} + \alpha\mathscr{F})(\mathbf{x}) = (\mathscr{I} - \alpha\mathscr{G})(\mathbf{x})$, $\alpha > 0$, which has the same fixed point as (4). Classical PG algorithms use the iterative rule $\mathbf{x}_{n+1} = (\mathscr{I} + \alpha\mathscr{F})^{-1}(\mathscr{I} - \alpha\mathscr{G})(\mathbf{x}_n)$ that converges to the fixed point of (4). In the linear measurement setting (1), this translates to

$$\mathbf{x}_{n+1} = \underbrace{(\mathscr{I} + \alpha\mathscr{F})^{-1}}_{\text{prox}_\alpha\phi}(\mathbf{x}_n - \underbrace{\alpha\lambda\mathbf{A}^H(\mathbf{A}\mathbf{x}_n - \mathbf{b})}_{\mathscr{G}(\mathbf{x}_n)})$$ (5)

Here, $\text{prox}_\alpha\phi$ is the proximal operator of ϕ.

A. Plug and play methods

The steepest descent update $\mathbf{z}_n = \mathbf{x}_n - \alpha\lambda\mathbf{A}^H(\mathbf{A}\mathbf{x}_n - \mathbf{b})$ improves the data consistency, while the proximal mapping $\mathbf{x}_{n+1} = (\text{prox}_\alpha\phi)(\mathbf{z}_n)$ in (5) can be viewed as *denoising* the current solution $\mathbf{x}_n - \alpha\lambda\mathbf{A}^H(\mathbf{A}\mathbf{x}_n - \mathbf{b})$, thus moving the iterate towards the maximum of prior $p(\mathbf{x})$. Plug and

play methods replace the proximal mapping with off-the-shelf or CNN denoisers $\mathscr{H}_\theta$ [6], [13], [17], [33]:

$$\mathbf{x}_{n+1} = \mathscr{H}_\theta\Big(\mathbf{x}_n - \alpha\lambda\mathbf{A}^H(\mathbf{A}\mathbf{x}_n - \mathbf{b})\Big) = \mathscr{T}(\mathbf{x}_n, \theta) \tag{6}$$

Here, θ denotes the learnable parameters.

There are PnP algorithms that use different optimization algorithms (e.g., ADMM, PG) with convergence guarantees to the fixed point $\mathbf{x}^* = \mathscr{H}_\theta(\mathbf{x}^* - \alpha\mathscr{G}(\mathbf{x}^*))$ [13]. The solutions obtained by these approaches often do not have a one-to-one correspondence to the MAP setting in (3); they may be better understood from the consensus equilibrium setting [7]. See [13] for a detailed review of the PnP framework and associated convergence guarantees.

B.   Unrolled algorithms

Unrolled optimization schemes [14]–[16], [18] aim to learn a CNN denoiser, either as a prior or to replace the proximal mapping as in (6). A key difference with PnP is in the training strategy. The alternation between the physics-based data consistency (DC) update and the CNN update is unrolled for a finite number of iterations and trained end-to-end, while PnP methods alternate between the DC update and the pre-trained CNN. Unrolled schemes learn the CNN parameters that offer improved reconstruction for a specific sampling operator $\mathbf{A}$. These schemes obtain a deep network with shared CNN blocks. The parameters θ of the CNN are optimized to minimize the loss $L = \sum_i \| \tilde{\mathbf{x}}_{\theta,i} - \mathbf{x}_i \|_2^2$, where $\mathbf{x}_i$ are the ground truth images and $\tilde{\mathbf{x}}_{\theta,i}$ are the output of the deep network. The main challenge of this scheme is the high memory demand of the unrolled algorithm, especially in higher-dimensional (e.g., 3D, 2D + time) settings. This is mainly due to memory required for backpropagation updates scaling linearly with the number of unrolls. While memory-efficient techniques [21], [22] have been proposed, these methods come at the cost of increased computational complexity during training. The choice of minimum number of unrolls to offer good performance is usually ten [15], [16], which is feasible for 2D problems. However, this approach is often infeasible for higher-dimensional applications (3D, 4D, 5D).

C.   Deep equilibrium models

To overcome the challenge associated with unrolled schemes, [24] adapted the elegant DEQ approach introduced in [23]. Deep equilibrium models assume that the forward iterations in (6) are run until convergence to the fixed point $\mathbf{x}^*$ that satisfies $\mathbf{x}^* = \mathscr{T}(\mathbf{x}^*, \theta)$. This approach allows one to compute the back-propagation steps using fixed-point iterations [23], [24] with just one physical layer.

Unlike unrolled methods, the convergence of the forward and backward iterations are key to maintaining the accuracy of backpropagation in DEQ methods. The convergence of such algorithms are analyzed in [10], [24]. In the general case with a full-rank $\mathbf{A}$ considered in [10], convergence of the forward-backward splitting (FBS) algorithm in (6) is guaranteed when the Lipschitz constant of $\mathscr{H}_\theta$ satisfies $L[\mathscr{H}_\theta] \leq 2\mu_{min}/(\mu_{max} - \mu_{min})$, where μ_{max} and μ_{min} are the maximum and minimum eigenvalues of $\mathbf{A}^H\mathbf{A}$ [10]. In many inverse problems (e.g.,

parallel MRI with high acceleration), $\mathbf{A}$ is ill conditioned $\mu_{\max} >> \mu_{\min}$. In these cases $L[\mathcal{H}_\theta]$ should be close to zero to ensure convergence. Similar issues exist with PnP-ADMM and PnP-PROX, as discussed in [24].

III. Monotone Operator Learning

The main goal of this work is to introduce DEQ algorithms that share the desirable properties of CS algorithms, including uniqueness of the solution, robustness to input perturbations, and guaranteed fast convergence. We constrain $\mathcal{F}(\mathbf{x})$ to be an m-monotone ($m > 0$) CNN network to achieve these goals.

A. Monotone Operators

We constrain the CNN module $\mathcal{F}$ to be m-monotone:

Assumption: The operator $\mathcal{F}: \mathbb{C}^M \to \mathbb{C}^M$ is m-monotone if:

$$\Re(\langle \mathbf{x} - \mathbf{y}, \mathcal{F}(\mathbf{x}) - \mathcal{F}(\mathbf{y})\rangle) \geq m \parallel \mathbf{x} - \mathbf{y} \parallel_2^2, \quad m > 0, \tag{7}$$

for all $\mathbf{x}, \mathbf{y} \in \mathbb{C}^M$. Here, $\Re(\cdot)$ denotes the real part. Monotone operators enjoy several desirable properties and have been carefully studied in the context of convex optimization algorithms [34], mainly due to its following relation with convex priors.

Lemma III.1.—[9], [11] Let $\phi: \mathbb{C}^M \to \mathbb{R}_+$ be a proper, continuously differentiable, strongly convex function with $m > 0$. Then $\mathcal{F} = \nabla\phi$ is an m-monotone operator.

While derivatives of convex priors are monotone, the converse is not true in general. Our results show that the parameter m plays an important role in the convergence, uniqueness, and robustness of the algorithm to perturbations. In many CS applications, $\mathbf{A}$ often has a large null space, and hence the data-consistency term is not strictly convex. The following result shows that constraining $\mathcal{F}$ to be m-monotone is necessary and sufficient to ensure the uniqueness of the fixed point of (4).

Proposition III.2.—The fixed point of (4) is unique for a specific $\mathbf{b}$, iff $\mathcal{F}$ is m-monotone with $m > 0$.

The proof is provided in the Appendix. The following result shows that monotone operators can be represented efficiently as a residual neural network.

Proposition III.3.—If $\mathcal{H}_\theta: \mathbb{C}^M \to \mathbb{C}^M$ is $a(1-m)$ Lipschitz function:

$$L[\mathcal{H}_\theta] = (1 - m); \quad 0 < m < 1. \tag{8}$$

Then, the residual function $\mathcal{F}: \mathbb{C}^M \to \mathbb{C}^M$

$$\mathcal{F} = \mathcal{I} - \mathcal{H}_\theta, \tag{9}$$

is m-monotone:

$$\Re(\langle \mathscr{F}(\mathbf{x}) - \mathscr{F}(\mathbf{y}), \mathbf{x} - \mathbf{y}\rangle) \geq m \parallel \mathbf{x} - \mathbf{y} \parallel_2^2 . \tag{10}$$

In addition, the Lipschitz constant of $\mathscr{F}$ is $2 - m$:

$$\parallel \mathscr{F}(\mathbf{x}) - \mathscr{F}(\mathbf{y}) \parallel_2 \leq (2 - m) \parallel \mathbf{x} - \mathbf{y} \parallel_2 . \tag{11}$$

This result allows us to construct a monotone operator as a residual CNN. Because $\mathscr{F}$ is a score network that predicts the noise, $\mathscr{H}_\theta$ can be viewed as a denoiser.

B.   Proposed iterative algorithm

We now introduce a novel forward-backward algorithm using m-monotone CNN $\mathscr{F}$. To obtain an algorithm with guaranteed convergence even when $\mathbf{A}$ is low-rank, we swap $\mathscr{F}$ and $\mathscr{G}$ in (5):

$$\mathbf{x}_{n+1} = \underbrace{(\mathscr{I} + \alpha\mathscr{G})^{-1}}_{\mathrm{prox}_\alpha\mathscr{D}}(\mathscr{I} - \alpha\mathscr{F})(\mathbf{x}_n) . \tag{12}$$

Therefore, this approach involves a gradient descent to improve the prior, followed by a proximal map of the data term. A similar swapping approach was introduced in [28] to explain MoDL [16]. When $\mathscr{F}$ is an m-monotone operator, the Lipschitz constant of the gradient descent step $(\mathscr{I} - \alpha\mathscr{F})(\mathbf{x})$ can be made lower than one as shown in Lemma IV.1, while that of $\mathrm{prox}_\alpha\mathscr{D}$ is upper-bounded by one. This ensures that the resulting algorithm converges.

The fixed points of the above relation are equal to the fixed points of (3) for all $\alpha > 0$. In the linear setting considered in (3), suppose we have $\mathbf{p} = (\mathrm{prox}_\alpha\mathscr{D})(\mathbf{u}) = (\mathscr{I} + \alpha\mathscr{G})^{-1}\mathbf{u}$, which is the solution of $\mathbf{p} + \alpha\underbrace{\left(\lambda\mathbf{A}^H(\mathbf{A}(\mathbf{p}) - \mathbf{b})\right)}_{\mathscr{G}(\mathbf{p})} = \mathbf{u}$, or

$$(\mathrm{prox}_\alpha\mathscr{D})(\mathbf{u}) = \underbrace{\left(\mathbf{I} + \alpha\lambda\mathbf{A}^H\mathbf{A}\right)^{-1}}_{\mathcal{Q}_\alpha}\left(\mathbf{u} + \alpha\lambda\mathbf{A}^H\mathbf{b}\right) \tag{13}$$

Combining with (9) and (12), we obtain the proposed MOL algorithm:

$$\mathbf{x}_{n+1} = \underbrace{\left(\mathbf{I} + \alpha\lambda\mathbf{A}^H\mathbf{A}\right)^{-1}\left((1 - \alpha)\mathbf{x}_n + \alpha\mathscr{H}_\theta(\mathbf{x}_n)\right)}_{\mathscr{T}_{\mathrm{MOL}}(\mathbf{x}_n)} +$$
$$\underbrace{\left(\mathbf{I} + \alpha\lambda\mathbf{A}^H\mathbf{A}\right)^{-1}\left(\alpha\lambda\mathbf{A}^H\mathbf{b}\right)}_{\mathbf{z}} \tag{14}$$

$$= \mathscr{T}_{\mathrm{MOL}}(\mathbf{x}_n) + \mathbf{z} \tag{15}$$

We show below that this iterative rule will converge to a unique fixed point $\mathbf{x}^*(\mathbf{b})$ specified by $\mathscr{T}_{\mathrm{MOL}}(\mathbf{x}^*(\mathbf{b})) + \mathbf{z} = \mathbf{x}^*(\mathbf{b})$, which is identical to (4).

C. Relation to existing algorithms

We now consider a special case, which has been introduced by other researchers. When $\alpha = 1$, the fixed-point algorithm in the iterative rule in (14) can be rewritten as

$$\mathbf{x}_{n+1} = \left(\mathbf{I} + \lambda \mathbf{A}^H \mathbf{A}\right)^{-1}\left(\mathscr{H}_\theta(\mathbf{x}_n) + \lambda \mathbf{A}^H \mathbf{b}\right). \tag{16}$$

We note that the above update rule is used by multiple algorithms [16], [28], [8, equation (37)]. This update rule has been used in fixed-point RED algorithm (see [8, equation (37)]). Model-based deep learning (MoDL) [16] has trained $\mathscr{H}_\theta$ by unrolling the iterative algorithm with a fixed number of iterations; it did not require the iterative rule to converge. Our analysis shows that the corresponding DEQ algorithms will converge only if the Lipschitz constant of $\mathscr{H}_\theta$ is very low, which would translate to poor performance. The update rule in (14) can be viewed as a damped version of MoDL or the fixed-point RED algorithm. As will be seen in our analysis later, the use of the damping factor $\alpha < 1$ enables us to relax the constraints on the CNN network $\mathscr{H}_\theta$ that are needed for convergence, which will translate to improved performance. For both MoDL and the proposed fixed-point algorithms, the scalar λ is kept trainable.

IV. Theoretical Analysis

The monotone nature of $\mathscr{F}$ allows us to characterize the fixed point of the iterative algorithm $\mathbf{x}_{n+1} = \mathscr{T}_{\mathrm{MOL}}(\mathbf{x}_n) + \mathbf{z}$ in (14). In particular, we will now analyze the convergence and the robustness of the solution to input perturbations.

A. Convergence of the algorithm to a fixed point

The algorithm specified by (14) converges if the Lipschitz constant of

$$\mathscr{T}_{\mathrm{MOL}}(\mathbf{x}) = \mathcal{Q}_\alpha(\mathbf{x} - \alpha\underbrace{(\mathscr{I} - \mathscr{H}_\theta)\mathbf{x}}_{\mathscr{F}}) = \mathcal{Q}_\alpha \circ \underbrace{(\mathscr{I} - \alpha\mathscr{F})}_{\mathscr{R}}\mathbf{x} \tag{17}$$

is less than one. We will now focus on the composition $\mathcal{Q}_\alpha \circ \mathscr{R}$. When $\mathbf{A}^H\mathbf{A}$ is full-rank, $\mathcal{Q}_\alpha = \left(\mathbf{I} + \alpha\lambda\mathbf{A}^H\mathbf{A}\right)^{-1}$ is a contraction. In many inverse problems including super-resolution and compressed sensing, the Lipschitz constant of $\mathcal{Q}_\alpha$ is 1. Assuming that $\mathscr{F}$ is m-monotone, we have the following result for $L[\mathscr{R}]$.

Lemma IV.1.—Let $\mathscr{F}: \mathbb{C}^M \to \mathbb{C}^M$ be an m-monotone operator. Then, the operator $\mathscr{R} = (\mathscr{I} - \alpha\mathscr{F})$ has a Lipschitz constant of

$$L[\mathscr{R}] \le \sqrt{1 - 2\alpha m + \alpha^2(2 - m)^2}. \tag{18}$$

From the above relation, we note that $\mathcal{R}$ is a contraction (i.e., $L[\mathcal{R}] < 1$) when the damping factor α satisfies

$$\alpha < \frac{2m}{(2-m)^2} = \alpha_{\max} . \tag{19}$$

Proposition IV.2.—Consider the algorithm specified by (14), where $\mathcal{F}$ is an m-monotone operator. Assume that (14) has a fixed point specified by $\mathbf{x}^*(\mathbf{b})$. Then,

$$\| \mathbf{x}_n - \mathbf{x}^*(\mathbf{b}) \|_2 \leq (L[\mathcal{T}_{\mathrm{MOL}}])^n \| \mathbf{x}_0 - \mathbf{x}^*(\mathbf{b}) \|_2 , \tag{20}$$

where $L[\mathcal{T}_{\mathrm{MOL}}] = \frac{1}{(1 + \lambda \mu_{\min})} L[\mathcal{R}]$. Here, $\mu_{\min}$ is the minimum eigenvalue of $\mathbf{A}^H \mathbf{A}$ and $L[\mathcal{R}]$ is specified by (18).

We note that $\mathcal{T}_{\mathrm{MOL}}$ being a contraction translates to geometric convergence with a factor of $L[\mathcal{T}_{\mathrm{MOL}}]$; this is faster than the sublinear convergence rates [9], [11] available for ISTA [31] and ADMM [30] in the CS setting ($\mu_{\min} = 0$).

B. Benefit of damping parameter α in the MOL algorithm (14)

We note from Section III-C that the algorithms [8], [16], [28] correspond to the special case of $\alpha = 1$. Setting $\alpha = 1$ in (19), we see from Lemma IV.1 and (19) that the DEQ algorithm will converge if

$$m \geq 3 - \sqrt{5} = 0.76 \tag{21}$$

or $L[\mathcal{H}_\theta] < 0.24$. As discussed previously, the denoising ability of a network is dependent on its Lipschitz bound; a smaller $L[\mathcal{H}_\theta]$ bound translates to poor performance of the resulting MOL algorithm. The use of the damping factor $\alpha < 1$ allows us to use denoising networks $\mathcal{H}_\theta$ with larger Lipschitz bounds and hence improved denoising performance. For instance, if we choose $m = 0.1$; $L[\mathcal{H}_\theta] = 0.9$, from (19), the algorithm will converge if $\alpha < 0.055$.

C. Robustness of the solution to input perturbation

The following result shows that the robustness of the proposed algorithm is dependent on m or, equivalently, $L[\mathcal{H}_\theta]$. We note that the link between the Lipschitz bound on the network and sensitivity to perturbations is straightforward in a direct inversion scheme (e.g., UNET). By contrast, such relations are not available in the context of DEQ-based image recovery algorithms, to the best of our knowledge.

Proposition IV.3.—Consider $\mathbf{z}_1$ and $\mathbf{z}_2$ to be measurements with $\delta = \mathbf{z}_2 - \mathbf{z}_1$ as the perturbation. Let the corresponding outputs of the MOL algorithm be $\mathbf{x}^*(\mathbf{z}_1)$ and $\mathbf{x}^*(\mathbf{z}_2)$, respectively, with $\Delta = \mathbf{x}^*(\mathbf{z}_2) - \mathbf{x}^*(\mathbf{z}_1)$ as the output perturbation,

$$\| \Delta \|_2 \leq \frac{\alpha\lambda/(1 + \lambda\mu_{\min})}{1 - \sqrt{1 - 2\alpha m + \alpha^2(2 - m)^2}} \| \delta \|_2 . \tag{22}$$

The above result shows that the norm of the perturbation in the reconstructed images is linearly dependent on the norm of the input perturbations. The constant factor is a function of the monotonicity parameter m and the step size α in Fig. 2, where we plot the constant term without the λ parameter for different values of m. The plots show that the constant decreases with m roughly at $1/m$ rate. We note that the algorithm will converge to the same fixed point as long as the damping parameter α satisfies the condition (19). Hence, we consider the case with small damping parameter $\alpha \to 0$ and set $\mu_{\min} = 0$ corresponding to the CS and super-resolution settings, when we obtain a simpler expression:

$$\lim_{\alpha \to 0} \| \Delta \|_2 \leq \frac{\lambda}{m} \| \delta \|_2 . \tag{23}$$

The above results show that the robustness of the algorithm is fundamentally related to m; a higher value of m translates to a more robust algorithm. However, note that the Lipschitz constant of the denoiser $\mathscr{H}_\theta$ is specified by $L[\mathscr{H}_\theta] = 1 - m; m > 0$. We need to choose a denoising network with a lower Lipschitz constant, which translates to lower performance, to make the resulting algorithm more robust to perturbations. There is a trade-off between robustness and performance of the algorithm, controlled by either the parameter m or the Lipschitz constant $L[\mathscr{H}_\theta]$.

V. IMPLEMENTATION DETAILS

A. DEQ: forward and backward propagation

During training and inference, we use the forward iteration rule, $\mathbf{x}_{n+1} = \mathscr{T}_{\mathrm{MOL}}(\mathbf{x}_n) + \mathbf{z}$. We terminate the algorithm at the nth iteration $\mathbf{x}_n$ if it satisfies,

$$e_n = \frac{\| \mathbf{x}_n - \mathbf{x}_{n-1} \|_2}{\| \mathbf{x}_{n-1} \|_2} \leq \kappa . \tag{24}$$

We set $\kappa = 1 \times 10^{-4}$ for the experiments. Please see the pseudo-code for forward propagation in Algorithm 1 of the supplementary material. We denote the fixed point of the algorithm as $\mathbf{x}^*(\mathbf{b})$, such that $\mathbf{x}^*(\mathbf{b}) \approx \mathscr{T}_{\mathrm{MOL}}(\mathbf{x}^*(\mathbf{b})) + \mathbf{z}$. DEQ schemes [24] rely on fixed-point iterations for back-propagating the gradients. The details are shown in Algorithm 1 and 2 in the supplementary material. The iterations are evaluated until convergence, using similar termination conditions as in (24).

B. Implementation of the monotone CNN operator

We note from (III.3) that a monotone $\mathscr{F} = \mathscr{I} - \mathscr{H}_\theta$ can be learned by constraining the Lipschitz constant of the denoiser network $\mathscr{H}_\theta$. We propose two different implementations of the MOL algorithm, which differs in the way the Lipschitz constraint is implemented.

1)   Spectral normalization: Similar to [35], we use normalization of the spectral norm of the CNN layers to constrain $L[\mathscr{H}_\theta]$. In particular, we bound the spectral norm of each layer to $\sqrt[s]{L[\mathscr{H}_\theta]}$, where s is the number of layers. We term this version of MOL as MOL-SN (MOL-spectral normalization). This approach can guarantee $\mathscr{H}_\theta$ to be a contraction, and hence the guarantees are satisfied exactly. However, the product of the spectral norms of the individual layers is a conservative estimate of the Lipschitz constant of the network. As shown by our experiments, the use of spectral normalization in our setting (MOL-SN) translates to lower performance. Another challenge with the spectral normalization approach is that it restricts the type of networks that can be used; architectures with skipped connections cannot be used with this strategy. We note that spectral normalization is indeed a conservative bound for the Lipschitz constant and hence may over constrain the network, translating to lower performance.

2)   Approximating monotone constraint using a Lipschitz penalty: Motivated by [36], we propose to train the MOL algorithm using a training loss which minimizes a constrained optimization problem. In [37], authors use Jacobian regularization instead to learn a contractive network. The estimation of the Lipschitz constant of $\mathscr{H}_\theta$ is posed as a maximization problem [36]:

$$l[\mathscr{H}_\theta] = \max_{\mathbf{x} \in S} \sup_{\boldsymbol{\eta}} \underbrace{\overbrace{\frac{\parallel \mathscr{H}_\theta(\mathbf{x}+\boldsymbol{\eta}) - \mathscr{H}_\theta(\mathbf{x}) \parallel_2^2}{\parallel \boldsymbol{\eta} \parallel_2^2}}^{P(\mathbf{x})}}_{p(\mathbf{x}, \boldsymbol{\eta})} \tag{25}$$

We denote the estimated Lipschitz constant as $l[\mathscr{H}_\theta]$ to differentiate it from the true Lipschitz constant $L[\mathscr{H}_\theta]$. Here, $\boldsymbol{\eta}$ is a perturbation, and S is the set of training data samples. We note that this estimate is less conservative than the one using spectral normalization. However, this is only an approximation of the true Lipschitz constant, even though our experiments show that the use of this estimate can indeed result in algorithms with convergence and robustness as predicted by the theory. We note that several researchers have recently introduced tighter estimates for the Lipschitz constant [38], [39], and they could be used to replace the above estimate. The theoretical results derived in the earlier sections will still hold, irrespective of the specific choice of the Lipschitz estimation strategy. We initialize η by a small random vector, which is then updated using steepest ascent. It is solved using a log-barrier approach which constrains the estimated Lipschitz of the CNN below a threshold value. The total training loss is a linear combination of the log-barrier term and the supervised mean squared error (MSE) loss. We call this method as MOL-LR (MOL-Lipschitz Regularized).

C.   Training the MOL-LR algorithm

In the supervised learning setting, we propose to minimize

$$\mathscr{C}(\theta) = \underbrace{\sum_{i=0}^{N_t} \| \mathbf{x}_i^* - \mathbf{x}_i \|_2^2}_{\mathscr{C}} \text{ such that}$$

(26)

$$\underbrace{P(\mathbf{x}_i^*)}_{\text{Local Lipschitz estimate}} \leq T; i = 0, \dots, N_t$$

Here, the threshold is selected as $T = 1 - m$ and $P(\mathbf{x})$ is indicated in (25). The above loss function is minimized with respect to parameters θ of the CNN $\mathscr{H}_\theta$. $\mathbf{x}_i^*$ is a fixed point of (4) described in Section V-A, which is dependent on the CNN parameters θ. $\mathbf{x}_i; i = 0, \dots, N_t$ and $\mathbf{b}_i, i = 0, \dots, N_t$ are the ground truth images in the training dataset and the corresponding under-sampled measurements, respectively. We solve the above constrained optimization scheme by using a log-barrier approach:

$$\theta^* = \underset{\theta}{\arg\min} \underbrace{\sum_{i=0}^{N_t} \left(\| \mathbf{x}_i^* - \mathbf{x}_i \|_2^2 - \beta \log(T - P(\mathbf{x}_i^*)) \right)}_{C_i}.$$

(27)

Here β is a parameter that decays with training epochs similar to conventional log-barrier methods. This optimization strategy ensures that the estimate does not exceed T. For implementation purposes, we evaluate the worst-case perturbations $\boldsymbol{\eta}_i$ for each $\mathbf{x}_i^*$ by maximizing (25) at each epoch. These perturbations are then assumed to be fixed to evaluate the above loss function, which is used to optimize θ. The training algorithm is illustrated in the pseudo-code shown in Algorithm 1, which is illustrated for a batch size of a single image and gradient descent for simplicity.

Algorithm 1:

Training: input=training data $\mathbf{x}_i; i = 1, \dots, N_t$

1:   **for**
     $\text{ep} = 1, 2, \dots$
     do

2:       **for**
         $i = 1, 2, \dots, N_t$
         do

3:           Determine
             $\mathbf{x}_i^*$
             using DEQ forward iterations

4:           Determine
             $\boldsymbol{\eta}_i^* = \arg\max_{\eta} p(\mathbf{x}_i^*, \boldsymbol{\eta})$

5:
             $C_i = \| \mathbf{x}_i - \mathbf{x}_i^* \|^2 - \frac{1}{m} \log(T - p(\mathscr{H}_\theta(\mathbf{x}_i^*), \boldsymbol{\eta}_i^*))$

6:           Determine
             $\nabla_\theta C_i$
             using DEQ backward iterations

7:
$$\theta \leftarrow \theta - \gamma \nabla_\theta C_i$$
, where
C_i
is the loss in (27)

8:    **end for**

9:    **end for**

D. Unrolled algorithms used for comparison

We compare the proposed MOL algorithm against SENSE [40], MoDL [16], ADMM-Net [19], DE-GRAD [24], and UNET [29]. SENSE is a CS-based approach that uses a forward model consisting of coil sensitivity weighting and undersampled Fourier transform. MoDL and ADMM-Net are unrolled deep learning algorithms, which alternate between the DC step and the CNN-based denoising step. Both approaches are trained in an end-to-end fashion for 10 iterations. DE-GRAD is a deep equilibrium network, where we use spectral normalization as described in [24]. UNET is a direct inversion approach, which uses a CNN without any DC steps. We choose five-layer CNNs for all the unrolled deep-learning and DEQ based algorithms used for comparisons. The CNN architecture is shown in Fig. 1 (b).

We consider two versions of MOL: MOL-SN, which relies on spectral normalization during training to constrain the Lipschitz constant of the overall network, and MOL-LR, which consists of an additional loss term computing the Lipschitz constant of the CNN. We also consider Lipschitz regularized versions of UNET, ADMM-Net, and MoDL for robustness experiments, and those are denoted by UNET-LR, ADMM-Net-LR, and MoDL-LR, respectively. The Lipschitz of the CNNs in these methods is regularized by the proposed training strategy in (27). For 2D+time experiments on cardiac data, we compare a 2D+time version of MOL-LR (with 3D convolutions) against the 2D MoDL (with 2D convolutions).

E. Architecture of the CNNs and training details

The MOL architecture is shown in Fig. 1. In our 2D experiments, the CNN $\mathcal{H}_\theta$ consists of five 2D convolution layers, each followed by rectified linear unit (ReLU) non-linearity, except for the last layer. The convolution layers consist of 64 filters with 3×3 kernels. The parameter λ in (14), weighing the DC term, is kept trainable. A SENSE reconstruction with $\lambda_0 = 100$ is performed initially on the undersampled image $\mathbf{A}^H \mathbf{b}$ to initialize the MOL network as $\mathbf{x}_0$. A similar approach is used for the other deep learning networks (MoDL, ADMM-Net, UNET) to ensure fair comparisons. We share the CNN weights across the iterations for all the unrolled deep learning algorithms (MoDL, ADMM-Net). We use a full-size UNET, consisting of four layers of pooling and unpooling. Note that the number of trainable parameters in the chosen UNET is at least twice the number of parameters in five-layer CNNs. In 2D+time experiments, a five-layer CNN is chosen; it is similar to the 2D case, with the exception of 3D convolution layers instead of 2D. We set $\alpha = 0.055$, which corresponds to $m = 0.1$ or $L[\mathcal{H}_\theta] = 0.9$ for all MOL algorithms. We keep this variable fixed and non-trainable because this parameter is chosen based on (19), which depends on the

bound $T = 1 - m$. We show in Fig. 3 that the MOL algorithm with $\alpha = 1$, which is the DEQ extension of MoDL [16] and RED [8], diverges.

All the trainings are performed on a 16 GB NVIDIA P100 GPU. The CNN weights are Xavier initialized and trained using an Adam optimizer for 200 epochs. The learning rates for updating CNN weights and λ are chosen empirically as 10^{-4} and $1{:}0$, respectively. MoDL and ADMM-Net are unrolled for 10 iterations; MOL, on the other hand, consumes memory equivalent to a single iteration. All the methods are implemented in PyTorch. During inference, the reconstruction results are quantitatively evaluated in terms of the Structural Similarity Index (SSIM) [41] and the Peak Signal-to-Noise Ratio (PSNR).

F. Computing worst-case (adversarial) perturbations

We determine the robustness of the networks to Gaussian as well as worst-case perturbations. We determine the worst-case perturbation γ by solving the following optimization problem:

$$\gamma^* = \max_{\gamma; \; \| \gamma \|_2 \; < \epsilon \; \cdot \; \| \mathbf{b} \|_2} \underbrace{\frac{\| \; \mathbf{x}^*(\mathbf{b} + \gamma) - \mathbf{x}^*(\mathbf{b}) \; \|_2^2}{}}_{U(\gamma)} \tag{28}$$

Here, γ is solved using a projected gradient algorithm; we alternate between gradient ascent steps and renormalization of γ to satisfy the constraint $\| \gamma \|_2 \; < \epsilon \; \cdot \; \| \mathbf{b} \|_2$. For MOL, we use the fixed-point iterations described in Section V-A to compute the gradient $\nabla_\gamma U(\gamma)$. We note that the fixed-point iterations for back-propagation are accurate as long as the forward and backward iterations converge. We track the maximum number of iterations and the termination criterion (24) to ensure that the iterations converge. In this work we relied on an ℓ_2 norm on the perturbations, while ℓ_∞ constraints have also been used in the literature [25].

G. 2D Brain and knee datasets

We used the 2D multi-coil brain data from the publicly available Calgary-Campinas Public (CCP) Dataset [42]. The dataset consists of twelve-coil T1-weighted brain data from 117 healthy subjects, collected on a 3.0 Tesla MRI scanner. The scan parameters are: TR (repetition time)/TE (echo time)/TI (inversion time) = 6.3 ms/2.6ms/650 ms or TR/TE/TI = 7.4ms/3.1ms/400ms. Matrix sizes are 256×208×170/180 with 256, 208, and 170/180 being the readout, phase encoding, and slice encoding directions, respectively. For the experiments, we choose subjects with fully sampled data (67 out of 117) and split them into training (45), validation (2), and testing (20) sets. The k-space measurements are retrospectively undersampled along the phase and slice encoding directions using a four-fold 2D non-uniform variable-density mask.

We also perform experiments on the multi-channel knee MRI datasets from the fastMRI challenge [43]. It consists of 15-coil coronal proton-density weighted knee data with or without fat suppression. The sequence parameters were: matrix size 320×320, in-plane resolution 0.5mm $\times$ 0.5mm, slice thickness 3mm, and repetition time (TR) varying from 2200 to 3000 ms, and echo time (TE) between 27 and 34 ms. We use the k-space measurements from 50 subjects for training, 5 for validation, and 20 for testing, respectively.

The data is retrospectively undersampled along the phase-encoding direction, for four-fold, using a 1D non-uniform variable density mask. In another set of experiments, we consider four-fold undersampling using 1D uniform mask.

H. Cardiac MRI datasets

Compressed sensing and low-rank methods have been extensively used to reduce the breath-hold duration in cardiac cine [44], [45]. Several authors have introduced unrolled algorithms for cardiac cine MRI acceleration. For instance, one of the initial works considered the independent recovery of 2D images using unrolled methods [18], together with data sharing. More recent works [46], [47] rely on a 15-iteration unrolled scheme, where they used separable (2+1)-D spatio-temporal convolutions to keep the memory demand manageable.

In this work, we show the preliminary utility of the proposed MOL approach with 3D CNN to accelerate cardiac cine MRI. We used the multi-coil cardiac data from the open-source OCMR Dataset [48]. We note that the high memory demand often restricts the training of unrolled algorithms such as MoDL [16] in the 2D+time setting. We chose data from 1.5 Tesla scans, which consists of a series of short-axis, long-axis, or four-chamber views. We use fifteen subjects for training, two for validation, and eight for testing. Each dataset consists of 20–25 time-frames per slice, with 1–3 slices per subject. We retrospectively undersample the k-t space data using a 1D non-uniform variable-density mask along the phase encode direction. In these experiments, we compare the MOL algorithm using a 3D network against the MoDL algorithm [16] using a 2D network. The MOL training is performed on 2D+time datasets. On the other hand, the MoDL algorithm [16] processes each of the time-frames independently, and hence is not capable of exploiting inter-frame dependencies.

VI. EXPERIMENTS AND RESULTS

The proposed method can handle wide range of inverse problems. We showcase its application in parallel MRI recovery from non-uniform and uniform undersampled acquisition settings in the subsequent sections. In addition, we also demonstrate it on image super-resolution problems. The super-resolution experiments and results are discussed at Section III in the supplementary material.

A. Characterization of the models

In Fig. 3, we show the characteristics of the different models during training. Fig. 3.(a) shows plots of Lipschitz constant against epochs for different methods. We note that the computed Lipschitz constant $l[\mathcal{H}_\theta]$ of the MOL-SN and DE-GRAD schemes that use spectral normalization stays around 0.7 and 0.4, respectively, which translates to lower performance seen from the testing error curves in (c). By contrast, the unrolled MoDL and ADMM-Net, and UNET, have no Lipschitz constraints and, therefore, have more flexibility in CNN weight updates. It is observed that the estimated Lipschitz constants of these networks often exceed 1. The proposed MOL-LR algorithm maintains the Lipschitz constant less than 0.9. The slightly lower performance of MOL-LR compared to that of other unrolled methods can be attributed to its lower Lipschitz constant.

The plot in Fig. 3.(b) shows the number of iterations of (14) (equivalently, the number of CNN function evaluations, denoted by nFE) needed to converge to the fixed point with a precision of $\kappa = 1 \times 10^{-4}$. As expected, MOL-SN and DE-GRAD, which have lower Lipschitz constants and hence higher m, converge more rapidly than MOL-LR, which has a lower value of m. We note that the MOL-LR for $\alpha = 1$ algorithm is essentially the MoDL algorithm [16] used in the DEQ setting. As predicted by theory, we note that this algorithm fails to converge, evidenced by the number of iterations (nFE) increasing rapidly around 100 epochs. Similarly, the MOL algorithm without Lipschitz constraints also diverges around 10 epochs. The proposed MOL-LR with $\alpha = 0.055$ converges at approximately nFE = 28.

B. Performance comparison in the parallel MRI setting

The comparison of performance of the algorithms on four-fold accelerated knee data is shown in Fig. 4 and Table I, respectively. Cartesian 1D non-uniform variable density mask has been used for undersampling the data. Table I reports the quantitative performance in terms of mean PSNR and SSIM on 20 subjects. We observe that the performance of MOL-LR is only marginally lower than ten-iteration MoDL and ADMM-Net. The marginally lower performance of MOL-LR can be attributed to the stricter Lipschitz constraint on the CNN blocks, compared to MoDL and ADMM-Net. This is also confirmed by our experiments on Calgary brain data in Fig. 6, where the performance of MoDL and ADMM-Net decreases even more with the addition of the Lipschitz constraint (see MoDL-LR and ADMM-Net-LR reconstructions with no input noise). We also note that the reduction in performance is higher for the MOL-SN and DE-GRAD, both of which use spectral normalization. The performance of the UNET is lower than that of the unrolled algorithms MoDL, ADMM-Net, and MOL-LR. The SENSE reconstruction performance is the lowest among all. The comparison of the average runtimes of the algorithms show that the MOL-LR scheme with around 25 DEQ iterations is around 2.5 times higher than MoDL and ADMM-Net with 10 unrolling steps. The qualitative comparisons are shown in Fig. 4. The error images show higher errors for MOL-SN, DE-GRAD, SENSE and UNET, while the error images from MOL-LR, MoDL, and ADMM-Net are comparable. We performed statistical tests to compare MOL-LR against the other methods in terms of PSNR and SSIM reported in Table I. This was done for PSNR using linear mixed model analysis with Dunnetts test for pairwise comparison of means. For SSIM values which did not meet normality assumptions, Friedmans test was used, with pairwise comparisons tested using Wilcoxon signed-rank test with Bonferroni adjustment applied to the p-values to account for multiplicity. The PSNR comparisons showed statistically significant differences between MOL-LR and each of the other methods. MoDL and ADMM-Net performed only slightly better than MOL-LR, with difference in mean PSNR of +0.40 (95% confidence interval (CI): 0.16, 0.64; p < 0.0001) and +0.29 (95% CI: 0.05, 0.53; p < 0.009), respectively. All the other methods underperformed compared to MOL-LR with a much larger mean difference in PSNR, from −1.81 (95% CI: −1.05, −2.05) for MoDL-LR to −5.30 (95% CI: −5.03, −5.54) for SENSE. Comparison of SSIM showed no significant difference between MOL-LR and MoDL (median difference 0.000; 95% CI: −0.001, +0.001; p = 1.00) and ADMM-Net (median difference 0.000; 95% CI: 0.000, +0.001; p = 1.00). All the other methods had significantly smaller SSIM compared to MOL-LR (p < 0.0001), from −0.003

(95% CI: −0.002, −0.004) for MoDL-LR and ADMM-Net-LR to −0.007 (95% CI: −0.006, −0.008) for SENSE and UNET-LR.

We also perform experiments to compare the proposed MOL-LR against unrolled method MoDL and the CS approach SENSE in uniform undersampling setting. The multi-channel knee data is four-fold undersampled using a uniform mask as shown in Fig. 5. Table II reports the quantitative results in terms of PSNR and SSIM on 20 subjects. The quality of MOL-LR is slightly lower than MoDL with no regularization, which is consistent. Both the methods significantly outperform SENSE which has visible aliasing in the reconstructed image as evident from Fig. 5. It is found through statistical analysis that MoDL is only slightly better than MOL-LR in PSNR (mean difference of +0.59; 95% CI: 0.43, 0.74; $p < 0.0001$), but with no significant difference in SSIM (median difference of 0.000; $p = 1.00$). SENSE underperformed compared to MOL-LR with a mean difference of −5.27 (95% CI: −5.11, − 5.43; $p < 0.0001$) for PSNR, and median difference of −0.007 (95% CI: −0.006, −0.008; $p < 0.0001$) for SSIM (See Table II). The statistical methods used in this case are the same as the ones used for non-uniform undersampling case, mentioned above.

Quantitative comparisons of the methods on four-fold accelerated Calagary brain data is reported by Table I in the supplementary material. For this case, cartesian 2D non-uniform variable density mask has been used. A similar trend is observed here in terms of performance metrics PSNR, SSIM and run-time of the algorithms. The reconstructed brain images for qualitative comparisons are shown in Fig. 6 in the main paper and also in Fig. 1 from the supplementary material. The first column of Fig. 6 shows reconstructions from different methods when no noise has been added. MOL-LR performs comparable to the unrolled algorithms MoDL and ADMM-Net. The Lipschitz constrained MoDL (MoDL-LR) and ADMM-Net (ADMM-Net-LR) show relatively lower performance due to constrained CNNs. Both UNET and UNET-LR show reduction in performance as compared to the unrolled algorithms while MOL-SN and DE-GRAD also show reduced performance due to Lipschitz constraint enforced through spectral normalization. Similar trends are also seen in a different slice in Fig. 1 of the supplementary material.

C. Robustness to input perturbations

We compare the robustness of the networks on four-fold accelerated brain data in Fig. 6 and Fig. 7 respectively. Specifically, we study the change in output with respect to the perturbations to input to determine the stability of the models.

The first column in Fig. 6 shows the reconstructions given by different methods when there is no additional noise in the measured k-space data (no-noise). The second and third column shows reconstructions when the measurements are corrupted by worst-case perturbations (adversarial) with energy as 10% and 15% of the energy of the measurements respectively. The fourth column shows reconstructed images when the measurements are corrupted by Gaussian noise with energy as 15% of the energy of the measurements. Here, MoDL, ADMM-Net and UNET are traditional methods with no Lipschitz constraint. By contrast, MOL-SN and DE-GRAD use spectral normalization, while MOL-LR uses the proposed Lipschitz constraint. MoDL-LR, ADMM-Net-LR, and UNET-LR correspond to the above traditional methods with the proposed Lipschitz constraints added to the CNN blocks.

The performance of MOL-LR is comparable to that of MoDL and ADMM-Net in the setting with no additional noise, which is also consistent with the findings in Table I and the PSNR plots in Fig. 7. The improved performance of these unrolled methods over UNET is well established. The MOL-SN and DE-GRAD schemes that use spectral normalization are associated with lower performance. We notice from the last column that the performance of all the Lipschitz-constrained methods only decrease by around 4 – 5 with the addition of 15% Gaussian noise. However, the performance of ADMM-Net, MoDL, and UNET drops by around 10 dB with adversarial perturbations of the same amount of norm. By contrast, the performance drop of the Lipschitz-constrained models are largely consistent between the Gaussian and the worst-case setting, indicating that the proposed constraint can stabilize unrolled methods as well. However, we note that ADMM-Net-LR, MoDL-LR, and UNET-LR are associated with a decreased performance in the case with no additional noise. The decrease in performance can be attributed to the more constrained CNN block. The MOL-LR scheme offers performance comparable to the competing methods in the case without additional noise, while it is also more robust to worst-case perturbations. The better performance of MOL-LR compared to other LR methods in the absence of additional noise perturbations can be attributed to the higher number of iterations (nFE=28) compared to the 10 unrolling steps used in those methods. These trends can also be appreciated from the plot of the PSNRs in Fig. 7. (a). The models without Lipschitz constraints (MoDL, ADMM-Net, UNET) exhibit a drastic drop with $\epsilon > 0.1$, whereas SENSE, MOL-LR, and MOL-SN have an approximately linear drop. We note that our theory predicts a linear drop in performance with MOL methods. MOL-SN and DE-GRAD (blue & red curves) are the flattest, which can be explained with the smaller Lipschitz constant. Although quantitative analysis of the reconstructions clearly show the proposed MOL-LR to be superior, a rigorous qualitative analysis by radiologists is needed to determine if the MOL-LR reconstructions are fit for clinical purposes.

D. Illustration in high-dimensional settings

The proposed MOL training approach only requires one physical layer to evaluate the forward and backward propagation; the memory demand of MOL-LR is ten times smaller than that of the unrolled networks with ten iterations. Our 2D experiments show that MOL-LR achieves performance similar to unrolled algorithms with a much lower memory footprint, which makes it an attractive option for large-scale problems. In this section, we illustrate the preliminary applicability of the MOL framework to large-scale problems. The joint recovery of 2D+time data using different undersampling patters for each time point can capitalize on the strong redundancy between the time-frames. However, it is challenging to use conventional unrolled optimization algorithms because of the high memory demand. We compare the performance of a 2D+time version of MOL-LR against a 2D MoDL (ten iterations) for recovery of time series of cardiac CINE MRI.

The reconstruction results for six-fold and four-fold accelerated CINE MRI recovery are shown in Fig. 8.(a) and Fig. 8.(b), respectively. The top two rows correspond to the diastole and systole phases, respectively. The third row corresponds to the time series. It is observed from the top two rows that the 2D+time MOL-LR approach is able to minimize spatial blurring when compared to the 2D MoDL approach. The frame-to-frame changes in aliasing

artifacts can also be appreciated from the time plots. Table III displays the mean PSNR and SSIM over eight test subjects. MOL-LR 3D outperforms MoDL 2D in terms of PSNR (by $\approx$ 2 dB) and SSIM at both the acceleration factors. The masks are shown in the bottom row. The four-fold and six-fold Poisson density sampling masks used in these experiments have eight lines in the center. While the preliminary experiments in this context are encouraging, more experiments are needed to compare the MOL in this setting to state-of-the-art dynamic MRI methods. We plan to pursue this in the future.

VII. CONCLUSION

We introduced a deep monotone operator learning framework for model-based deep learning to solve inverse problems in imaging. The proposed approach learns a monotone CNN in a deep equilibrium algorithm. The DEQ formulation enables forward and backward propagation using a single physical layer, thus significantly reducing the memory demand. The monotone constraint on the CNN allows us to introduce guarantees on the uniqueness of the solution, rapid convergence, and stability of the solution to input perturbations. We introduced two implementations that differ in the way the monotone constraint is imposed. The first approach relies on an exact spectral normalization strategy, while the second method relies on an approximate regularization approach. Our experiments show that both approaches result in convergent algorithms that are more robust to input perturbations than other deep learning approaches. However, the less conservative regularization-based implementation offers improved performance compared to the more constrained spectral normalization approach. The validations in the context of parallel MRI show that the proposed MOL framework provides performance similar to the unrolled MoDL algorithms, but with significantly reduced memory demand and improved robustness to worst-case input perturbations. The memory efficiency of the proposed scheme enabled us to demonstrate the preliminary utility of this scheme in a larger-scale (2D+time) problem. The preliminary comparisons in the super-resolution setting also show that the proposed method is broadly applicable to other linear inverse problems.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgments

VIII.: APPENDIX

A.   Proof of Proposition III.2

Proof. Assume that there exist two fixed points $\mathbf{x} \neq \mathbf{y}$ for a specific $\mathbf{b}$:

$$\lambda \mathbf{A}^H(\mathbf{A}\mathbf{x} - \mathbf{b}) + \mathscr{F}(\mathbf{x}) = 0 \tag{29}$$

$$\lambda \mathbf{A}^H (\mathbf{A}\mathbf{y} - \mathbf{b}) + \mathcal{F}(\mathbf{y}) = 0 \tag{30}$$

which gives

$$\mathbf{z} = \lambda \mathbf{A}^H (\mathbf{A}(\mathbf{x} - \mathbf{y})) + \mathcal{F}(\mathbf{x}) - \mathcal{F}(\mathbf{y}) = \mathbf{0}. \tag{31}$$

Setting, $\mathbf{v} = \mathbf{x} - \mathbf{y}$, we consider

$$\Re(\langle \mathbf{z}, \mathbf{v} \rangle) = \underbrace{\Re\left(\left\langle \lambda \mathbf{A}^H \mathbf{A}\mathbf{v}, \mathbf{v} \right\rangle\right)}_{\geq \lambda \mu_{\min} \|\mathbf{v}\|_2^2} +$$
$$\underbrace{\Re(\langle \mathcal{F}(\mathbf{x}) - \mathcal{F}(\mathbf{y}), \mathbf{v} \rangle)}_{\geq m \|\mathbf{v}\|_2^2}$$
$$\geq (\lambda \mu_{\min} + m) \|\mathbf{v}\|_2^2$$

where $\mu_{\min} \geq 0$ is the minimum eigenvalue of $\mathbf{A}^H\mathbf{A}$ operator and $\mathcal{F}$ is m-monotone. The above relation is true only if $\mathbf{v} = 0$ or $\mathbf{z} \neq 0$. The first condition is true if $\mathbf{x} = \mathbf{y}$, while the second condition implies that (31) is not true for $\mathbf{v} \neq 0$ or $\mathbf{x} \neq \mathbf{y}$.

We will now present a counter-example to show that the constraint is necessary. Suppose $\mathcal{F}$ is a linear non-monotone operator, denoted by a symmetric matrix $\mathbf{F}$ that has a null-space $\mathcal{N}(\mathbf{F})$ which overlaps with null-space $\mathcal{N}(\mathbf{A})$ of $\mathbf{A}$. Since $\mathbf{F}$ is not monotone, it will not satisfy $\Re(\langle \mathbf{F}\mathbf{v}, \mathbf{v} \rangle) > 0$, which implies that $\mathbf{F}$ is not positive definite. If the null-spaces of $\mathbf{A}$ and $\mathbf{F}$ overlap, we can choose a $\mathbf{v} \in \mathcal{N}(\mathbf{A}) \cap \mathcal{N}(\mathbf{F})$ such that $\langle \mathbf{z}, \mathbf{v} \rangle = 0$. This counter-example shows that there exist non-monotone operators such that the fixed points are not unique. $\square$

B.   Proof of Proposition III.3

Proof. Let the Lipschitz constant of $\mathcal{H}_\theta$ is $1 - m$:

$$\|\mathcal{H}_\theta(\mathbf{x}) - \mathcal{H}_\theta(\mathbf{y})\|_2 \leq \underbrace{(1 - m)}_{\epsilon} \|\mathbf{x} - \mathbf{y}\|_2, \quad \epsilon > 0. \tag{32}$$

Using Cauchy Schwartz, $-\|\mathbf{a}\|_2 \cdot \|\mathbf{b}\|_2 \leq \Re(\langle \mathbf{a}, \mathbf{b} \rangle)$ and (32), we have

$$-(1 - m)\|\mathbf{x} - \mathbf{y}\|_2^2 \leq \Re(\langle \mathcal{H}_\theta(\mathbf{x}) - \mathcal{H}_\theta(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle) \tag{33}$$

We consider the inner product,

$$\begin{aligned}
s &= \Re(\langle \mathcal{F}(\mathbf{x}) - \mathcal{F}(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle) \\
s &= \Re(\langle (\mathcal{I} - \mathcal{H}_\theta)(\mathbf{x}) - (\mathcal{I} - \mathcal{H}_\theta)(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle) \\
&= \|\mathbf{x} - \mathbf{y}\|_2^2 - \Re(\langle \mathcal{H}_\theta(\mathbf{x}) - \mathcal{H}_\theta(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle). \\
&\geq m \|\mathbf{x} - \mathbf{y}\|_2^2
\end{aligned} \tag{34}$$

In the second step, we used (33). The relation (34) shows that F is m-monotone. The second relation specified by (11) can be derived using the triangle equality,

$$\begin{aligned} \| \mathscr{F}(\mathbf{x}) - \mathscr{F}(\mathbf{y}) \|_2 &= \| \mathbf{x} - \mathscr{H}_\theta(\mathbf{x}) - \mathbf{y} + \mathscr{H}_\theta(\mathbf{y}) \|_2 \\ &\leq \| \mathbf{x} - \mathbf{y} \|_2 + \| \mathscr{H}_\theta(\mathbf{x}) - \mathscr{H}_\theta(\mathbf{y}) \|_2 \\ &\leq (2 - m) \| \mathbf{x} - \mathbf{y} \|_2 . \end{aligned}$$

$\square$

C. Proof of Lemma IV.1

Proof. Using $\mathscr{R} = (\mathscr{I} - \alpha\mathscr{F})$,

$$\begin{aligned} \| \mathscr{R}(\mathbf{x}) - \mathscr{R}(\mathbf{y}) \|_2^2 &= \| \mathbf{x} - \mathbf{y} - \alpha\mathscr{F}(\mathbf{x}) + \alpha\mathscr{F}(\mathbf{y}) \|_2^2 \\ &= \| \mathbf{x} - \mathbf{y} \|_2^2 + \alpha^2 \underbrace{\| \mathscr{F}(\mathbf{x}) - \mathscr{F}(\mathbf{y}) \|_2^2}_{< (2-m)^2 \| \mathbf{x} - \mathbf{y} \|_2^2} + \\ &\quad \underbrace{-2\alpha\mathfrak{R}(\langle \mathbf{x} - \mathbf{y}, \mathscr{F}(\mathbf{x}) - \mathscr{F}(\mathbf{y}) \rangle)}_{< -2\alpha m \| \mathbf{x} - \mathbf{y} \|_2^2} . \end{aligned} \tag{35}$$

$$\leq \| \mathbf{x} - \mathbf{y} \|_2^2 \sqrt{1 + \alpha^2(2 - m)^2 - 2\alpha m},$$

which shows that $L[\mathscr{R}] = \sqrt{1 + \alpha^2(2 - m)^2 - 2\alpha m}$. The first inequality in (35) follows from the Lipschitz bound for $\mathscr{F}$ in (11), while the second one is from the m-monotonicity (7) condition on $\mathscr{F}$. $\square$

D. Proof of Proposition IV.2

Proof. We first show that the operator $\mathscr{T}_{\text{MOL}}$ in the iterative relation (14):

$$\mathbf{x}_{n+1} = \mathscr{T}_{\text{MOL}}(\mathbf{x}_n) + \mathbf{z} \tag{36}$$

is a contraction. In particular, the Lipschitz constant of $\mathcal{Q}_\alpha$ in (13) is given by $L[\mathcal{Q}_\alpha] = \frac{1}{(1 + \lambda\mu_{\min})}$, where $\mu_{\min} \geq 0$ is the minimum eigenvalue of $\mathbf{A}^H\mathbf{A}$.

Under the conditions of the theorem, the Lipschitz constant $L[\mathscr{I} - \alpha\mathscr{F}]$ is less than one. Combining the two, we have $L[\mathscr{T}_{\text{MOL}}] < 1$. If $\mathbf{x}^*$ is a fixed point, we have $\mathbf{x}^* = \mathscr{T}_{\text{MOL}}(\mathbf{x}^*) + \mathbf{z}$. The result follows by the straightforward application of the Banach fixed-point theorem. $\square$

E. Proof of Proposition IV.3

Proof. Consider the iterative rule in (14),

$$\begin{aligned}
\mathbf{x}_n &= \underbrace{\mathcal{Q}_\alpha(\mathcal{I} - \alpha\mathcal{F})}_{\mathcal{T}}(\mathbf{x}_{n-1}) + \alpha\lambda\mathcal{Q}_\alpha\underbrace{\left(\mathbf{A}^H\mathbf{b}\right)}_{\mathbf{w}} \\
&= \mathcal{T}^2(\mathbf{x}_{n-2}) + (\mathcal{T} + I)\alpha\lambda\mathcal{Q}_\alpha(\mathbf{w}) \\
&= \mathcal{T}^n(\mathbf{x}_0) + \left(\mathcal{T}^{n-1} + \mathcal{T} + \ldots + I\right)\alpha\lambda\mathcal{Q}_\alpha(\mathbf{w})
\end{aligned}$$
(37)

The Lipschitz bound of $\mathcal{Q}_\alpha$ is,

$$L_{\mathcal{Q}_\alpha} = \frac{1}{1 + \lambda\mu_{\min}},$$
(38)

$$L[\mathcal{T}] = L_{\mathcal{Q}_\alpha}\sqrt{1 + \alpha^2(2 - m)^2 - 2\alpha m}.$$
(39)

Consider $\mathbf{z}_1$ and $\mathbf{z}_2$ as two input measurements with $\delta = \mathbf{z}_2 - \mathbf{z}_1$ as the perturbation in the input. Let the corresponding outputs be $\mathbf{x}_{1,n}$ and $\mathbf{x}_{2,n}$, respectively, with $\Delta_n = \mathbf{x}_{2,n} - \mathbf{x}_{1,n}$ as the perturbation in the output. Thus, the perturbation in the output can be written as

$$\begin{aligned}
\|\Delta_n\|_2 &= \|\mathcal{T}(\mathbf{x}_{1,n-1}) - \mathcal{T}(\mathbf{x}_{2,n-1}) + \alpha\lambda\mathcal{Q}_\alpha(\delta)\|_2 \\
&\leq \|\mathcal{T}(\mathbf{x}_{1,n-1}) - \mathcal{T}(\mathbf{x}_{2,n-1})\|_2 + \alpha\lambda\|\mathcal{Q}_\alpha(\delta)\|_2 \\
&\leq L[\mathcal{T}]\|\Delta_{n-1}\|_2 + \alpha\lambda L_{\mathcal{Q}_\alpha}\|\delta\|_2
\end{aligned}$$

Using (37), we can expand the above relation as

$$\begin{aligned}
\|\Delta_n\|_2 \leq\ & (L[\mathcal{T}])^n \|\Delta_0\|_2 + \\
& \alpha\lambda\left((L[\mathcal{T}])^{n-1} + (L[\mathcal{T}])^{n-2} + \ldots 1\right)L_{\mathcal{Q}_\alpha}\|\delta\|_2.
\end{aligned}$$

When $L[\mathcal{T}] < 1$, the first term vanishes, and we have

$$\lim_{n \to \infty} \|\Delta_n\|_2 \leq \frac{\alpha\lambda L_{\mathcal{Q}_\alpha}}{1 - L[\mathcal{T}]}\|\delta\|_2.$$

We thus have

$$\lim_{n \to \infty} \|\Delta_n\|_2 = \|\Delta\|_2 \leq \frac{\alpha\lambda/(1 + \lambda\mu_{\min})}{1 - \sqrt{1 + \alpha^2(2 - m)^2 - 2\alpha m}}\|\delta\|_2.$$
(40)

□

REFERENCES

[1]. Fessler JA, "Model-based image reconstruction for MRI," IEEE signal processing magazine, vol. 27, no. 4, pp. 81–89, 2010. [PubMed: 21135916]

[2]. Elbakri IA and Fessler JA, "Statistical image reconstruction for polyenergetic X-ray computed tomography," IEEE transactions on medical imaging, vol. 21, no. 2, pp. 89–99, 2002. [PubMed: 11929108]

[3]. Verhaeghe J, Van De Ville D, Khalidov I, D'Asseler Y, Lemahieu I, and Unser M, "Dynamic PET reconstruction using wavelet regularization with adapted basis functions," IEEE Transactions on Medical Imaging, vol. 27, no. 7, pp. 943–959, 2008. [PubMed: 18599400]

[4]. Aguet F, Van De Ville D, and Unser M, "Model-based 2.5-D deconvolution for extended depth of field in brightfield microscopy," IEEE Transactions on Image Processing, vol. 17, no. 7, pp. 1144–1153, 2008. [PubMed: 18586622]

[5]. Lustig M, Donoho D, and Pauly JM, "Sparse MRI: The application of compressed sensing for rapid MR imaging," Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine, vol. 58, no. 6, pp. 1182–1195, 2007.

[6]. Venkatakrishnan SV, Bouman CA, and Wohlberg B, "Plug-and-play priors for model based reconstruction," in 2013 IEEE Global Conference on Signal and Information Processing. IEEE, 2013, pp. 945–948.

[7]. Buzzard GT, Chan SH, Sreehari S, and Bouman CA, "Plug-and-play unplugged: Optimization-free reconstruction using consensus equilibrium," SIAM Journal on Imaging Sciences, vol. 11, no. 3, pp. 2001–2020, 2018.

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

[9]. Sun Y, Wohlberg B, and Kamilov US, "An online plug-and-play algorithm for regularized image reconstruction," IEEE Transactions on Computational Imaging, vol. 5, no. 3, pp. 395–408, 2019.

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

[11]. Sun Y, Wu Z, Xu X, Wohlberg B, and Kamilov US, "Scalable plug-and-play ADMM with convergence guarantees," IEEE Transactions on Computational Imaging, vol. 7, pp. 849–863, 2021.

[12]. Dabov K, Foi A, Katkovnik V, and Egiazarian K, "BM3D image denoising with shape-adaptive principal component analysis," in SPARS'09-Signal Processing with Adaptive Sparse Structured Representations, 2009.

[13]. Kamilov US, Bouman CA, Buzzard GT, and Wohlberg B, "Plug-and-Play Methods for Integrating Physical and Learned Models in Computational Imaging: Theory, algorithms, and applications," IEEE Signal Processing Magazine, vol. 40, no. 1, pp. 85–97, 2023.

[14]. Gregor K. and LeCun Y, "Learning fast approximations of sparse coding," in Proceedings of the 27th international conference on international conference on machine learning, 2010, pp. 399–406.

[15]. Hammernik K. et al. , "Learning a variational network for reconstruction of accelerated MRI data," Magnetic resonance in medicine, vol. 79, no. 6, pp. 3055–3071, 2018. [PubMed: 29115689]

[16]. Aggarwal HK, Mani MP, and Jacob M, "MoDL: Model-based deep learning architecture for inverse problems," IEEE transactions on medical imaging, vol. 38, no. 2, pp. 394–405, 2018. [PubMed: 30106719]

[17]. Xiang J, Dong Y, and Yang Y, "FISTA-net: Learning a fast iterative shrinkage thresholding network for inverse problems in imaging," IEEE Transactions on Medical Imaging, vol. 40, no. 5, pp. 1329–1339, 2021. [PubMed: 33493113]

[18]. Schlemper J, Caballero J, Hajnal JV, Price AN, and Rueckert D, "A Deep Cascade of Convolutional Neural Networks for Dynamic MR Image Reconstruction," IEEE Transactions on Medical Imaging, vol. 37, no. 2, pp. 491–503, 2018. [PubMed: 29035212]

[19]. Yang Y, Sun J, Li H, and Xu Z, "Deep ADMM-Net for compressive sensing MRI," Advances in neural information processing systems, vol. 29, 2016.

[20]. Ongie G, Jalal A, Metzler CA, Baraniuk RG, Dimakis AG, and Willett R, "Deep learning techniques for inverse problems in imaging," IEEE Journal on Selected Areas in Information Theory, vol. 1, no. 1, pp. 39–56, 2020.

[21]. Chen T, Xu B, Zhang C, and Guestrin C, "Training deep nets with sublinear memory cost," arXiv preprint arXiv:1604.06174, 2016.

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

[23]. Bai S, Kolter JZ, and Koltun V, "Deep equilibrium models," Advances in Neural Information Processing Systems, vol. 32, 2019.

[24]. Gilton D, Ongie G, and Willett R, "Deep equilibrium architectures for inverse problems in imaging," IEEE Transactions on Computational Imaging, vol. 7, pp. 1123–1133, 2021.

[25]. Antun V, Renna F, Poon C, Adcock B, and Hansen AC, "On instabilities of deep learning in image reconstruction and the potential costs of AI," Proceedings of the National Academy of Sciences, vol. 117, no. 48, pp. 30 088–30 095, 2020.

[26]. Darestani MZ, Chaudhari AS, and Heckel R, "Measuring robustness in deep learning based compressive sensing," in International Conference on Machine Learning. PMLR, 2021, pp. 2433–2444.

[27]. Genzel M, Macdonald J, and Marz M, "Solving inverse problems with deep neural networks-robustness included," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022.

[28]. Hammernik K. and Knoll F, "Machine learning for image reconstruction," in Handbook of medical image computing and computer assisted intervention. Elsevier, 2020, pp. 25–64.

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

[30]. Yang J, Zhang Y, and Yin W, "A fast alternating direction method for TVL1-L2 signal reconstruction from partial Fourier data," IEEE Journal of Selected Topics in Signal Processing, vol. 4, no. 2, pp. 288–297, 2010.

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

[32]. Beck A. and Teboulle M, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," SIAM journal on imaging sciences, vol. 2, no. 1, pp. 183–202, 2009.

[33]. Zhang J. and Ghanem B, "ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 1828–1837.

[34]. Ryu EK and Boyd S, "Primer on monotone operator methods," Appl. Comput. Math, vol. 15, no. 1, pp. 3–43, 2016.

[35]. Miyato T, Kataoka T, Koyama M, and Yoshida Y, "Spectral normalization for generative adversarial networks," arXiv preprint arXiv:1802.05957, 2018.

[36]. Bungert L, Raab R, Roith T, Schwinn L, and Tenbrinck D, "CLIP: Cheap Lipschitz training of neural networks," in International Conference on Scale Space and Variational Methods in Computer Vision. Springer, 2021, pp. 307–319.

[37]. Pesquet J-C, Repetti A, Terris M, and Wiaux Y, "Learning maximally monotone operators for image recovery," SIAM Journal on Imaging Sciences, vol. 14, no. 3, pp. 1206–1237, 2021.

[38]. Latorre F, Rolland P, and Cevher V, "Lipschitz constant estimation of neural networks via sparse polynomial optimization," arXiv preprint arXiv:2004.08688, 2020.

[39]. Jordan M. and Dimakis AG, "Exactly computing the local lipschitz constant of relu networks," Advances in Neural Information Processing Systems, vol. 33, pp. 7344–7353, 2020.

[40]. Pruessmann KP, Weiger M, Scheidegger MB, and Boesiger P, "SENSE: sensitivity encoding for fast MRI," Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine, vol. 42, no. 5, pp. 952–962, 1999.

[41]. Wang Z, Bovik AC, Sheikh HR, and Simoncelli EP, "Image quality assessment: from error visibility to structural similarity," IEEE transactions on image processing, vol. 13, no. 4, pp. 600–612, 2004. [PubMed: 15376593]

[42]. Souza R. et al. , "An open, multi-vendor, multi-field-strength brain MR dataset and analysis of publicly available skull stripping methods agreement," NeuroImage, vol. 170, pp. 482–494, 2018. [PubMed: 28807870]

[43]. Zbontar J, Knoll F, Sriram A, Murrell T, Huang Z, Muckley MJ, Defazio A, Stern R, Johnson P, Bruno M. et al. , "fastMRI: An open dataset and benchmarks for accelerated MRI," arXiv preprint arXiv:1811.08839, 2018.

[44]. Vincenti G, Monney P, Chaptinel J, Rutz T, Coppo S, Zenge MO, Schmidt M, Nadar MS, Piccini D, Che P`vre et al. , "Compressed sensing single–breath-hold CMR for fast quantification of LV function, volumes, and mass," JACC: Cardiovascular Imaging, vol. 7, no. 9, pp. 882–892, 2014. [PubMed: 25129517]

[45]. Zhao B, Haldar JP, and Liang Z-P, "PSF model-based reconstruction with sparsity constraint: Algorithm and application to real-time cardiac MRI," in 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology. IEEE, 2010, pp. 3390–3393.

[46]. Sandino CM, Lai P, Vasanawala SS, and Cheng JY, "Accelerating cardiac cine MRI using a deep learning-based ESPIRiT reconstruction," Magnetic Resonance in Medicine, vol. 85, no. 1, pp. 152–167, 2021. [PubMed: 32697891]

[47]. Zucker EJ, Sandino CM, Kino A, Lai P, and Vasanawala SS, "Free-breathing Accelerated Cardiac MRI Using Deep Learning: Validation in Children and Young Adults," Radiology, vol. 300, no. 3, pp. 539–548, 2021. [PubMed: 34128724]

[48]. Chen C. et al. , "OCMR (v1. 0)–Open-Access Multi-Coil k-Space Dataset for Cardiovascular Magnetic Resonance Imaging," arXiv preprint arXiv:2008.03410, 2020.
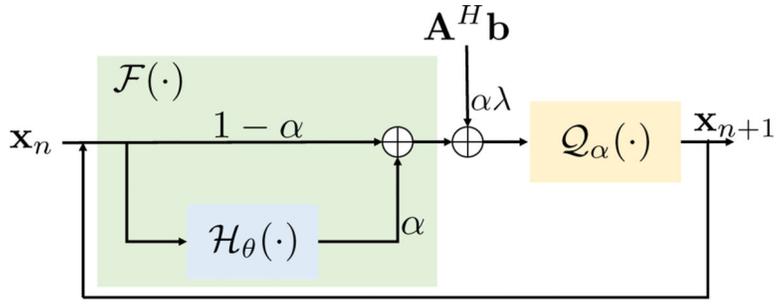
(a) Iterative Rule
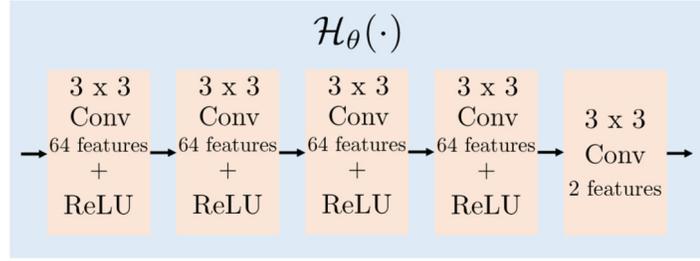
(b) CNN architecture

Fig. 1.

Figure (a) shows fixed-point iterative rule of the proposed MOL algorithm from (14) and
(b) shows the architecture of the five-layer CNN $\mathcal{H}_\theta(\cdot)$ used in the experiments. When
$\alpha = 1$, the approach reduces to the MoDL [16], which was originally introduced for unrolled
optimization, or RED [8], which was introduced for PnP models. Our analysis shows that
using $\alpha = 1$ in the DEQ setting requires highly constrained networks for convergence, which
translates to poor performance.
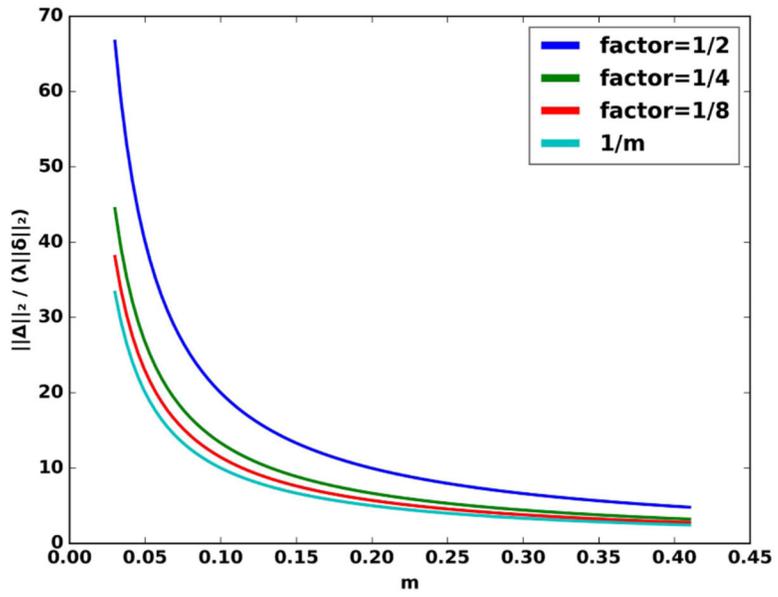
Fig. 2.

Plot of $\dfrac{\alpha}{1 - \sqrt{1 - 2\alpha m + \alpha^2 (2 - m)^2}}$ in (IV.3) vs m. Here, we choose $\alpha = \text{factor} \times \alpha_{\max}$, where $\alpha_{\max}$ is specified by (19) in the paper. We note that all the curves roughly decay with m with a $1/m$ decay rate. As $\alpha \to 0$ or equivalently low values of factor, the curves approach $1/m$.
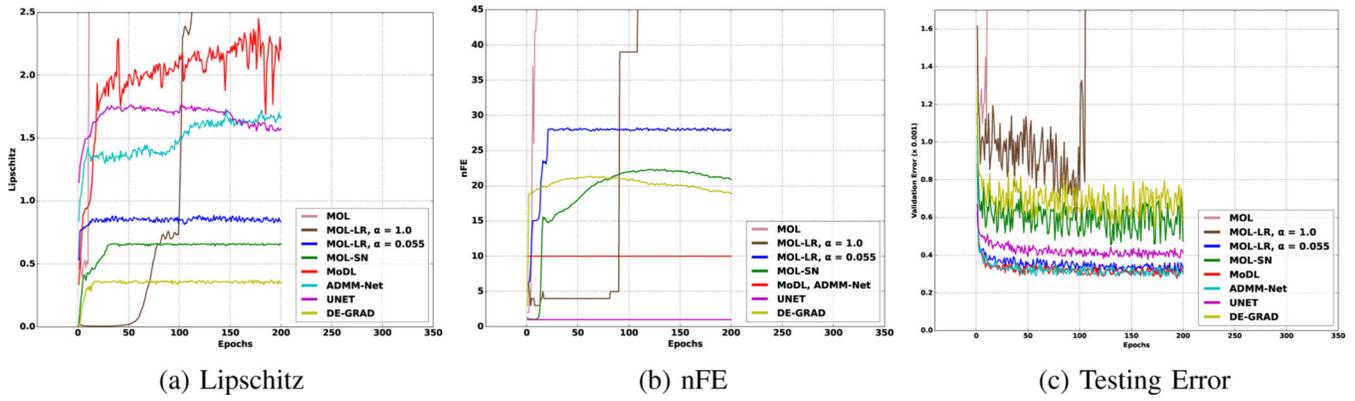
(a) Lipschitz　　　　　　(b) nFE　　　　　　(c) Testing Error

Fig. 3.

Convergence behaviour of MOL and other algorithms. MOL corresponds to the proposed algorithm without any Lipschitz constraint, MOL-LR, $\alpha = 1$ is the DEQ version of MoDL/RED [8], [16] with Lipschitz constraint, and MOL-LR, $\alpha = 0.055$ is the proposed scheme. Graphs are plotted with respect to epochs during training. (a) shows the evolution of the Lipschitz constant of the CNN module. (b) plots the number of iterations used in the algorithm. (c) plots the testing error on the validation datasets used during training. For MoDL and ADMM-Net, we have ten unrolls, and UNET consists of a single CNN. The DEQ forward and backward algorithms are run until the difference between the subsequent terms satisfies the convergence criterion from (24). We note that the MOL and MOL-LR with $\alpha = 1$ diverges as the training proceeds, as predicted by the theory. By contrast, the proposed MOL-LR scheme converges with ≈ 28 iterations or number of function evaluations (nFEs). We note that MOL-LR requires more forward iterations to converge than MOL-SN, mainly because of the higher Lipschitz constant of $\mathcal{H}_\theta$.
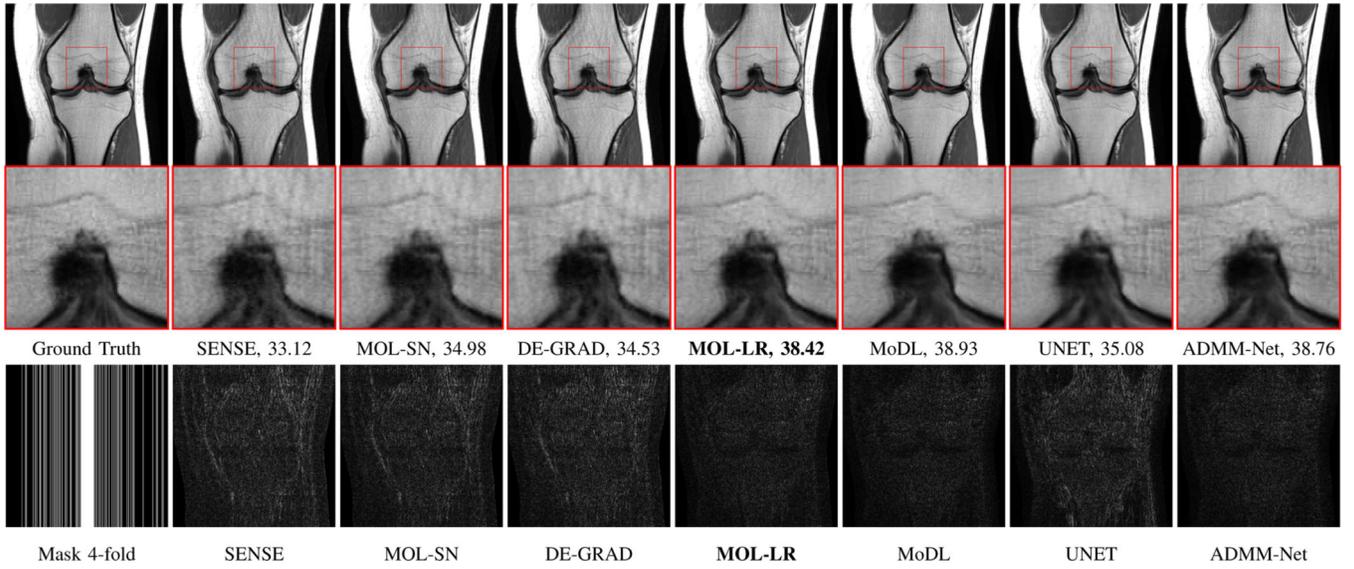
Fig. 4.
Reconstruction results of 4x accelerated multi-channel fastMRI knee data with variable density sampling. PSNR (dB) values are reported for each case. The image in the first row of the first column was undersampled along the phase-encoding direction using a Cartesian 1D non-uniform variable-density mask as shown in the second row of the first column. The top row shows reconstructions (magnitude images), while the bottom row shows corresponding error images. We note that the quality of the **MOL-LR** reconstructions is comparable to unrolled methods MoDL and ADMM-Net. MOL-SN and DE-GRAD show significantly lower performance due to spectral normalization of weights, resulting in stricter bounds on the Lipschitz constant of its CNN.
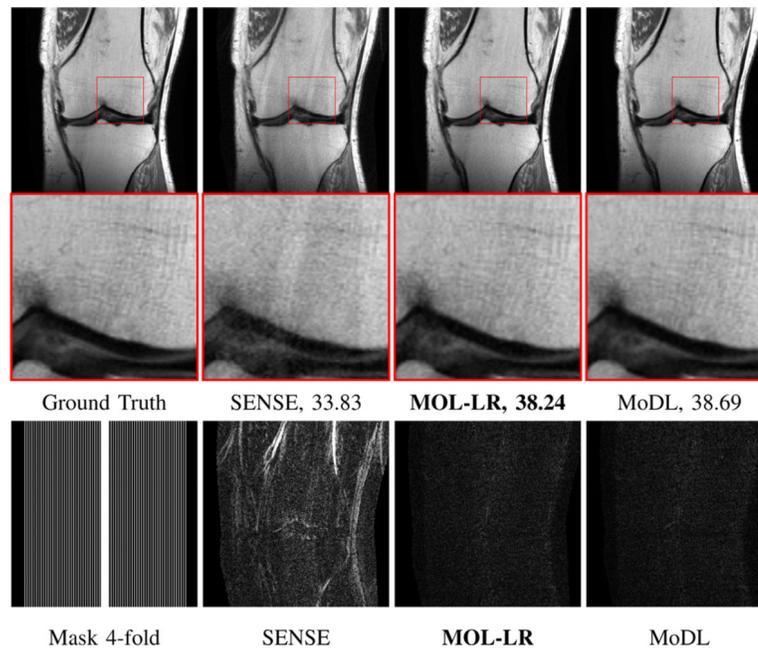
Fig. 5.

Reconstruction results of 4x accelerated knee data with uniform undersampling. PSNR (dB) values are reported for each case. The image in the first row and column was undersampled along phase encoding direction using a Cartesian 1D uniform mask as shown in the first column, second row. The top row shows reconstructions (magnitude images), while the bottom row shows corresponding error images. We note that the quality of the **MOL-LR** reconstruction is comparable to the unrolled method MoDL. SENSE reconstruction has visible aliasing and performs significantly poorer in this setting.
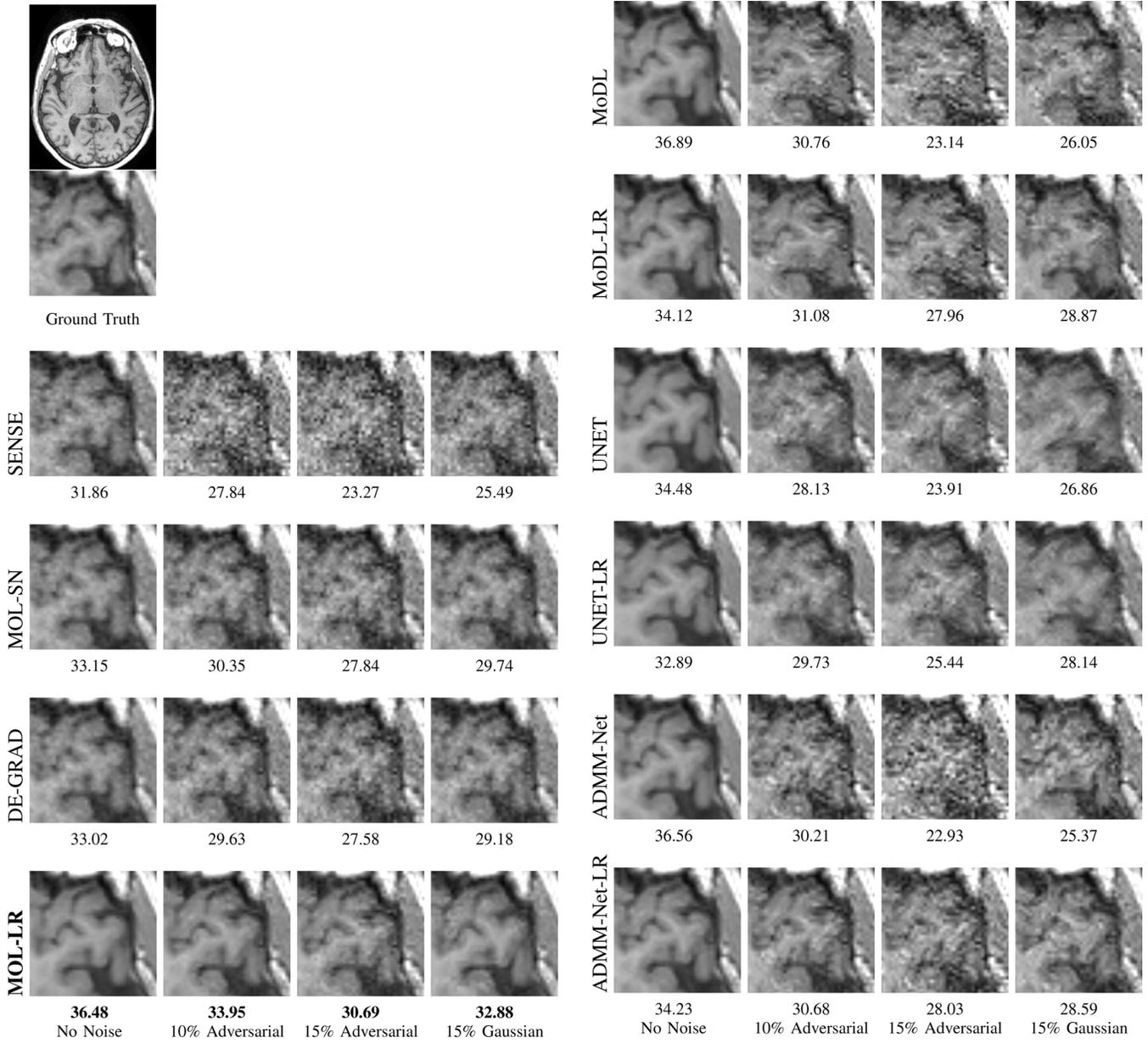
Fig. 6.

Sensitivity of the algorithms to input perturbations: The rows correspond to reconstructed images from 4x accelerated multi-channel brain data using different methods. The data was undersampled using a Cartesian 2D non-uniform variable-density mask. The columns correspond to recovery from noiseless, worst-case (adversarial) perturbation whose norm is 10% and 15% of the measured data, and Gaussian noise whose norm is also 15% of the measured data, respectively. The PSNR (dB) values of the reconstructed images are reported for each case. We observe that the performance of the MOL-LR algorithm is comparable to that of the other unrolled networks (MoDL and ADMM-Net), which is superior to that of UNET and SENSE. We note that the performance of all Lipschitz constrained methods degrade gracefully in the presence of Gaussian noise. The experiments show that

MOL-LR, which uses the Lipschitz regularization, is less sensitive to adversarial noise than are MoDL and ADMM-Net. We note that the robustness of the original MoDL and ADMM-Net implementations can be improved by Lipschitz regularization (MoDL-LR and ADMM-Net-LR), albeit with a decrease in performance in the noiseless setting. By contrast, the larger number of iterations in MOL translates to improved performance in the noiseless condition, while being robust to input perturbations.
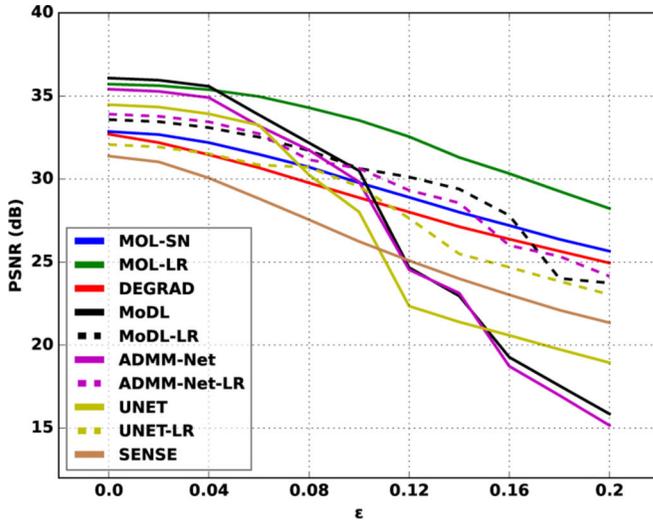
(a) Adversarial Noise
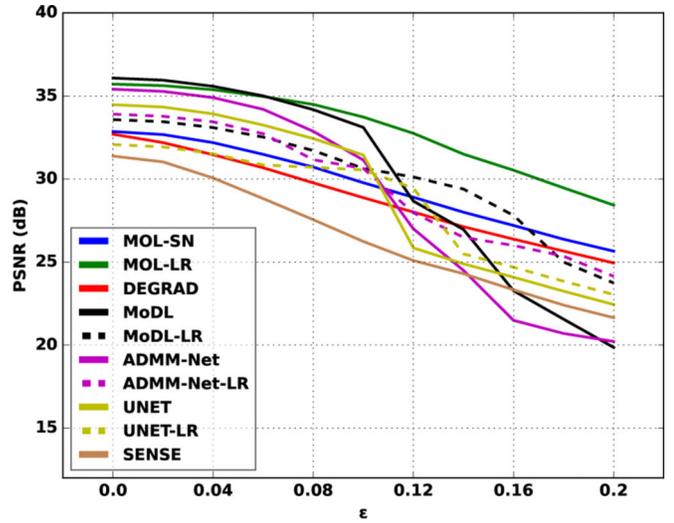
(b) Gaussian Noise

Fig. 7.

Quantitative comparison of the robustness of different algorithms to worst-case and Gaussian input perturbations. (a) shows the plot between PSNR and ϵ, which is the ratio of the norm of perturbations and the norm of the measurements, and (b) shows a similar plot for Gaussian noise. In the case of Gaussian noise in (b), all the models show an almost linear decrease in performance. By contrast, we note that the PSNR of the models trained without any Lipschitz constraints (solid curves that denote MoDL, ADMM-Net, UNET) drop significantly at around $\epsilon = 10\%$. The performance of the Lipschitz-constrained algorithms exhibit linear decay in (a), similar to the Gaussian setting in (b). However, the Lipschitz-constrained versions are associated with a decrease in performance in the noiseless setting, resulting from the constrained CNN. The proposed MOL-LR curves in the Gaussian and worst-case setting are roughly similar, while offering performance similar to the unrolled methods without Lipschitz constraint.
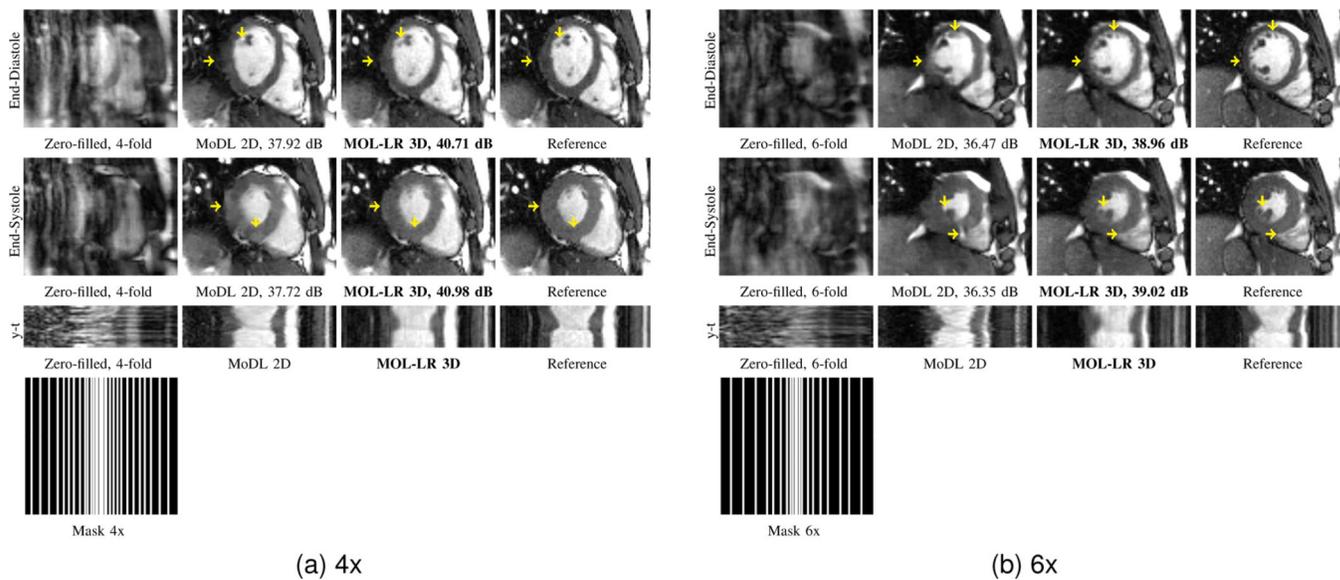
Fig. 8.

MOL recovery of 2D+time cine data at 4x and 6x accelerations. The PSNR (dB) values are reported for each case. The data is retrospectively undersampled using a Poisson density sampling pattern. For both figures (a) and (b), the top and middle rows correspond to the diastole and systole phases, respectively. For MoDL 2D and MOL-LR 3D, reconstructions (magnitude images) are shown. MOL-LR 3D reconstructs 2D+time data using 3D CNN while MoDL 2D processes each of the temproral frame independently using 2D CNN. Thus, MoDL 2D does not exploit inter-frame depedencies. MoDL 2D has more errors in the boundary of the myocardium than does MOL-LR 3D, as indicated by the yellow arrows. MoDL 2D reconstructs images in the spatial domain only, whereas MOL-LR 3D exploits redundancies in the additional temporal domain, leading to lower errors. The short axis cut looks sharper for MOL-LR 3D and shows improved preservation of wall details.

TABLE I

QUANTITATIVE COMPARISONS ON 2D DATASETS WITH 4-FOLD UNDERSAMPLING USING CARTESIAN 1D NON-UNIFORM VARIABLE-DENSITY MASK. PSNR IN dB, SSIM, AND MEAN RUN-TIME PER SLICE IN SECONDS ARE REPORTED. THE PSNR AND SSIM VALUES ARE IN MEAN ± STANDARD DEVIATION FORMAT.

	Four-fold Knee MRI		
Methods	**PSNR**	**SSIM**	**Run-time**
SENSE	33.04 ± 1.44	0.986 ± 0.023	**0.07s**
MOL-SN	34.86 ± 1.26	0.987 ± 0.019	0.32s
DE-GRAD	34.47 ± 1.39	0.987 ± 0.021	0.26s
MOL-LR	38.34 ± 0.83	**0.993 ± 0.005**	0.47s
MoDL	**38.74 ± 0.77**	**0.993 ± 0.005**	0.19s
MoDL-LR	36.53 ± 1.01	0.990 ± 0.010	0.19s
ADMM-Net	38.63 ± 0.78	**0.993 ± 0.005**	0.20s
ADMM-Net-LR	36.45 ± 1.02	0.990 ± 0.010	0.20s
UNET	35.12 ± 1.19	0.988 ± 0.013	0.08s
UNET-LR	33.76 ± 1.40	0.986 ± 0.022	0.08s

TABLE II

QUANTITATIVE COMPARISONS ON 2D DATASETS WITH 4-FOLD UNDERSAMPLING USING CARTESIAN 1D UNIFORM MASK. PSNR IN DB AND SSIM VALUES ARE REPORTED. THE FORMAT IS MEAN ± STANDARD DEVIATION.

	Four-fold Knee MRI	
Methods	**PSNR**	**SSIM**
SENSE	32.76 ± 1.56	0.985 ± 0.026
MOL-LR	38.03 ± 0.85	0.992 ± 0.007
MoDL	**38.62 ± 0.80**	**0.992 ± 0.006**

TABLE III

Quantitative comparisons on 2D+time CINE data.

2D vs 2D+time Cardiac CINE MRI Recovery				
Acceleration	4x		6x	
Methods	PSNR	SSIM	PSNR	SSIM
Zero-filled	25.25	0.764	23.64	0.712
MoDL 2D	38.21	0.975	36.94	0.959
MOL-LR 2D+t	**40.68**	**0.988**	**39.16**	**0.976**