This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCI.2021.3094714, IEEE Transactions on Computational Imaging

Model Adaptation for Inverse Problems in Imaging

Davis Gilton, Gregory Ongie, and Rebecca Willett.

***Abstract*— Deep neural networks have been applied successfully to a wide variety of inverse problems arising in computational imaging. These networks are typically trained using a forward model that describes the measurement process to be inverted, which is often incorporated directly into the network itself. However, these approaches are sensitive to changes in the forward model: if at test time the forward model varies (even slightly) from the one the network was trained for, the reconstruction performance can degrade substantially. Given a network trained to solve an initial inverse problem with a known forward model, we propose two novel procedures that adapt the network to a change in the forward model, even without full knowledge of the change. Our approaches do not require access to more labeled data (i.e., ground truth images). We show these simple model adaptation approaches achieve empirical success in a variety of inverse problems, including deblurring, super-resolution, and undersampled image reconstruction in magnetic resonance imaging.**

I. Introduction

Repeated studies have illustrated that neural networks can be trained to solve inverse problems in imaging, including problems such as image reconstruction in MRI, inpainting, superresolution, deblurring, and more. Recent reviews and tutorials on this topic [4], [27] have described various approaches to this problem. For concreteness, we focus on the case of *linear* inverse problems in imaging. In the general framework of interest, an unknown n-pixel image (in vectorized form) $x \in \mathbb{R}^n$ (or $\mathbb{C}^n$) is observed via m noisy linear measurements $y \in \mathbb{R}^m$ (or $\mathbb{C}^m$) according to the model

$$y = A_0 x + \varepsilon, \tag{1}$$

where the matrix $A_0 \in \mathbb{R}^{m \times n}$ (or $\mathbb{C}^{m \times n}$) is the *forward model* and ε represents a vector of noise. The goal is to recover x from y.

In this paper, we focus on the setting in which the forward model A_0 is known and used during training. Past work has illustrated that leveraging knowledge of A_0 during training can reduce the sample complexity [12]. This paradigm is particularly common in applications such as medical imaging, where A_0 represents a model of the imaging system. For instance, in magnetic resonance imaging (MRI), A_0 reflects which k-space measurements are collected.

Unfortunately, these methods can be surprisingly fragile in the face of *model drift*, which occurs when, at test time, we are provided samples of the form

$$y = A_1 x + \varepsilon' \tag{2}$$

for some new forward model $A_1 \neq A_0$ and/or a change in the noise distribution (*i.e.*, the noise ε' is distributed differently than ε). That is, assume we have trained a solver that is a function of both the original forward model A_0 and a learned neural network. One might try to reconstruct x from y using this solver, but it will perform poorly because it is using a misspecified model (A_0 instead of A_1). Alternatively, we might attempt to use the same general solver where we replace A_0 with A_1 but leave the learned component intact. In this case, the estimate x computed from y may also be poor, as illustrated in [3] and [17]. The situation is complicated even further if we do not have a precise model of A_1 at test time.

These are real challenges in practice. For example, in MRI reconstruction there is substantial variation in the forward model depending on the type of acquisition – e.g., Cartesian versus non-Cartesian k-space sampling trajectories, different undersampling factors, different number of coils and coil sensitivity maps, magnetic field inhomogeneity maps, and other calibration parameters [10] – all which need to be accounted for during training and testing. A network trained for one of these forward models may need to be retrained from scratch in order to perform well on even a slightly different setting (e.g., from six-fold to four-fold undersampling of k-space). Furthermore, training a new network from scratch may not always be feasible after deployment due to a lack of access to ground truth images. This could be either due to privacy concerns of sharing patient data between hospitals and researchers, or because acquiring ground truth images is difficult for the new inverse problem.

This leads us to formulate the problem of *model adaptation*: given a reconstruction network trained on measurements from one forward model adapt/retrain/modify the network to reconstruct images from measurements reflecting a new forward model. We consider a few variants of this problem: (a) the new forward model A_1 is known, along with one or more unlabeled training samples y_i reflecting A_1, and (b) A_1 is unknown or only partially known, and we only have one or

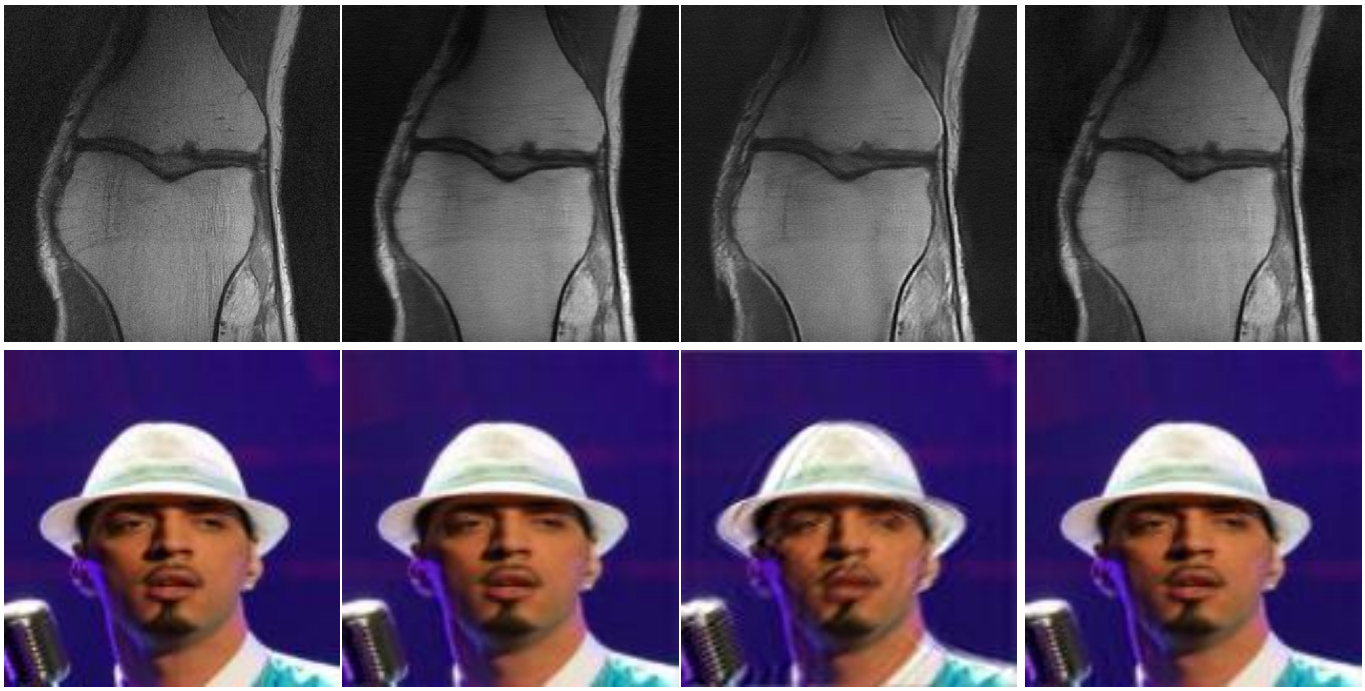
The authors gratefully acknowledge funding from NSF Awards DMS-1925101, DMS-2023109, and OAC-1934637 and AFOSR FA9550-18-1-0166.

D. Gilton is with the Department of Electrical and Computer Engineering at the University of Wisconsin-Madison, 1415 Engineering Dr, Madison, WI 53706 USA.

G. Ongie is with the Department of Mathematical and Statistical Sciences at Marquette University, 1250 W Wisconsin Ave, Milwaukee, WI 53233 USA.

R. Willett is with the Departments of Computer Science and Statistics at the University of Chicago, 5747 S Ellis Ave, Chicago, IL 60637 USA.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCI.2021.3094714, IEEE Transactions on Computational Imaging



(a) Ground truth (b) No model drift (c) Model drift (d) Model drift w/model adaptation

Fig. 1: Small perturbations in measurements for deep learning-based image reconstruction operators can lead to both subtle and obvious artifacts in reconstructions across problems and domains. In the top row, we present results for undersampled MRI reconstruction of knee images, and the second row illustrates deblurring images of human faces. (a) Ground truth image. (b) No model drift. Training and test data correspond to same model, A_0, yielding accurate reconstruction via learned model. (c) Model drift but no model adaptation. Training assumes model A_0 but at test time we have model A_1. Reconstruction using trained network *without model adaptation* gives significant distortions. (d) Model drift and model adaptation. Training assumes model A_0 but at test time we have model A_1. Reconstruction using model adaptation prevents distortions and compares favorably to the setting without model drift. The MRI example demonstrates our Reuse and Regularize method (Alg. 2), and the deblurring example demonstrates our Parameterize and Perturb method (Alg. 1). Experimental details are in Section IV.

more unlabeled training samples reflecting A_1. These training samples are unlabeled in the sense that they are not paired with "ground truth" images used to generate the y_i's. Our proposed model adaptation methods allow a reconstruction network to be trained for a known forward model and then adapted to a related forward model without access to ground truth images, and without knowing the exact parameters of the new forward model.

Model drift as stated above is a particular form of *distribution drift*, in which the distribution of $Y|X = x$ changes between training and deployment and we know Y has a linear dependence on X before and after the drift (even if we do not know the parameters of those linear relationships, represented as A_0 and A_1). That is, if we assume a noise model $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$, then the training distribution is $Y|X = x \sim \mathcal{N}(A_0 x, \sigma^2 I)$ and the distribution at deployment (assuming the same noise model) is $Y|X = x \sim \mathcal{N}(A_1 x, \sigma^2 I)$. In general, distribution drift challenges may be addressed using transfer learning [28], [37], [41] and domain adaptation [20], [26], [39]. One of the methods we explore in the body of the paper, Parameterize and Perturb, shares several features with transfer learning methodology. However, since in our setting we have a specific form of distribution drift, it is possible to design more targeted methods with better performance, as illustrated by existing specialized methods for image inpainting [9], as well as our general-purpose Reuse and Regularize method (detailed below).

A. Related Work

A broad collection of recent works, as surveyed by [4] and [27], have explored using machine learning methods to help solve inverse problems in imaging. The current paper is motivated in part by experiments presented in [3], which show that deep neural networks trained to solve inverse problems are prone to several types of instabilities. Specifically, they showed that model drift in the form of slight changes in the forward model (even "beneficial" ones, like increasing the number of k-space samples in MRI) often have detrimental impacts on reconstruction performance. While [3] is mostly empirical in nature, a follow-up mathematical study [13] provides theoretical support to this finding, implying that instability arises naturally from training standard deep learning inverse solvers. However, recent work also shows that the instabilities observed in in [3] can be mitigated to some extent by adding noise to measurements during training, though such techniques are not sufficient to resolve artifacts arising from substantial model drift.

To address a subset of these issues, [30] and [23] propose adversarial training frameworks that increases the robustness

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCI.2021.3094714, IEEE Transactions on Computational Imaging

of inverse problem solvers. However, [30] and [23] focus on robustness to adversarial perturbations in the measurements for a fixed forward model, and do not address a global change in the forward model, which is the focus of this work.

Similar to this work, a recent paper [15] has proposed domain adaptation techniques to transfer a reconstruction network from one inverse problem setting to another, e.g., adapting a network trained for CT reconstruction to perform MRI reconstruction. However, the focus of that approach is on adapting to *changes in the image distribution*, whereas our approaches focus on *changes to the forward model* assuming the image distribution is unchanged. Additionally, to our knowledge, no existing domain adaptation approaches consider the scenario where the new forward model depends on unknown calibration parameters, as we do in this work.

Another line of work explores learned methods for image reconstruction with automatic parameter tuning; see [40] and references therein. However, this work focuses on learning regularization and optimization parameters, not parameters of a drifting forward model. Also, [42] describes a unrolling approach to learning a forward model in an imaging context, but with the goal of designing a forward model that optimizes reconstruction quality, rather than estimating a correction to the forward model from measurements. Some recent studies have used pre-trained generative models to solve inverse problems with unknown calibration parameters [2]; this line of work can be viewed as an extension of compressed sensing with general models framework introduced in [8].

II. Problem Formulation

Here we formalize the problem of *model adaptation* as introduced above.

Suppose we have access to an estimator $\widehat{x} = f_0(y)$ that has been designed/trained to solve the inverse problem

$$y = A_0 x + \varepsilon, \quad x \sim P_X,\ \varepsilon \sim P_{N_0} \tag{P0}$$

where A_0 is a known (linear) forward model, P_X denotes the distribution of images x and P_{N_0} denotes the distribution of the noise ε. We assume the trained estimator "solves" the inverse problem in the sense that it produces an estimate $\hat{x} = f_0(y)$ such that the mean-squared error (MSE) $\mathbb{E}_{x,\varepsilon}[\|\hat{x} - x\|^2]$ is small.

Now assume that the forward model has changed from A_0 to a new model A_1 and/or the noise distribution has changed from P_{N_0} to a new noise distribution P_{N_1}, resulting in the new inverse problem

$$y = A_1 x + \varepsilon', \quad x \sim P_X,\ \varepsilon' \sim P_{N_1}. \tag{P1}$$

We consider both the case where A_1 is *known* (*i.e.*, we have an accurate estimate of A_1) and the case where A_1 is partially unknown, in the sense that it belongs to a class of parameterized forward models, *i.e.*, $A_1 \in \{A(\sigma) : \sigma \in \mathbb{R}^q\}$, where the parameters $\sigma \in \mathbb{R}^q$ are unknown.

The goal of *model adaptation* is to adapt/retrain/modify the estimator $\widehat{x} = f_0(y)$ that was designed to solve the original inverse problem (P0) to solve the new inverse problem (P1). We will consider two variants of this problem:

- *Model adaptation without calibration data:* In this setting, we assume access to only one measurement vector y generated according to (P1).
- *Model adaptation with calibration data:* In this setting we assume access to a new set of measurement vectors $\{y_i\}_{i=1}^N$ generated according to (P1), but without access to the paired ground truth images (i.e., the corresponding x_i's).

While the above discussion centers around a general estimator $\widehat{x} = f_0(y)$, we are particularly interested in estimators that combine a trained trained deep neural network component depending on a vector of weights/parameters θ_0, along with the original forward model A_0; we will call such an estimator a *reconstruction network*. Specifically, we assume that the forward model A_0 (or other derived quantities, such as its transpose $A_0^\top$, psuedo-inverse $A_0^\dagger$, etc.) is embedded in the reconstruction network, either in an initialization layer and/or in multiple subsequent layers. This is the case for networks based on unrolling of iterative algorithms (see, for example, [4], [14], [24], [27], [36], and references therein), in which A_0 appears repeatedly in the network in "data-consistency" layers that approximately re-project the intermediate outputs of the network onto the set of data constraints $\{x \in \mathbb{R}^n : A_0 x = y\}$. In general, we will assume the reconstruction network can be parametrized as $f_0(\cdot) = f(\cdot; \theta_0, A_0)$ where $\theta_0 \in \mathbb{R}^p$ is the vector of pre-trained neural network weights/parameters and A_0 is the original forward model.

Finally, to simplify the presentation, we will assume an additive white Gaussian noise model for both (P0) and (P1), *i.e.*, $P_{N_0} = \mathcal{N}(0, \sigma_0^2 I)$ and $P_{N_1} = \mathcal{N}(0, \sigma_1^2 I)$ with known variances σ_0^2 and σ_1^2. In this case the negative log-likelihood of x given y under measurement model (P1) is $\frac{1}{2\sigma_1^2}\|A_1 x - y\|_2^2$, which justifies our use of quadratic data-consistency terms in the development below.

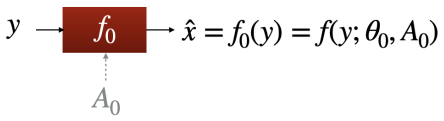
A. The feasibility of model adaptation

To compute an accurate reconstruction under the original forward model, A_0, the learned solver must reconstruct components of the image that lie in the null space $N(A_0)$: for superresolution, these are high-frequency details lost during downsampling, and in inpainting, these are the pixels removed by A_0.

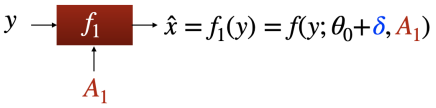
Reconstructing under a different forward model, A_1, requires reconstructing different components of the image in the null space $N(A_1)$. The general intuition behind model adaptation is that if A_0 and A_1 are similar, then the mapping represented by f_0 can inform the new mapping that we need to learn from image components in the range of A_1 to components in $N(A_1)$. For example, in an inpainting setting, the learned network not only represents the missing pixels, but it also represents some function of the observed pixels that are relevant to filling in the missing pixels. Thus if A_1 has a similar null space (*e.g.*, an offset in the collection of missing pixels), it is reasonable to expect that the original network has learned to represent some information about image components in the null space of A_1 but not in the null space of A_0. As the null spaces of A_0 and A_1 get further apart, model adaptation becomes less effective.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCI.2021.3094714, IEEE Transactions on Computational Imaging

(a) Original reconstruction network.



(b) Model adaptation by Parametrize and Perturb (P&P).



(c) Model adaptation by Reuse & Regularize (R&R).

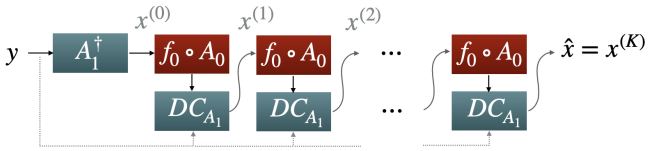


Fig. 2: Three basic paradigms of reconstruction under "model drift". (a) If the training data is generated using the model $y = A_0 x + \varepsilon$, this can be used to learn a reconstruction network $f(y; \theta_0, A_0)$ which is parameterized by weights or parameters θ_0 and may also explicitly depend on forward model A_0. (b) **Parametrize and Perturb (P&P):** If at test time we are presented with data corresponding to the model $y = A_1 x + \varepsilon'$, we may not only use the new forward model A_1 but also learn a perturbation δ to the original network parameters θ_0 to compensate for the model drift. (c) **Reuse and Regularize (R&R):** Alternatively to P&P, we may reuse the pre-trained network f_0 as an implicit regularizer in an iterative model-based reconstruction scheme. The proposed scheme alternates between applying $f_0 \circ A_0$, which denoises and/or removes artifacts, and a data-consistency step (denoted by DC_{A_1} above) that enforces the estimated image $\widehat{x}$ satisfies $A_1 \widehat{x} \approx y$.

This is similar to the widely-noted behavior of transfer learning, where transfer learning efficacy depends on the similarity of the training and target distributions. This intuition is supported by our empirical results, which illustrate that when A_0 and A_1 correspond to different blur kernels or perturbed k-space sampling patterns in MRI, the learned mapping f_0 does contain information about image components in the null space of A_1 that can be leveraged to improve reconstruction accuracy, even without additional training samples drawn using the model A_1.

III. Proposed Approaches

We propose two distinct model adaptation approaches, *Parameterize & Perturb (P&P)* and *Reuse & Regularize (R&R)*, as detailed below.

A. Parametrize and Perturb: A transfer learning approach

Let f_0 be a reconstruction network trained to solve inverse problem (P0). Suppose we can explicitly parameterize f_0 both in terms of the trained weights/parameters θ_0 and the original forward model A_0, *i.e.*, we may write $f_0(\cdot) = f(\cdot; \theta, A_0)$. Given a new measurement vector y under the measurement model (P1), a "naive" approach to model adaptation is to simply to replace substitute the new forward model A_1 for A_0 in this parametrization, and estimate the image as $f(y; \theta_0, A_1)$. However, as illustrated in Figure 1, this can lead to artifacts in the reconstruction due to model mismatch.

Instead, we propose estimating the image as $f(y; \theta_1, A_1)$ where θ_1 is a perturbed set of of network parameters obtained by solving the optimization problem:

$$\min_{\theta} \|y - A_1 f(y; \theta, A_1)\|_2^2 + \mu \|\theta - \theta_0\|_2^2. \tag{3}$$

where $\mu > 0$ is a tunable parameter. The first term enforces data consistency, i.e., the estimated image $\widehat{x}$ should satisfy $y \approx A_1 \widehat{x}$, while the second term $\|\theta - \theta_0\|_2^2$ ensures the retrained parameters θ stay close to the original network parameters θ_0. This term is necessary to avoid degenerate solutions, which we demonstrate in the Supplement. Our use of a proximity term of this form is also inspired in part by its success in other transfer learning applications (see, *e.g.*, [43]).

If the forward model A_1 is also unknown, we propose optimizing for it as well in the above formulation, which gives:

$$\min_{\theta, A \in \mathcal{A}} \|y - A f(y; \theta, A)\|_2^2 + \mu \|\theta - \theta_0\|_2^2. \tag{4}$$

where $\mathcal{A}$ denotes a constraint set. We assume the forward model is parameterized such that the constraint set is given by $\mathcal{A} = \{A(\sigma) : \sigma \in \mathbb{R}^q\}$, where $A(\sigma)$ denotes a class of of forward models parametrized by a vector $\sigma \in \mathbb{R}^q$ with $q \ll m \cdot n$ (e.g., in a blind deconvolution setting, $A(\sigma)$ corresponds convolution with an unknown kernel σ). In particular, we propose optimizing over the parameters σ, which is possible with first-order methods such as stochastic gradient descent, provided the map $\sigma \mapsto A_\sigma$ is first-order differentiable.

Algorithm 1 Parameterize & Perturb (P&P)

Require: Original forward model A_0, new forward model A_1, pre-trained reconstruction network $f_0(\cdot) = f(\cdot; \theta_0, A_0)$, regularization parameter μ, new measurements y.

1: Modify the reconstruction network f_0 by internally changing A_0 to A_1, to obtain the estimate $f(y; \theta_0, A_1)$

2: Fine-tune the network weights as $\theta_1 = \theta_0 + \delta$ where δ is a perturbation learned by solving (3)

The preceding discussion focused on the case of reconstructing single measurement vector y at test time, *i.e.*, model adaptation without calibration data. Additionally, we consider a P&P approach in the case where we have access to calibration data $y_1, ..., y_N$ generated according to (P1). In this case we propose retraining the network by minimizing the sum of data-consistency terms over the calibration set:

$$(\theta_1, \tilde{A}_1) = \arg\min_{\theta, A \in \mathcal{A}} \frac{1}{N} \sum_{i=1}^{N} \|y_i - A f(y_i; \theta, A)\|_2^2 + \lambda \|\theta - \theta_0\|_2^2. \tag{5}$$

At deployment, we propose using the retrained network $\widehat{x} = f(y; \theta_1, \tilde{A}_1)$ as our estimator.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCI.2021.3094714, IEEE Transactions on Computational Imaging

It is worth noting that the P&P model adaptation technique presented above bears similarities to the deep image prior (DIP) approach to solving inverse problems as introduced in [38]. However, P&P differs from DIP in two key aspects: First, in DIP the reconstruction network is initialized with random weights, whereas in P&P we start with a network whose initial weights θ_0 are obtained by training to solve the initial inverse problem (P0). Second, we explicitly enforce proximity to the initial weights to prevent overfitting to the data, and do not rely on early stopping heuristics as is the case DIP. The P&P approach also shares similarities to the "fine-tuning" step proposed in the Σ-net MRI reconstruction framework [19], where a loss similar to (3) is minimized to enforce data consistency at test time. However, different from P&P, the fine-tuning approach in [19] regularizes the reconstruction by minimizing the loss between initial reconstruction and the new network output in the SSIM metric. As demonstrated in Figure 1, this initial reconstruction can have severe artifacts in certain settings due to model mismatch, in which case enforcing proximity in image space to an initial reconstruction is less justified.

B. Reuse & Regularize: Model adaptation without retraining

One drawback of the P&P approach is that it requires fine-tuning the network for each input y, which is computationally expensive relative to a feed-forward reconstruction approach. Additionally, the P&P approach is somewhat indirect, relying only on the inductive bias of the network architecture and its original parameter configuration to impart a regularization effect for the new inverse problem (P1). Here we propose a different model adaptation approach that does not require retraining the original reconstruction network, and explicitly makes use of the fact that the original network is designed to solve (P0).

Suppose we are given a reconstruction network $f_0(y)$ trained to solve (P0). The key idea we exploit is that the composition of f_0 with the original forward model A_0, should act as an *auto-encoder*, *i.e.*, if we define the map $g : \mathbb{R}^d \to \mathbb{R}^d$ by $g(x) = f_0(A_0 x)$ then by design we should have $g(x) \approx x$ for any image x sampled from the image distribution P_X. See Figure 3 for an illustration in the case of undersampled MRI reconstruction.

Given this fact, one simple approach to reconstructing a measurement vector y under (P1) is to start from an initial guess, *e.g.*, the least squares solution $x^{(0)} = A_1^\dagger y$, and attempt to find a fixed-point of $g(\cdot)$ by iterating:

$$x^{(k+1)} = g(x^{(k)}), \quad k = 0, 1, 2, ... \tag{6}$$

However, this approach only uses knowledge of the new forward model A_1 in the initialization step. Also, unless we can guarantee the map $g(\cdot)$ is non-expansive (i.e., its Jacobian is 1-Lipschitz), these iterations could diverge.

Instead, building off the intuition that g acts as an auto-encoder, we propose using g as a regularizer in an iterative model-based reconstruction scheme. In particular, we adopt a regularization-by-denoising (RED) approach, which allows

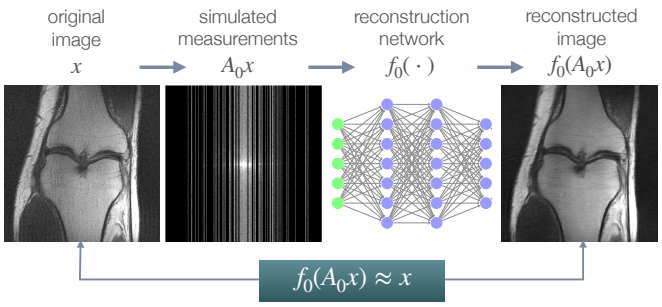


Fig. 3: Illustration of the auto-encoding property of the map $f_0 \circ A_0$ as used in the proposed R&R model adaptation approach, illustrated in an undersampled MRI reconstruction setting.

one to convert an arbitrary denoiser/de-artifacting map into a regularizer [32]. The RED approach is motivated by the following cost function:

$$\min_x \frac{1}{2}\|A_1 x - y\|_2^2 + \lambda \rho(x) \tag{7}$$

where the function $\rho(x) := x^\top (x - g(x))$ can be interpreted as a regularizer induced by the map $g(x)$ and $\lambda > 0$ is a regularization parameter. Under appropriate conditions on the function g, one can show $\nabla \rho(x) = x - g(x)$. This fact is used in [32] to derive a variety of iterative algorithms based on first-order methods (see also [31] for further analysis of RED, including convergence guarantees).

For simplicity, we focus on a RED approach with proximal gradient descent (see, e.g., [29]) as the base algorithm with stepsize $\tau > 0$. This results in an alternating scheme:

$$z^{(k)} = (1 - \tau)x^{(k)} + \tau g(x^{(k)})$$
$$x^{(k+1)} = \arg\min_x \frac{1}{2\lambda}\|A_1 x - y\|_2^2 + \frac{1}{2\tau}\|x - z^{(k)}\|_2^2$$

The x-update above has the closed-form expression

$$x^{(k+1)} = \left(A_1^\top A_1 + \tfrac{\lambda}{\tau} I\right)^{-1} \left(A_1^\top y + \tfrac{\lambda}{\tau} z^{(k)}\right) \tag{8}$$

For simplicity of implementation and to reduce the number of tuning parameters, we fix the stepsize to $\tau = 1$ in all our experiments. We summarize these steps in Algorithm 2.

Algorithm 2 Reuse & Regularize (R&R)

Require: Pre-trained reconstruction network $f_0(\cdot)$, original forward model A_0, new forward model A_1, regularization parameter $\lambda > 0$, max iterations K, new measurements y.

$x \leftarrow A_1^\dagger y$ ▷ *least-squares initialization*
for $k = 1, 2, ..., K$ **do**
  $z \leftarrow f_0(A_0 x)$ ▷ *regularize using pre-trained network*
  $x \leftarrow (A_1^\top A_1 + \lambda I)^{-1}(A_1^\top y + \lambda z)$ ▷ *data consistency*
end for
return x

Note that in the limit as $\lambda \to \infty$, Algorithm 2 reduces to the fixed-point scheme (6), and in the limit as $\lambda \to 0$ Algorithm 2 will return the initialization $x = A_1^\dagger y$. In general, the output from Algorithm 2 will interpolate between these

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCI.2021.3094714, IEEE Transactions on Computational Imaging

two extremes: x will be an approximate fixed point of g and will approximately satisfy data consistency, *i.e.*, $y \approx A_1 x$.

For certain types of forward models the x-update in (8) can be computed efficiently (e.g., if A_1 corresponds to a 2-D discrete convolution with circular boundary conditions, then $A_1^\top A_1$ diagonalizes under the 2-D discrete Fourier transform). However, in general, the matrix inverse $(A_1^\top A_1 + \lambda I)^{-1}$ may be expensive. Therefore, in practice we propose approximating (8) with a fixed number of conjugate gradient iterations.

A notable aspect of the R&R approach is that it has potential to improve the accuracy of network-based reconstructions *even in the absence of model drift*, *i.e.*, even if $A_1 = A_0$. This is because data-consistency is not guaranteed by certain reconstruction networks (e.g., U-Nets). However, we are less likely to see a benefit in the case where the reconstruction network already incorporates data-consistency layers, such as networks inspired by unrolling iterative optimization algorithms. We explore this aspect empirically in Section IV-E in the context of MRI reconstruction.

The R&R approach can also be extended the case where the new forward model A_1 depends on unknown parameters. First, we define an estimator $\hat{x}(y; A_1)$ by unrolling of a fixed number of iterates of Algorithm 2, *i.e.*, we take $\hat{x}(y; A_1) = x^{(K)}$ where $x^{(K)}$ is the Kth iterate of Algorithm 2 with input y for some small fixed value of K (e.g., $K = 5$). Supposing A_1 belongs to a parameterized class of forward models $\mathcal{A} = \{A(\sigma) : \sigma \in \mathbb{R}^q\}$, we propose optimizing a data-fit term over A:

$$\tilde{A}_1 = \arg\min_{A \in \mathcal{A}} \|A\hat{x}(y; A) - y\|_2^2. \tag{9}$$

The resulting image estimate is then taken to be $\hat{x}(y; \tilde{A}_1)$.

Finally, we also consider a combination of the P&P and R&R approaches where we additionally fine-tune the weights θ_0 of the reconstruction network f_0 embedded in the unrolled R&R estimator. We call this approach R&R+. Writing the R&R+ estimator as $\widehat{x}_{R\&R+}(y; \theta, A_1)$, similar to the P&P approach we propose "fine-tuning" the weights θ_0 by approximately minimizing the cost function, with $\mu > 0$ a tunable parameter,

$$\min_\theta \|A_1 \widehat{x}_{R\&R+}(y; \theta, A_1) - y\|_2^2 + \mu||\theta_1 - \theta||_2^2, \tag{10}$$

to obtain the updated network parameters $\theta_1 = \theta + \delta$ where δ is some small perturbation. The estimated image is then given by $x = \widehat{x}_{R\&R+}(y; \theta, A_1)$.

Empirically we see consistent improvement in reconstruction accuracy from R&R+ over R&R without any fine-tuning (see Figures 7 and 8). However, this comes at the additional computational cost of having to retrain the reconstruction network parameters at test time.

If we have access to calibration data $y_1, ..., y_N$ generated according to (P1) we can train R&R by minimizing the sum of data-consistency terms over the calibration set:

$$(\theta_1, \tilde{A}_1) = \arg\min_{\theta, A \in \mathcal{A}} \frac{1}{N} \sum_{i=1}^{N} \|y_i - A\widehat{x}_{R\&R+}(y_i; \theta, A)\|_2^2 + \mu||\theta - \theta_0||_2^2. \tag{11}$$

At deployment, we propose using Algorithm 2 with the retrained $f(y; \theta_1, \tilde{A}_1)$ in place of f_0.

Algorithm 3 Reuse & Regularize with fine-tuning (R&R+)

Require: Pretrained reconstruction network $f_0(\cdot) = f(\cdot; \theta_0)$, original forward model A_0, new forward model A_1, regularization parameter λ, new measurements y.

1: Construct an estimator $\widehat{x}(y; \theta_0)$ by unrolling K iterations of Algorithm 2

2: Fine-tune the network weights as $\theta_1 = \theta_0 + \delta$ where δ is a perturbation learned by approximately minimizing the cost (10) via SGD.

3: **return** $x = \widehat{x}(y; \theta_1)$

IV. Experiments

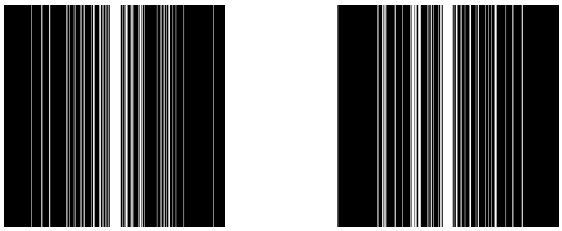
In this section we empirically demonstrate our approach to model adaptation on three types of inverse problems with two example reconstruction network architectures. We have chosen these comparison points for their simplicity and to illustrate the broad applicability of our proposed approaches. In particular, our approaches to model adaptation are not tied to a specific architectural design.

A. Methods and datasets used

We demonstrate our approaches on three inverse problems: motion deblurring, superresolution, and undersampled single-coil MRI reconstruction.

For motion deblurring, our initial model A_0 corresponds to a 10° motion blur with a 7×7 kernel, and A_1 is a 20° motion blur with a 7×7 kernel, with angle given with respect to the horizontal axis. In superresolution, our initial model is a bilinear downsampling with rate $2\times$, and A_1 corresponds to $2\times$ bicubic downsampling.

MRI reconstruction is performed with a $6\times$ undersampling of k-space in the phase encoding direction for both A_0 and A_1. The sampling maps are shown in Fig 4.



(a) Original k-space sampling pattern (A_0)

(b) Resampled k-space sampling pattern (A_1)

Fig. 4: Visualization of k-space masks used for MRI experiments. Each mask represents a 6-fold Cartesian undersampling with 4% of the center k-space lines fully sampled, and the remaining lines sampled according to a Gaussian variable density scheme. The A_1 mask contains the same center lines, but the higher frequency k-space lines are sampled separately.

We use two datasets in our experiments. First, for motion deblurring and superresolution, we train and test on 128x128-pixel aligned photos of human faces from the CelebA dataset [21].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCI.2021.3094714, IEEE Transactions on Computational Imaging

The data used in the undersampled MRI experiments were obtained from the NYU fastMRI Initiative [45]. The primary goal of the fastMRI dataset is to test whether machine learning can aid in the reconstruction of medical images. We trained and tested on a subset of the single-coil knee dataset, which consist of simulated single-coil measurements. In all tests, we use complex-valued data, which interfaces with our deep networks by treating the real and imaginary parts of the images as separate channels. We measure reconstruction accuracy with respect to the center 320×320 pixels of the complex IFFT of the fully-sampled k-space data. For the purpose of visualization, we display only the magnitude images in the following sections.

Learning rates and regularization parameters (*i.e.*, μ in Algorithm 1 and λ in Algorithm 2) were tuned via cross-validation on a hold out validation set of 512 images for CelebA, and 64 MR images for fastMRI. Batch sizes were fixed in advance to be 128 for the motion blur and superresolution settings, and 8 for the MRI setting. Hyperparameters were tuned via grid search on a log scale. For R&R, we use $K = 5$ iterations in the main loop of Algorithm 2. During training, we add Gaussian noise with $\sigma = 0.01$ to all measurements, as suggested by [11] to improve robustness.

We compare the performance of two reconstruction network architectures across all datasets. First, we utilize the U-Net architecture [33]. Our U-Net implementation takes as input the adjoint of the measurements under the forward model $A_0^\top y$ or $A_1^\top y$, which is then passed through several CNN layers before obtaining a reconstructed image $\widehat{x}$.

We also utilize the MoDL architecture [1], a learned architecture designed for solving inverse problems with known forward models. MoDL is an iterative or "unrolled" architecture, which alternates between a data-consistency step and a trained CNN denoiser, with weights tied across unrolled iterations. We use a U-Net architecture as the denoiser in our implementation of MoDL, ensuring that the overall number of parameters (except for a learned scaling factor in MoDL) is the same in both architectures.

To compare to deep learning-based approaches which do not require training on particular forward models, we compare to the Image-Adaptive GAN (IAGAN) [18] and to Regularization by Denoising (RED) [32]. IAGAN leverages a pretrained GAN to reconstruct arbitrary linear measurements by fitting the latent code input to the GAN, while also tuning the GAN weights in a way similar to our proposed P&P approach and the Deep Image Prior approach [38]. We utilize early stopping in this optimization process by choosing a fixed early stopping point based on a held-out validation set.

RED requires only a pretrained denoiser, which we implement by pretraining a set of residual U-Net denoisers on the fastMRI and CelebA training sets, with a variety of different Gaussian noise levels. Specifically, we train 15 denoisers for each problem setting, with σ ranging from 10^{-4} to 10^{1} on a logarithmic scale. All results shown are tuned on the validation set to ensure the optimal denoisers are used.

We also compare to a penalized least squares approach with total variation regularization [34], a classical approach that does not use any learned elements. While more complex regularizers are possible, total variation (TV) is used because of its status as a simple, widely-used conventional baseline.

B. Parametrizing forward models

Both of our proposed model adaptation methods permit the new forward model to be unknown during training, provided it has a known parametrization. In this case, the parameters describing the forward model are learned along with the reconstruction. Here we describe the parametrizations of the forward models that are used.

For the deblurring task, the unknown blur kernel is parametrized as a 7x7 blur kernel, initialized with the weights used for the ground-truth kernel during the initial stage of training. Practically, this is identical to a standard convolutional layer with a fixed initialization and only one learned kernel.

A similar approach is used for superresolution. The forward model can be efficiently represented by strided convolution, and the adjoint is represented by a standard "convolution transpose" layer, again with the weights initialized to match the forward operator in the initial pre-training phase.

In the case of MRI, we use two choices of A_1, depending on whether we assume A_1 is fully known or not. In the case A_1 is fully known, we utilize another $6\times$ undersampled k-space mask, but with resampled high-frequency lines. We display the original and new k-space sampling masks in Figure 4. To illustrate the utility of our approach under miscalibration of the forward model in an MRI reconstruction setting, we also consider a unknown random perturbation of the original k-space lines, which we attempt to learn during reconstruction. The vertical k-space lines are still fully sampled, as are the center 4% of frequencies, but all high frequency lines are perturbed uniformly at random with a continuous value from -2 to 2. We wish to emphasize that this experiment is not meant to reflect clinical practice, since such miscalibration of k-space sampling locations is not typically encountered in anatomical imaging with Cartesian k-space sampling trajectories. However, we include this experiment simply to illustrate that our approach could be extended to unknown parametric changes in the forward model in an MR reconstruction setting.

C. Main results

In Table I we present our main results. We present sample reconstructions for the deblurring problem and MRI reconstruction problem in Figs. 7 and 8. For reference, the ground truth, inputs to the networks, a total variation regularized reconstruction, and a RED reconstruction are presented in Figs. 5 and 6. We also provide in the Appendix a table of SSIM values as well as the full version of Table I, which contains the standard deviations of PSNR.

While the magnitude of the improvements vary across domains and problems, we find that retraining the network with the proposed model adaptation techniques significantly improve performance by several dBs in the new setting. This effect is particularly striking in the case of MRI reconstruction with MoDL, where the "naive" approach of replacing A_0 with A_1 in the network gives catastrophic results (a roughly 9 dB drop in reconstruction PSNR), while the proposed model adaptation approaches give reconstruction PSNRs within 1-2 dB of the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCI.2021.3094714, IEEE Transactions on Computational Imaging

		Baselines					
		TV	RED		Train w/A_0 Test w/A_0	Train w/A_0 Test w/A_1	Train w/A_1 Test w/A_1
Blur		27.61	30.23	U-Net	34.15	25.42	33.98
				MoDL	36.25	23.91	36.13
SR		28.33	28.59	U-Net	30.74	26.3	31.22
				MoDL	31.32	22.27	31.98
MRI		25.09	27.76	U-Net	31.51	27.47	32.33
				MoDL	31.88	22.82	31.79
		Proposed Model Adaptation Methods					
		Known A_1			Unknown A_1		
		P&P (Alg. 1)	R&R (Alg. 2)	R&R+ (Alg. 3)	P&P (Alg. 1)	R&R (Alg. 2)	R&R+ (Alg. 3)
Blur	U-Net	33.01	32.11	**33.50**	29.18	27.67	30.05
	MoDL	30.08	33.82	**34.73**	29.89	27.81	27.94
SR	U-Net	28.00	29.95	**29.99**	27.77	26.98	29.35
	MoDL	24.59	28.18	**29.83**	23.14	24.93	25.29
MRI	U-Net	29.07	29.71	**31.43**	28.92	28.06	29.54
	MoDL	30.63	30.25	**31.44**	26.64	23.46	27.67

TABLE I: Comparison of performance of various baseline methods for inverse problems across a variety of datasets and forward models. The metric presented is the mean PSNR. SSIM values can be found in Table IV.

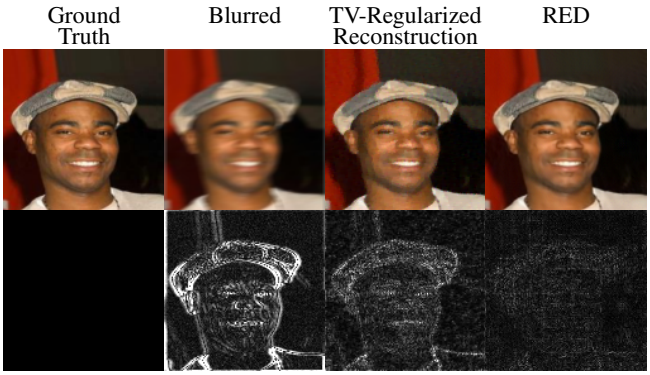


Fig. 5: Comparison figures for the deblurring methods in Figure 7. We present the ground truth, the blurred image (with Gaussian noise with $\sigma = 0.01$ added), a total variation (TV) regularized reconstruction, and a comparison to Regularization by Denoising (RED), a model-agnostic method leveraging a deep denoiser. Below each of the above is the residual image, multiplied by $5\times$ for ease of visualization.

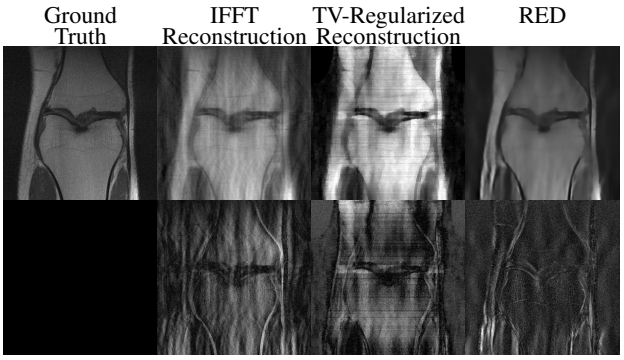


Fig. 6: Comparison figures for the MRI reconstruction methods in Figure 8. We present the IFFT with all k-space data maintained, the naïve IFFT reconstruction after k-space masking, a total variation (TV) regularized reconstruction (with PSNR 27.3 dB), and a RED reconstruction (with PSNR 28.4 dB). We also present the residuals relative to the fully-sampled IFFT, multiplied by $5\times$ for ease of visualization.

baseline approach of training and testing with the same forward model in the case where A_1 is known.

D. Learning multiple forward models

In this section we explore an alternative approach to model adaptation. In this setting, we assume that a set of candidate forward models are known during training time. During test time, a single forward model is used for measurement, but the test-time forward model is not known during training. This case represents the setting where the forward model might be parametrized, and so a reasonable approach may be to train the learned network using a number of different forward models to improve robustness.

In simple settings, training on multiple models might be reasonable. However, when the forward model parameterization is high-dimensional, learning to invert all possible forward models may be difficult.

We demonstrate this setting with a deblurring example, in which the same network is trained using a number of blur kernels. The blur kernels are the same kernels used for comparisons in [16]. For consistency, we resize all 50 blur kernels to 7x7, and normalize the weights to sum to 1. We compare reconstruction accuracy when the ground truth blur kernel is included in the set of kernels used for training, as well as when the reconstruction network has never seen data blurred with the testing kernel.

The results are shown in Fig 9. Experimentally, we find that training on multiple blur kernels simultaneously incurs a performance penalty as the number of blur kernels used in training increases. In this setting, where the forward model has many degrees of freedom and data is limited, attempting to learn to solve all models simultaneously is worse than transferring a single learned model, even in the absence of further ground truth data for calibration.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCI.2021.3094714, IEEE Transactions on Computational Imaging

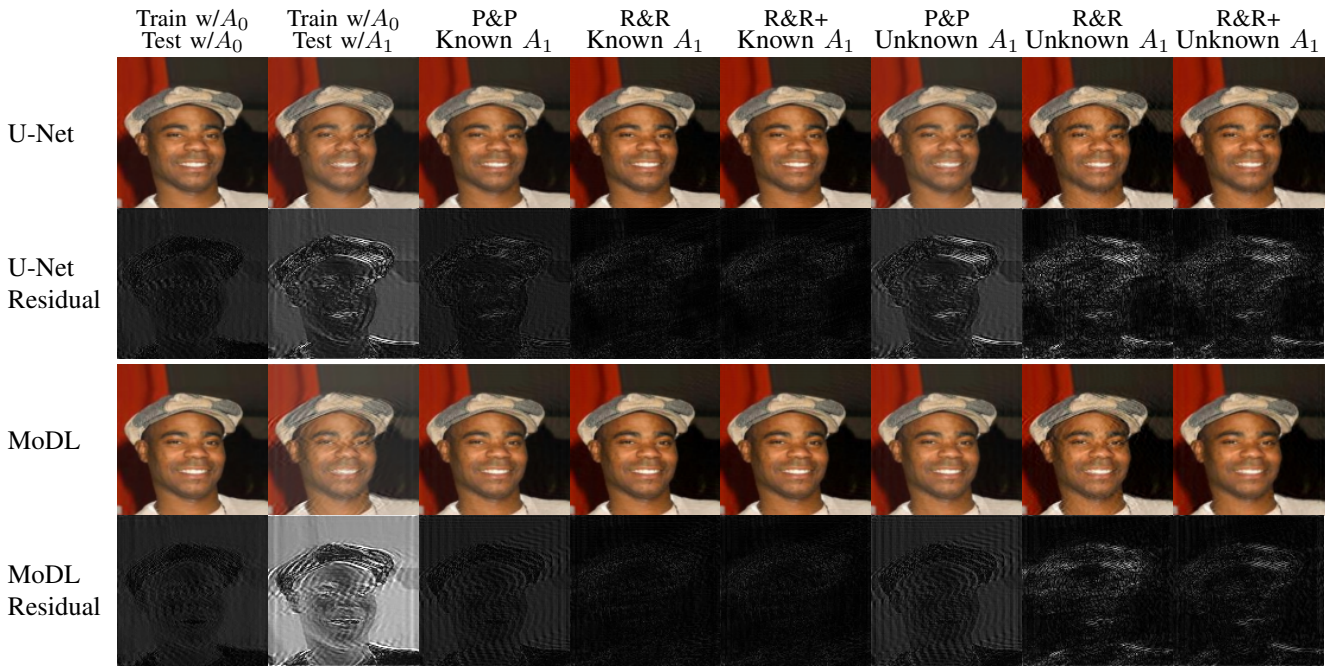


Fig. 7: Visual examples of reconstruction quality for the motion deblurring inverse problem solved by U-Net and MoDL, as well as the associated residuals. Each residual is multiplied by 5 for ease of inspection. The initial forward model A_0 is a 7x7 motion blur with angle $10°$, and the A_1 model has a 7x7 motion blur kernel with angle $20°$. The analogous figure for the superresolution problem, and further examples, are available in the Supplement. Best viewed electronically.

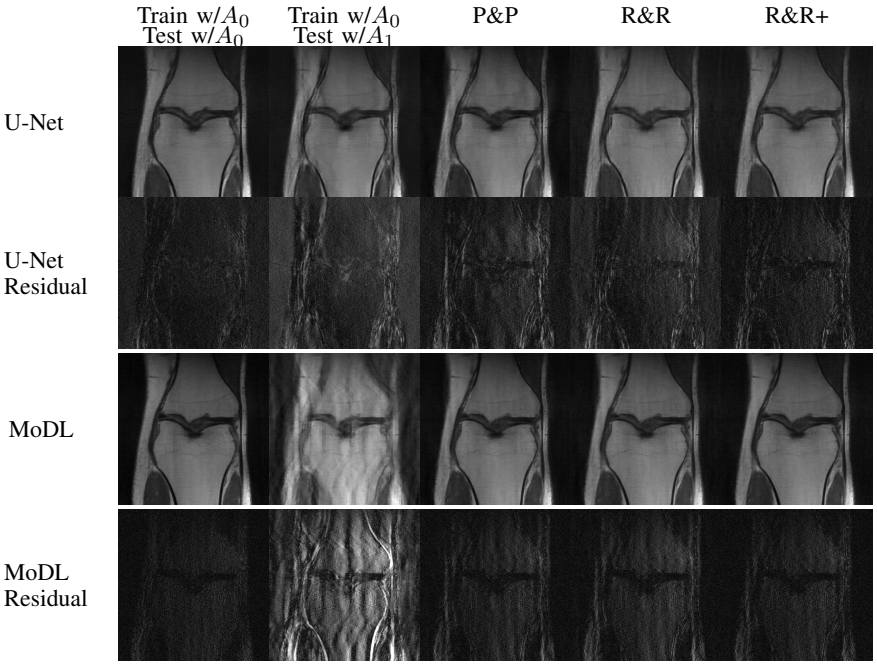


Fig. 8: Visual examples of different reconstruction approaches for the MRI inverse problem under model drift, along with associated residuals. All residual images are scaled by 5x for ease of inspection. Best viewed electronically.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCI.2021.3094714, IEEE Transactions on Computational Imaging

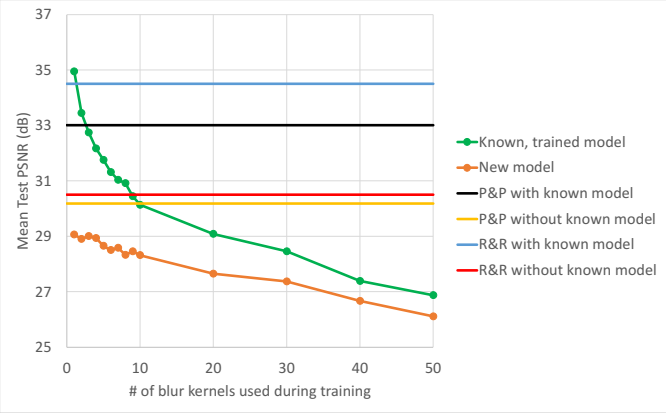


Fig. 9: Naïvely learning to deblur with a single network and multiple blur kernels sacrifices performance on all blurs. In **green**, the test-time accuracy of a network trained to deblur multiple blurs, tested on a known kernel. In **orange**, the same network, but tested on a new blur that was not used during training. In **black**, our proposed P&P approach (Alg. 1), with a known model, and in **yellow** the same with a learned forward model. **Blue** and **red** show the performance of our R&R approach (Alg. 2), with and without a known forward model.

Adaptation	Training A	Deployment Acceleration Factor α 2×	4×	6×	8×	12×
None	α	35.74	32.53	31.51	30.69	29.48
None	6×	27.02	30.20	31.51	27.76	26.15
None	All α	33.99	31.62	30.48	29.25	28.35
R&R	6×	35.11	**32.61**	**31.73**	29.40	27.34
R&R	All α	**35.80**	32.35	30.81	**29.60**	**28.61**

TABLE II: Comparison of reconstruction PSNR for a variety of MRI acceleration factors for several different approaches. "All α" refers to a U-Net trained for reconstruction on all shown sampling rates, whereas each column of the "α" row results represents a network trained for that particular sampling pattern, i.e. the 2× column is tested on a network trained for 2× acceleration. The grayed-out "α" row represents the "oracle" setting where the deployment forward model is known at train time. 6× refers to the network shown in other experiments. Training the 6× model consistently performs well for that particular forward model, but without model adaptation it has lower performance on accelerations it was not trained for, even for higher sampling rates. The All α approach sacrifices performance on any one forward model, but most of the difference can be removed by augmenting the All α network with our R&R method.

E. Adapting to variable sampling rates in single-coil MRI

A particular concern raised in [3] is related to the stability of a learned solver with respect to the level of undersampling at measurement time. In particular, the authors of that work observe that an image reconstruction system trained to recover images sampled at a particular rate would experience a degradation in reconstruction accuracy for *higher* sampling rates than the one the system was trained on.

In Fig. 10 we explore this problem in the MRI setting using a U-Net as the reconstruction method, and demonstrate that our R&R method can adapt to this setting as well. By using R&R during inference, the learned network was trained at a 6× accceleration acquisition setting, but was safely deployed for other accelerations *without significant degradation in reconstruction quality*, and *comparing favorably to networks trained explicitly for other sampling rates*.

For comparison purposes, we also train a U-Net using multiple sampling rates α. During training, the multiple-α solver is trained to reconstruct MRIs that are measured using the five different sampling patterns demonstrated in Fig. 10. We present the mean PSNR on the test set in Table II, along with the mean test PSNR for applying R&R to the multiple-model solver, assuming at test time that the sampling pattern is known. Reconstructions from the multiple-model solver can be found in the Supplement. We observe that training with multiple models means that at test time all models produce reasonable reconstructions, but at the cost of reconstruction quality compared to networks trained for single sampling patterns.

In this experiment, we also observe an interesting side-effect of R&R: when R&R is used to "adapt" to a forward model A_0 that the original network was trained on, we tend to see an improvement in reconstruction quality. This effect is most pronounced for the U-Net trained to reconstruct multiple sampling patterns, but is also true for the "dedicated" solver, demonstrated in Fig. 10.

F. Model adaptation under variable model overlap

In this section we explore how varying the distance between the forward models A_0 and A_1 affects reconstruction quality, and how our proposed R&R method deals with different amounts of overlap between A_0 and A_1. The forward model under investigation is 6× single-coil MRI reconstruction.

To explore variable levels of model drift in the single-coil MRI reconstruction case, we vary which k-space frequencies are sampled in a Cartesian pattern. Specifically, we construct a list of "non-sampled" frequencies and a list of "sampled" frequencies under A_0. We create A_1 by swapping n "sampled" frequencies for n frequencies in the original "non-sampled" list, to ensure that the new A_1 contains exactly n frequencies that were not sampled under A_0. We do not swap the 4% center frequencies in any test.

In Fig. 11 we plot n vs the mean PSNR over 10 separate instantiations of the above experiment for a no-adaptation approach as well as our R&R method. We run 10 separate instances since the frequencies that are swapped, as well as what frequencies they are swapped to, is random, introducing some variance to the process. We also visually represent the maximum and minimum PSNR across all instances with shading.

Note that the new A_1 in this subsection and Fig. 11 is different from the A_1 shown in Fig. 4b that is used in other MRI experiments in Table III. The A_1 in Fig. 4b corresponds to a new random selection of frequencies, some of which might also be represented in A_0 (in Fig. 4a). In contrast, the A_1 used for the results in Fig. 11 may be "harder" for model

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCI.2021.3094714, IEEE Transactions on Computational Imaging

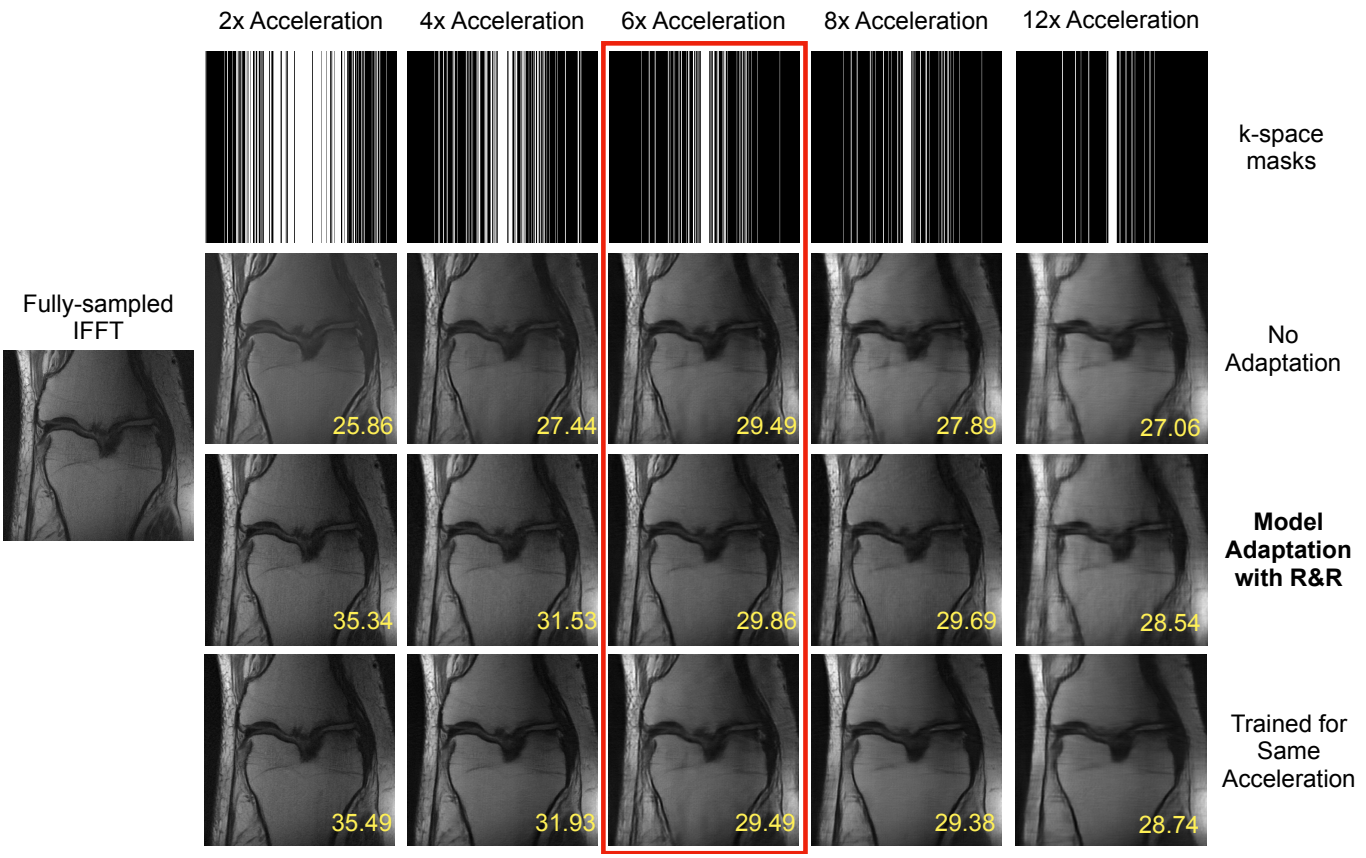


Fig. 10: Using the R&R model adaptation approach permits using a U-Net trained for $6\times$ acceleration on MRI reconstruction across a range of acceleration parameters. The various k-space sampling patterns used in these experiments are shown in the top row. Without adaptation (second row), the reconstruction quality decreases when changing the acceleration factor, *even when more k-space measurements are taken*, as originally observed in [3]. The R&R reconstructions (third row) compare favorably to the performance of networks trained on each particular k-space sampling pattern (bottom row). The PSNR of each image is presented in dB in yellow on each image.

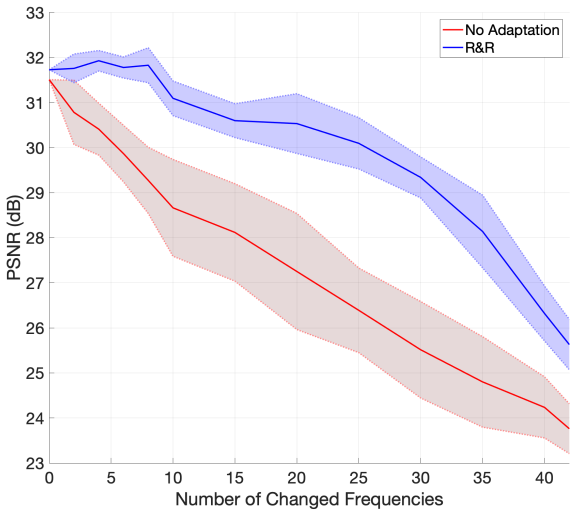


Fig. 11: Comparison of the mean PSNR for the R&R method and no adaptation for single-sample MRI reconstruction vs. the number of frequencies that differ between A_0 and A_1. The shaded areas represent the standard deviation of mean test PSNR over 10 runs, since frequencies are replaced randomly.

adaptation because the new frequencies were explicitly chosen to have zero joint support with those in A_0. In this experiment, the number of changed frequencies acts as a proxy for the difference between $N(A_0)$ and $N(A_1)$, the null spaces of A_0 and A_1.

G. Sample complexity

Our other experiments assume that model adaptation is performed at the level of individual test samples. However, in the case where we have access to a *calibration set* of measurements under the new forward model A_1 that we can leverage to retrain the network using the P&P approach. In this case, the optimization is performed over an objective function that is the mean of Eq. 3 for all y_i in the calibration set.

In the transfer learning setting, a key concern is the size of the transfer learning set necessary to achieve high-quality results. In this section we compare the performance of P&Pacross different calibration set sizes.

In Fig. 12 we explore the effect of the number of samples observed under the new forward model on the adapted model. We observe that even without knowing the forward model, a single calibration sample is sufficient to give improvement

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCI.2021.3094714, IEEE Transactions on Computational Imaging

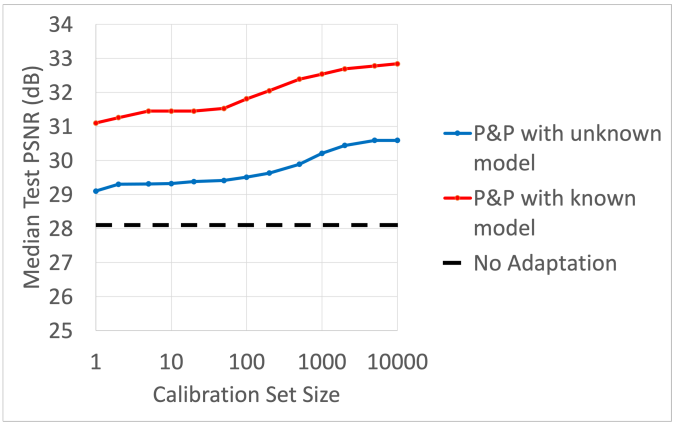


Fig. 12: Performance of the P&P model adaptation approach for motion deblurring as a function of the number of calibration samples (blurred images) under the new forward model. Both of our approaches outperform a naive approach ("No Adaptation"), even without exact knowledge of the new forward model.

over the "naive" method that replaces A_0 with A_1 without further retraining. When the forward model is known during calibration and testing, a single example image can result in a 2 dB improvement in PSNR.

H. Model-blind reconstruction with generative networks

Recent work [2], [5], [8], [18] has explored solving inverse problems using generative networks, which permit reconstruction under arbitrary forward models assuming an expressive enough generative network. In particular, [2] and [5] consider the case where the forward model is either partially or entirely unknown, and hence may be learned by parameterizing and jointly optimizing over both the forward model and the latent code for the generative network.

In Fig. 13 we provide an illustration of reconstructions on a $2x$ superresolution with bicubic downsampling problem obtained by the method of [18], compared to our proposed R&R approach. In our demonstration, as in [18], the generative network under consideration is a pretrained Boundary Equilibrium GAN (BEGAN) [6]. The R&R approach uses a U-Net trained for $2\times$ superresolution with bilinear downsampling. The reconstruction quality is higher when a model-specific network is used, especially when examining fine details and textures.

In the absence of (x_i, y_i) pairs, a generative approach may be reasonable. However, learning the data manifold in its entirety requires a great deal of data at minimum, along with a sufficiently large and well-tuned generator. The authors of [44] also note this fundamental limitation: for smaller or simpler applications, learning a high-quality GAN is straightforward, but for more complex applications it is difficult to train GAN models that are sufficiently accurate to rely on for high-quality reconstructions.

V. Discussion and Conclusion

This paper explores solutions to the fragility of learned inverse problem solvers in the face of model drift. We

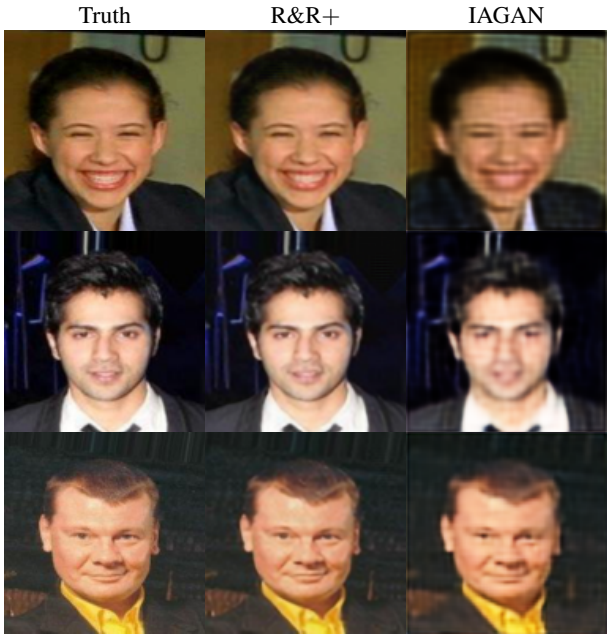


Fig. 13: Comparison of model adaptation (R&R+) with a model-blind GAN-based reconstruction approach (IAGAN [18]) for $2\times$ super-resolution with bicubic downsampling. The R&R approach adapts a U-Net trained on a forward model of $2\times$ bilinear downsampling. While a GAN-based approach only requires learning a single generative network for all forward models, our results suggest that a network trained for a specific forward model with the same number of training samples gives better reconstructions. Best viewed electronically.

demonstrate across a range of simple, practical applications that using a learned image reconstruction network in settings even slightly different than they were trained in results in significant reconstruction errors, both subtle and obvious. We propose two model adaptation procedures: the first is based on a transfer learning approach that attempts to learn a perturbation to the pre-trained network parameters, which we call Parametrize and Perturb (P&P); the second reuses the network as an implicitly defined regularizer in an iterative model-based reconstruction scheme, which we call Reuse and Regularize (R&R). We also look at a hybrid approach combining these techniques we call R&R+.

We show that our model adaptation techniques enable reuse of learned solvers under a change in the forward model, even when the change in forward model is not known. In addition, we demonstrate that just learning to invert a variety of forward models at once is not necessarily the solution to the problem of model drift: directly training on many forward models empirically appears to cause reconstruction quality to fall across all learned models. We also show that our approach is superior to one that requires learning a model of the entire image space via a generative model.

The proposed P&P, R&R, and R&R+ model adaptation approaches each have different trade-offs, and may be useful in different scenarios. In general, we observe that R&R+ produces superior reconstructions over R&R and P&P, but incurs significant computation and time costs associated with

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCI.2021.3094714, IEEE Transactions on Computational Imaging

network retraining specified in (10). The P&P approach also incurs similar costs associated with network retraining. However, when a calibration set is available (as in Section IV-G), the P&P approach only needs to be retrained once, and computation cost at deployment matches the original solver. However, we observe two significant benefits of the R&R approach. First, empirically we observe that only few iterations of R&R (see Algorithm 2) tend to be required to give accurate results (namely, 5 iterations in all our experiments), which increases computational cost by only a constant factor relative to the original reconstruction network. In addition, in the R&R approach only one new parameter is introduced, in contrast to several parameters related to the optimization required for P&P and R&R+. Finally, our experiments suggest that the improvement offered by R&R+ tends to be marginal relative to the improvement seen by going from no adaptation to R&R. Therefore, in situations where reconstruction time is crucial, model adaptation by R&R may be preferred over R&R+.

One surprising benefit of the R&R approach is that even in the absence of model drift (i.e., $A_0 = A_1$) the reconstruction accuracy improves relative to the output from the reconstruction network. This is because R&R iteratively modifies the output of the network to enforce data-consistency at test time. This may potentially resolve the issue raised in [35] about whether learned image reconstruction networks are truly "solving" a given inverse problem, *i.e.*, give a well-defined inverse map of the measurement model. However, to show this would require a much more detailed analysis of the estimator defined by the R&R approach that is beyond the scope of this work.

Adapting learned inverse problem solvers to function under new forward models is just one step towards robustifying these powerful approaches. Our approach for unknown A_1 assumes an explicit parametrization of the forward model, but such a parametrization is not always straightforward or realistic. How best to adapt to complex changes in the forward model that are not easily parametrized is an important open question for future work; see [22] for one recent approach to learning non-parametric (and potentially non-linear) changes to forward models in an iterative reconstruction framework.

Other crucial aspects of robustness of learned inverse problem solvers have been proposed by [3] and [13]. In particular, the tendency of learned solvers to be sensitive to small perturbations in the input and the possibility of nonphysical "hallucinations" remain important open problems to solve. This work is directed towards solving the brittleness of learned solvers to changes in the forward model, but we do not address these other important elements of robustness. Exploring the impact of model drift on such hallucinations is an important future direction.

While this work focused on linear inverse problem, many of the principles introduced in this work extend also to non-linear inverse problems. For example, the R&R approach, which is based on the regularization-by-denoising technique (RED), is readily adapted to non-linear problems amenable to a RED approach, which includes phase retrieval [25] among others.

Our empirical evidence suggests that successful model adaptation is possible provided the nullspace (or approximate nullspaces) of A_0 and A_1 are close in some sense. However, in settings where nullspaces of A_0 and A_1 are far apart, model adaptation may lead to artifacts or hallucinated details in the reconstructions. In order to understand these limitations of model adaptation, recent methods introduced to quantify hallucinations induced by neural-network based reconstructions may prove to be useful [7].

Finally, while we focused our attention on model drift, an important open problem is how to adapt to simultaneous model and data distribution drift, and the extent to which these effects can be treated independently. We hope to address these questions in future work.

APPENDIX

In this Appendix we present companion tables to Table I which contain further information, including corresponding mean SSIM over the test set, as well as the standard deviations of PSNR and SSIM.

		Baselines					
		TV	RED		Train w/A_0 Test w/A_0	Train w/A_0 Test w/A_1	Train w/A_1 Test w/A_1
Blur		27.61 ± 2.57	30.23 ± 2.98	U-Net	34.15 ± 2.33	25.42 ± 1.74	33.98 ± 2.15
				MoDL	36.25 ± 2.25	23.91 ± 2.02	36.13 ± 2.19
SR		28.33 ± 2.42	28.59 ± 2.09	U-Net	30.74 ± 2.59	26.3 ± 1.65	31.22 ± 2.71
				MoDL	31.32 ± 2.65	22.27 ± 2.04	31.98 ± 2.61
MRI		25.09 ± 2.50	27.76 ± 3.37	U-Net	31.51 ± 2.83	27.47 ± 2.47	32.33 ± 2.64
				MoDL	31.88 ± 2.85	22.82 ± 2.75	31.79 ± 2.81
		Proposed Model Adaptation Methods					
		Known A_1			Unknown A_1		
		P&P (Alg. 1)	R&R (Alg. 2)	R&R+ (Alg. 3)	P&P (Alg. 1)	R&R (Alg. 2)	R&R+ (Alg. 3)
Blur	U-Net	33.01 ± 1.85	32.11 ± 2.64	33.50 ± 2.47	29.18 ± 1.81	27.67 ± 2.23	30.05 ± 2.73
	MoDL	30.08 ± 1.59	33.82 ± 1.73	34.73 ± 1.82	29.89 ± 1.66	27.81 ± 2.3	27.94 ± 2.4
SR	U-Net	28.00 ± 1.83	29.95 ± 2.49	29.99 ± 2.48	27.77 ± 2.15	26.98 ± 2.39	29.35 ± 2.36
	MoDL	24.59 ± 2.31	28.18 ± 1.64	29.83 ± 2.00	23.14 ± 2.01	24.93 ± 2.04	25.29 ± 2.33
MRI	U-Net	29.07 ± 2.72	29.71 ± 2.75	31.43 ± 2.99	28.92 ± 3.04	28.06 ± 2.63	29.54 ± 2.53
	MoDL	30.63 ± 2.85	30.25 ± 3.10	31.44 ± 2.75	26.64 ± 2.60	23.46 ± 2.54	27.67 ± 2.62

TABLE III: Comparison of performance of various baseline methods for inverse problems across a variety of datasets and forward models. The metric presented is the mean PSNR ± the standard deviation.

		Baselines					
		TV	RED		Train w/A_0 Test w/A_0	Train w/A_0 Test w/A_1	Train w/A_1 Test w/A_1
Blur		0.94 ± 0.06	0.96 ± 0.05	U-Net	0.98 ± 0.05	0.89 ± 0.09	0.98 ± 0.05
				MoDL	0.98 ± 0.03	0.84 ± 0.08	0.98 ± 0.04
SR		0.95 ± 0.06	0.96 ± 0.03	U-Net	0.97 ± 0.03	0.92 ± 0.09	0.97 ± 0.02
				MoDL	0.97 ± 0.04	0.89 ± 0.06	0.98 ± 0.04
MRI		0.79 ± 0.04	0.80 ± 0.05	U-Net	0.82 ± 0.06	0.74 ± 0.06	0.82 ± 0.06
				MoDL	0.83 ± 0.06	0.65 ± 0.08	0.83 ± 0.06
		Proposed Model Adaptation Methods					
		Known A_1			Unknown A_1		
		P&P (Alg. 1)	R&R (Alg. 2)	R&R+ (Alg. 3)	P&P (Alg. 1)	R&R (Alg. 2)	R&R+ (Alg. 3)
Blur	U-Net	0.81 ± 0.09	0.81 ± 0.07	0.82 ± 0.07	0.75 ± 0.05	0.79 ± 0.09	0.77 ± 0.08
	MoDL	0.82 ± 0.07	0.81 ± 0.09	0.83 ± 0.07	0.72 ± 0.09	0.68 ± 0.07	0.75 ± 0.10
SR	U-Net	0.94 ± 0.09	0.96 ± 0.04	0.96 ± 0.04	0.94 ± 0.09	0.94 ± 0.08	0.96 ± 0.06
	MoDL	0.92 ± 0.06	0.94 ± 0.02	0.95 ± 0.02	0.91 ± 0.04	0.92 ± 0.02	0.94 ± 0.01
MRI	U-Net	0.81 ± 0.09	0.81 ± 0.07	0.82 ± 0.07	0.75 ± 0.05	0.79 ± 0.09	0.77 ± 0.08
	MoDL	0.82 ± 0.07	0.81 ± 0.09	0.83 ± 0.07	0.72 ± 0.09	0.68 ± 0.07	0.75 ± 0.10

TABLE IV: Comparison of performance of various baseline methods for inverse problems across a variety of datasets and forward models. The metric presented is the mean SSIM ± the standard deviation.

REFERENCES

[1] Hemant K Aggarwal, Merry P Mani, and Mathews Jacob. MoDL: Model-based deep learning architecture for inverse problems. *IEEE transactions on medical imaging*, 38(2):394–405, 2018.

[2] Rushil Anirudh, Jayaraman J Thiagarajan, Bhavya Kailkhura, and Timo Bremer. An unsupervised approach to solving inverse problems using generative adversarial networks. *arXiv preprint arXiv:1805.07281*, 2018.

[3] Vegard Antun, Francesco Renna, Clarice Poon, Ben Adcock, and Anders C Hansen. On instabilities of deep learning in image reconstruction and the potential costs of ai. *Proceedings of the National Academy of Sciences*, 2020.

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

[5] Kalliopi Basioti and George V Moustakides. Image restoration from parametric transformations using generative models. *arXiv preprint arXiv:2005.14036*, 2020.

[6] David Berthelot, Thomas Schumm, and Luke Metz. Began: Boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*, 2017.

[7] Sayantan Bhadra, Varun A Kelkar, Frank J Brooks, and Mark A Anastasio. On hallucinations in tomographic image reconstruction. *arXiv preprint arXiv:2012.00646*, 2020.

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

[9] Alhussein Fawzi, Horst Samulowitz, Deepak Turaga, and Pascal Frossard. Image inpainting through neural networks hallucinations. In *2016 IEEE 12th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP)*, pages 1–5. Ieee, 2016.

[10] Jeffrey A Fessler. Model-based image reconstruction for MRI. *IEEE signal processing magazine*, 27(4):81–89, 2010.

[11] Martin Genzel, Jan Macdonald, and Maximilian März. Solving inverse problems with deep neural networks–robustness included? *arXiv preprint arXiv:2011.04268*, 2020.

[12] Davis Gilton, Greg Ongie, and Rebecca Willett. Neumann networks for linear inverse problems in imaging. *IEEE Transactions on Computational Imaging*, 2019.

[13] Nina M Gottschling, Vegard Antun, Ben Adcock, and Anders C Hansen. The troublesome kernel: why deep learning for inverse problems is typically unstable. *arXiv preprint arXiv:2001.01258*, 2020.

[14] Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th international conference on international conference on machine learning*, pages 399–406, 2010.

[15] Yoseob Han, Jaejun Yoo, Hak Hee Kim, Hee Jung Shin, Kyunghyun Sung, and Jong Chul Ye. Deep learning with domain adaptation for accelerated projection-reconstruction mr. *Magnetic resonance in medicine*, 80(3):1189–1205, 2018.

[16] Michal Hradiš, Jan Kotera, Pavel Zemcık, and Filip Šroubek. Convolutional neural networks for direct text deblurring. In *Proceedings of BMVC*, volume 10, page 2, 2015.

[17] Shady Abu Hussein, Tom Tirer, and Raja Giryes. Correction filter for single image super-resolution: Robustifying off-the-shelf deep super-

resolvers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1428–1437, 2020.

[18] Shady Abu Hussein, Tom Tirer, and Raja Giryes. Image-adaptive gan based reconstruction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3121–3129, 2020.

[19] Hammernik Kerstin, Schlemper Jo, Qin Chen, Duan Jingming, Summers Ronald M., and Rueckert Daniel. σ-net Systematic Evaluation of Iterative Deep Neural Networks for Fast Parallel MR Image Reconstruction. *arXiv preprint arXiv:1912.09278*, 2019.

[20] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017.

[21] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[22] Sebastian Lunz, Andreas Hauptmann, Tanja Tarvainen, Carola-Bibiane Schönlieb, and Simon Arridge. On learned operator correction in inverse problems. *SIAM Journal on Imaging Sciences*, 14(1):92–127, 2021.

[23] Sebastian Lunz, Ozan Öktem, and Carola-Bibiane Schönlieb. Adversarial regularizers in inverse problems. In *Advances in Neural Information Processing Systems*, pages 8507–8516, 2018.

[24] Morteza Mardani, Qingyun Sun, David Donoho, Vardan Papyan, Hatef Monajemi, Shreyas Vasanawala, and John Pauly. Neural proximal gradient descent for compressive imaging. In *Advances in Neural Information Processing Systems*, pages 9573–9583, 2018.

[25] Christopher Metzler, Phillip Schniter, Ashok Veeraraghavan, et al. prdeep: robust phase retrieval with a flexible deep network. In *International Conference on Machine Learning*, pages 3501–3510. PMLR, 2018.

[26] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pages 10–18. PMLR, 2013.

[27] Gregory Ongie, Ajil Jalal, Christopher A Metzler, Richard G Baraniuk, Alexandros G Dimakis, and Rebecca Willett. Deep learning techniques for inverse problems in imaging. *arXiv preprint arXiv:2005.06001*, 2020.

[28] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[29] Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends in optimization*, 1(3):127–239, 2014.

[30] Ankit Raj, Yoram Bresler, and Bo Li. Improving robustness of deep-learning-based image reconstruction. *arXiv preprint arXiv:2002.11821*, 2020.

[31] Edward T Reehorst and Philip Schniter. Regularization by denoising: Clarifications and new interpretations. *IEEE Transactions on Computational Imaging*, 5(1):52–67, 2018.

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

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

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

[35] Emil Sidky, Iris Lorente, Jovan G Brankov, and Xiaochuan Pan. Do CNNs solve the CT inverse problem. *IEEE Transactions on Biomedical Engineering*, 2020.

[36] Jian Sun, Huibin Li, Zongben Xu, et al. Deep ADMM-Net for compressive sensing MRI. In *Advances in neural information processing systems*, pages 10–18, 2016.

[37] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning. In *International conference on artificial neural networks*, pages 270–279. Springer, 2018.

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

[39] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018.

[40] Kaixuan Wei, Angelica Aviles-Rivero, Jingwei Liang, Ying Fu, Carola-Bibiane Schnlieb, and Hua Huang. Tuning-free plug-and-play proximal algorithm for inverse imaging problems. *arXiv preprint arXiv:2002.09611*, 2020.

[41] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3(1):1–40, 2016.

[42] Shanshan Wu, Alex Dimakis, Sujay Sanghavi, Felix Yu, Daniel Holtmann-Rice, Dmitry Storcheus, Afshin Rostamizadeh, and Sanjiv Kumar. Learning a compressed sensing measurement matrix via gradient unrolling. In *International Conference on Machine Learning*, pages 6828–6839. PMLR, 2019.

[43] LI Xuhong, Yves Grandvalet, and Franck Davoine. Explicit inductive bias for transfer learning with convolutional networks. In *International Conference on Machine Learning*, pages 2825–2834. PMLR, 2018.

[44] Raymond A Yeh, Chen Chen, Teck Yian Lim, Alexander G Schwing, Mark Hasegawa-Johnson, and Minh N Do. Semantic image inpainting with deep generative models. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5485–5493, 2017.

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