Domain Expansion via Network Adaptation for Solving Inverse Problems

Nebiyou Yismaw, *Student Member, IEEE*, Ulugbek S. Kamilov, *Senior Member, IEEE*, 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 adapt pretrained networks to such shifts. We demonstrate the effectiveness of our method for a number of natural image, MRI, and CT reconstructions tasks under domain, measurement model, and noise-level shifts. Our experiments demonstrate that our method provides significantly better performance and parameter efficiency compared to existing 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:

$$\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + h_\theta(\mathbf{x}), \tag{2}$$

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]. For instance, signal is sparse or low-rank in some representation space or belongs to a manifold of natural images [3]–[6].

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 [7]. 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 [8], [9]. 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 [10]–[12] and plug-and-play (PnP) [1], [13] methods use artifact-removal or image denoising networks that are trained to map a noisy or corrupted estimate of an image onto a clean image manifold [7], [11], [12]. Despite recent success of such deep unrolled or PnP methods, they are also sensitive to shifts in the data distribution [14]. Figure 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 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 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. Given a single deep unrolled network, we learn a compact set of modulation parameters for each domain/measurement/noise setting, and adapt the weights of the network according to the specific problem at the inference time. 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 Figure 1 shows

This paper is partially based on work supported by the NSF CAREER awards under grants CCF-2043134 and CCF-2046293.

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

Ulugbek Kamilov is with Washington University in St. Louis (e-mail: kamilov@wustl.edu).

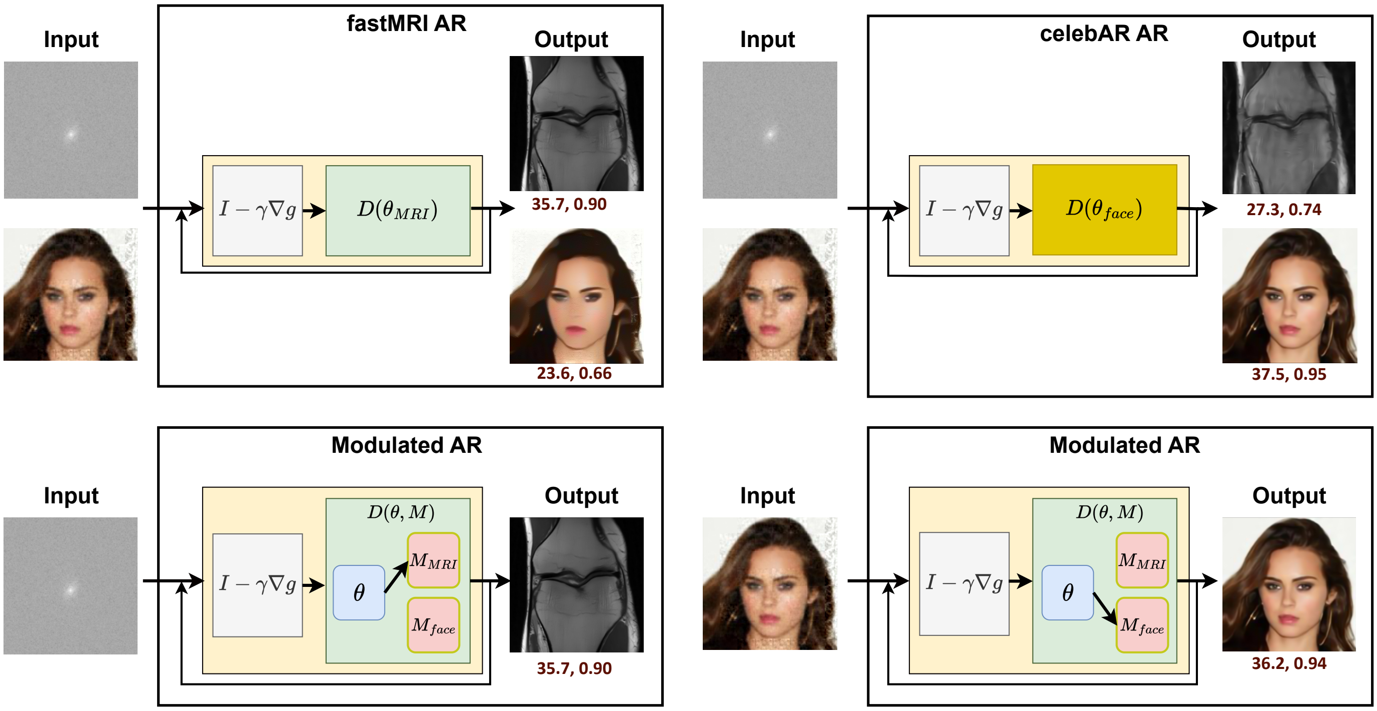

Fig. 1: Artifact removal (AR) networks trained on MRI scans (fastMRI AR) and face images (celebA AR) suffer from performance degradation under domain shifts, resulting in poor reconstruction quality (as indicated by PSNR and SSIM values under each image). Our proposed network (Modulated AR) adapts fastMRI 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.

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 [15] and requires a large number of parameters for every new domain [16]. Several parameter efficient domain adaptation techniques have been proposed in [17], [18]. 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 [19].

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. Inverse problems and structured priors.

Hand-crafted priors for inverse problems usually assume the signal is sparse in some transform domain. ℓ_1 norm has been widely used as a sparsity-promoting regularizer [20], [21]. Total variation (TV) minimization is used as a regularization approach in [22], [23] to solve denoising and deblurring problems. An iterative algorithm that minimizes the image total variation (TV) for CT reconstruction was proposed in [24]. These hand-crafted priors, however, have limited ability to represent the true underlying image and may lead to sub-optimal solutions [25].

B. Deep networks for inverse problems

Generative models learn to map a low-dimensional code into an image. Following [26], several methods have successfully applied generative networks as priors when solving inverse problems including MRI compressed sensing [27], super-resolution [28], blind image deconvolution [29], and phase retrieval [30], [31].

End-to-end trained networks are purely data driven methods that learn to directly map measurements to signals. A denoising network that directly maps corrupted images to clean images was proposed in [32]. The method was applied to MRI measurements captured under different acquisition setups [33]. Other approaches such as [34], [35] 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 minimizes data consistency and regularization terms. PnP-ADMM [1] was the first plug-and-play iterative algorithm that used pretrained denoisers as priors. This method is based on the ADMM algorithm [2]. PnP-FISTA [36] 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 [37], [38]

Deep unrolled networks learn the denoiser network in PnP algorithms in a supervised manner [11], [12], [39]. 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. Deep unrolled methods show remarkable results in several inverse problems such as super-resolution [40], image restoration [41], MRI [42] and CT [43] reconstruction.

C. 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 [44], [45]. 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 [46]–[48] 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 [49]–[51]. 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 [16], [52]. [53] proposed a modular-network that learns new tasks without compromising performance on previous tasks. 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 [54]. 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 we can perform successful task/domain adaptation.

Tuning specific layers such as the BatchNorm [55], the final classification head [56], and LayerNorm [57] 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 [58]. In [18], [59], a network reparametrization technique was proposed to learn shared and task-specific modules, enabling a single network to adapt to various settings. Hyperdomain Networks [17] use modulated convolution to adapt generator networks to new domains. An adaptation method for shifts in domain and forward-models when solving inverse problems was proposed in [60]. The method proposes a fine-tuning and regularization technique adopted from RED [61]. Domain-specific batch normalization layers were proposed in [62] for a segmentation network that can handle brain MR scans across scanners and protocols. Unlike R&R [61], the method proposed in [62] 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 [63]–[65]. 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 [66] 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], [66], [67] 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.

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

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

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

where γ is the step size, superscript $k = 1, \dots, K$ 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 [12], 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 [68].

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 [46], [69], we assume the intrinsic dimension of the objective in (7) is small. We parameterize M_d such that its trainable parameters remains significantly smaller than the number of parameters in the base network.

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 $\theta$, $\{\beta_k\}_{k\geq 0}$, $\gamma$, $\alpha$
Output: Recovered image $\mathbf{x}^K$ and 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 $\mathbf{x}^K, M_d$
```

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 $k \times k$ 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 are 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.

Figure 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

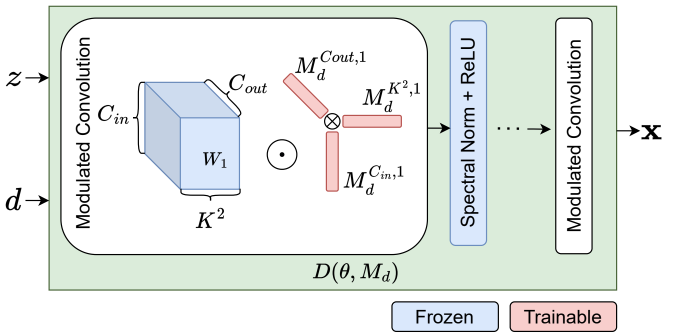

Fig. 2: Overview of our factorized network that uses modulated convolutions for domain adaptation. Our network follows the DNCNN [68] architecture that leverages modulated convolution for domain adaptation. After trained on the source domain, the network learns low-rank modulations for each domain while 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.

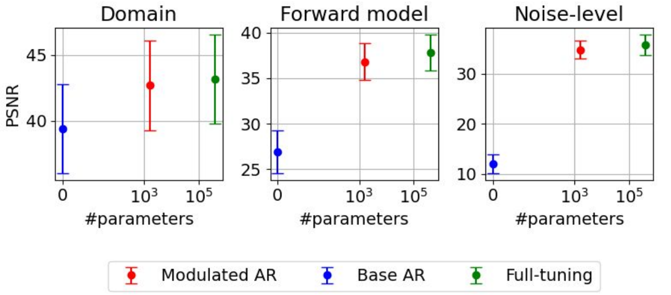

Fig. 3: Comparison of our modulated AR, fully-tuned AR, and the Base AR networks in terms of accuracy and number of additional parameters they require. Base AR requires no additional parameter and provides worst performance. Fully-tune AR provides best performance using a large number of parameters.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

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 Fourier measurements without any measurement noise. Base AR uses spectral normalization proposed in [70] along with the ReLU activation functions. We implement our AR network using a 12-layer DnCNN [68] network. We will provide training details as well as hyper-parameters used in our experiments in the supplementary material.

A. *Parameter efficiency for adaptation*

Figure 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 worst performance. Full training learns a new network for every domain/distribution shifts and provides 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 1.6K 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.

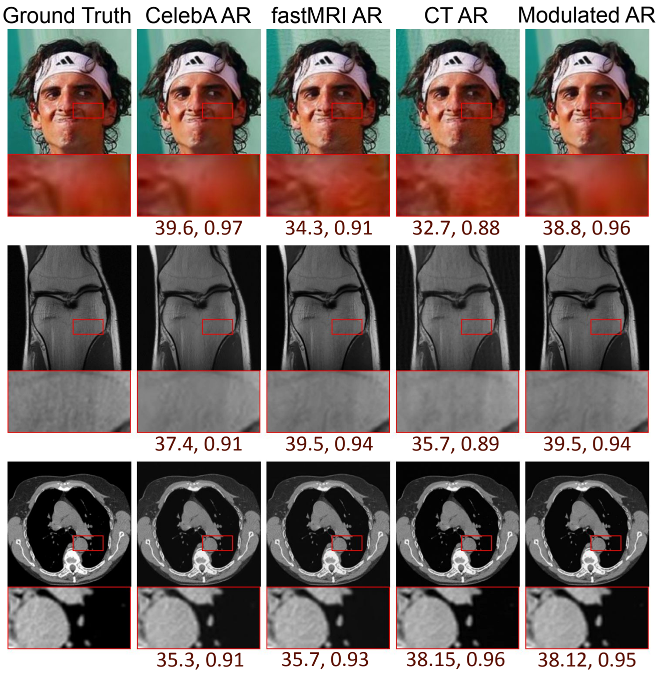

Fig. 4: We present sample ground truth images in the first column and reconstruction of these images using three AR networks trained on Face, MR, and CT images in the subsequent three columns . Our modulated AR, shown in the last column effectively removes this artifacts and closes the performance gap.

TABLE I: Average PSNR of AR networks under domain shift. 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.

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

TABLE II: Comparison of our method with existing domain adaptation techniques. 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.

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

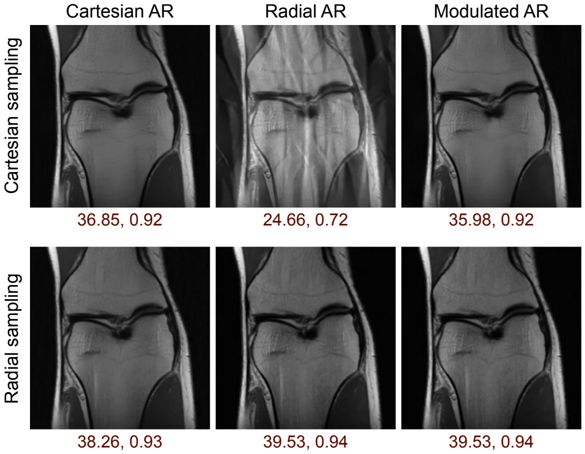

Fig. 5: Reconstruction results under sampling pattern shifts. AR trained on radial pattern performs poorly when tested on Cartesian sampled patterns. Our Modulated AR applies low-rank modulations to adapt Radial AR to Cartesian samples.

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 [71] for natural images, NYU fastMRI dataset for [72] knee MRI scans, and a subset of TCGA-LUAD dataset [73] for CT scans. The first three columns of Table I show the performance 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. Figure 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 [52], RCM [18], Hyperdomain Modulation [17], and Full-tuning. We evaluate these methods using the same training and testing procedure as our proposed approach. Supsup [52] learns binary masks to find domain specific sub-networks. RCM [18] reparameterizes convolutions using domain-specific feature transformations. Hyperdomain [17] 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 while requiring fewer additional trainable parameters.

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.

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 Figure 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.

TABLE III: Sampling pattern shift adaptation results. Our Modulated AR achieves competitive in-domain performance to ARs trained on specific patterns. Moreover, it shows an overall superior performance across all patterns.

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**

TABLE IV: Sampling ratio shift adaptation results.

Test ratio	4x AR	8x AR	10x AR	Modualted 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**

Sampling ratio shifts. We compared the performance of different AR networks trained on three sampling ratios and presented the results in Table IV. The 4× AR network exhibits poor performance when tested with 8× and 10× radially subsampled measurements. Additionally, the AR network trained on the 8× ratio did not perform well with 10× ratio. To address this, we applied our modulation technique to adapt the Base AR model to 8× and 10× sampling ratios. On average, the modulated network outperforms AR networks trained on specific sampling ratios. Figure 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.

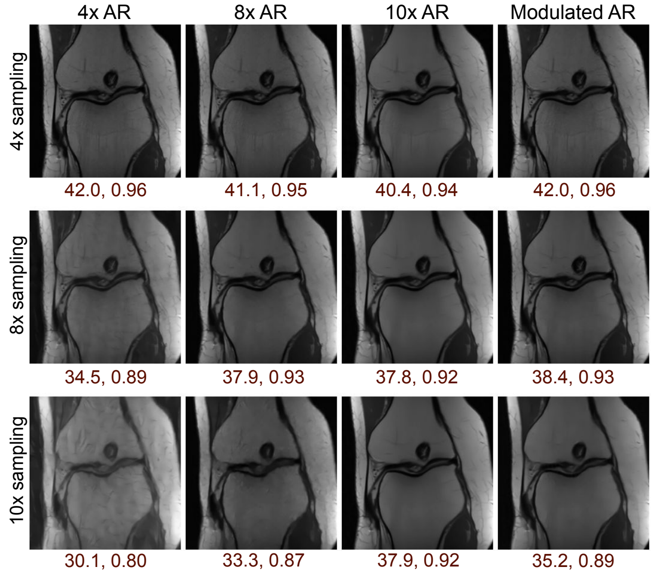

Fig. 6: Examples of image reconstruction under sampling ratio shifts. Our Modulated AR shows an average superior performance when compared to the 4×, 8×, and 10× AR networks.

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 techniques and is only one dB less than full-tuning, which requires significantly larger number of parameters.

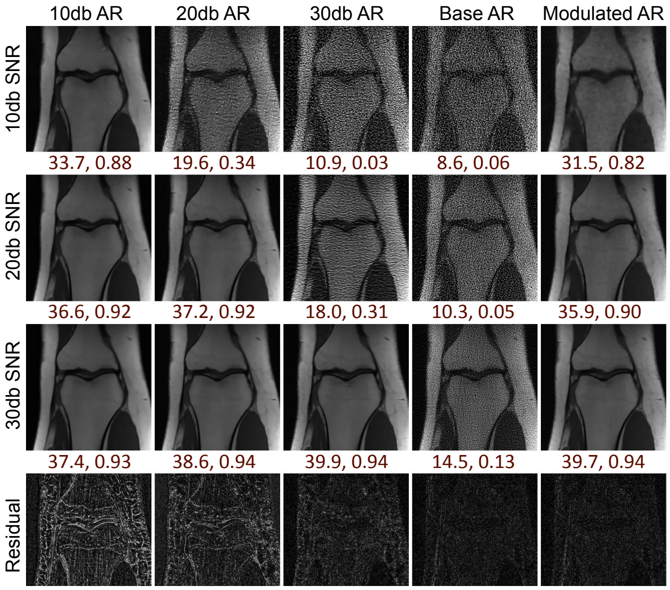

Fig. 7: Visual results of models trained at specific noise levels and our modulated network under measurement level. The last row shows the 20×amplified residual of the reconstructed image under no measurement noise.

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 different noise levels on the performance. Figure 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 10dB 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 Figure 7

TABLE V: Comparison of domain adaptation methods under forward model shifts. Our proposed method achieves competitive performance to full-tuning with significantly fewer parameters. It outperforms related domain adaptation methods in terms of performance and parameter efficiency.

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	37.10
4x to 10x	**35.34**	33.62	34.49	34.56	34.73
Fourier to Gaussian	**38.59**	36.31	38.49	38.45	38.55
Avg	**37.71**	35.43	36.50	36.53	36.79

TABLE VI: Noise level shift adaptation results. Modulated AR that learns low-rank factors for each noise-level outperforms networks trained for a specifc noise.

Test SNR	10db AR	20db AR	30db AR	Base AR	Modualted 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	40.93	40.93	40.93	40.93
Avg	35.21	33.30	28.11	19.30	36.33

TABLE VII: Comparison of various domain adaptation methods under noise level shifts. Our proposed method can achieve competitive performance to RCM and Full-tuning while requiring significantly fewer number of additional parameters.

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

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 AR 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 [52] 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 [18] 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 [74]. We believe that these limitations will serve as inspiration for several future studies.

VI. Conclusion

We proposed a simple and parameter-efficient method to adapt networks 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 *2013 IEEE global conference on signal and information processing*. IEEE, 2013, pp. 945–948.

[2] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[3] E. J. Candes, J. K. Romberg, and T. Tao, “Stable signal recovery from incomplete and inaccurate measurements,” *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, vol. 59, no. 8, pp. 1207–1223, 2006.

[4] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, “Online dictionary learning for sparse coding,” in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 689–696.

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

[6] M. Elad and M. Aharon, “Image denoising via sparse and redundant representations over learned dictionaries,” *IEEE Transactions on Image processing*, vol. 15, no. 12, pp. 3736–3745, 2006.

[7] 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 Journal on Selected Areas in Information Theory*, vol. 1, no. 1, pp. 39–56, 2020.

[8] V. Antun, F. Renna, C. Poon, B. Adcock, and A. C. Hansen, “On instabilities of deep learning in image reconstruction and the potential costs of ai,” *Proceedings of the National Academy of Sciences*, vol. 117, no. 48, pp. 30 088–30 095, 2020.

[9] N. M. Gottschling, V. Antun, B. Adcock, and A. C. Hansen, “The troublesome kernel: why deep learning for inverse problems is typically unstable,” *arXiv preprint arXiv:2001.01258*, 2020.

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

[11] V. Monga, Y. Li, and Y. C. Eldar, “Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing,” *IEEE Signal Processing Magazine*, vol. 38, no. 2, pp. 18–44, 2021.

[12] J. Liu, M. S. Asif, B. Wohlberg, and U. S. Kamilov, “Recovery analysis for plug-and-play priors using the restricted eigenvalue condition,” in *Advances in Neural Information Processing Systems (NeurIPS)*, Dec. 2021, in press.

[13] Y. Sun, B. Wohlberg, and U. S. Kamilov, “An online plug-and-play algorithm for regularized image reconstruction,” *IEEE Transactions on Computational Imaging*, vol. 5, no. 3, pp. 395–408, 2019.

[14] M. Z. Darestani, A. S. Chaudhari, and R. Heckel, “Measuring robustness in deep learning based compressive sensing,” in *International Conference on Machine Learning*. PMLR, 2021, pp. 2433–2444.

[15] M. McCloskey and N. J. Cohen, “Catastrophic interference in connectionist networks: The sequential learning problem,” in *Psychology of learning and motivation*. Elsevier, 1989, vol. 24, pp. 109–165.

[16] A. Mallya, D. Davis, and S. Lazebnik, “Piggyback: Adapting a single network to multiple tasks by learning to mask weights,” in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 67–82.

[17] A. Alanov, V. Titov, and D. P. Vetrov, “Hyperdomainnet: Universal domain adaptation for generative adversarial networks,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 29 414–29 426, 2022.

[18] 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 *European Conference on Computer Vision*. Springer, 2020, pp. 689–707.

[19] C. Riquelme, J. Puigcerver, B. Mustafa, M. Neumann, R. Jenatton, A. Susano Pinto, D. Keysers, and N. Houlsby, “Scaling vision with sparse mixture of experts,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 8583–8595, 2021.

[20] E. J. Candès, J. Romberg, and T. Tao, “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information,” *IEEE Transactions on information theory*, vol. 52, no. 2, pp. 489–509, 2006.

[21] D. L. Donoho, “Compressed sensing,” *IEEE Transactions on information theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

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

[23] L. I. Rudin, S. Osher, and E. Fatemi, “Nonlinear total variation based noise removal algorithms,” *Physica D: Nonlinear Phenomena*, vol. 60, no. 1, pp. 259–268, 1992.

[24] E. Y. Sidky, C.-M. Kao, and X. Pan, “Accurate image reconstruction from few-views and limited-angle data in divergent-beam ct,” *Journal of X-ray Science and Technology*, vol. 14, no. 2, pp. 119–139, 2006.

[25] A. Qayyum, I. Ilahi, F. Shamshad, F. Boussaid, M. Bennamoun, and J. Qadir, “Untrained neural network priors for inverse imaging problems: A survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[26] A. Bora, A. Jalal, E. Price, and A. G. Dimakis, “Compressed sensing using generative models,” in *International conference on machine learning*. PMLR, 2017, pp. 537–546.

[27] A. Jalal, M. Arvinte, G. Daras, E. Price, A. G. Dimakis, and J. Tamir, “Robust compressed sensing mri with deep generative priors,” in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 14 938–14 954.

[28] S. Menon, A. Damian, S. Hu, N. Ravi, and C. Rudin, “Pulse: Self-supervised photo upsampling via latent space exploration of generative models,” in *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 2020, pp. 2437–2445.

[29] M. Asim, F. Shamshad, and A. Ahmed, “Blind image deconvolution using deep generative priors,” *IEEE Transactions on Computational Imaging*, vol. 6, pp. 1493–1506, 2020.

[30] P. Hand, O. Leong, and V. Voroninski, “Phase retrieval under a generative prior,” *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[31] R. Hyder, V. Shah, C. Hegde, and M. S. Asif, “Alternating phase projected gradient descent with generative priors for solving compressive phase retrieval,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 7705–7709.

[32] 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.

[33] J. Lehtinen, J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, and T. Aila, “Noise2noise: Learning image restoration without clean data,” in *International Conference on Machine Learning*. PMLR, 2018, pp. 2965–2974.

[34] A. Sriram, J. Zbontar, T. Murrell, A. Defazio, C. L. Zitnick, N. Yakubova, F. Knoll, and P. Johnson, “End-to-end variational networks for accelerated mri reconstruction,” in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part II 23*. Springer, 2020, pp. 64–73.

[35] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser, “Deep convolutional neural network for inverse problems in imaging,” *IEEE transactions on image processing*, vol. 26, no. 9, pp. 4509–4522, 2017.

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

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

[38] R. Ahmad, C. A. Bouman, G. T. Buzzard, S. Chan, S. Liu, E. T. Reehorst, and P. Schniter, “Plug-and-play methods for magnetic resonance imaging: Using denoisers for image recovery,” *IEEE signal processing magazine*, vol. 37, no. 1, pp. 105–116, 2020.

[39] J. Zhang and B. Ghanem, “Ista-net: Interpretable optimization-inspired deep network for image compressive sensing,” in *CVPR*, 2018, pp. 1828–1837.

[40] K. Zhang, L. V. Gool, and R. Timofte, “Deep unfolding network for image super-resolution,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[41] C. Mou, Q. Wang, and J. Zhang, “Deep generalized unfolding networks for image restoration,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 17 399–17 410.

[42] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5270–5279.

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

[44] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?” *Advances in neural information processing systems*, vol. 27, 2014.

[45] M. Long, Y. Cao, Z. Cao, J. Wang, and M. I. Jordan, "Transferable representation learning with deep adaptation networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 12, pp. 3071–3085, 2018.

[46] E. J. Hu, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen *et al.*, "Lora: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2021.

[47] X. He, C. Li, P. Zhang, J. Yang, and X. E. Wang, "Parameter-efficient model adaptation for vision transformers," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 1, pp. 817–825, Jun. 2023.

[48] Y.-C. Liu, C.-Y. MA, J. Tian, Z. He, and Z. Kira, "Polyhistor: Parameter-efficient multi-task adaptation for dense vision tasks," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 36 889–36 901.

[49] S.-A. Rebuffi, H. Bilen, and A. Vedaldi, "Efficient parametrization of multi-domain deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[50] W.-H. Li, X. Liu, and H. Bilen, "Cross-domain few-shot learning with task-specific adapters," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 7161–7170.

[51] S. Chen, C. Ge, Z. Tong, J. Wang, Y. Song, J. Wang, and P. Luo, "Adaptformer: Adapting vision transformers for scalable visual recognition," *Advances in Neural Information Processing Systems*, vol. 35, pp. 16 664–16 678, 2022.

[52] M. Wortsman, V. Ramanujan, R. Liu, A. Kembhavi, M. Rastegari, J. Yosinski, and A. Farhadi, "Supermasks in superposition," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 15 173–15 184.

[53] A. Shaker, F. Alesiani, and S. Yu, "Modular-relatedness for continual learning," in *International Symposium on Intelligent Data Analysis*. Springer, 2022, pp. 290–301.

[54] T. Veniat, L. Denoyer, and M. Ranzato, "Efficient continual learning with modular networks and task-driven priors," in *9th International Conference on Learning Representations, ICLR 2021*, 2021.

[55] J. Frankle, D. J. Schwab, and A. S. Morcos, "Training batchnorm and only batchnorm: On the expressive power of random features in cnns," in *International Conference on Learning Representations*, 2020.

[56] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9729–9738.

[57] S. Basu, D. Massiceti, S. X. Hu, and S. Feizi, "Strong baselines for parameter efficient few-shot fine-tuning," *arXiv e-prints*, pp. arXiv–2304, 2023.

[58] D. Lian, D. Zhou, J. Feng, and X. Wang, "Scaling & shifting your features: A new baseline for efficient model tuning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 109–123, 2022.

[59] A. Rosenfeld and J. K. Tsotsos, "Incremental learning through deep adaptation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 3, pp. 651–663, 2018.

[60] D. Gilton, G. Ongie, and R. Willett, "Model adaptation for inverse problems in imaging," *IEEE Transactions on Computational Imaging*, vol. 7, pp. 661–674, 2021.

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

[62] N. Karani, K. Chaitanya, C. Baumgartner, and E. Konukoglu, "A lifelong learning approach to brain mr segmentation across scanners and protocols," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2018, pp. 476–484.

[63] 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 *International Conference on Machine Learning*. PMLR, 2022, pp. 4754–4776.

[64] B. Song, L. Shen, and L. Xing, "Piner: Prior-informed implicit neural representation learning for test-time adaptation in sparse-view ct reconstruction," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2023, pp. 1928–1938.

[65] S. Goyal, M. Sun, A. Raghunathan, and J. Z. Kolter, "Test time adaptation via conjugate pseudo-labels," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 6204–6218.

[66] 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 Processing Magazine*, vol. 40, no. 1, pp. 85–97, 2023.

[67] N. Parikh, S. Boyd *et al.*, "Proximal algorithms," *Foundations and trends® in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[68] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[69] C. Li, H. Farkhoor, R. Liu, and J. Yosinski, "Measuring the intrinsic dimension of objective landscapes," in *International Conference on Learning Representations*, 2018.

[70] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in *International Conference on Learning Representations*, 2018.

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

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

[73] K. Clark, B. Vendt, K. Smith, J. Freymann, J. Kirby, P. Koppel, S. Moore, S. Phillips, D. Maffitt, M. Pringle, L. Tarbox, and F. Prior, "The cancer imaging archive (TCIA): Maintaining and operating a public information repository," *Journal of Digital Imaging*, vol. 26, no. 6, pp. 1045–1057, jul 2013.

[74] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 2017, pp. 1126–1135.

[75] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[76] A. F. Agarap, "Deep learning using rectified linear units (relu)," *arXiv preprint arXiv:1803.08375*, 2018.

[77] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

Supplementary Material

We present additional material and details to complement our main paper. We provide a detailed description of our training and hyper-parameter tuning procedures. Additionally, we present further experiments, analyze the effects of modulation in each layer, and showcase visual results. Finally, we address the limitations of our work and suggest potential directions for future research.

VII. Training details

We used PyTorch [75] to implement our proposed method on a single NVIDIA GeForce RTX 2080 Ti GPU with 12GB memory. Our artifact removal (AR) prior is implemented using a DnCNN [68] network with 12 blocks. Each block comprises a convolution layer, a spectral normalization layer [70], and a ReLU activation layer [76]. Within the intermediate blocks, our convolution consists of 64 filter kernels with a size of 3×3. The number of filters at the input and output layers is set to match the number of features in the target dataset.

The AR operator is implemented as the residual of our DnCNN network, utilizing an α-averaged operator similar to [12]. We set $\alpha = 0.2$ for all experiments. With this operator, we obtained the best performance when setting the acceleration parameter $q_k = 1$ for all $k \geq 1$. We used the ADAM optimizer [77], setting the learning rate to 10^{-4} for the base network weights during full-tuning and 10^{-2} for the modulation factors. Our experiments were trained for 100 epochs. After 50 epochs, we adjusted the learning rate for the modulations, reducing it by a factor of 2. We maintained default settings for the remaining optimizer configurations. We reported the results using the best-performing model on the validation dataset. Throughout all experiments, we incorporated $K = 33$ unrolled iterations. For MR image reconstruction tasks, we employed a step size of $\gamma = 1.5$, while $\gamma = 1.2$ was utilized for face image reconstruction tasks. In the comparison experiments, we adapted the source code available on the GitHub pages of RCM [18], Supsup [52], and Hyperdomain Modulation [17] to align with our framework.

We utilized PyTorch's Torchvision library as the data source for CelebA [71] and adhered to the official train-val-test set split. For the multi-coil Knee dataset provided by NYU FastMRI [72], we partitioned the original training set into training and validation sets using an 85-15 split. Subsequently, we used the validation dataset as our test set. In the case of CT scans, we used a subset of the TCGA-LUAD dataset [73]. We performed a 75-15-10 split for the training, validation, and test sets. All images were normalized within the range of $[0, 1]$.

A. Initialization.

To ensure stability, we implement our modulation as $W^l \odot (1 + M_d^l)$. The factors of M_d^l are initialized using a uniform distribution near zero, such that $M_d^{i,l} \sim U\left[-\frac{1}{\sqrt{f}}, \frac{1}{\sqrt{f}}\right]$, where f corresponds to the number of channels or kernel size to be modulated at the l^{th} layer. This approach ensures that our modulated weights start with values close to the pre-trained weights and allows them to be updated to an optimal value.

VIII. Additional experiments

A. Analyzing the layer-wise effects of weight modulation

We conducted an analysis of the effects of weight modulation at each layer. This analysis provides us with better insight into which layers play more crucial roles in bridging performance gaps. Subsequently, we can use this information to apply dynamic adaptation to different layers, thereby further reducing additional computation and the necessary parameters.

Figure 8 illustrates the ratio of ℓ_2 norms of modulations $|M_d^l|$ to pre-trained weight norms $|W^l|$ at the l^{th} layer, corresponding to domain, forward model, and noise-level adaptation experiments. We normalize the norm ratios independently for each adaptation, scaling them between 0 and 1. The figure suggests that modulation power is primarily concentrated in the final layers of the network, with minimal impact observed in the initial layers. We confirm this observation by conducting partial modulation experiments for noise-level adaptation, where we apply modulation to a subset of layers in our network. We report our findings in Table VIII. We note that modulating the top half of the layers (requiring only 0.88k parameters) resulted in a 7% performance decrease, and modulating the last 4 layers (requiring only 0.67k parameters) led to only a 12.8% drop. In contrast, modulating the first half and the first four layers resulted in a significant performance reduction. In future works, we can utilize these insights when designing parameter-efficient domain adaptation techniques.

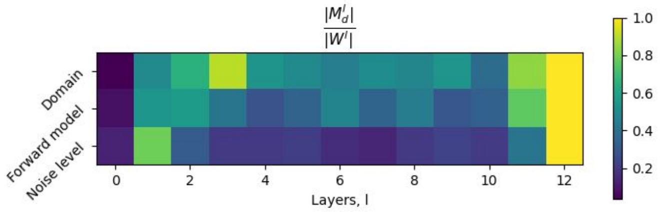

Fig. 8: We show most of the modulation occurs at the final layers of our artifact removal network. We present the ratio of ℓ_2 norms of modulations to weights at each layer for domain, forward model, and noise level shifts. We have normalized each row to enhance visualization.

TABLE VIII: We experimentally demonstrate that modulating the top half and the last four layers of our network retains the majority of the performance achieved through full modulation. In contrast, we show that the same level of performance cannot be attained by solely modulating the initial layers.

Trained blocks	Noise-level Adaptation		# Trained	
	PSNR	SSIM	Params	Layers
None	9.18	0.02	1.6k	13
All	31.40	0.79	1.6k	13
Blocks 0-4	10.67	0.04	0.67k	5
Blocks 4-8	16.87	0.20	0.67k	5
Blocks 8-12	27.15	0.49	0.67k	5
Blocks 0-6	19.97	0.28	0.88k	7
Blocks 6-12	29.14	0.64	0.88k	7

B. Adapting to domain and forward model shifts

In the main paper, we focused on experiments involving a single type of shift that can arise when solving inverse problems. Now, we are expanding this concept by applying our technique to accommodate multiple shifts. Specifically, we will adapt a network trained for face image reconstruction using Gaussian sampling to MR image reconstruction with Fourier sampling, encompassing both domain and forward model shifts. Table IX shows the performance of these AR networks. We observe an 18% drop in performance on CelebA and a 30.37% drop on FastMRI ARs due to this shift. To address this challenge, we applied our domain adaptation technique and conducted a comparison with relevant methods. We present the average performance in Table X. Similar to our previous experiments, our method outperforms other domain adaptation techniques. Additional visual results for this experiment are shown in figure 9.

TABLE IX: The CelebA AR, which is trained to reconstruct faces from random projections, exhibits poor performance in MR reconstruction. Similarly, the fastMRI AR is unable to reconstruct face images, despite its competence in its own domain. We provide the average PSNR values for these ARs.

Test dataset	CelebA AR	FastMRI AR
CelebA	35.05	24.58
FastMRI	28.46	35.34

TABLE X: Comparison results upon adapting CelebA AR to MR and CT reconstruction tasks. We have highlighted the best-performing method in boldface and second best with an underscore.

Methods	# Trainable parameters	MRI Knee	CT
Full-tuning	407k	**35.34**	**32.63**
Supsup	407k	32.57	28.82
Reparametrized AR	50.6k	34.14	30.43
Hyperdomain modulation	0.7k	33.81	31.37
Modulated AR	1.6k	34.17	32.11

C. Adapting to sampling type shifts

We consider Fourier and Gaussian measurement operators as domains of different sampling types. When testing our Base AR with Gaussian sub-sampled MRI measurements, we observed an average PSNR drop of approximately 2.7 dBs on our test set. As shown in the second row of the first column of Figure 10, the Base AR is incapable of recovering fine details and produces a smoothed output. Using our proposed modulation technique, we were able to enhance the reconstruction results and achieve outcomes similar to those of the fully-tuned AR.

Lastly, in Figure 11, we present the ground truth images for the samples utilized in our forward model adaptation experiments. The corresponding reconstruction outputs of these images are displayed in the main paper.

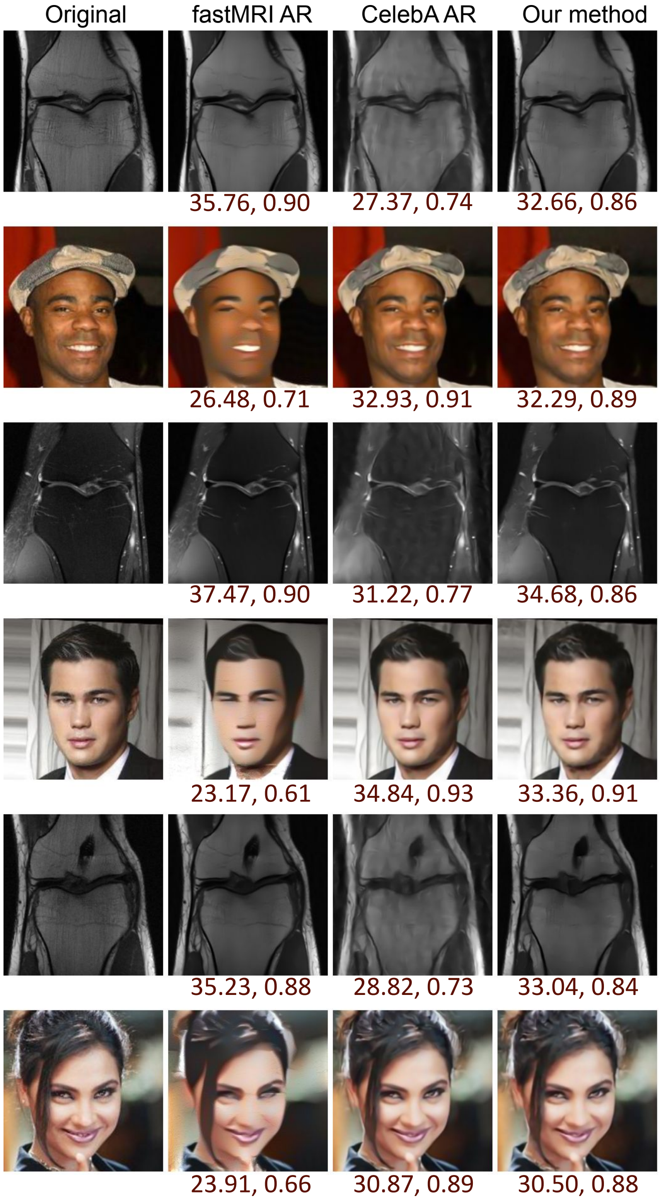

Fig. 9: Examples of image reconstruction under domain and forward model shifts. Second and third columns show reconstructed images with fastMRI AR and celebA AR, respectively. Reconstruction quality degrades with domain shifts (PSNR and SSIM reported under each image). Our proposed network adaptation method, where we adapt the mis-matched ARs (recovering MR images using CelebA AR and recovering faces using fastMRI AR) to recover an targets images in shown below.

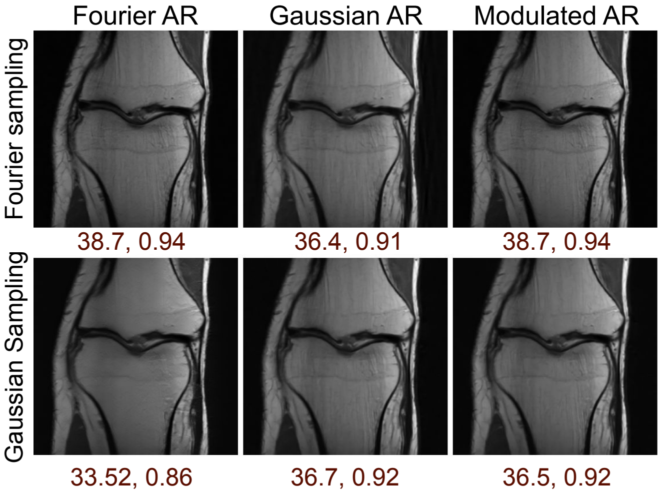

Fig. 10: The Fourier (Base) AR is unable to recover fine details under Gaussian sampling (first column, second row). On the other hand, the Modulated AR can achieve performance comparable to fully-tuned networks when reconstructing images from both Fourier and Gaussian samples (last column).

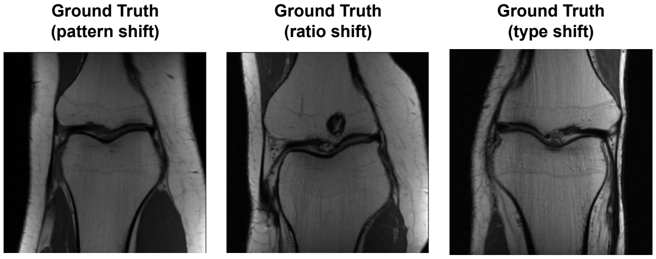

Fig. 11: Ground truth images for forward model shifts experiments.