Domain Expansion via Network Adaptation for Solving Inverse Problems

Nebiyou Yismaw, *Graduate Student Member, IEEE*, Ulugbek S. Kamilov, *Senior Member, IEEE*, and M. Salman Asif, *Senior Member, IEEE*

***Abstract*—Deep learning-based methods deliver state-of-the-art performance for solving inverse problems that arise in computational imaging. These methods can be broadly divided into two groups: (1) learn a network to map measurements to the signal estimate, which is known to be fragile; (2) learn a prior for the signal to use in an optimization-based recovery. Despite the impressive results from the latter approach, many of these methods also lack robustness to shifts in data distribution, measurements, and noise levels. Such domain shifts result in a performance gap and in some cases introduce undesired artifacts in the estimated signal. In this paper, we explore the qualitative and quantitative effects of various domain shifts and propose a flexible and parameter efficient framework that adapts pretrained networks to such shifts. We demonstrate the effectiveness of our method for a number of reconstruction tasks that involve natural image, MRI, and CT imaging domains under distribution, measurement model, and noise level shifts. Our experiments demonstrate that our method achieves competitive performance compared to independently fully trained networks, while requiring significantly fewer additional parameters, and outperforms several domain adaptation techniques.**

***Index Terms*—Inverse problems, image recovery, domain adaptation, unrolled networks.**

I. Introduction

LINEAR inverse problems arise in many real-world applications. For instance, image enhancement and restoration tasks in denoising, deblurring, and super-resolution or medical image reconstruction from indirect measurements in computed tomography (CT) and magnetic resonance imaging (MRI). We can model such inverse problems as the recovery of an unknown signal $\mathbf{x}$ from a set of measurements:

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

where $\mathbf{y}$ represents measurements, $\mathbf{A}$ represents an $m \times n$ measurement matrix or forward operator, and η represents noise. The unknown signal and measurements can be real- or complex-valued. To recover $\mathbf{x}$, we can solve an optimization problem of the following form:

$$\min_{\mathbf{x}} g(\mathbf{x}) + h_\theta(\mathbf{x}), \tag{2}$$

where $g(\mathbf{x})$ is a data fidelity term (e.g., $g(\mathbf{x}) = \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2$), $h_\theta(\cdot)$ denotes a regularization function that enforces some prior constraint on the unknown signal, and θ denotes the regularization function parameters [1], [2].

In the deep learning era, we can recover $\mathbf{x}$ by either training a deep (reconstruction) network that maps measurements to the signal estimate or solving an iterative optimization problem (similar to the one in (2)) that can also be represented as an unrolled network [3]. While training a reconstruction network in an end-to-end manner is possible, it usually requires a large set of input-output training pairs $(\mathbf{y}, \mathbf{x})$. Furthermore, since these networks do not explicitly use the forward model in (1), they are known to be sensitive to small changes in the data distribution, measurement operators, and noise [4], [5]. Solving the optimization problem in (2) with an appropriate choice of regularization function $h(\cdot)$ is often considered a flexible and relatively robust option.

In recent years, deep networks are often used to represent $h(\cdot)$ instead of hand-designed functions (e.g., ℓ_1 norm or total variation). For instance, deep unrolling [6], [7], [8] and plug-and-play (PnP) [1], [9] methods use artifact removal (AR) or image denoising networks that are trained to map a noisy or corrupted estimate of an image onto a clean image manifold [3], [7], [8]. Despite recent success of deep learning-based methods, they are sensitive to shifts in the data distribution [10].

Fig. 1 illustrates this effect for deep unrolling with artifact removal (AR) networks under domain and forward model shifts. The fastMRI AR is trained while solving (2) for MR image reconstruction from radially under-sampled simulated k-space measurements. CelebA AR is trained while solving (2) to reconstruct face images from measurements obtained using a Gaussian sampling matrix. Note that reconstructing MR images using the CelebA AR and vice versa results in a significant performance degradation.

In this paper, we propose a parameter-efficient method to adapt pretrained networks to multiple domains, measurement models, and noise levels with little to no drop in performance. In particular, we propose a domain-specific modulation of network weights using low-rank (or rank-one) factors.

Manuscript received 9 October 2023; revised 30 January 2024; accepted 27 February 2024. Date of publication 13 March 2024; date of current version 10 April 2024. This work was supported by the NSF CAREER awards under Grant CCF-2043134 and Grant CCF-2046293. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Mariya Doneva. *(Corresponding author: M. Salman Asif.)*

Nebiyou Yismaw and M. Salman Asif are with the University of California Riverside, Riverside, CA 92521 USA (e-mail: nyism001@ucr.edu; sasif@ucr.edu).

Ulugbek S. Kamilov is with the Washington University in St. Louis, Saint Louis, MO 63130-4899 USA (e-mail: kamilov@wustl.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/TCI.2024.3377101, provided by the authors.

Digital Object Identifier 10.1109/TCI.2024.3377101

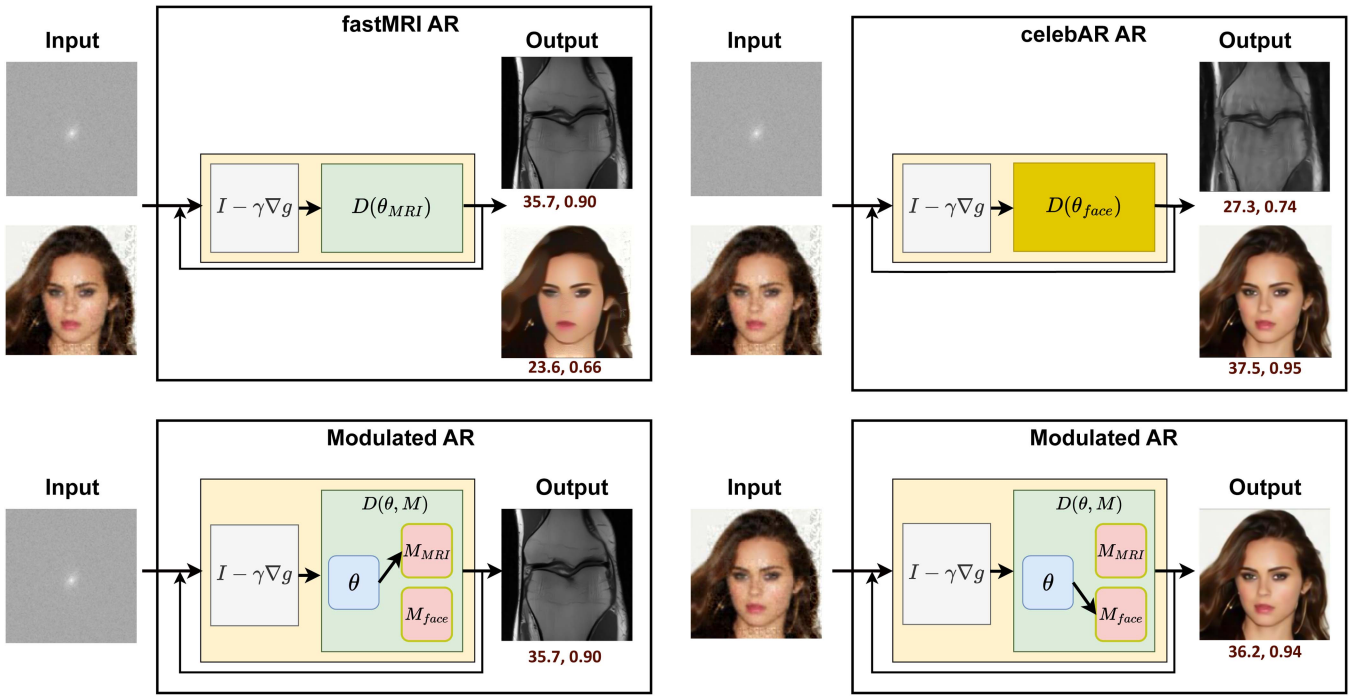

Fig. 1. Artifact removal (AR) networks trained on MRI scans (fastMRI AR) and face images (celebA AR) suffer from performance degradation under domain shifts, resulting in poor reconstruction quality (as indicated by PSNR and SSIM values under each image). Our proposed network (Modulated AR) adapts fastMRI AR for face image reconstruction by learning rank-one factors (modulations). The network stores shared and domain-specific modulations separately. During inference, it applies the correct modulation according to the specified domain. Our proposed network retains the performance of fastMRI AR on MR images and achieves competitive reconstruction quality with celebA AR on face images.

Given a single deep unrolled network, we learn a compact set of modulation parameters for each domain/measurement/noise setting in a supervised manner. At inference time, we apply the learned modulations to adapt the network weights to the specific target problem. In the remainder of the paper, we use the term domain shift and domain adaptation to refer to changes in data/measurement/noise distributions. We present a set of experiments to demonstrate the effectiveness of our method in adapting the deep unrolled network for shifts in data distribution/domain ($\mathbf{x}$), forward models ($\mathbf{A}$), and noise levels (η). The modulated AR in Fig. 1 shows an application of our method, where we adapt a pretrained fastMRI AR to celebA. It applies the learned modulations when recovering celebA images and will use the pretrained weights when reconstructing MRI scans. This network recovers images that qualitatively and quantitatively resemble results of the networks trained for the correct domains. The number of parameters needed to adapt the pretrained network is less than 0.5% of the parameters in the pretrained network.

Our method can be viewed as an example of domain adaptation or domain expansion technique, where we update a network trained for a source domain to perform well on several target domains. Fine-tuning pre-trained networks is a widely used method for domain adaptation but suffers from catastrophic forgetting [11] and requires a large number of parameters for every new domain [12]. Several parameter efficient domain adaptation techniques have been proposed in [13], [14]. Our method resembles some of these methods in spirit and separates the network into shared and domain-specific modules. By limiting the number of parameters for the domain-specific modules, our method provides a parameter-efficient method to learn multiple tasks and domains. Furthermore, conditional computation is efficient during training and inference compared to independent networks [15].

Contributions: We summarize the contributions of this paper as follows.

- We proposed a simple parameter-efficient domain expansion technique to modulate weights of a pretrained network with rank-one factors. Our method expands the domain of the networks and adapts to a variety of data/model shifts that arise in inverse problems.
- Our method requires a small number of domain-specific parameters (less than 0.5% of a single network) that can be stored separately from the shared network weights. This enables the network to continuously adapt to new domains without forgetting previous knowledge; therefore, we call it domain expansion.
- We present a detailed set of experiments that analyze the effects of domain, forward model, and noise-level shifts in natural and medical image recovery problems using deep unrolled methods.

II. Related Work

A. Deep Networks for Inverse Problems

Generative models, which learn to map a low-dimensional code into an image, have become increasingly prominent in various inverse problems. Following [16], several methods have

successfully applied generative networks as priors when solving inverse problems including MRI compressed sensing [17], super-resolution [18], blind image deconvolution [19], and phase retrieval [20], [21].

End-to-end methods in imaging inverse problems are typically trained in a supervised manner using training data. In this training process, the model learns to map input data, often representing measurements or corrupted images, to their corresponding ground truth or a high-quality representation. AUTOMAP [22] proposed a framework that learns a direct mapping from the measurement space to the image space using a set of training data. Noise2Noise [23] proposed a method that learns to directly map corrupted images to clean images. The method was applied to MRI measurements captured under different acquisition setups. Other approaches such as [24], [25] use end-to-end networks to estimate artifact free signals from initial states.

Plug-and-play (PnP) methods are at the intersection of data driven and model based methods that alternatively minimize data consistency and regularization terms. PnP-ADMM [1] was the first plug-and-play iterative algorithm that used pre-trained denoisers as priors. This method is based on the ADMM algorithm [2]. PnP-FISTA [26] is a PnP variant that replaces the proximal operator [2] of the data fidelity with the gradient. These methods have been applied to solve inverse problems [27], [28]

Deep unrolled networks learn the denoiser network in PnP algorithms in a supervised manner [7], [8], [29]. These methods truncate the PnP algorithm for a fixed number of iterations and share the same network through the iterations. They perform updates using the reconstruction output of the final iteration. ISTA-Net [29] introduced a learnable deep network designed to perform ISTA [30] updates for compressive sensing reconstruction tasks involving natural images and MRI. A deep unrolled artifact removal (AR) network that performs PnP iterations was proposed in [8]. The proposed method demonstrated remarkable recovery performance across various image reconstruction tasks. Overall, deep unrolled methods show remarkable results in several inverse problems such as super-resolution [31], image restoration [32], MRI [33] and CT [34] reconstruction.

B. Domain Expansion and Adaptation

Developing a single network that can handle multiple domains as well as adapt to new target domains has been an active area of research. Deep neural networks can learn transferable features and fine-tuning to a new dataset improves generalization performance [35], [36]. Despite its success, fine-tuning a network or parts of it force the network to lose previously learned domain or task, which requires storing multiple networks per domain and task. Parameter-efficient fine-tuning methods [37], [38], [39] propose networks that can achieve competitive performance to fully-tuned networks while requiring few number of additional parameters. Adapter-based techniques that learn efficient modules have been proposed in [40], [41], [42]. These modules are added to a pretrained network and enable it to adapt to new tasks.

Domain specific sub-network selection using binary masks was proposed in [12], [43]. Supsup [43] starts from a fixed base network that is randomly initialized and finds a sub-network that can perform well on a specific task. It learns task-specific binary masks sequentially without interference. The binary mask has the same number of parameters as the network weights and is applied using an element-wise product. In scenarios where task identity is given during training and inference, the binary masks are the only learned parameters during training time and will be applied to the base network parameters during test time.

A continual learning technique that utilizes modular task relatedness for sequential task learning was proposed in [44]. The proposed method was successfully applied to a rehearsal-based continual learning method. Such methods, however, require a replay buffer, which is a subset of training samples from previous tasks. A modular-network for continuous task adaptation that does not require replay buffers was proposed in [45]. Up on arrival of a new task/domain, the method creates trainable modules at every layer and finds the optimal way to add them to a frozen backbone network. These added modules are required to match the base-network in terms of parameters. After training, modules that are not part of the optimal path way will be discarded. This method is computationally demanding and parameter inefficient. Later, we will show that modules with significantly fewer parameters compared to the base-network modules can perform successful task/domain adaptation.

Tuning specific layers such as the BatchNorm [46], the final classification head [47], and LayerNorm [48] are proven to be effective adaptation techniques. A related approach that scales and shifts features to achieve the performance of full-tuning was proposed in [49]. A technique that parameterizes a network into learnable shared and task-specific modules, enabling a single network to adapt to various settings, was proposed in [14], [50]. RCM [14] proposed to reparametrize a standard convolution layer into a shared module that is non-trainable and a trainable task-specific module (modulator). The number of trainable parameters in the modulator is a fraction of the parameters of the shared layers. The proposed convolution operation is implemented as a stack of two 2D convolutional layers. The first stack refers to convolution with the shared modules. The second stack, which uses task-specific modulators, performs a 1×1 convolution by keeping the number of output feature channels fixed. Since the convolution filter size of these modulators is one, the number of trainable parameters in this stack is significantly less than the parameters in the first (shared) stack. Hyperdomain Networks [13] use modulated convolution to adapt generator networks to new domains. This approach introduces an intermediate domain modulation step to the original convolution modulation and demodulation operations proposed in StyleGAN2 [51]. The Hyperdomain modulation operation utilizes a domain-specific vector per layer to scale each input feature map independently. At each convolution layer, the number of trainable parameters is equal to the number of input features.

An adaptation method for shifts in domain and forward-models when solving inverse problems was proposed in [52]. The method proposes a fine-tuning and regularization technique adopted from RED [53]. Domain-specific batch normalization layers were proposed in [54] for a segmentation network that can handle brain MR scans across different scanners and protocols.

Unlike R&R [53], the method proposed in [54] can adapt to new domains without forgetting previous domains. Several test-time adaptation techniques have been proposed to close performance gaps resulting from domain shifts [55], [56], [57]. While many of these methods are proposed for purely data drive approaches, we focus on methods that fuse data-driven and model based techniques. In addition, our aim is to find parameter efficient domain adaptation techniques without introducing catastrophic forgetting.

III. Methods

In this section, we present details of our proposed domain expansion method for deep unrolling-based reconstruction. We first briefly discuss deep unrolled networks (readers may refer to [58] for further details). Then we discuss how we adapt the network weights using rank-one factors to perform domain expansion/adaptation.

A. Deep Unrolled Network

A deep unrolled network in its simplest form represents a fixed number of iterations for solving the optimization problem in (2). Plug and play (PnP) methods based on accelerated proximal gradients [1], [58], [59] offer a flexible and efficient framework for solving such problems. Key steps of PnP with a deep denoiser at iteration k can be described as follows.

$$\mathbf{z}^k = \mathbf{x}^{k-1} - \gamma \nabla g(\mathbf{x}^{k-1}) \tag{3}$$

$$\mathbf{s}^k = \mathcal{D}(\mathbf{z}^k; \theta) \tag{4}$$

$$\mathbf{x}^k = \mathbf{s}^k + \beta_k(\mathbf{s}^k - \mathbf{s}^{k-1}), \tag{5}$$

where γ is the step size, superscript $k = 1, \ldots, K$ denotes iteration number, $\nabla g(\cdot)$ denotes gradient of data fidelity with respect to $\mathbf{x}$, $\mathcal{D}(\cdot; \theta)$ denotes a denoiser or artifact removal network with weights θ, $\beta_k = (q_{k-1} - 1)/q_k$, and $q_k = (1/2)(1 + \sqrt{1 + 4q_{k-1}^2})$. We can initialize the estimate as $\mathbf{x}^0 = \mathbf{A}^H\mathbf{y}$, where $\mathbf{A}^H$ denotes Hermitian transpose of the measurement operator. Similar to [8], we implement $\mathcal{D}$ as an artifact removal network: $\mathcal{D}(\mathbf{x}; \theta) = \mathbf{x} - \mathbf{f}(\mathbf{x}; \theta)$, where $\mathbf{f}$ is a DnCNN-based residual network [60].

We can view each iteration of PnP as one layer of the unrolled network with predefined parameters. The output of an unrolled network with denoiser $\mathcal{D}(\cdot, \theta)$ and K iterations can be denoted as $\mathbf{x}^K(\theta)$. Since all operations are differentiable, we can further improve the performance by minimizing the reconstruction error on some training images with respect to θ. We can define such an optimization problem as

$$\min_{\theta} \sum_{\mathbf{x} \in \mathcal{X}} \mathcal{L}(\mathbf{x}, \mathbf{x}^K(\theta)), \tag{6}$$

where $\mathcal{X}$ denotes the set of training images.

B. Factorized Network Adaptation

Our method primarily adapts the prior in the unrolled network using domain/task-specific rank-one factors as the data, measurement, or noise distribution changes. We start with a pretrained network $\mathcal{D}(\cdot; \theta)$ with parameters θ. Then we learn domain-specific modulations denoted as $\{M_d\}_{d=1}^D$ for D domains. Each M_d represents a set of domain-specific modulation parameters that we use to adapt base network parameters to $\theta \odot M_d$, where $\odot$ represents element-wise multiplication. In order for this multiplication to be defined, we require θ and M_d to have identical number of elements. In practice, we do not create a new set of modulated weights; instead we keep the M_d and θ separate. This allows us to fix the base network and adapt to multiple new domains without forgetting previous domains. We represent the domain-specific network for dth domain as $\mathcal{D}(\cdot, \theta, M_d)$ and the output of the unrolled network as $\mathbf{x}^K(\theta, M_d)$. To learn the modulation parameters for dth domain, we keep θ unchanged and solve the following optimization problem for M_i:

$$\min_{M_d} \sum_{\mathbf{x} \in \mathcal{X}_d} \mathcal{L}(\mathbf{x}, \mathbf{x}^K(\theta, M_d)), \tag{7}$$

where $\mathcal{X}_d$ denotes the set of training images for the dth domain.

Even though we do not explicitly discuss measurement operator $\mathbf{A}$ and noise η in the unrolled network, any mismatch between training and test time settings of domain, measurements, and noise can cause performance degradation. We can consider any variation in data, measurements, or noise as a new domain and use the same procedure described above to learn the domain-specific modulations.

Rank-one factorization: Inspired by [37], [61], we assume the intrinsic dimension of the objective in (7) is small. We parameterize M_d such that its trainable parameters remain significantly smaller than the number of parameters in the base network.

To achieve the goal of parameter efficiency, we represent modulation weights for each layer as a rank-one tensor. Let us assume lth convolution layer has weights W^l with kernels of size

Algorithm 1: Factorized Network Adaptation.

Input: Training images $\mathbf{x} \in \mathcal{X}_d$ with measurements $\mathbf{y}$, and operator $\mathbf{A}$ for domain indicator d
Base network parameters θ, $\{\beta_k\}_{k \geq 0}$, γ, α
Output: Domain-specific M_d

$M_d \leftarrow$ `initialModulation`(d)
repeat
  for every $\mathbf{x} \in \mathcal{X}_d$ and $\mathbf{y}$
  initialize $\mathbf{x}^0 \leftarrow \mathbf{A}^H\mathbf{y}$
  for $k \in \{1, \ldots, K\}$ **do**
    $\mathbf{z}^k \leftarrow \mathbf{x}^{k-1} - \gamma \nabla g(\mathbf{x}^{k-1})$
    $\mathbf{s}^k \leftarrow \mathcal{D}(\mathbf{z}^k; \theta, M_d)$
    $\mathbf{x}^k \leftarrow \mathbf{s}^k + \beta_k(\mathbf{s}^k - \mathbf{s}^{k-1})$
  end for
  Calculate loss for all training samples in a minibatch and compute gradient w.r.t. M_d;
  $M_d \leftarrow M_d - \alpha \nabla_{M_d} \sum_{\mathbf{x} \in \mathcal{X}_d} \mathcal{L}(\mathbf{x}^K, \mathbf{x})$;
until Convergence of M_d
return M_d

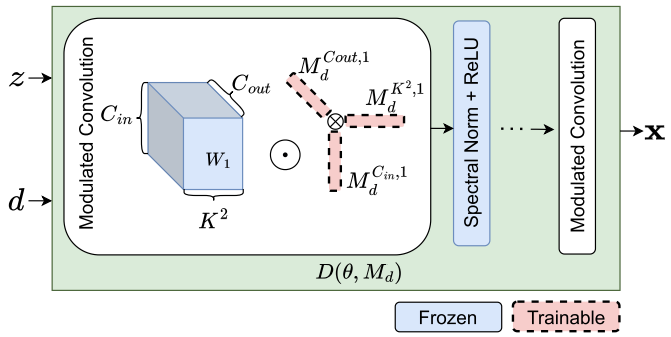

Fig. 2. Overview of our factorized network that uses modulated convolutions for domain adaptation. Our network follows the DNCNN [60] architecture that leverages modulated convolution for domain adaptation. After trained on the source domain, the network learns low-rank modulations for each domain while keeping the base network parameters frozen. Using a domain identifier, the network selects the appropriate low-rank factors during inference and applies them to the pretrained network through element-wise multiplication.

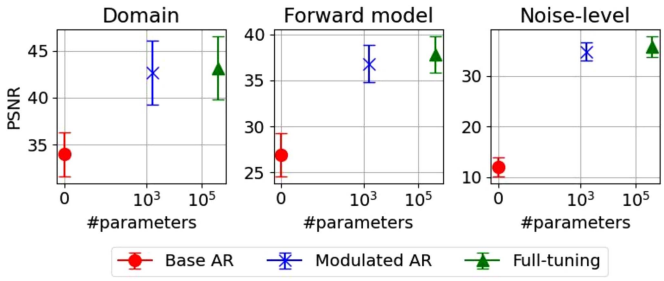

Fig. 3. Comparison of our modulated AR, fully-tuned AR, and the Base AR networks in terms of accuracy and number of additional parameters they require. Base AR requires no additional parameter and provides worst performance. Fully-tune AR provides best performance using a large number of parameters. Our proposed method, Modulated AR, shows performance comparable to Fully-tuned AR with a fraction of additional parameters.

$k \times k$ with C_{in} input and C_{out} output channels. We represent the modulation weights for dth domain and lth layer as an outer product of four vectors as

$$M_d^l = M_d^{1,l} \otimes M_d^{2,l} \otimes M_d^{3,l} \otimes M_d^{4,l}, \tag{8}$$

where $M_d^{1,l} \in \mathbb{R}^k, M_d^{2,l} \in \mathbb{R}^k, M_d^{3,l} \in \mathbb{R}^{C_{in}}, M_d^{4,l} \in \mathbb{R}^{C_{out}}$. Thus, we need $k + k + C_{in} + C_{out}$ parameters to adapt a layer with $k^2 C_{in} C_{out}$ parameters. We apply the rank-one factorization and modulation on the convolution layers as follows. For an input U with C_{in} channels, we can represent ith output channel of the convolution layer as

$$V(:,:,i) = \sum_{j=1}^{C_{in}} W^l(:,:,j,i) * U(:,:,j), \tag{9}$$

where $*$ represents 2D convolution. Modulated weights for domain d and layer l can be represented as $W_d^l = W^l \odot M_d^l$. We can represent the convolution operation as

$$V(:,:,i) = M_d^{4,l}(i) \left[\sum_{j=1}^{C_{in}} \widetilde{W}^l(:,:,j,i) * \widetilde{U}(:,:,j) \right], \tag{10}$$

where $\widetilde{W}^l(:,:,j,i) = W^l(:,:,j,i) \odot (M_d^{1,l} \otimes M_d^{2,l})$ represents a modulated version of (j,i) slice of weight tensor and $\widetilde{U}(:,:,j) = U(:,:,j) \odot M_d^{3,l}$ represents a modulated version of the jth input channel. In summary, even though we represent modulation weights as a rank-one tensor, we do not need to modulate the weights of the base network. We can implement the same procedure by modulating input channels, 2D filters, and output channels.

Fig. 2 illustrates how our proposed unrolled multi-domain network applies low-rank factors to the pretrained network. We implement (10) by first combining the low-rank factors as formulated in (8) and applying them to the base convolution weights using an element-wise product. We then use these updated weights to perform regular convolution during the forward pass. When performing backward propagation, we compute gradients with respect to the low-rank factors and update them while keeping the remaining parameters of the network frozen.

A pseudocode for factorized adaptation with the unrolled network is provided in Algorithm 1. The algorithm begins by initializing domain-specific modulations using an outer product of the low-rank factors. These low-rank factors are real-valued and randomly initialized. After computing the initial estimates $\mathbf{x}^0$, we perform K unrolled iterations containing data-consistency and artificial-removal updates. Finally, we use the output from the last iteration, $\mathbf{x}^K$, to compute the reconstruction loss. This loss is used to compute gradients with respect to the low-rank factors and to perform updates. Further details and hyper-parameter setups are provided in the supplementary material.

IV. Experiments and Results

We performed a number of experiments to analyse the effects of shifts in different parts of the inverse problem in (1). The shifts can occur in the data distribution $\mathbf{x}$, the forward model $\mathbf{A}$, and the measurement noise η. We test our proposed adaptation technique for all these shifts. In all our experiments, we start with a fixed base network, which we refer to as Base AR, and learn domain-specific rank-one modulations. Base AR is trained to reconstruct MR images from $4\times$ radially sub-sampled simulated Fourier measurements without any additional measurement noise.

Base AR uses spectral normalization proposed in [62] along with the ReLU activation functions. We implement our AR network using a 12-layer DnCNN [60] network. We provide details on training and dataset preparation, along with the hyperparameters used in our experiments, in the supplementary material.

A. Parameter Efficiency for Adaptation

Fig. 3 compares the performance of a base network, full training, and our proposed modulation-based adaptation for shifts in data distribution/domain, forward model, and noise level. Base network does not require any additional parameter for different domains, but it provides the worst performance.

TABLE I
AVERAGE PSNR OF ARTIFACT REMOVAL (AR) NETWORKS UNDER DOMAIN SHIFT

Test domain	AR Trained on MRI	AR Trained on CelebA	AR Trained on CT	Modulated AR (Ours)
MRI	**40.93**	39.22	37.14	**40.93**
CelebA	40.34	**44.29**	35.44	42.97
CT	37.68	38.56	41.97	**42.25**
Avg	39.65	40.69	38.18	**42.05**

ARs trained for specific domain (MRI, CelebA, and CT ARs) do not perform well on out-of-domain samples. In contrast, our Modulated AR network, that applies learned modulations for each target domain, has the best average performance across all domains.

Full training learns a new network for every domain/distribution shifts and provides the best performance, but at the expense of a large number of parameters per domain. Our proposed network adaptation approach requires a small number of parameters (nearly 1.6 K additional parameters) and achieves performance close to full training method. The additional parameters are unique for each domain and are stored separately from the base network. In this manner, the pre-trained model can be adapted to learn new domains while retaining previously learned knowledge.

B. Domain Shift

For experiments with domain/data distribution shifts in $\mathbf{x}$, we consider natural image, MRI, and CT scans. We use CelebA dataset [63] for natural images, NYU fastMRI dataset for [64] knee MRI scans, and a subset of TCGA-LUAD dataset [65] for CT scans. In this experiment, we independently train artifact removal (AR) networks on each dataset to reconstruct the ground truth images from radially sub-sampled simulated Fourier measurements. We refer to these networks as single domain (domain specific) AR networks. To asses the effects of domain shift, we test each of these networks on samples obtained from datasets that were not used to train them. The first three columns of Table I show the performance of the single domain AR networks. We present the reconstruction PSNR of these AR networks evaluated under the domain shifts. The last column shows the performance of our modulated network that uses weights of the Base AR trained on MRI and learned modulations for each target domain. Quantitatively we observe that performance drops as domains change (off-diagonal entries in columns 2,3,4). Our proposed method for modulated AR offers best overall performance. Fig. 4 shows example reconstructed images for our domain shift experiments. Our modulated network effectively removes artifacts introduced by fastMRI AR and CT AR on CelebA images.

Comparison with existing domain adaptation methods: We compare our proposed approach with the following related domain adaptation techniques: Supsup [43], RCM [14], Hyperdomain Modulation [13], and Full-tuning. We evaluate these methods using the same training and testing procedure as our proposed approach. Supsup [43] learns binary masks to find domain specific sub-networks. RCM [14] reparameterizes convolutions using domain-specific feature transformations. Hyperdomain [13] learns domain-specific modulation for input channel of every convolution operation. Full-tuning retrains the entire network for each target domain and is considered as an upper-bound. Table II shows comparison of these methods and our proposed method outperforms other adaptation techniques. In addition, it requires fewer additional parameters than Supsup [43] and RCM [14].

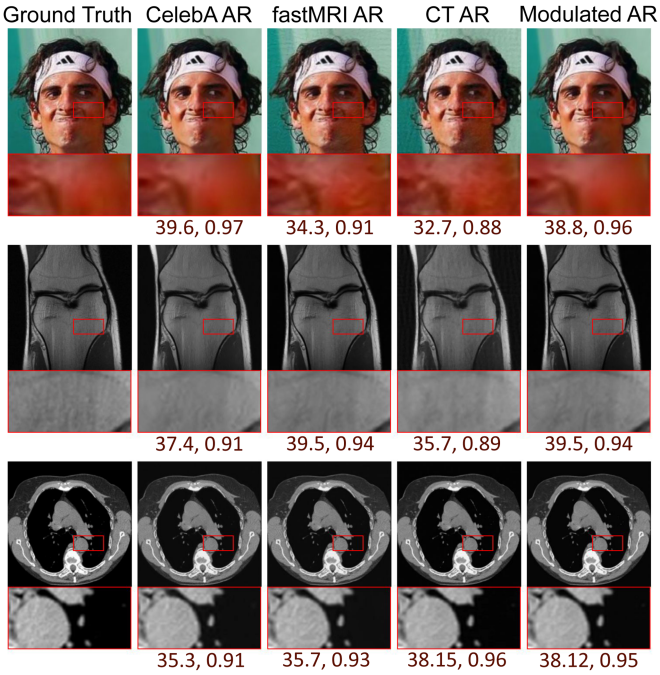

Fig. 4. We present sample ground truth images in the first column and reconstruction of these images using three AR networks trained on Face, MR, and CT images in the subsequent three columns. Our modulated AR, shown in the last column effectively removes artifacts from the face images and improves the reconstruction performance on the CT images.

C. Forward Model Shifts

To evaluate the performance with shifts in the forward model, $\mathbf{A}$, we consider sampling types, ratio, and patterns as domains that can induce shifts. The sampling type can be either Fourier or Gaussian sampling. In the case of Fourier sampling, we can have Cartesian, Radial, Gaussian, or Spiral patterns. The sampling ratio determines the rate at which measurements are captured. We consider reconstruction from $4\times$, $8\times$ and $10\times$ under-sampled measurements. We will now examine the effects of each of these shifts and utilize our proposed method to adapt our Base AR.

TABLE II
COMPARISON OF OUR METHOD WITH EXISTING DOMAIN ADAPTATION TECHNIQUES

Target domain	Full-tuning	Supsup	RCM	Hyperdomain	Modulated AR (Ours)
	407k	407k	50.6k	0.7k	1.6k
CelebA	**44.29**	42.49	43.88	42.73	42.97
CT	41.97	40.57	40.99	41.57	**42.25**
Avg	**43.13**	41.53	42.49	42.15	42.61

We have highlighted the best-performing method in boldface And second best with an underscore. Additionally, we provide the count of additional parameters required by each method. Our Modulated ar outperforms other methods and is comparable to full tuning.

TABLE III
SAMPLING PATTERN SHIFT ADAPTATION RESULTS

Test pattern	Radial AR	Cartesian AR	Gaussian AR	Spiral AR	Modulated AR (Ours)
Radial	**40.93**	37.75	40.55	40.83	**40.93**
Cartesian	29.74	**39.21**	28.39	29.12	37.10
Gaussian	41.91	40.19	42.05	42.04	**42.10**
Spiral	41.24	39.57	41.26	41.36	**41.38**
Avg	38.46	39.18	38.06	38.34	**40.38**

Our modulated ar achieves competitive in-domain performance to ars trained on specific Patterns. Moreover, it shows an overall superior performance across all patterns.

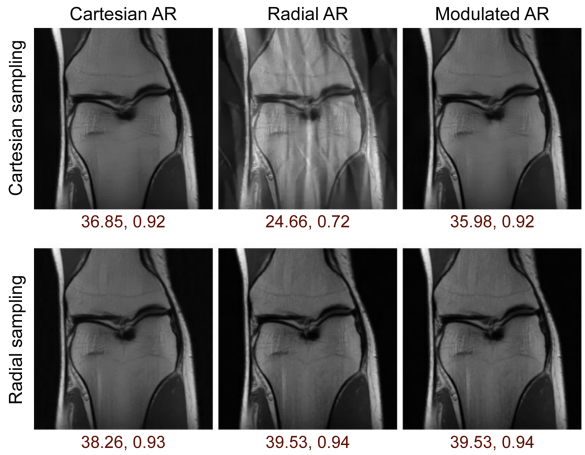

Fig. 5. Reconstruction results under sampling pattern shifts. AR trained on radial pattern performs poorly when tested on Cartesian sampled patterns. Our Modulated AR applies low-rank modulations to adapt Radial AR to Cartesian samples.

Sampling pattern shifts: Table III shows the performance of AR networks trained on single sampling patterns when tested on all available patterns in the first four columns. The last column shows the performance of our modulated AR. We observed a significant performance drop when our Base AR was tested on samples from Cartesian samples. This drop is also evident qualitatively in Fig. 5, where visible artifacts appear in the output. Our modulated AR successfully eliminates these artifacts and bridges the performance gap. Moreover, our method provides overall superior performance compared to networks trained for individual patterns.

Sampling ratio shifts: We compared the performance of different AR networks trained on three sampling ratios and

TABLE IV
SAMPLING RATIO SHIFT ADAPTATION RESULTS

Test ratio	4x AR	8x AR	10x AR	Modulated AR (Ours)
4x	**40.93**	40.23	39.61	**40.93**
8x	34.98	37.13	37.05	**37.32**
10x	31.00	33.63	**35.34**	34.73
Avg	35.64	37.00	37.33	**37.66**

presented the results in Table IV. The $4\times$ AR network exhibits poor performance when tested with $8\times$ and $10\times$ radially subsampled measurements. Additionally, the AR network trained on the $8\times$ ratio did not perform well with $10\times$ ratio. To address this, we applied our modulation technique to adapt the Base AR model to $8\times$ and $10\times$ sampling ratios. On average, the modulated network outperforms AR networks trained on specific sampling ratios. Fig. 6 illustrates the reconstruction results of the networks trained at various sampling ratios, including our modulated network. On average, the modulated network outperforms AR networks trained on specific sampling ratios.

Comparison with existing domain adaptation methods: We now compare our method with some of the existing domain adaptation techniques under the forward model shifts discussed above. We report the average PSNR along with the number of trainable parameters with in each method in Table V. Our proposed method outperforms all domain adaptation techniques and is only one dB less than full-tuning, which requires significantly larger number of parameters.

D. Noise-Level Shifts

Noise-level shifts can also cause significant performance degradation in AR networks. We model the noise as an additive Gaussian noise $\eta \sim \mathcal{N}(0, \sigma^2)$ and analyze the effects of

TABLE V
COMPARISON OF DOMAIN ADAPTATION METHODS UNDER FORWARD MODEL SHIFTS

Sampling shifts	Full-tuning	Supsup	RCM	Hyperdomain	Modulated AR (Ours)
	407k	407k	50.6k	0.7k	1.6k
Radial to Cartesian	**39.21**	36.37	36.52	36.57	<u>37.10</u>
4x to 10x	**35.34**	33.62	34.49	34.56	<u>34.73</u>
Fourier to Gaussian	**38.59**	36.31	38.49	38.45	<u>38.55</u>
Avg	**37.71**	35.43	36.50	36.53	<u>36.79</u>

Our proposed method achieves competitive performance to Full-tuning with significantly fewer parameters. It outperforms related domain adaptation methods in terms of performance.

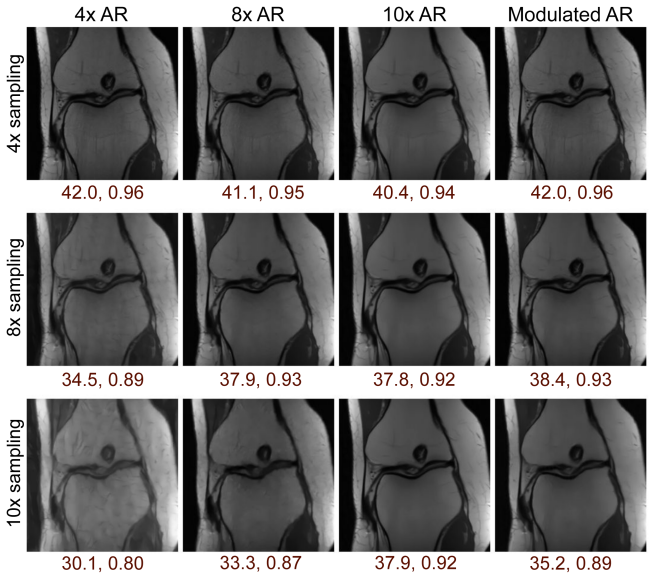

Fig. 6. Examples of image reconstruction under sampling ratio shifts. Our Modulated AR shows an average superior performance when compared to the 4×, 8×, and 10× AR networks.

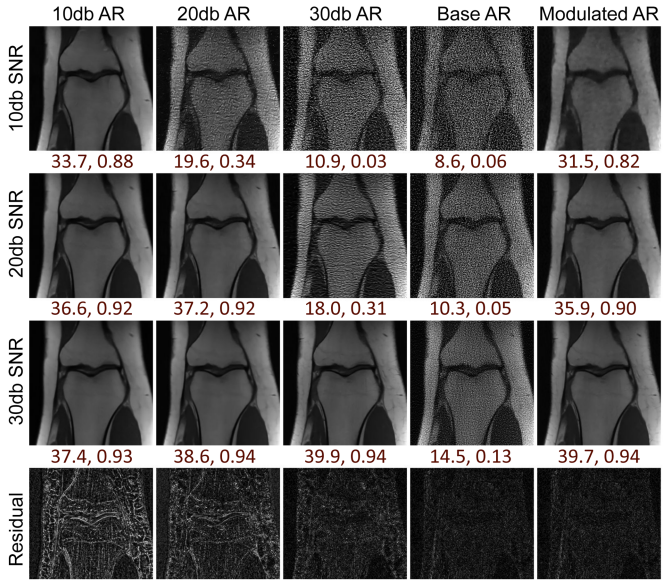

Fig. 7. Visual results of models trained at specific noise levels and our modulated network under measurement level. The last row shows the 20×amplified residual of the reconstructed image under no measurement noise.

different noise levels on the performance. Fig. 7 shows sample reconstructed images with the Base AR, our Modulated AR, and AR networks trained for 10, 20, and 30 dB SNR. We observed that the Base AR is unable to reconstruct the MRI scans from the noisy measurements. This is also shown quantitatively in Table VI, where the performance of the Base AR is severely degraded in the presence of noise. The AR network trained on 10 dB SNR performs well on higher noise settings but fails to recover fine details when tested with noise-free or low noise measurements. The last row of Fig. 7 shows the 20× amplified reconstruction residual of each model when reconstructing noise-free measurements. From this row, we can infer that AR networks trained on higher noise-levels fail to recover fine details when tested with lower noise-levels. To the contrary, our Modulated AR has the ability to reconstruct fine details when the measurement noise is low and maintains comparable performance to noise-specific ARs as the noise level increases.

Comparison with existing domain adaptation methods: Table VII reports comparison of our proposed method with related domain adaptation techniques. Although Full-tuning and Supsup [43] show slight performance improvement (less than 1 dB), they require a significant number of trainable parameters. Furthermore, Full-tuning does not have the ability to retrain previously learned knowledge. Our method achieves competitive performance to RCM [14] while requiring a fraction of the additional trainable parameters.

V. LIMITATIONS

While our proposed method is able to continuously adapt to new domains, it requires domain selectors/identifiers during inference to apply the correct modulations. In some cases, this is not a major limitation since we can partially infer the domain from the available measurements or context. In principle, we can parameterize the network modulations as a function of the input and construct a multi-domain network that can infer the domain without the need for explicit identifiers. Another limitation of our current method and experiments is the incremental adaptation to target domains. We start from a fixed base network and subsequently adapt it to multiple domains independently. We can further improve the efficiency of our method by adapting the network to multiple domains jointly. Achieving rapid and generalized multi-domain adaptation is feasible following meta-learning techniques as outlined in [66]. We believe that these limitations will serve as inspiration for several future studies.

TABLE VI
NOISE LEVEL SHIFT ADAPTATION RESULTS

Test SNR	10db AR	20db AR	30db AR	Base AR	Modulated AR (Ours)
10db	**33.37**	23.86	12.56	9.18	31.40
20db	35.39	**35.82**	22.86	11.20	35.09
30db	35.97	36.84	**38.04**	15.90	37.91
No noise	36.10	36.97	38.97	**40.93**	**40.93**
Avg	35.21	33.30	28.11	19.30	**36.33**

Modulated ar that learns low-rank factors for each noise-level outperforms networks trained For a specifc noise.

TABLE VII
COMPARISON OF VARIOUS DOMAIN ADAPTATION METHODS UNDER NOISE LEVEL SHIFTS

Test SNR	Full-tuning	Supsup	RCM	Hyperdomain modulation	Modulated AR (Ours)
	407k	407k	50.6k	0.7k	1.6k
10db	**33.37**	31.94	32.80	29.11	31.40
20db	**35.82**	35.53	35.70	34.32	35.09
30db	38.04	37.75	**38.08**	37.58	37.91
Avg	**35.74**	35.07	35.53	33.67	34.80

Our proposed method can achieve competitive performance To rcm and full-tuning while requiring significantly fewer number of additional parameters.

VI. CONCLUSION

We proposed a simple and parameter-efficient method for domain adaptation and expansion. Our method uses a fixed base network and learns separate (domain-specific) rank-one modulation parameters. This capability allows our method to continually learn new domains while retaining previously acquired knowledge. We focused on shifts that arise in solving inverse problems for imaging, including shifts in data distribution, forward model, and noise level. We demonstrated the effectiveness of our approach in adapting to all these shifts.

REFERENCES

[1] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, "Plug-and-play priors for model based reconstruction," in *Proc. IEEE Glob. Conf. Signal Inf. Process.*, 2013, pp. 945–948.

[2] S. Boyd et al., "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[3] G. Ongie, A. Jalal, C. A. Metzler, R. G. Baraniuk, A. G. Dimakis, and R. Willett, "Deep learning techniques for inverse problems in imaging," *IEEE J. Sel. Areas Inf. Theory*, vol. 1, no. 1, pp. 39–56, May 2020.

[4] V. Antun, F. Renna, C. Poon, B. Adcock, and A. C. Hansen, "On instabilities of deep learning in image reconstruction and the potential costs of AI," *Proc. Nat. Acad. Sci.*, vol. 117, no. 48, pp. 30088–30095, 2020.

[5] N. M. Gottschling, V. Antun, B. Adcock, and A. C. Hansen, "The troublesome kernel: Why deep learning for inverse problems is typically unstable," 2020, *arXiv:2001.01258*.

[6] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proc. 27th Int. Conf. Int. Conf. Mach. Learn.*, 2010, pp. 399–406.

[7] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," *IEEE Signal Process. Mag.*, vol. 38, no. 2, pp. 18–44, Mar. 2021.

[8] J. Liu, M. S. Asif, B. Wohlberg, and U. S. Kamilov, "Recovery analysis for plug-and-play priors using the restricted eigenvalue condition," in *Adv. Neural Inf. Process. Syst.*, 2021, pp. 5921–5933.

[9] Y. Sun, B. Wohlberg, and U. S. Kamilov, "An online plug-and-play algorithm for regularized image reconstruction," *IEEE Trans. Comput. Imag.*, vol. 5, no. 3, pp. 395–408, Sep. 2019.

[10] M. Z. Darestani, A. S. Chaudhari, and R. Heckel, "Measuring robustness in deep learning based compressive sensing," in *Proc. 38th Int. Conf. Mach. Learn.*, 2021, pp. 2433–2444.

[11] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," *Psychol. Learn. Motivation*, 1989, vol. 24, pp. 109–165.

[12] A. Mallya, D. Davis, and S. Lazebnik, "Piggyback: Adapting a single network to multiple tasks by learning to mask weights," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 67–82.

[13] A. Alanov, V. Titov, and D. P. Vetrov, "Hyperdomainnet: Universal domain adaptation for generative adversarial networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, vol. 35, pp. 29414–29426.

[14] M. Kanakis, D. Bruggemann, S. Saha, S. Georgoulis, A. Obukhov, and L. V. Gool, "Reparameterizing convolutions for incremental multi-task learning without task interference," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 689–707.

[15] C. Riquelme et al., "Scaling vision with sparse mixture of experts," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, vol. 34, pp. 8583–8595.

[16] A. Bora, A. Jalal, E. Price, and A. G. Dimakis, "Compressed sensing using generative models," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 537–546.

[17] A. Jalal, M. Arvinte, G. Daras, E. Price, A. G. Dimakis, and J. Tamir, "Robust compressed sensing MRI with deep generative priors," in *Adv. Neural Inf. Process. Syst.*, 2021, pp. 14938–14954.

[18] S. Menon, A. Damian, S. Hu, N. Ravi, and C. Rudin, "Pulse: Self-supervised photo upsampling via latent space exploration of generative models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2437–2445.

[19] M. Asim, F. Shamshad, and A. Ahmed, "Blind image deconvolution using deep generative priors," *IEEE Trans. Comput. Imag.*, vol. 6, pp. 1493–1506, 2020.

[20] P. Hand, O. Leong, and V. Voroninski, "Phase retrieval under a generative prior," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, vol. 31, pp. 9154–9164.

[21] R. Hyder, V. Shah, C. Hegde, and M. S. Asif, "Alternating phase projected gradient descent with generative priors for solving compressive phase retrieval," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 7705–7709.

[22] B. Zhu, J. Z. Liu, S. F. Cauley, B. R. Rosen, and M. S. Rosen, "Image reconstruction by domain-transform manifold learning," *Nature*, vol. 555, no. 7697, pp. 487–492, 2018.
[23] J. Lehtinen et al., "Noise2Noise: Learning image restoration without clean data," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 2965–2974.
[24] A. Sriram et al., "End-to-end variational networks for accelerated MRI reconstruction," in *23rd Int. Conf., Med. Image Comput. Computer Assist. Interv.*, 2020, pp. 64–73.
[25] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser, "Deep convolutional neural network for inverse problems in imaging," *IEEE Trans. Image Process.*, vol. 26, no. 9, pp. 4509–4522, Sep. 2017.
[26] U. S. Kamilov, H. Mansour, and B. Wohlberg, "A plug-and-play priors approach for solving nonlinear imaging inverse problems," *IEEE Signal Process. Lett.*, vol. 24, no. 12, pp. 1872–1876, Dec. 2017.
[27] S. H. Chan, X. Wang, and O. A. Elgendy, "Plug-and-play ADMM for image restoration: Fixed-point convergence and applications," *IEEE Trans. Comput. Imag.*, vol. 3, no. 1, pp. 84–98, Mar. 2017.
[28] R. Ahmad et al., "Plug-and-play methods for magnetic resonance imaging: Using denoisers for image recovery," *IEEE Signal Process. Mag.*, vol. 37, no. 1, pp. 105–116, Jan. 2020.
[29] J. Zhang and B. Ghanem, "ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing," in *Proc. IEEE/CVF Conf. Computer Vis. Pattern Recognit.*, 2018, pp. 1828–1837.
[30] A. Beck and M. Teboulle, "Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems," *IEEE Trans. Image Process.*, vol. 18, no. 11, pp. 2419–2434, Nov. 2009.
[31] K. Zhang, L. V. Gool, and R. Timofte, "Deep unfolding network for image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 3217–3226.
[32] C. Mou, Q. Wang, and J. Zhang, "Deep generalized unfolding networks for image restoration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 17399–17410.
[33] Y. Jun, H. Shin, T. Eo, and D. Hwang, "Joint deep model-based mr image and coil sensitivity reconstruction network (joint-icnet) for fast MRI," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 5270–5279.
[34] D. Wu, K. Kim, and Q. Li, "Computationally efficient deep neural network for computed tomography image reconstruction," *Med. Phys.*, vol. 46, no. 11, pp. 4763–4776, 2019.
[35] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks ?," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, vol. 27, pp. 3320–3328.
[36] M. Long, Y. Cao, Z. Cao, J. Wang, and M. I. Jordan, "Transferable representation learning with deep adaptation networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 12, pp. 3071–3085, Dec. 2019.
[37] E. J. Hu et al., "LoRA: Low-rank adaptation of large language models," in *Int. Conf. Learn. Representations*, 2021.
[38] X. He, C. Li, P. Zhang, J. Yang, and X. E. Wang, "Parameter-efficient model adaptation for vision transformers," in *Proc. AAAI Conf. Artif. Intell.*, 2023, pp. 817–825.
[39] Y.-C. Liu, C.-Y. MA, J. Tian, Z. He, and Z. Kira, "Polyhistor: Parameter-efficient multi-task adaptation for dense vision tasks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 36889–36901.
[40] S.-A. Rebuffi, H. Bilen, and A. Vedaldi, "Efficient parametrization of multi-domain deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8119–8127.
[41] W.-H. Li, X. Liu, and H. Bilen, "Cross-domain few-shot learning with task-specific adapters," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 7161–7170.
[42] S. Chen et al., "Adaptformer: Adapting vision transformers for scalable visual recognition," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, vol. 35, pp. 16664–16678.
[43] M. Wortsman et al., "Supermasks in superposition," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 15173–15184.
[44] A. Shaker, F. Alesiani, and S. Yu, "Modular-relatedness for continual learning," in *Int. Symp. Intell. Data Anal.*, 2022, pp. 290–301.
[45] T. Veniat, L. Denoyer, and M. Ranzato, "Efficient continual learning with modular networks and task-driven priors," in *Proc. 9th Int. Conf. Learn. Representations*, 2021.
[46] J. Frankle, D. J. Schwab, and A. S. Morcos, "Training batchnorm and only batchnorm: On the expressive power of random features in CNNs," in *Int. Conf. Learn. Representations*, 2020.
[47] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9729–9738.
[48] S. Basu, D. Massiceti, S. X. Hu, and S. Feizi, "Strong baselines for parameter efficient few-shot fine-tuning," in *Proc. AAAI Conf. Artif. Intell.*, 2024, pp. 11024–11031.
[49] D. Lian, D. Zhou, J. Feng, and X. Wang, "Scaling & shifting your features: A new baseline for efficient model tuning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, vol. 35, pp. 109–123.
[50] A. Rosenfeld and J. K. Tsotsos, "Incremental learning through deep adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 3, pp. 651–663, Mar. 2020.
[51] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of StyleGaN," in *Proc. IEEE/CVF Conf. Comput. Vis. pattern Recognit.*, 2020, pp. 8110–8119.
[52] D. Gilton, G. Ongie, and R. Willett, "Model adaptation for inverse problems in imaging," *IEEE Trans. Comput. Imag.*, vol. 7, pp. 661–674, 2021.
[53] Y. Romano, M. Elad, and P. Milanfar, "The little engine that could: Regularization by denoising (red)," *SIAM J. Imag. Sci.*, vol. 10, no. 4, pp. 1804–1844, 2017.
[54] N. Karani, K. Chaitanya, C. Baumgartner, and E. Konukoglu, "A lifelong learning approach to brain mr segmentation across scanners and protocols," in *Int. Conf. Med. Image Comput. Comput.- Assist. Interv.*, 2018, pp. 476–484.
[55] M. Z. Darestani, J. Liu, and R. Heckel, "Test-time training can close the natural distribution shift performance gap in deep learning based compressed sensing," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 4754–4776.
[56] B. Song, L. Shen, and L. Xing, "PINER: Prior-informed implicit neural representation learning for test-time adaptation in sparse-view CT reconstruction," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2023, pp. 1928–1938.
[57] S. Goyal, M. Sun, A. Raghunathan, and J. Z. Kolter, "Test time adaptation via conjugate pseudo-labels," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 6204–6218.
[58] U. S. Kamilov, C. A. Bouman, G. T. Buzzard, and B. Wohlberg, "Plug-and-play methods for integrating physical and learned models in computational imaging: Theory, algorithms, and applications," *IEEE Signal Process. Mag.*, vol. 40, no. 1, pp. 85–97, Jan. 2023.
[59] N. Parikh et al., "Proximal algorithms," *Foundations Trends Optim.*, vol. 1, no. 3, pp. 127–239, 2014.
[60] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.
[61] C. Li, H. Farkhoor, R. Liu, and J. Yosinski, "Measuring the intrinsic dimension of objective landscapes," in *Proc. Int. Conf. Learn. Representations*, 2018.
[62] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in *Int. Conf. Learn. Representations*, 2018.
[63] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proc. Int. Conf. Comput. Vis.*, 2015, pp. 3730–3738.
[64] J. Zbontar et al., "fastMRI: An open dataset and benchmarks for accelerated MRI," 2018. [Online]. Available: https://github.com/facebookresearch/fastMRI
[65] K. Clark et al., "The cancer imaging archive (TCIA): Maintaining and operating a public information repository," *J. Digit. Imag.*, vol. 26, no. 6, pp. 1045–1057, Jul. 2013.
[66] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, vol. 70, pp. 1126–1135.

Nebiyou Yismaw (Graduate Student Member, IEEE) received the B.Sc. degree in electrical and computer engineering from Addis Ababa University, Addis Ababa, Ethiopia, 2017, and the M.Sc. degree in electrical and computer engineering from Carnegie Mellon University Africa, Kigali, Rwanda, in 2019. He is currently working toward the Ph.D. degree with the University of California, Riverside, Riverside, CA, USA, under the supervision of Prof. M. Salman Asif. His current research interests include robust computational imaging, inverse problems and computer vision.

Ulugbek S. Kamilov (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in communication systems and the Ph.D. degree in electrical engineering from the Swiss Federal Institute of Technology Lausanne, Lausanne, Switzerland, in 2011 and 2015, respectively. He is currently the Director of Computational Imaging Group, Washington University, St. Louis, St. Louis, MO, USA, where he is also an Associate Professor with the Departments of Electrical and Systems Engineering and Computer Science and Engineering. He is currently a Visiting Research Faculty with Google Research, CA, USA. From 2015 to 2017, he was a Research Scientist with Mitsubishi Electric Research Laboratories, Cambridge, MA, USA. His primary research interests include computational imaging, machine learning, and optimization. He is currently a Senior Member of the Editorial Board of *IEEE Signal Processing Magazine* and IEEE Signal Processing Society's Bioimaging and Signal Processing Technical Committee. He was an Associate Editor for IEEE TRANSACTIONS ON COMPUTATIONAL IMAGING and IEEE Signal Processing Society's Computational Imaging Technical Committee. He was the recipient of the NSF CAREER Award and IEEE Signal Processing Society's 2017 Best Paper Award. He was a Fellow among 55 early-career researchers in the USA selected in 2021 for the Scialog initiative on Advancing Bioimaging. His Ph.D. thesis was selected in 2016 as a finalist for the EPFL Doctorate Award. He was also the recipient of the Outstanding Teaching Award in 2023 from the Department of Electrical and Systems Engineering at Washington University.

M. Salman Asif (Senior Member, IEEE) received the B.Sc. degree from the University of Engineering and Technology, Lahore, Pakistan, and the M.S and Ph.D. degrees from the Georgia Institute of Technology, Atlanta, GA, USA. He is currently an Associate Professor with the University of California Riverside, Riverside, CA, USA. Prior to that, he was a Postdoctoral Researcher with Rice University, Houston, TX, USA, and a Senior Research Engineer with Samsung Research America, Dallas. His research interests include computational imaging, signal/image processing, computer vision, and machine learning. He was the recipient of the NSF CAREER Award, Google Faculty Award, Hershel M. Rich Outstanding Invention Award, and UC Regents Faculty Fellowship and Development Awards. He is an Associate Editor for IEEE TRANSACTIONS ON COMPUTATIONAL IMAGING and a member of IEEE Signal Processing Society's Computational Imaging Technical Committee.