Received: 27 September 2019 | Revised: 12 November 2019 | Accepted: 6 December 2019

DOI: 10.1002/mrm.28148

FULL PAPER

Magnetic Resonance in Medicine

A Transfer-Learning Approach for Accelerated MRI Using Deep Neural Networks

Salman Ul Hassan Dar[1,2] | **Muzaffer Özbey**[1,2] | **Ahmet Burak Çatlı**[1,2] | **Tolga Çukur**[1,2,3]

[1]Department of Electrical and Electronics Engineering, Bilkent University, Ankara, Turkey

[2]National Magnetic Resonance Research Center (UMRAM), Bilkent University, Ankara, Turkey

[3]Neuroscience Program, Sabuncu Brain Research Center, Bilkent University, Ankara, Turkey

Correspondence
Tolga Çukur, Department of Electrical and Electronics Engineering, Room 304, Bilkent University, Ankara, TR-06800, Turkey.
Email: cukur@ee.bilkent.edu.tr

Funding information
This work was supported in part by the following: Marie Curie Actions Career Integration grant (PCIG13-GA-2013-618101), European Molecular Biology Organization Installation grant (IG 3028), TUBA GEBIP fellowship, TUBITAK 1001 grant (118E256), and BAGEP fellowship awarded to T. Çukur. We also gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research

Purpose: Neural networks have received recent interest for reconstruction of undersampled MR acquisitions. Ideally, network performance should be optimized by drawing the training and testing data from the same domain. In practice, however, large datasets comprising hundreds of subjects scanned under a common protocol are rare. The goal of this study is to introduce a transfer-learning approach to address the problem of data scarcity in training deep networks for accelerated MRI.

Methods: Neural networks were trained on thousands (upto 4 thousand) of samples from public datasets of either natural images or brain MR images. The networks were then fine-tuned using only tens of brain MR images in a distinct testing domain. Domain-transferred networks were compared to networks trained directly in the testing domain. Network performance was evaluated for varying acceleration factors (4-10), number of training samples (0.5-4k), and number of fine-tuning samples (0-100).

Results: The proposed approach achieves successful domain transfer between MR images acquired with different contrasts (T_1- and T_2-weighted images) and between natural and MR images (ImageNet and T_1- or T_2-weighted images). Networks obtained via transfer learning using only tens of images in the testing domain achieve nearly identical performance to networks trained directly in the testing domain using thousands (upto 4 thousand) of images.

Conclusion: The proposed approach might facilitate the use of neural networks for MRI reconstruction without the need for collection of extensive imaging datasets.

KEYWORDS

accelerated MRI, compressive sensing, deep learning, image reconstruction, transfer learning

1 | INTRODUCTION

The unparalleled soft-tissue contrast in MRI has rendered it a preferred modality in many diagnostic applications, but long scan durations limit its clinical use. Acquisitions can be accelerated by undersampling in k-space, and a tailored reconstruction can be used to recover unacquired data. Because MR images are inherently compressible, a popular framework for accelerated MRI has been compressive sensing (CS).[1,2] CS has offered improvements in scan efficiency in many applications, including structural,[2] angiographic,[3] functional,[4] diffusion,[5] and parametric imaging.[6] Yet, the CS

[Correction added after online publication 6 March 2020. The author has updated section 3.1.2 to change "T2-domain transfer" to "T_2-domain transfer."]

framework is not without limitation. First, CS involves nonlinear optimization algorithms that scale poorly with growing data size and hamper clinical workflow. Second, CS commonly assumes that MRI data are sparse in fixed transform domains, such as finite differences or wavelet transforms. Recent studies highlight the need for learning the transform domains specific to each dataset to optimize performance.[7] Lastly, CS requires careful parameter tuning (e.g., for regularization) for optimal performance. Whereas several approaches were proposed for data-driven parameter tuning,[8,9] these methods can induce further computational burden.

Neural network (NN) architectures that reconstruct images from undersampled data have recently been proposed to address the abovementioned limitations. Improved image quality over traditional CS has readily been demonstrated for several applications, including angiographic,[10] cardiac,[11-13] brain,[13-33] abdominal,[34-36] and musculoskeletal imaging.[37-41] The common approach is to train a network off-line using a relatively large set of fully sampled MRI data, and then use it for online reconstruction of undersampled data. Reconstructions can be achieved in several hundred milliseconds, significantly reducing computational burden.[38,39] The NN framework also alleviates the need for ad hoc selection of transform domains. For example, a recent study used a cascade of convolutional neural networks (CNNs) to recover images directly from zero-filled Fourier reconstructions of undersampled data.[11,22,39] The trained CNN layers reflect suitable transforms for image reconstruction. The NN framework introduces more tunable hyperparameters (e.g., number of layers, units, activation functions) than would be required in CS. However, previous studies demonstrate that hyperparameters optimized during the training phase generally perform well during the testing phase.[39] Taken together, these advantages render the NN framework a promising avenue for accelerated MRI.

A common strategy to enhance network performance is to boost model complexity by increasing the number of layers and units in the architecture. A large set of training data must then be used to reliably learn the numerous model parameters.[42] Previous studies either used an extensive database of MR images comprising several tens to hundreds of subjects,[12,24,38] or data augmentation procedures to artificially expand the size of training data.[11,12] For instance, an early study performed training on T_1-weighted brain images from nearly 500 subjects in the Human Connectome Project database and testing on T_2-weighted images.[24] Yet, it remains unclear how well a network trained on images acquired with a specific type of tissue contrast generalizes to images acquired with different contrasts. Furthermore, for optimal reconstruction performance the network must be trained on images acquired with the same scan protocol that it later will be tested on. However, large databases such as those provided by the Human Connectome Project may not be readily available in many applications, potentially rendering NN-based reconstructions suboptimal.

In this study, we propose a transfer-learning approach to address the problem of data scarcity in network training for accelerated MRI (Figure 1). In transfer learning, network training is performed in a domain where large datasets are available, and knowledge captured by the trained network is then transferred to a different domain where data are scarce.[43,44] Domain transfer was previously used to suppress coherent aliasing artifacts in projection reconstruction acquisitions,[15] to perform non-Cartesian to Cartesian interpolation in k-space,[24] and to assess the robustness of network reconstructions to variations in SNR and undersampling patterns.[40] In contrast, we employ transfer learning to enhance NN-based reconstructions of randomly undersampled acquisitions in the testing domain. A deep CNN architecture with multiple subnetworks is taken as a model network.[11] For reconstruction of multi-coil data, calibration consistency (CC), data consistency (DC) and CNN blocks are incorporated to synthesize missing samples. In the training domain using several thousand images, the network is pretrained to reconstruct reference images from zero-filled reconstructions of undersampled data. The trained network is then fine-tuned end to end in the testing domain using tens of images.

To demonstrate the proposed approach, comprehensive evaluations were performed across a broad range of acceleration factors (R = 4-10) on T_1- and T_2-weighted brain images, considering both single-coil data from a public database and multi-coil data acquired on a 3 Tesla (T) scanner. Separate network models were learned for domain transfer between natural and MR images (ImageNet and T_1- or T_2-weighted). Domain-transferred networks were quantitatively compared against networks trained in the testing domain and against conventional CS reconstructions in the single-coil setting[1,2] and iTerative Self-consistent Parallel Imaging Reconstruction (SPIRiT) in the multi-coil setting.[45] We find that domain-transferred networks fine-tuned with tens of images achieve nearly identical performance to networks trained directly in the testing domain using thousands (upto 4 thousand) of images, and that networks outperform conventional image reconstruction methods.

A preliminary version of this work was presented at the 26th Annual Meeting of International Society for Magnetic Resonance in Medicine under the title Transfer Learning for Reconstruction of Accelerated MRI Acquisitions via Neural Networks.[46]

2 | METHODS

2.1 | MRI reconstruction via compressed sensing

2.1.1 | Single-coil data

In accelerated MRI, an undersampled acquisition is followed by a reconstruction to recover missing k-space

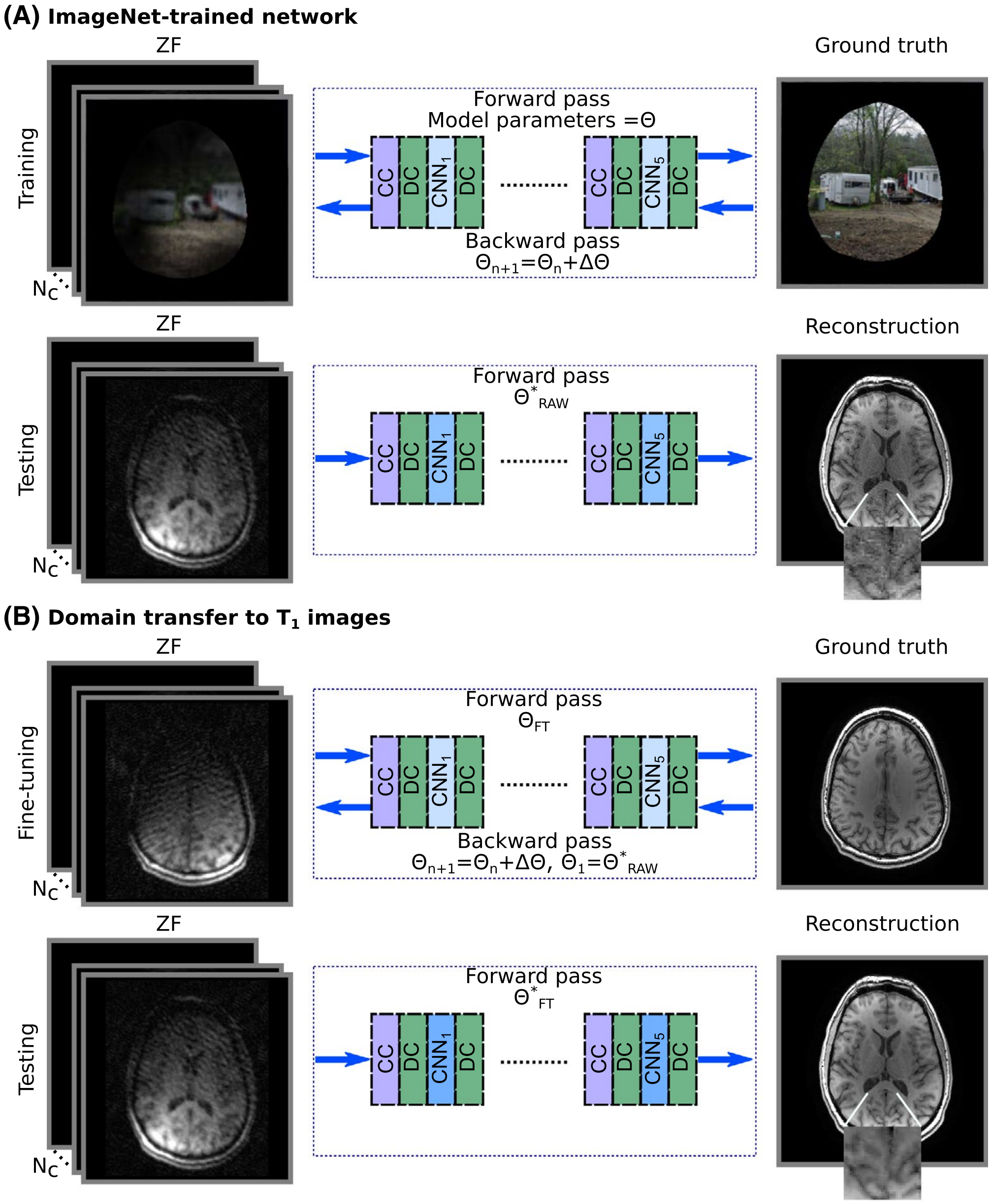

FIGURE 1 Proposed transfer-learning approach for NN-based reconstructions of multi-coil (N_c coils) undersampled acquisitions. A deep architecture with multiple subnetworks is used. The subnetworks consist of CC and CNN blocks, each followed by a DC block. (A) Each CNN block is trained sequentially to reconstruct synthetic multi-coil natural images from ImageNet, given zero-filled Fourier reconstructions of their undersampled versions. Due to differences in the characteristics of natural and MR images, the ImageNet-trained network will yield suboptimal performance when directly tested on MR images. (B) For domain transfer, the ImageNet-trained network is fine-tuned end to end in the testing domain using tens of images. This approach enables successful domain transfer between natural and MR images. CC, calibration-consistency; CNNs, convolutional neural networks; DC, data consistency; NN, neural networks

samples. This recovery can be formulated as a linear inverse problem:

$$F_u x = y_u, \tag{1}$$

where x denotes the image to be reconstructed; F_u is the partial Fourier transform operator at the sampled k-space locations; and y_u denotes acquired k-space data. Because Equation 1 is underdetermined, additional prior information is typically incorporated in the form of a regularization term:

$$x_{rec} = \min_x \|F_u x - y_u\|_2 + R(x). \tag{2}$$

Here, the first term enforces consistency between acquired and reconstructed data, whereas $R(x)$ enforces prior information to improve reconstruction performance. In CS, $R(x)$ typically corresponds to L_1-norm of the image in a known transform domain (e.g., wavelet transform or finite differences transform).

The solution of Equation 2 involves nonlinear optimization algorithms that are often computationally complex. This reduces clinical feasibility as reconstruction time becomes prohibitive with increasing size of data. Furthermore, assuming ad hoc selection of fixed transform domains leads to suboptimal reconstructions in many applications.[7] Lastly, it is often challenging to find a set of reconstruction parameters that work optimally across subjects.[47]

2.1.2 | Multi-coil data

For reconstruction of multi-coil data, a hybrid parallel imaging/compressed sensing approach is commonly used. In the common SPIRiT method, k-space samples are synthesized as a weighted linear combination of acquired samples across neighboring k-space locations and coils.[45] The synthesis operation can be formulated as:

$$\widehat{y_m} = \sum_{j=1}^{N_C} g_{mj} \otimes y_j, \tag{3}$$

where $\otimes$ is the convolution operator; g_{mj} denotes weights of the interpolation kernel that takes as input data for the jth coil (y_j) and outputs data for the mth coil ($\widehat{y_m}$); and N_C denotes the number of coils. For each coil, the interpolation kernel is estimated from calibration data y^c, a fully sampled central k-space region. For the mth coil, the estimation is performed via Tikhonov regularized regression as follows:

$$g_m = (Y^* Y + \beta I)\, Y^* y_m^c, \tag{4}$$

where g_m is obtained by aggregating g_{mj} across coils; y_m^c are calibration data from the mth coil; Y is obtained by aggregating calibration data y_j^c in form of a matrix; and β is the Tikhonov regularization parameter.

Given the entire k-space data y, Equation 3 can be expressed in matrix form with the use of an interpolation operator G as follows:

$$\hat{y} = Gy, \tag{5}$$

where $\hat{y}$ is the recovered k-space data, and G is the operator that performs interpolation in matrix form.[45]

In SPIRiT,[45] the recovery problem in Equation 2 can be reformulated as:

$$x_{rec} = \min_x \left\| F_u x - y_u \right\|_2 + \left\| (G-I)\, Fx \right\|_2 + R(x), \tag{6}$$

where x denotes multi-coil images to be reconstructed; y_u denotes acquired multi-coil k-space data; F is the forward Fourier transform operator; and G denotes the interpolation operator that synthesizes unacquired samples in terms of acquired samples across neighboring k-space and coils. To enforce sparsity, $R(x)$ can be selected as the L_1-norm of wavelet coefficients. One efficient way to solve Equation 6 is via the projection onto convex sets algorithm.[48] Projection onto convex sets alternates among a calibration-consistency (CC) projection that applies G, a sparsity projection that enforces sparsity in the transform domain, and a data-consistency (DC) projection.

2.2 | MRI reconstruction via neural networks

2.2.1 | Single-coil data

In the NN framework, a network architecture is used for reconstruction instead of explicit transform-domain constraints. Network training is performed via a supervised learning procedure with the aim to find the set of network parameters that yield accurate reconstructions of undersampled acquisitions. This procedure is performed on a large set of training data (with $\mathrm{N_{train}}$ samples) in which fully sampled reference acquisitions are retrospectively undersampled. Network training typically amounts to minimizing the following loss function[14]:

$$\min_\theta \sum_{n=1}^{\mathrm{N_{train}}} \frac{1}{\mathrm{N_{train}}} \left\| C\left(x_{un};\theta\right) - x_{refn} \right\|_2, \tag{7}$$

where x_{un} represents the Fourier reconstruction of nth undersampled acquisition; x_{refn} represents the respective Fourier reconstruction of the fully sampled acquisition; and $C\left(x_{un};\theta\right)$ denotes the output of the network given the input image x_{un} and the network parameters θ. To reduce sensitivity to outliers, here we minimized a hybrid loss that includes both mean-squared error and mean-absolute error terms. To minimize overfitting, we further added an L_2-regularization term on the network parameters. Therefore, neural network training was performed with the following loss function:

$$\begin{aligned} \min_\theta \sum_{n=1}^{\mathrm{N_{train}}} \frac{1}{\mathrm{N_{train}}} \left\| C\left(x_{un};\theta\right) - x_{refn} \right\|_2 \\ + \sum_{n=1}^{\mathrm{N_{train}}} \frac{1}{\mathrm{N_{train}}} \left\| C\left(x_{un};\theta\right) - x_{refn} \right\|_1 + \gamma_\Phi \left\| \theta \right\|_2, \end{aligned} \tag{8}$$

where γ_Φ is the regularization parameter for network parameters.

A network trained on a sufficiently large set of training examples can then be used to reconstruct an undersampled acquisition from an independent test dataset. This reconstruction can be achieved by reformulating the problem in Equation 2[14]:

$$x_{rec}=\min_{x}\lambda\left\|F_u x-y_u\right\|_2+\left\|C\left(x_u;\theta^*\right)-x\right\|_2, \tag{9}$$

where $C\left(x_u;\theta^*\right)$ is the output of the trained network with optimized parameters θ^*. Note that the problem in Equation 9 has the following closed-form solution[14]:

$$y_{rec}(k)=\begin{cases}\frac{[FC(x_u;\theta^*)](k)+\lambda y_u(k)}{1+\lambda}, & \text{if } k\epsilon\Omega\\ \left[FC\left(x_u;\theta^*\right)\right](k), & \text{otherwise}\end{cases},\qquad x_{rec}=F^{-1}y_{rec} \tag{10}$$

where k denotes k-space location; Ω represents the set of acquired k-space locations; F and F^{-1} are the forward and backward Fourier transform operators; and x_{rec} is the reconstructed image. The solution outlined in Equation 10 performs 2 separate projections during reconstruction. The first projection calculates the output of the trained neural network $C\left(x_u;\theta^*\right)$ given the input image x_u, the Fourier reconstruction of undersampled data. The second projection enforces DC. The parameter λ in Equation 10 controls the relative weighting between data samples that are originally acquired and those that are recovered by the network. Here we used $\lambda=\infty$ to enforce DC strictly. Given an input x_{in} in the image domain, the DC projection outlined in Equation 10 can be compactly expressed as 11:

$$f_{DC}\{x_{in}\}=F^{-1}\Lambda F x_{in}+\frac{\lambda}{1+\lambda}x_u, \tag{11}$$

where Λ is a diagonal matrix:

$$\Lambda_{kk}=\begin{cases}\frac{1}{1+\lambda}, & \text{if } k\epsilon\Omega\\ 1, & \text{otherwise}\end{cases}. \tag{12}$$

Conventional optimization algorithms for CS run iteratively to progressively minimize the loss function. A similar approach can also be adopted for NN-based reconstructions.[11,22,39] Here, we cascaded several CNN blocks in series with DC projections interleaved between consecutive CNN blocks.[11] In this architecture, the input x_{ip} to the pth CNN block was formed as:

$$x_{ip}=\begin{cases}x_{un}, & \text{if } p=1\\ f_{DC}\left\{C_{p-1}\left(f_{DC}\left\{C_{p-2}(f_{DC}\dots C_1\left(x_{un};\theta_1^*\right)\right\};\dots\theta_{p-1}^*\right)\right\}, & \text{if } p>1\end{cases}, \tag{13}$$

where θ_p^* denotes the parameters of the pth CNN block. Starting with the initial network with $p=1$, each CNN block was trained sequentially by solving the following optimization problem:

$$\min_{\theta_p}\sum_{n=1}^{\mathrm{N_{train}}}\frac{1}{\mathrm{N_{train}}}\left\|C\left(x_{ip};\theta_p\right)-x_{refn}\right\|_2+\sum_{n=1}^{\mathrm{N_{train}}}\frac{1}{\mathrm{N_{train}}}\left\|C\left(x_{ip};\theta_p\right)-x_{refn}\right\|_1+\gamma_{\Phi}\left\|\theta_p\right\|_2. \tag{14}$$

While training the pth CNN block, the parameters of preceding networks and thus the input x_{ip} are assumed to be fixed.

2.2.2 | Multi-coil data

Similar to SPIRiT, for multi-coil reconstructions, here we reformulate Equation 6 as:

$$x_{rec}=\min_{x}\left\|F_u x-y_u\right\|_2+\|(G-I)Fx\|_2+\left\|C\left(A^*x_u;\theta^*\right)-A^*x\right\|_2, \tag{15}$$

where x denotes the multi-coil images to be reconstructed; A denotes coil-sensitivity profiles using ESPIRiT[49] and A^* denotes its adjoints; and G denotes the interpolation operator in SPIRiT as in Equation 6. The network C has been trained to recover fully sampled coil-combined images given undersampled coil-combined images as outlined in Equation 8. The trained network regularizes the reconstruction in Equation 15 given undersampled coil-combined images A^*x_u. The optimization problem in Equation 15 is solved by alternating projections for CC, DC, and CNN blocks (see Supporting Information Figure S1 for details). CNN blocks are cascaded in series with DC and calibration consistency projections. Given an input x_{in} in the image domain, the calibration-consistency projection can be compactly expressed as:

$$f_{CC}\left\{x_{in}\right\}=F^{-1}GFx_{in}, \tag{16}$$

where F and F^{-1} are the forward and backward Fourier transform operators. Note that the input and output of the CC blocks are in the image domain.

In this multi-coil implementation, the input x_{ip} to the pth CNN block was formed as:

$$x_{ip}=\begin{cases}f_{DC}\left\{f_{CC}\{x_{un}\}\right\}, & \text{if } p=1\\ f_{DC}\{f_{CC}\{f_{DC}\{AC_{p-1}(A^*f_{DC}\{f_{CC}\dots f_{DC}\{AC_1(A^*f_{DC}\{f_{CC}\{x_{un}\}\};\theta_1^*)\};\dots\theta_{p-1}^*)\}\}\}, & \text{if } p>1\end{cases}. \tag{17}$$

Note that CNN blocks receive coil-combined images, and CC and DC blocks receive multi-coil images as input. A^* converts multi-coil images into a coil-combined image, and A back projects the coil-combined image onto individual coils. CC and CNN blocks are both followed by a DC block.

2.3 | Datasets

2.3.1 | Single-coil magnitude images

For demonstrations on single-coil data, 2 distinct types of datasets were used: MR brain images and natural images. The details are listed below.

MR brain images

Training deep neural networks for MR image reconstruction typically requires large datasets containing thousands of images that may be difficult to acquire. Yet, in this study we wanted to systematically examine the interaction between the number of training and fine-tuning samples for domain-transferred neural networks. To comprehensively examine this issue, we opted for the publicly available MIDAS dataset with multi-contrast MR images from nearly 100 subjects.

T_1-weighted images: We assembled a total of 6500 T_1-weighted images (58 subjects) from the MIDAS database.[50] These images were divided into 4580 training images (42 subjects), 720 fine-tuning images (6 subjects), and 1200 testing images (10 subjects). During the training phase, for CNN block training 4000 images (34 subjects) were used for training, and 240 images (2 subjects) were reserved for validation. During the end-to-end network training, 100 images (4 subjects) were used for training, and 240 images (2 subjects) were reserved for validation. During the fine-tuning phase, 480 images (4 subjects) were used for fine-tuning, and 240 images (2 subjects) were reserved for validation. There was no overlap between subjects included in the training, validation, and testing sets. T_1-weighted images analyzed here were collected on a 3T scanner via the following parameters: a 3D gradient-echo sequence, TR = 14 ms, TE = 7.7 ms, flip angle = 25º, matrix size = 256 × 176, 1 mm isotropic resolution.

T_2-weighted images: We assembled a total of 6100 T_2-weighted images (64 subjects) from the MIDAS database.[50] These images were divided into 4500 training images (48 subjects); 600 fine-tuning images (6 subjects); and 1000 testing images (10 subjects), with no subject overlap between training, validation, and testing sets. During the training phase, for CNN block training 4000 images (40 subjects) were used for training, and 200 images (2 subjects) were reserved for validation. During the end-to-end network training, 100 images (4 subjects) were used for training, and 200 images (2 subjects) were reserved for validation. During the fine-tuning phase, 400 images (4 subjects) were used for fine-tuning, and 200 images (2 subjects) were used for validation. T_2-weighted images that were analyzed here were collected on a 3T scanner via the following parameters: a 2D spin-echo sequence, TR = 7730 ms, TE = 80 ms, flip angle = 180º, matrix size = 256 × 192, 1 mm isotropic resolution.

For the fine-tuning phase, images from 4 subjects were reserved. Cross-section images from the reserved subjects were aggregated, and 100 images were randomly selected from within the aggregate set. Therefore, the selected images during the fine-tuning phase contained images from multiple different subjects.

Note that the MIDAS dataset contains DICOM images with only magnitude information. Therefore, all analyses were performed for magnitude-only reconstructions.

Natural images

To perform domain transfer from natural images to single-coil magnitude MR images, we assembled 5100 natural images from the validation set used during the ImageNet Large Scale Visual Recognition Challenge 2011 (ILSVRC2011).[51] Four thousand images were used for training; 100 images were used for end-to-end training; and 1000 images were used for validation. All images were either cropped or zero-padded to yield consistent dimensions of 256 × 256. Color RGB images were first converted to LAB color space using rgb2lab function of MATLAB 2015b, and the L-channel was extracted to obtain grayscale images.

2.3.2 | Multi-coil complex images

MR brain images

The proposed approach was also demonstrated on multi-coil complex k-space data. Images from 10 subjects were acquired. Within each subject, 60 central cross-sections containing sizeable amount of brain tissue were selected. Images were then divided into 360 training images (6 subjects), 60 validation images (1 subject), and 180 testing images (3 subjects), with no subject overlap. Images were collected on a 3T Siemens Magnetom scanner (maximum gradient strength of 45 mT/m and slew rate of 200 T/m/s) using a 32-channel receive-only head coil at Bilkent University, Ankara, Turkey.

1. T_1-weighted images: The images were collected via the following parameters: a 3D MPRAGE sequence, TR = 2000 ms, TE = 5.53 ms, flip angle = 20º, matrix size = 256 × 192 × 80, 1 mm × 1 mm × 2 mm resolution.

2. T_2-weighted images: The images were collected via the following parameters: a 3D spin-echo sequence, TR = 1000ms, TE = 118 ms, flip angle = 90º, matrix size = 256 × 192 × 80, 1 mm × 1 mm × 2 mm resolution.

Imaging protocols were approved by the local ethics committee at Bilkent University, Ankara, Turkey, and all participants provided written informed consent. To reduce computational complexity, geometric-decomposition coil compression was performed to reduce number of coils from 32 to 8.[52]

Natural images

The multi-coil data mentioned in section 2.3.2.1 consisted of complex T_1- and T_2-weighted images acquired on a 3T scanner. However, the ImageNet dataset consisted of magnitude images. Therefore, to perform domain transfer from natural images to multi-coil MR images, complex natural images were simulated from 2420 magnitude images in ImageNet by adding sinusoidal phase at random spatial frequencies along each axis varying from $-\pi$ to $+\pi$. The amplitude of the sinusoids was normalized between 0 and 1. Fully sampled multi-coil T_1-weighted acquisitions from 2 training subjects were selected to extract coil-sensitivity maps using ESPIRiT.[49] Each multi-coil complex natural image was then simulated by utilizing coil-sensitivity maps of a randomly selected cross-section from the 2 reserved subjects (see Supporting Information Figure S2 for sample multi-coil complex natural images). Please note that this phase simulation procedure was also demonstrated to enable successful domain transfer in other recent studies on image reconstruction.[24,40] From the simulated 2420 images, 2000 images were used for initial CNN block training; 360 images were used for end-to-end training; and 60 images were used for validation.

2.3.3 | Single-coil complex images

Single-coil reconstructions on the MIDAS dataset (see subsection 2.3.1) were performed on magnitude images that were Fourier-transformed and undersampled in k-space. To demonstrate the proposed approach on single-coil complex images, we conducted additional experiments using the multi-coil complex MRI data (subsection 2.3.2). To do this, multi-coil images were combined via coil-sensitivity maps estimated using ESPIRiT. For domain transfer from natural images to single-coil complex MR images, complex natural images were synthesized from 2420 ImageNet images by adding sinusoidal phase at random spatial frequencies along each axis varying from $-\pi$ to $+\pi$. Note that for domain transfer experiments in the multi-coil case, natural images were multiplied with coil sensitivity maps estimated from actual MRI data to synthesize multi-coil images. This multiplication intrinsically restricts the spatial extent of objects in natural images. When performing domain transfer in the single-coil complex case, we wanted to match the simulation procedures as closely as possible. Therefore, the synthesized images were spatially restricted by utilizing brain masks extracted from coil sensitivity maps of a randomly selected cross-section from 2 subjects reserved for this purpose. From the simulated 2420 images; 2000 images were used for initial CNN block training; 360 images were used for end-to-end training; and 60 images were used for validation.

Data augmentation is a common method to increase data size for network training. Yet, artificially created samples are inherently correlated with the original samples. Because a central aim of the current study was to examine the interaction between the number of training and fine-tuning samples, no data augmentation was employed to minimize bias due to sample correlation.

Undersampling patterns: Images in each dataset were undersampled via variable-density Poisson-disc sampling.[45] All datasets were undersampled for varying acceleration factors (R = 4, 6, 8, 10). Fully sampled images were first Fourier transformed and then retrospectively undersampled. To ensure reliability against mask selection, 100 unique undersampling masks were generated and used during the training phase. A different set of 100 undersampling masks was used during the testing phase.

2.4 | Network training and fine-tuning

We adopted a cascade of neural networks as inspired by Ref. 11. Five subnetworks were cascaded in series. For single-coil magnitude data, the CNN block within each subnetwork contained an input layer, 4 convolutional layers, and an output layer. The input layer consisted of 2 channels for real imaginary parts of undersampled images. Each convolution operation in the convolutional layers was passed through a rectified linear unit activation. The hidden layers consisted of 64 channels. The output layer consisted of only a single channel for a magnitude reconstruction. For multi-coil complex data, undersampled multi-coil data were combined prior to CNN blocks using coil-sensitivity maps estimated via ESPIRiT. Real and imaginary parts of coil-combined images were then reconstructed using 2 separate networks, and each network consisted of a single input and output channel. The network outputs were joined to form a coil-combined complex image. Note that the DC block operates on individual-coil data. Thus, prior to the DC block, the coil-combined complex image was back-projected onto individual coils, again using coil-sensitivity maps.

2.4.1 | CNN block training

CNN blocks were trained on N$_{\text{train}}$ images in the source domain via the back-propagation algorithm.[53] In the forward passes, a batch of 50 samples in the single-coil case and 10 samples in the multi-coil case were passed through the network to calculate the respective loss function. In the backward passes, network parameters were updated according to the gradients of this function with respect to the parameters. The gradient of the loss function with respect to parameters of the mth hidden layer (θ_m) can be calculated using chain rule:

$$\frac{\partial L}{\partial \theta_m} = \frac{\partial L}{\partial o_l} \frac{\partial o_l}{\partial a_l} \frac{\partial a_l}{\partial \theta_l} \frac{\partial o_{l-1}}{\partial a_{l-1}} \cdots \frac{\partial o_m}{\partial a_m} \frac{\partial a_m}{\partial \theta_m}, \tag{18}$$

where l is the output layer of the network; a_l is the output of the lth layer; and o_l is the output of the lth layer passed through the activation function. The parameters of the mth layer are only updated if the loss-function gradient flows through all subsequent layers (i.e., gradients are non-zero). Each subnetwork was trained individually for 20 epochs. In the CNN block training, the network parameters were optimized using the ADAM optimizer with a learning rate of $\eta = 10^{-4}$, decay rate for first moment of gradient estimates of $\beta_1 = 0.9$, and decay rate for the second moment of gradient estimate of $\beta_2 = 0.999$.[54] Connection weights were L$_2$-regularized with a regularization parameter of $\gamma_\Phi = 10^{-6}$.

2.4.2 | End-to-end network training

Networks formed by sequential training of the CNN blocks were then trained end to end on N$_{\text{end-to-end}}$ images in the source domain. For single-coil magnitude data, this end-to-end training was performed on only 100 images from the source domain (i.e., N$_{\text{end-to-end}}$ = 100). For single-coil and multi-coil complex data, a relatively smaller set of images was used for initial training (360 images); thus, end-to-end training was performed on 360 images from the source domain (i.e., N$_{\text{end-to-end}}$ = 360). In the forward passes, a batch of 20 samples in the single-coil case and 4 samples in the multi-coil case were passed through the network to calculate the respective loss function.

To perform end-to-end training, the gradients must be calculated through the CNN, DC, and CC blocks. The gradient flow through the convolutional network layers that contain basic arithmetic operations and rectified linear unit activation functions are well known.[55] The gradient flow through DC in Equation 11 with respect to its input x_{in} is given as:

$$\frac{\partial f_{DC}}{\partial x_{in}} = F^{-1} \Lambda F \tag{19}$$

due to the linearity of the Fourier operator (F). Similarly, the gradient flow through CC in Equation 16 with respect to its input x_{in} is given as:

$$\frac{\partial f_{CC}}{\partial x_{in}} = GF. \tag{20}$$

Based on Equations 19 and 20, the gradient of the loss function with respect to output of the jth CNN block is given as:

$$\frac{\partial L}{\partial \mathrm{C}_j} = \frac{\partial L}{\partial \mathrm{C}_l} \frac{\partial \mathrm{C}_{l-1}}{\partial f_{DC,(l-1)2}} \frac{\partial f_{DC,(l-1)2}}{\partial f_{CC,l-1}} \frac{\partial f_{CC,l-1}}{\partial f_{DC,(l-1)1}} \frac{\partial f_{DC,(l-1)1}}{\partial \mathrm{C}_{l-1}} \cdots \frac{\partial f_{DC,(j+1)1}}{\partial \mathrm{C}_j}, \tag{21}$$

where l corresponds to the last subnetwork; $f_{DC,(l-1)2}$ corresponds to the DC layer posterior to the $(l-1)$th CC block; and $f_{DC,(l-1)1}$ corresponds to the DC block posterior to the $(l-1)$th subnetwork. Once we have the gradient of the loss function with respect to output of the jth CNN block, the gradients of the mth hidden layer (θ_m) within the jth CNN block can be calculated using chain rule.

$$\frac{\partial L}{\partial \theta_m} = \frac{\partial L}{\partial \mathrm{C}_j} \frac{\partial \mathrm{C}_j}{\partial \theta_l} \frac{\partial \theta_{l-1}}{\partial \theta_{l-2}} \cdots \frac{\partial \theta_{m+1}}{\partial \theta_m}, \tag{22}$$

where l corresponds to the last layer. If we define the gradient $\frac{\partial L}{\partial \theta_m}$ at the kth iteration as g_m^k, estimates of the first and second moments of the gradients at the kth iteration can be expressed as:

$$\begin{aligned} m_m^k &= m_m^{k-1} \beta_1 + \left(1 - \beta_1\right) g_m^k \\ v_m^k &= v_m^{k-1} \beta_2 + \left(1 - \beta_2\right) g_m^{k\,2} \end{aligned}, \tag{23}$$

where m_m^k is the estimate of the first moment of the gradient at the kth iteration; β_1 is the decay rate for m_m^k; v_m^k is the estimate of the second moment of the gradient at the kth iteration; and β_2 is the decay rate for v_m^k. The update for the parameters of the mth hidden layer (θ_m) in the kth iteration can then be expressed as:

$$\theta_m^k = \theta_m^{k-1} - \eta \frac{m_m^k}{\sqrt{v_m^k + \varepsilon}}, \tag{24}$$

where η is the learning rate and ε is a small constant that avoids division by 0 (set to 10^{-8}).

During the end-to-end training phase, the ADAM optimizer was used with identical parameters to those used in the subnetwork training, apart from a lower learning rate of 10^{-5} and a total of 100 epochs.

2.4.3 | Network fine-tuning

A network trained in 1 domain might lead to suboptimal performance in a different target domain. For this purpose, end-to-end fine-tuning was performed on a small number of images from the target domain. We will refer to the number of fine-tuning images as N_{tune}. Gradient calculation and parameter updates were identical to end-to-end network training, as described in subsection 2.4.2.

During the fine-tuning phase, the ADAM optimizer was used with identical parameters to those used in subnetwork training, apart from a lower learning rate of 10^{-5} and a total of 100 epochs.

2.5 | Network validation

During both the training and fine-tuning phases, the number of epochs and learning rate were selected based on reconstruction error (mean absolute error + mean square error) on the validation set. Training and fine-tuning phases exercised early stopping based on network performance on the validation set. During the course of model training, prediction errors will initially decrease on both training and validation sets. Yet, continued training will reduce training error at the expense of elevated validation error. This transition serves as a hallmark symptom of overfitting. To catch the onset of overfitting, we stopped network training based on a convergence criterion. Convergence was taken as the number of epochs in which the percentage change in validation error across consecutive epochs fell below 0.1% of the initial validation error. We found that for CNN block training all CNN blocks converged within 20 epochs, and for end-to-end training all networks converged within 100 epochs (see Supporting Information Figure S3). Learning rate was selected to facilitate convergence while preventing undesirable oscillations in the validation error. We observed the resulting learning rates to be 10^{-4} in the subnetwork training phase, 10^{-5} in the end-to-end training phase, and 10^{-5} in the fine-tuning phase. During the fine-tuning phase, because fine-tuning is performed on a few samples, proper selection of the learning rate is more critical. An excessive learning rate can cause the networks to overfit to the fine-tuning samples. This overfitting can be observed in the form of undesirable oscillations and increase in validation error (see Supporting Information Figure S4).

During the fine-tuning phase, validation data were again used to select the number of epochs and learning rate, and additionally to determine the number of fine-tuning samples required for successful domain transfer. Peak SNR (PSNR) values obtained on the validation images were used to assess domain transfer performance. The PSNR convergence point was used to select the number of fine-tuning samples. Convergence was taken as the number of fine-tuning samples in which the percentage change in PSNR by incrementing number of fine-tuning samples fell below 0.05% of PSNR for the network trained in the target domain.

Both training and fine-tuning phases consisted of separate validation datasets. During the training phase, validation data were exclusively selected from the source domain. For example, the validation set for the ImageNet-trained network contained ImageNet images, whereas the validation set for the T_1-trained network contained T_1-weighted images. In contrast, the validation set during the fine-tuning phase contained data exclusively from the target domain. For example, when T_1 was the target domain, the validation set for both domain-transferred and T_1-trained networks contained an identical set of T_1-weighted images.

2.6 | Performance analyses

2.6.1 | Single-coil magnitude data

We first evaluated the performance of networks under implicit domain transfer (i.e., without fine-tuning in the target domain). We reasoned that a network trained and tested in the same domain should outperform networks trained and tested on different domains. To investigate this issue, we reconstructed undersampled T_1-weighted acquisitions using the ImageNet-trained and T_2-trained networks for varying acceleration factors ($R = 4, 6, 8, 10$). The reconstructions obtained via these 2 networks were compared with reference reconstructions obtained from the network trained directly on T_1-weighted images. To ensure that our results were not biased by the selection of a specific MR contrast as the test set, we also reconstructed undersampled T_2-weighted acquisitions using the ImageNet-trained and T_1-trained networks. The reconstructions obtained via these 2 networks were compared with reference reconstructions obtained from the network trained directly on T_2-weighted images.

Next, we evaluated the performance of network under explicit domain transfer (i.e., with fine-tuning in the target domain). Networks were fine-tuned end to end in the testing domain. When T_1-weighted images were the testing domain, ImageNet-trained and T_2-trained networks were fine-tuned using a small set of T_1-weighted images (N_{tune}) with size ranging in [0 100]. When T_2-weighted images were the testing domain, ImageNet-trained and T_1-trained networks were fine-tuned using a small set of T_2-weighted images (N_{tune}) with size ranging in [0 100]. In both cases, the performance of fine-tuned networks was compared with the networks trained and further fine-tuned end to end directly in the testing domain on N_{tune} images. We also compared the performance of the fine-tuned networks with limited networks that were obtained via end-to-end training only on N_{tune} images.

Reconstruction performance of a fine-tuned network likely depends on the number of both training and fine-tuning images. To examine potential interaction between the number of training and fine-tuning samples, separate networks were trained using training sets of varying size (N_{train}) in [500 4000]. Each network was then fine-tuned using sets of varying size (N_{tune}) in [0 100]. Performance was evaluated to determine the number of fine-tuning samples that are required to achieve near-optimal performance for each separate size of training set. Optimal performance was taken as the PSNR of a network trained directly in the testing domain.

Please note that for all aforementioned analyses, networks were also end-to-end trained using a set of 100 images in the source domain (i.e., $N_{end\text{-}to\text{-}end} = 100$).

NN-based reconstructions were also compared to those obtained by conventional CS (SparseMRI).[2] Single-coil CS reconstructions were implemented via a nonlinear conjugate gradient method. Daubechies-4 wavelets were selected as the sparsifying transform. Parameter selection was performed to maximize PSNR on the validation images from the fine-tuning set. Consequently, an L_1-regularization parameter of 10^{-3}, 80 iterations for T_1-weighted acquisitions, and 120 iterations for T_2-weighted acquisitions were observed to yield near-optimal performance broadly across R.

2.6.2 | Multi-coil complex data

We also demonstrated the proposed approach on multi-coil MR images. For this purpose, a network was trained in which initial CNN block training was performed on 2000 (N_{train}) multi-coil complex natural images, and end-to-end training was performed on 360 ($N_{end\text{-}to\text{-}end}$) additional multi-coil complex natural images (see section 2.3.2 for details). The network was then fine-tuned using a set of multi-coil images (N_{tune}) from the target domain (T_1- or T_2-weighted) with varying size in [0 100]. Here, cross-sections from the training set in the target domain were aggregated, and 100 images were then randomly selected. Reconstruction performance was compared with networks trained using 360 multi-coil MR images from the target domain (6 subjects) and L_1-SPIRiT.[45] A projection onto convex sets implementation of SPIRiT was used. For each R, parameter selection was performed to maximize PSNR on validation images drawn from the multi-coil MR image dataset. For T_1-weighted images, an interpolation kernel width of 7, a Tikhonov regularization parameter of 10^{-2} for calibration, and an L_1-regularization parameter of 10^{-3} were observed to yield near-optimal performance across R. Meanwhile, the optimal number of iterations varied based on acceleration factor. For R = [4, 6, 8, 10], the following number of iterations = [30, 45, 65, 80] were selected. For T_2-weighted images, an interpolation kernel width of 7, a Tikhonov regularization parameter of 10^{-2} for calibration, and an L_1-regularization parameter of 10^{-4} were observed to yield near-optimal performance across R. Meanwhile, the optimal number of iterations varied based on acceleration factor. For R = [4, 6, 8, 10], the following number of iterations = [45, 70, 80, 80] were selected. The interpolation kernels optimized for SPIRiT were used in the calibration-consistency blocks of the networks that contained 5 consecutive CC projections.

We also inspected the degree of change in model weights following fine-tuning. To inspect the changes, we computed percentage change in coefficients of convolution kernels in CNN layers for R = 4-10. Measurements were averaged across neurons within each layer and across R.

2.6.3 | Single-coil complex data

We also demonstrated the proposed approach on single-coil complex images. For this purpose, initial CNN block training was performed on 2000 (N_{train}) synthetic single-coil complex natural images, and end-to-end training was performed on 360 ($N_{end\text{-}to\text{-}end}$) additional synthetic single-coil complex natural images (see section 2.3.3 for details). The network was then fine-tuned using a set of single-coil complex images (N_{tune}) from the target domain (T_1- or T_2-weighted) with varying size in [0 100]. Here, cross-sections from the training set in the target domain were aggregated, and 100 images were then randomly selected. Reconstruction performance was compared with networks trained using 360 single-coil MR complex images from the target domain (6 subjects).

To quantitatively compared alternative methods, we measured the structural similarity index (SSIM) and PSNR between the reconstructed and fully sampled reference images. For multi-coil data, the reference image was taken as the coil-combined image obtained via weighted linear combination using coil sensitivity maps from ESPIRiT. The training and testing of NN architectures were performed in the TensorFlow framework[56] using 2 NVIDIA Titan X Pascal GPUs (12 GB video RAM). Single-coil CS reconstructions were performed via libraries in the SparseMRI V0.2 toolbox available at https://people.eecs.berkeley.edu/~mlustig/Software.html. Multi-coil CS reconstructions were performed via libraries in the SPIRiT V0.3 toolbox available at https://people.eecs.berkeley.edu/~mlustig/Software.html.

3 | RESULTS

3.1 | Single-coil magnitude data

3.1.1 | T_1-domain transfer

A network trained on the same type of images with which it later will be tested should outperform networks that are trained and tested on different types of images. However, this performance difference should diminish following successful domain transfer between the training and testing domains. To test this prediction, we first investigated generalization performance for implicit domain transfer (i.e., without fine-tuning) in a single-coil setting. The training domain contained natural images from the ImageNet database or T_2-weighted images, and the testing domain contained T_1-weighted images. Figure 2 displays reconstructions of an undersampled T_1-weighted acquisition via the ImageNet-trained, T_2-trained, and T_1-trained networks for R = 4. As expected, the T_1-trained network yields sharper and more accurate reconstructions compared to the raw ImageNet–trained and T_2-trained networks. Next, we examined explicit domain transfer in which ImageNet-trained and T_2-trained networks were fine-tuned. In this case, all networks yielded visually similar reconstructions. Furthermore, when compared against conventional compressive sensing (CS), all network models yielded superior performance. Figure 3 displays reconstructions of an undersampled T_1-weighted acquisition via the ImageNet-trained, T_2-trained, and T_1-trained networks, and CS for R = 4. The ImageNet-trained network produces images of similar visual quality to other networks and outperforms CS in terms of image sharpness and residual aliasing artifacts.

Reconstruction performance of domain-transferred networks may depend on the sizes of both training and fine-tuning sets. To examine interactions between the number of training (N_{train}) and fine-tuning (N_{tune}) samples, we trained networks using training sets in the range [500 4000] and fine-tuning sets in the range [0 100]. Figure 4 shows average PSNR values for a reference T_1-trained network trained on 4000 and fine-tuned on 100 images, and domain-transferred networks for R = 4-10. Without fine-tuning, the T_1-trained network outperforms both domain-transferred networks. As the number of fine-tuning samples increases, the PSNR differences decay gradually to a negligible level. Consistently across R, domain-transferred networks trained on smaller training sets require more fine-tuning samples to yield similar performance.

Figure 5 displays the number of fine-tuning samples required for the PSNR values for ImageNet-trained networks to converge for R = 4-10. Convergence was taken as the number of fine-tuning samples in which the percentage change in PSNR by incrementing number of fine-tuning samples fell below 0.05% of PSNR for the T_1-trained network. Across R, networks trained on fewer samples require more fine-tuning samples for convergence. However, the required number of fine-tuning samples is greater for higher R. Averaged across R, N_{tune}= 68 for N_{train} = 500; N_{tune} = 72 for N_{train} = 1000; N_{tune} = 35 for N_{train} = 2000; and N_{tune} =3 8 for N_{train} = 4000.

To corroborate the visual observations, reconstruction performance was quantitively assessed for both implicit and explicit domain transfer across R = 4-10. PSNR and SSIM measurements across the test set are listed in Table 1 and Supporting Information Table S1. (For reconstruction performance when N_{tune} is fixed to 100, please refer to Supporting Information Table S3.) For implicit domain transfer, the T_1-trained networks outperform domain-transferred networks and CS consistently across all R. For explicit domain transfer, the differences between the T_1-trained and domain-transferred networks diminish. Following fine-tuning, the average differences in (PSNR, SSIM) across R between ImageNet and T_1-trained networks diminish from (1.61 dB, 1.50%) to (0.35 dB, 0.50%), and difference between T_2-trained and T_1-trained networks diminish from (1.96 dB, 2.50%) to (0.20 dB, 0.25%). Furthermore, the domain-transferred networks outperform CS consistently across R by an average of 3.70 dB PSNR and 6.13% SSIM and outperform limited networks by an average of 7.63 dB PSNR and 8.38% SSIM.

3.1.2 | T_2-domain transfer

Next, we repeated the analyses for implicit and explicit domain transfer when the testing domain contained T_2-weighted images. Supporting Information Figure S5 displays reconstructions of an undersampled T_2-weighted acquisition via the ImageNet-, T_1-, and T_2-trained networks for acceleration factor R = 4. Again, the network trained directly in the testing domain (T_2-weighted) outperforms domain-transferred networks. After fine-tuning with as few as 20 images, the domain-transferred networks yield visually similar reconstructions to the T_2-trained network. Supporting Information Figure S6 displays reconstructions of an undersampled T_2-weighted acquisition via the ImageNet-trained, T_2-trained, and T_1-trained networks, and CS for R = 4. The ImageNet-trained network produces images of similar visual quality to other networks and outperforms CS in terms of image sharpness and residual aliasing artifacts.

We also examined interactions between the number of training and fine-tuning samples when the target domain contained T_2-weighted images. Supporting Information Figure S7 shows average PSNR values for a reference T_2-trained network trained on 4000 and fine-tuned on 100 images, and

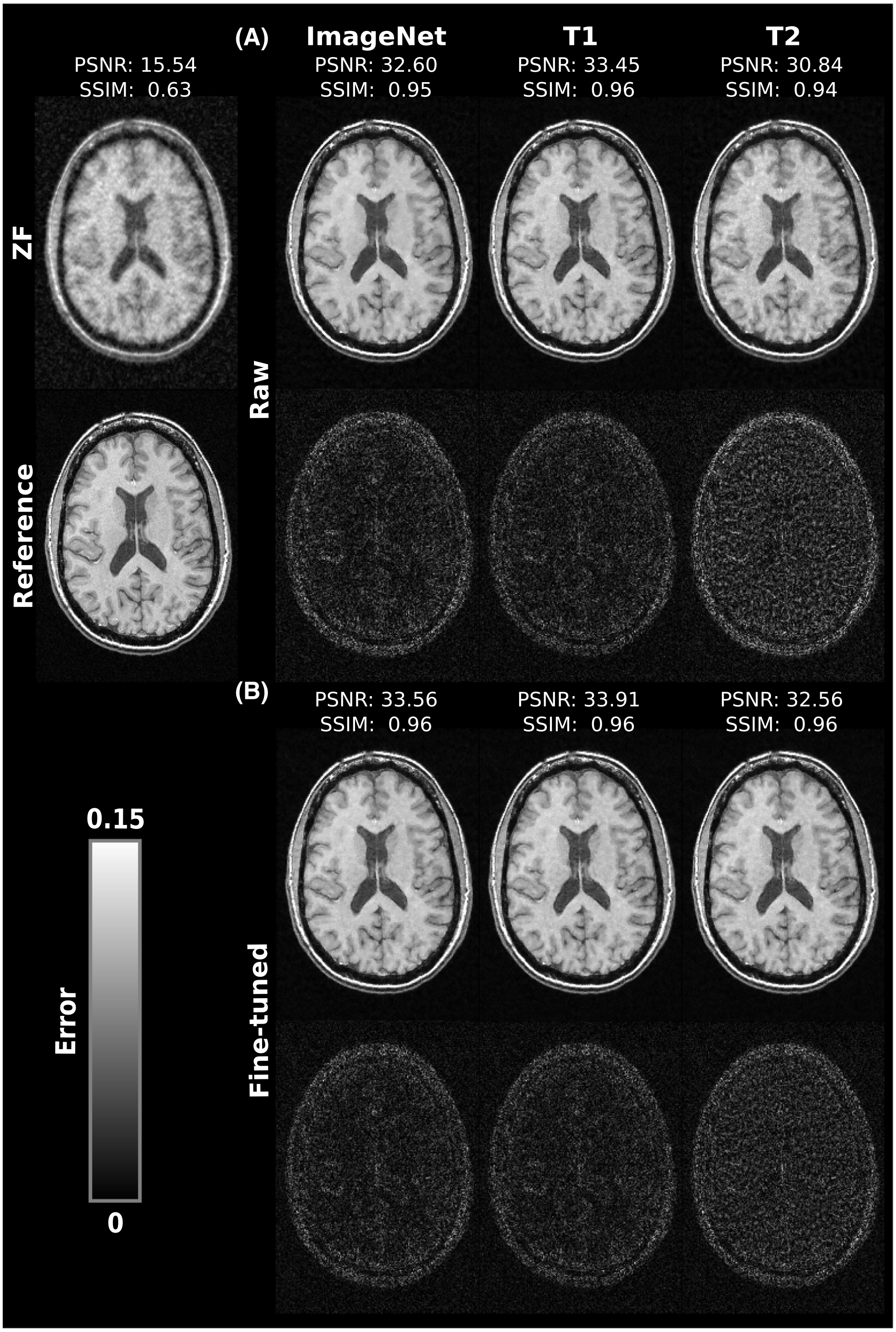

FIGURE 2 Representative reconstructions of a T_1-weighted acquisition at acceleration factor R = 4. Reconstructions were performed via the ZF method and ImageNet-trained, T_2-trained, and T_1-trained networks. (A) Reconstructed images and error maps for raw networks (see color bar). (B) Reconstructed images and error maps for fine-tuned networks. The fully sampled reference image is also shown. Network training was performed on a training dataset of 2000 images and fine-tuned on a sample of 20 T_1-weighted images. Following fine-tuning, ImageNet-trained and T_2-trained networks yield reconstructions of highly similar quality to the T_1-trained network. ZF, zero-filled Fourier reconstruction

domain-transferred networks for R = 4-10. Compared to the case of T_1-weighted images, interaction between number of training and fine-tuning samples is weaker. Yet, a greater number of fine-tuning samples is still required for reconstructions at higher R. Supporting Information Figure S8 displays the number of fine-tuning samples required for convergence of ImageNet-trained networks. Averaged across R = 4-10, N_{tune} = 53 for N_{train} = 500;

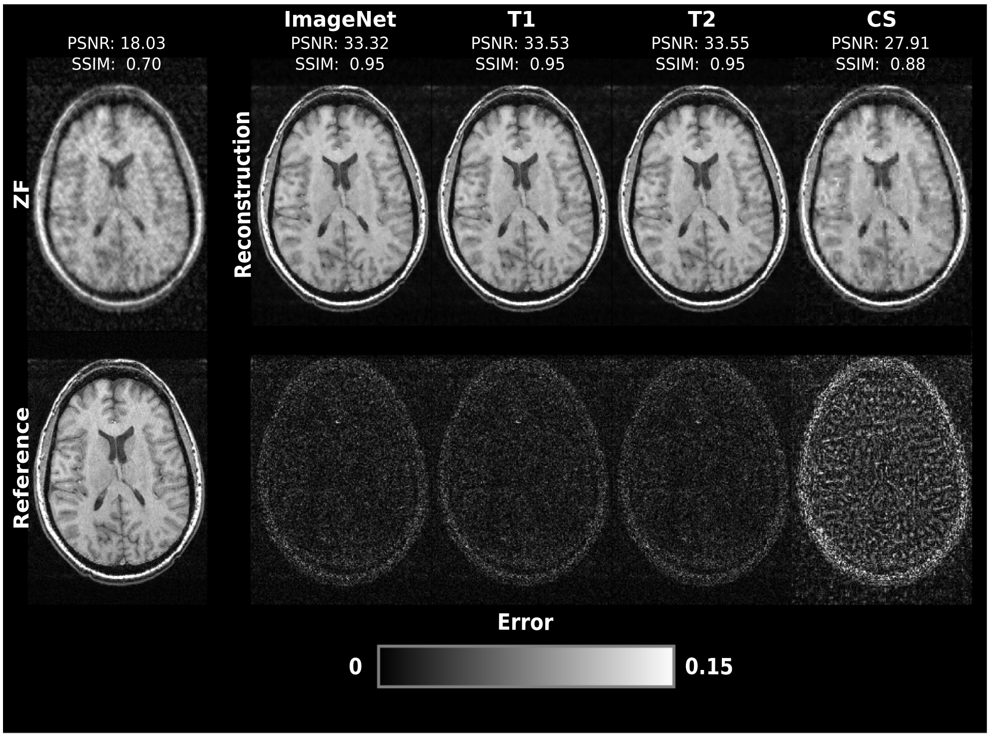

FIGURE 3 Reconstructions of a T_1-weighted acquisition with R = 4 via ZF; conventional CS; and ImageNet-trained, T_1-trained and T_2-trained networks along with the fully sampled reference image. Error maps for each reconstruction are shown below (see color bar). Networks were trained on 2000 images and fine-tuned on 20 images acquired with the test contrast. The domain-transferred networks maintain nearly identical performance to the networks trained directly in the testing domain. Furthermore, the domain-transferred networks reconstructions outperform conventional CS in terms of image sharpness and residual aliasing artifacts. CS, compressed sensing

$N_{tune} = 50$ for $N_{train} = 1000$; $N_{tune} = 48$ for $N_{train} = 2000$; and $N_{tune} = 49$ for $N_{train} = 4000$.

PSNR and SSIM measurements on T_2-weighted reconstructions across the test set are listed in Supporting Information Table S2. (For results using $N_{tune} = 100$, please refer to Supporting Information Table S4.) Following fine-tuning, average (PSNR, SSIM) differences between ImageNet and T_2-trained networks diminish from (1.23 dB, 2.00%) to (0.29 dB, 0.50%), and difference between T_1-trained and T_2-trained networks diminish from (0.67 dB, 1.25%) to (0.15 dB, 0.50%). Across R, the domain-transferred networks also outperform CS by 5.20 dB PSNR and 8.50% SSIM, and limited networks by 2.67 dB PSNR and 1.50% SSIM.

3.2 | Multi-coil complex data

3.2.1 | T_1-domain transfer

Next, we demonstrated the proposed approach on multi-coil T_1-weighted images. We compared ImageNet- and T_1-trained networks at R = 4-10. Figure 6 displays average PSNR values for the T_1-trained network (trained and fine-tuned on 360 images) and ImageNet-trained network ($N_{train} = 2000$ and $N_{end\text{-}to\text{-}end} = 360$ multi-coil natural images, and $N_{tune} \in [0, 100]$ T_1-weighted images). As N_{tune} increases, the PSNR differences between T_1- and ImageNet-trained networks start diminishing. Figure 7 displays the number of fine-tuning samples required for the PSNR values for ImageNet-trained networks to converge. Averaged across R = 4-10, ImageNet-trained networks require $N_{tune} = 18$ for convergence. We also compared the proposed transfer learning approach with L_1-regularized SPIRiT. Figure 8 shows representative reconstructions obtained via the ImageNet-trained network, T_1-trained network, and SPIRiT for R = 10. The ImageNet-trained network produces images of similar visual quality to the T_1-trained network and outperforms SPIRiT in terms of residual aliasing artifacts.

Quantitative assessment of multi-coil reconstructions for the ImageNet-trained network, T_1-trained network, and SPIRiT across R = 4-10 are listed in Table 2. For implicit domain transfer, the T_1-trained network performs better than the ImageNet-trained network. Following fine-tuning, the average differences in (PSNR, SSIM) across R between ImageNet and T_1-trained networks diminish from (2.10 dB, 1.43%) to (0.62 dB, 0.15%). Furthermore, the ImageNet-trained network outperforms SPIRiT in all cases. On average across R, the ImageNet-trained network improves performance over SPIRiT by 0.93 dB PSNR and 0.60% SSIM.

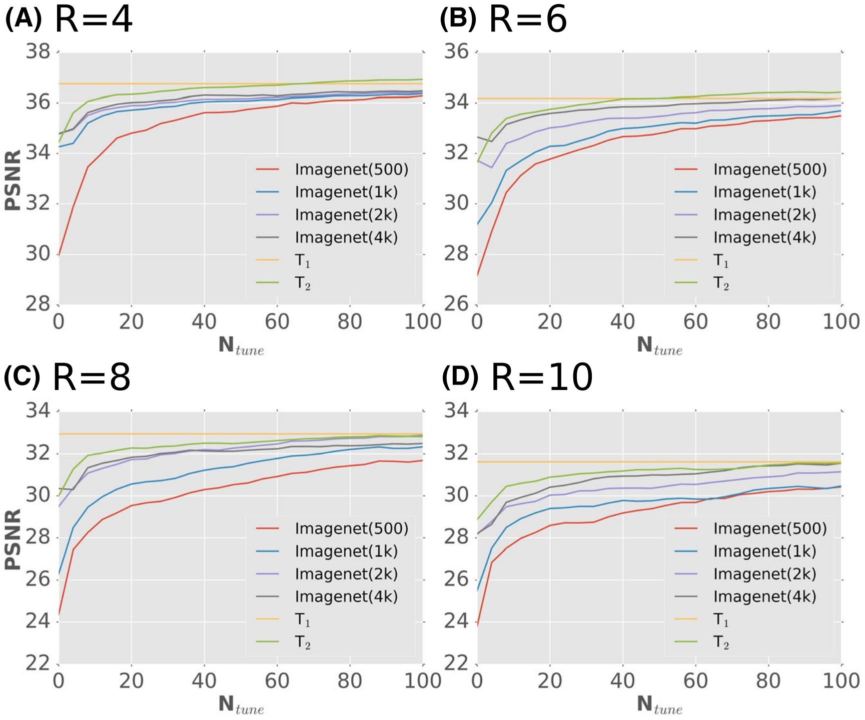

FIGURE 4 Reconstruction performance was evaluated for undersampled T_1-weighted acquisitions. Average PSNR values across T_1-weighted validation images were measured for the T_1-trained network (trained on 4k images and fine-tuned on 100 images), ImageNet-trained networks (trained on 500, 1000, 2000, or 4000 images), and T_2-trained network (trained on 4000 images). Results are plotted as a function of number of fine-tuning samples for acceleration factors (A) R = 4, (B) R = 6, (C) R = 8, and (D) R = 10. Without fine-tuning, the T_1-trained network outperforms all domain-transferred networks. As the number of fine-tuning samples increases, the PSNR differences decay gradually to a negligible level. Domain-transferred networks trained on fewer samples require more fine-tuning samples to yield similar performance consistently across R. PSNR, peak SNR

3.2.2 | T_2-domain transfer

We also demonstrated the proposed approach when the testing domain contained multi-coil T_2-weighted images. Supporting Information Figure S9 displays average PSNR values for the T_2-trained network (trained and fine-tuned on 360 images) and ImageNet-trained network (N_{train} = 2000 and $N_{end\text{-}to\text{-}end}$ = 360 multi-coil natural images, and $N_{tune} \in [0, 100]$ T_2-weighted images). Supporting Information Figure S10 displays the number of fine-tuning samples required for the ImageNet-trained network to converge. Averaged across R, ImageNet-trained networks require N_{tune} = 28 for convergence. Supporting Information Figure S11 shows representative reconstructions obtained via the ImageNet-trained network, T_2-trained network, and SPIRiT for R = 10. The ImageNet-trained network produces images of similar visual quality to the T_2-trained network while outperforming SPIRiT in terms of residual artifacts. Meanwhile, Supporting Information Table S5 lists quantitative assessments of reconstruction quality across R = 4-10. Following fine-tuning, the average differences in (PSNR, SSIM) between ImageNet and T_2-trained networks diminish from (2.33 dB, 0.70%) to (0.37 dB, 0.05%). On average across R, the ImageNet-trained network improves performance over SPIRiT by 1.53 dB PSNR and 1.07% SSIM.

We also computed percentage change in coefficients of convolution kernels in CNN layers for R = 4-10. Supporting Information Figure S12 demonstrates percentage change in network weights as a function of network depth for multi-coil ImageNet to T_1 and T_2 domain transfer, averaged across R. Overall, the percentage change in weights is higher for earlier versus later layers of the network. For ImageNet to T_1 domain transfer, percentage change varies from 2.27% to 0.56%, and for ImageNet to T_2 domain transfer percentage change varies from 3.28% to 0.47%. The difference in the level of weight change across layers can be attributed to the level of residual artifacts present in inputs to each layer. Because the inputs to earlier layers contain more domain-specific residual artifacts, they might undergo greater change during fine-tuning compared to later layers.

3.3 | Single-coil complex data

3.3.1 | T_1-domain transfer

Next, we demonstrated the proposed approach on single-coil complex T_1-weighted images, obtained by combining

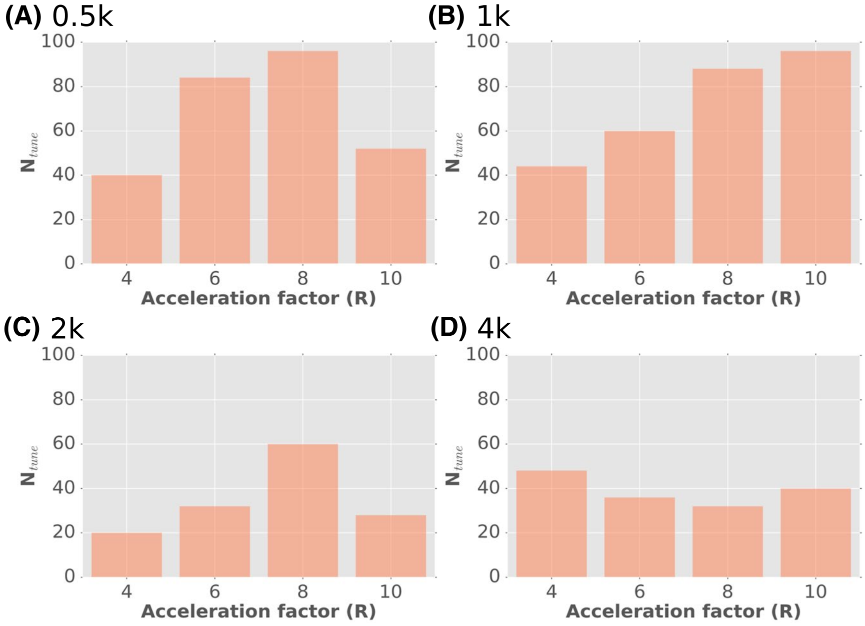

FIGURE 5 Number of fine-tuning samples required for the PSNR values for ImageNet-trained networks (trained on single-coil magnitude images) to converge. Average PSNR values across T_1-weighted validation images were measured for the ImageNet-trained networks trained on (A) 500, (B) 1000, (C) 2000, and (D) 4000 images. Convergence was taken as the number of fine-tuning samples where the percentage change in PSNR by incrementing N_{tune} fell below 0.05% of the average PSNR for the T_1-trained network (see Figure 4). Domain-transferred networks trained on fewer samples require more fine-tuning samples for the PSNR values to converge. Furthermore, at higher values of R, more fine-tuning samples are required for convergence

multi-coil images via coil sensitivity maps estimated using ESPIRiT. Supporting Information Figure S13 displays average PSNR values for the T_1-trained network (trained and fine-tuned on 360 images) and ImageNet-trained network (N_{train} = 2000 and $N_{end\text{-}to\text{-}end}$ = 360 single-coil natural images, and $N_{tune} \in [0, 100]$ T_1-weighted images); and Supporting Information Figure S14 displays the number of fine-tuning samples required for the ImageNet-trained network. Averaged across R, ImageNet-trained networks require N_{tune} = 29 for convergence. Quantitative assessment for the ImageNet-trained network and T_1-trained network across R = 4-10 are listed in Supporting Information Table S6. Following fine-tuning, the average differences in (PSNR, SSIM) across R between ImageNet and T_1-trained networks diminish from (4.23 dB, 5.95%) to (1.65 dB, 1.25%).

3.3.2 | T_2-domain transfer

Finally, we demonstrated the proposed approach when the testing domain contained single-coil complex T_2-weighted images. Supporting Information Figure S15 displays average PSNR values for the T_2-trained network (trained and fine-tuned on 360 images) and ImageNet-trained network (N_{train} = 2000 and $N_{end\text{-}to\text{-}end}$ = 360 single-coil natural images, and $N_{tune} \in [0, 100]$ T_2-weighted images); and Supporting Information Figure S16 displays the number of fine-tuning samples required for the ImageNet-trained network. Averaged across R, ImageNet-trained networks require N_{tune} = 42 for convergence. Quantitative assessment for the ImageNet-trained network and T_2-trained network across R = 4-10 are listed in Supporting Information Table S7. Following fine-tuning, the average differences in (PSNR, SSIM) across R between ImageNet T_2-trained networks diminish from (2.48 dB, 2.00%) to (0.79 dB, 0.50%).

4 | DISCUSSION

Neural networks for MRI reconstruction involve many free parameters to be learned; thus, an extensive amount of training samples is typically needed.[57] In theory, network performance should be optimized by drawing the training and testing samples from the same domain acquired under a common MRI protocol. In practice, however, compiling large public datasets can require coordinated efforts among multiple imaging centers; therefore, such datasets are rare. As an alternative, several recent studies trained neural networks on a collection of multi-contrast images.[25] When needed, data augmentation procedures were used to further expand the training dataset.[11,12] Although these approaches gather more samples for training, it remains unclear how well a network trained on images acquired with a specific type of tissue contrast generalizes to images acquired with different contrasts. Thus, variability in MR contrasts can lead to suboptimal reconstruction performance.

TABLE 1 Reconstruction quality for single-coil magnitude T_1-weighted images undersampled at R = 4, 6, 8, 10. Reconstructions were performed via ImageNet-trained, T_1-trained, T_2-trained, and limited networks. PSNR and SSIM values are reported as mean ± SD across test images. Results are shown for raw networks trained on 2000 training images (raw) and fine-tuned networks tuned with tens of T_1-weighted images (tuned)

		ImageNet-Trained		**T_1-Trained**		**T_2-Trained**	
		PSNR	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**
R = 4	Raw	34.92 ±3.57	0.96 ± 0.02	36.01 ± 3.17	0.97 ± 0.02	34.11 ± 3.64	0.95 ± 0.03
	Tuned	36.06 ± 3.10	0.97 ± 0.01	36.47 ± 3.27	0.97 ± 0.01	36.22 ± 3.13	0.97 ± 0.02
		Limited					
		PSNR			**SSIM**		
		29.79 ± 3.95			0.93 ± 0.03		
R = 6	Raw	31.97 ± 3.33	0.94 ± 0.02	33.49 ± 3.29	0.95 ± 0.02	31.27 ± 3.78	0.93 ± 0.04
	Tuned	33.53 ± 3.02	0.95 ± 0.02	33.99 ± 3.29	0.96 ± 0.02	33.71 ± 3.15	0.96 ± 0.02
		Limited					
		PSNR			**SSIM**		
		27.02 ± 4.35			0.90 ± 0.05		
R = 8	Raw	29.78 ± 3.75	0.92 ± 0.03	31.64 ± 3.37	0.94 ± 0.03	29.72 ± 3.75	0.91 ± 0.05
	Tuned	32.56 ± 3.15	0.95 ± 0.02	32.32 ± 3.36	0.95 ± 0.03	32.39 ± 3.45	0.95 ± 0.02
		Limited					
		PSNR			**SSIM**		
		24.02 ± 4.46			0.85 ± 0.07		
R = 10	Raw	28.52 ± 3.86	0.91 ± 0.04	30.50 ± 3.37	0.93 ± 0.03	28.70 ± 3.70	0.90 ± 0.05
	Tuned	30.37 ± 3.34	0.93 ± 0.03	31.12 ± 3.38	0.94 ± 0.03	30.78 ± 3.14	0.93 ± 0.03
		Limited					
		PSNR			**SSIM**		
		21.45 ± 4.52			0.79 ± 0.08		

Abbreviations: PNSR, peak SNR; SSIM, structural similarity index.

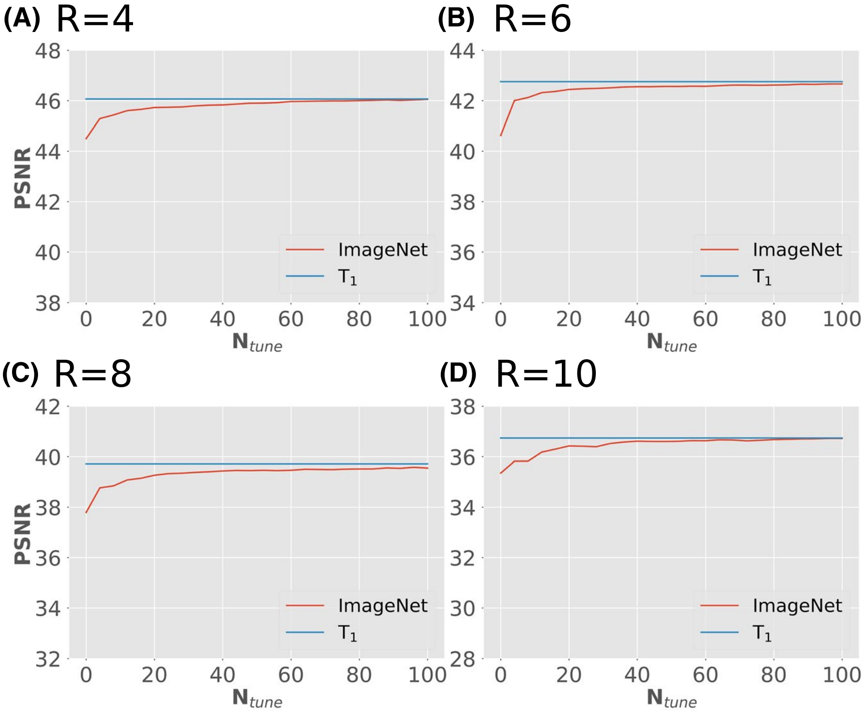

FIGURE 6 Reconstruction performance was evaluated for undersampled multi-coil T_1-weighted acquisitions. Average PSNR values across T_1-weighted validation images were measured for the T_1-trained network (trained and fine-tuned on 360 images) and ImageNet-trained network trained on 2000 images. Results are plotted as a function of number of fine-tuning samples for acceleration factors (A) R = 4, (B) R = 6, (C) R = 8, and (D) R = 10. Without fine-tuning, the T_1-trained network outperforms the domain-transferred network. As the number of fine-tuning samples increases, the PSNR differences decay gradually to a negligible level

Here, we first questioned the generalizability of neural network models across different contrasts. We find that a network trained on MR images of a given contrast (e.g., T_1-weighted) yields suboptimal reconstructions on images of a different contrast (e.g., T_2-weighted). This confirms that the best strategy is to train and test networks in the same domain. Yet, it may not be always feasible to gather a large collection of images from a desired contrast. To address the problem of data scarcity, we proposed a transfer-learning approach for accelerated MRI. The proposed approach trains neural networks using training samples from a large public dataset of natural images. The network is then fine-tuned end to end using only tens of MR images. Reconstructions obtained via the ImageNet-trained network are of nearly identical quality to reconstructions obtained by networks trained directly in the testing domain using thousands (upto 4 thousand) of MR images.

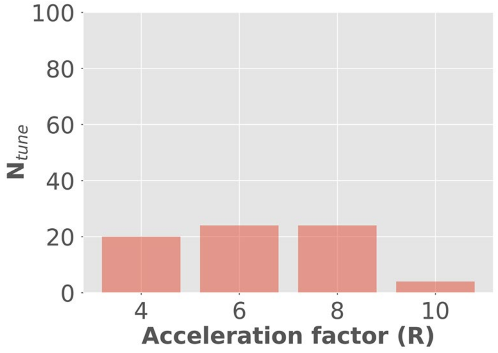

FIGURE 7 Number of fine-tuning samples required for the PSNR values for ImageNet-trained networks (trained on multi-coil complex images) to converge. Average PSNR values across T_1-weighted validation images were measured for the ImageNet-trained network trained on 2000 images. Convergence was taken as the number of fine-tuning samples where the percentage change in PSNR by incrementing N_{tune} fell below 0.05% of the average PSNR for the T_1-trained network (see Figure 6). At higher values of R, more fine-tuning samples are required for convergence

In the current study, we proposed an explicit domain transfer approach in which networks are initially trained using a large number of images in a source domain and then fine-tuned using fewer samples in the target domain. If the source and target domains were structurally dissimilar, the domain-transferred networks would not be expected to perform successfully in the target domain. Note that previous studies reported natural and MR images to have similar early-to-intermediate visual features[40] and similar energy spectrums in the Fourier domain.[2,58] Thus, the comparable performance of domain-transferred and target-domain networks here can be attributed to such shared visual features. In T_1 reconstructions, we observed that the ImageNet-trained and T_2-trained networks performed similarly at R = (8, 10); however, the ImageNet-trained network was superior at relatively low R = (4, 6). At high R, high-spatial-frequency samples in the target domain are largely missing, and the networks aim to synthesize missing samples primarily based on low-frequency samples. Thus, the general similarity of energy spectrums between natural and MR images might lead to similar performance for ImageNet- and T_2-trained networks. At low R, however, additional high-frequency information is available in the target domain, and success of implicit

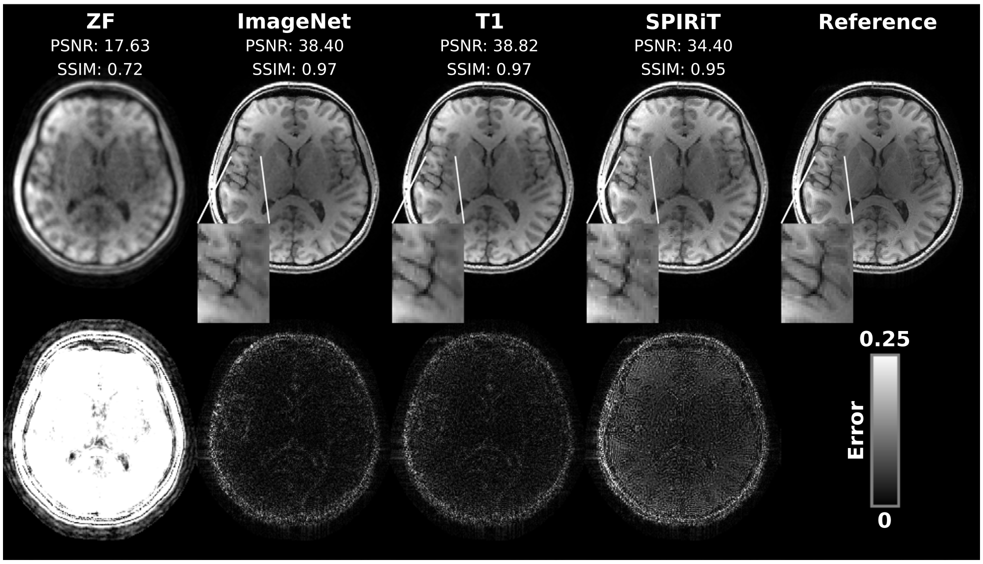

FIGURE 8 Representative reconstructions of a multi-coil T_1-weighted acquisition at acceleration factor R = 10. Reconstructions were performed via ZF, ImageNet-trained and T_1-trained networks, and SPIRiT (top row). Corresponding error maps are also shown (see color bar; bottom row) along with the fully sampled reference (top row). Network training was performed on a training dataset of 2000 images and fine-tuned on a sample of 20 T_1-weighted images. The ImageNet-trained network maintains similar performance to the T_1-trained network trained directly on the images from the test domain. Furthermore, the domain-transferred network outperforms conventional SPIRiT in terms of residual aliasing artifacts. SPIRiT, iterative self-consistent parallel imaging reconstruction

TABLE 2 Reconstruction quality for multi-coil complex T_1-weighted images undersampled at R = 4, 6, 8, 10. Reconstructions were performed via ImageNet-trained and T_1-trained networks as well as SPIRiT. PSNR and SSIM values are reported as mean ± SD across test images. Results are shown for raw networks trained on 2000 training images (raw) and fine-tuned networks tuned with tens of T_1-weighted images (tuned)

		ImageNet-Trained		T_1-Trained		SPIRiT	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
R = 4	Raw	43.48 ± 1.89	0.984 ± .006	45.36 ± 1.75	0.989 ± .004	44.60 ± 1.75	0.987 ± .004
	Tuned	44.82 ± 1.77	0.989 ± .004				
R = 6	Raw	39.77 ± 1.84	0.968 ± .010	42.06 ± 1.85	0.981 ± .006	40.62 ± 1.73	0.975 ± .007
	Tuned	41.72 ± 1.87	0.980 ± .007				
R = 8	Raw	36.64 ± 1.66	0.951 ± .014	39.14 ± 1.75	0.971 ± .009	37.11 ± 1.71	0.961 ± .012
	Tuned	38.50 ± 1.77	0.969 ± .010				
R = 10	Raw	34.46 ± 1.59	0.941 ± .016	36.19 ± 1.85	0.960 ±.012	34.23 ± 1.72	0.948 ± .017
	Tuned	35.24 ± 1.82	0.957 ± .013				

Abbreviation: SPIRiT, iterative self-consistent parallel imaging reconstruction.

domain transfer might rely more critically on the similarity of high-frequency structure between source and target domains. In the datasets reported here, natural and T_1-weighted images have sharper object boundaries, whereas T_2-weighted images have broadened tissue transitions. This difference might have contributed to the superior performance of the ImageNet-trained network at low R. Please also note that in T_2 reconstructions the T_1-trained networks were superior to ImageNet-trained network at all acceleration factors. These results suggest that T_2-weighted images are structurally closer to T_1-weighted images than to natural images.

Here, we demonstrated successful domain transfer from natural images to brain MR images. A future research direction is to examine the success of this approach for domain transfer between MR images of different organs. In these images, the anatomy of interest might occupy different portions of the FOV due to inherent shape and size differences among organs. These differences might in turn limit reconstruction performance of domain-transferred networks. That said, our domain-transfer experiments from natural to MR images indicate that such performance loss is not significant. In the single-coil magnitude case, we demonstrated successful domain transfer with ImageNet-trained networks that were trained on natural images spanning across the entire FOV. This result implies that the proposed transfer learning approach should generalize well between images of different organs.

An important concern for complex models trained on relatively restricted datasets is overfitting. Several precautions were employed here to minimize potential bias due to overfitting. First, a limited learning rate was used to fine-tune domain-transferred networks in the target domain to limit the range of parameter updates. Second, early stopping based on validation errors were used during both the training and fine-tuning phases. Lastly, a cross-validation procedure was used with a 3-way split of training/validation and test data for which test data are exclusively reserved for assessment of model performance. Note that the high performance of domain-transferred networks reported here imply that networks are not unduly biased by overfitting.

Another important concern is characteristic failures of domain-transferred networks such as hallucination of features from the source domain. Hallucination poses an important limitation particularly for generative network architectures designed to draw new samples of data from a learned distribution.[59,60] Unlike generative models, the deterministic CNN architectures are considered less prone to hallucination, and no significant hallucination was observed in the reconstructions reported here. However, reconstruction artifacts were visible for networks under implicit domain transfer, and these artifacts were alleviated following fine-tuning in the target domain.

Several recent studies have considered domain transfer to enhance performance in NN-based MRI reconstruction.[14,15,24,40,61] A group of studies have aimed to perform implicit domain transfer across MRI contrasts without fine-tuning. One proposed method was to train networks on MR images in a given contrast and then to directly use the trained networks on images of different contrasts.[24] Although this method yields successful reconstructions, our results suggest that network performance can be further boosted with additional fine-tuning in the testing domain. Another method to enhance generalizability was to compound datasets containing a mixture of distinct MRI contrasts during network training.[14] This approach enforces the network to better adapt to variations in tissue contrast. Yet, in the absence of contrast-specific fine-tuning, networks may deliver suboptimal performance for some individual contrasts. A recent study proposed implicit domain transfer from natural images to MR images.[61] Our study differs from Ref. 61 in several ways: First, we propose explicit domain transfer via end-to-end fine-tuning in the target domain, which is shown to significantly enhance success of domain-transferred networks.

Second, we introduce pretraining and domain-transfer approaches capable of reconstructing multi-coil MRI data. Third, unlike Ref. 61, where the number of training samples was 118k in the natural-image domain versus 0.3k to 1k in the MRI domain, we here maintain comparable sizes of training data across different domains with the ratio of natural to MR images ranging in [1 5.5]. Fourth, unlike Ref. 61, which that reports domain-transferred networks to occasionally outperform networks trained in the target domain, networks obtained via implicit domain transfer perform suboptimally in all cases examined here. This apparent contradiction might be attributed to the differences in the relative size of training data between the natural image and MRI domains.

A second group of studies have attempted explicit domain transfer across training and testing domains via fine-tuning. A recent proposed method trained a deep residual network to remove streaking artifacts from CT images, and the trained network was then used to suppress aliasing artifacts in projection-reconstruction MRI.[15] This method leverages the notion that the characteristic structure of artifacts due to polar sampling should be similar in CT and MRI. Here, we considered random sampling patterns on a Cartesian grid; therefore, the domain transfer method proposed in[15] is not directly applicable to our reconstructions that possess incoherent artifacts. Another recent, independent effort examined the reliability of reconstructions from a variational network to deviations in undersampling patterns and SNR between the training and testing domains.[40] Mismatch in patterns or SNR between the 2 domains caused suboptimal performance even for modest acceleration factors. They also assessed the generalization capability by performing implicit domain transfer between PD-weighted knee images with and without fat suppression. A network trained on PD-weighted knee images without fat suppression was observed to yield relatively poor reconstructions of images with fat suppression and vice versa. Consistent with these observations, we also find that, without fine-tuning, networks trained on MR images of a given contrast (e.g., T_1-weighted) do not generalize well to images of a different contrast (e.g., T_2-weighted). That said, a distinct contribution of our work was to address the issue of data scarcity by training a network in a domain with ample data and transferring the network to a domain with fewer samples.

An alternative approach proposed to train neural networks for MRI reconstruction with small datasets is robust artificial-neural-networks for k-space interpolation (RAKI).[13] RAKI aims to train a neural network for each individual subject that learns to synthesize missing k-space samples from acquired data. Unlike traditional k-space parallel imaging methods,[45,62] RAKI estimates a nonlinear interpolation kernel from central calibration data. Such nonlinear interpolation was shown to boost reconstruction performance beyond linear methods. However, RAKI might yield suboptimal performance when the optimal interpolation kernel shows considerable variation across k-space. Our proposed architecture for multi-coil reconstructions leverages a linear interpolation kernel; thus, the output of calibration-consistency blocks in our network can manifest similar reconstruction errors. Yet, the remaining CNN blocks are trained to recover fully sampled reference images given images with residual artifacts at the output of CC blocks.

5 | CONCLUSION

Here, we demonstrated domain transfer based on a cascade architecture with multiple CNNs interleaved with data- and calibration-consistency blocks. The proposed approach might facilitate the use of neural networks for MRI reconstruction in applications in which data are relatively scarce. It might also benefit other types of architectures that have been proposed for accelerated MRI,[22,25,39] in particular architectures that require extensive datasets for adequate training.[12,42] Here, the calibration-consistency projections were based on the SPIRiT method. These projections can also be replaced with other k-space methods for parallel imaging such as GRAPPA or RAKI. Note that the current study examined the generalization capability of networks trained on natural images to T_1-weighted and T_2-weighted images of the brain. ImageNet-trained networks could also be beneficial for reconstruction of MR images acquired with more specialized contrasts such as angiograms, and images acquired in other body parts.

ACKNOWLEDGMENT

This work was supported in part by the following: Marie Curie Actions Career Integration grant (PCIG13-GA-2013-618101), European Molecular Biology Organization Installation grant (IG 3028), TUBA GEBIP fellowship, TUBITAK 1001 grant (118E256), and BAGEP fellowship awarded to T. Çukur. We also gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research. The MR brain images from the MIDAS dataset were collected and made available by the CASILab at The University of North Carolina at Chapel Hill and were distributed by the MIDAS Data Server at Kitware, Inc.

ORCID

Tolga Çukur https://orcid.org/0000-0002-2296-851X

REFERENCES

1. Block KT, Uecker M, Frahm J. Undersampled radial MRI with multiple coils. Iterative image reconstruction using a total variation constraint. *Magn Reson Med*. 2007;57:1086–1098.
2. Lustig M, Donoho D, Pauly JM. Sparse MRI: the application of compressed sensing for rapid MR imaging. *Magn Reson Med*. 2007;58:1182–1195.

3. Çukur T, Lustig M, Nishimura DG. Improving non-contrast-enhanced steady-state free precession angiography with compressed sensing. *Magn Reson Med*. 2009;61:1122–1131.
4. Jung H, Ye JC. Performance evaluation of accelerated functional MRI acquisition using compressed sensing. In Proceedings IEEE International Symposium on Biomedical Imaging: From Nano to Macro. 2009. p. 702–705.
5. Menzel MI, Tan ET, Khare K, et al. Accelerated diffusion spectrum imaging in the human brain using compressed sensing. *Magn Reson Med*. 2011;66:1226–1233.
6. Doneva M, Börnert P, Eggers H, Stehning C, Sénégas J, Mertins A. Compressed sensing reconstruction for magnetic resonance parameter mapping. *Magn Reson Med*. 2010;64:1114–1120.
7. Ravishankar S, Bresler Y. MR image reconstruction from highly undersampled k-space data by dictionary learning. *IEEE Trans Med Imaging*. 2011;30:1028–1041.
8. Khare K, Hardy CJ, King KF, Turski PA, Marinelli L. Accelerated MR imaging using compressive sensing with no free parameters. *Magn Reson Med*. 2012;68:1450–1457.
9. Weller DS, Ramani S, Nielsen J-F, Fessler JA. Monte Carlo SURE-based parameter selection for parallel magnetic resonance imaging reconstruction. *Magn Reson Med*. 2014;71:1760–1770.
10. Jun Y, Eo T, Kim T, Jang J, Hwang D. Deep convolutional neural network for acceleration of magnetic resonance angiography (MRA). In Proceedings of the 25th Annual Meeting of ISMRM, Honolulu, HI, 2017. p. 0686.
11. Schlemper J, Caballero J, Hajnal JV, Price A, Rueckert D. A deep cascade of convolutional neural networks for MR image reconstruction. In Proceedings 25th Biennial International Conference on Information Processing in Medical Imaging (IPMI), Boone, NC, 2017. p. 647–658.
12. Yang G, Yu S, Dong H, et al. DAGAN: deep de-aliasing generative adversarial networks for fast compressed sensing MRI reconstruction. *IEEE Trans Med Imaging*. 2018;37:1310–1321.
13. Akçakaya M, Moeller S, Weingärtner S, Uğurbil K. Scan-specific robust artificial-neural-networks for k-space interpolation (RAKI) reconstruction: Database-free deep learning for fast imaging. *Magn Reson Med*. 2019;81:439–453.
14. Wang S, Su Z, Ying L, et al. Accelerating magnetic resonance imaging via deep learning. In Proceedings IEEE 13th International Symposium on Biomedical Imaging (ISBI), Prague, CZ, 2016. p. 514–517.
15. Han Y, Yoo J, Kim HH, Shin HJ, Sung K, Ye JC. Deep learning with domain adaptation for accelerated projection-reconstruction MR. *Magn Reson Med*. 2018;80:1189–1205.
16. Han Y, Ye JC. k-Space deep learning for accelerated MRI. *IEEE Trans Med Imaging*. 2020;39:377–386.
17. Kwon K, Kim D, Seo H, Cho J, Kim B, Park H. Learning-based reconstruction using artificial neural network for higher acceleration. In Proceedings of the 24th Annual Meeting of ISMRM, Singapore, 2016. p. 1801.
18. Gong E, Zaharchuk G, Pauly J. Improving the PI+CS reconstruction for highly undersampled multi-contrast MRI using local deep network. In Proceedings of the 25th Annual Meeting of ISMRM, Honolulu, HI, 2017. p. 5663.
19. Wang S, Huang N, Zhao T, Yang Y, Ying L, Liang D. 1D partial Fourier parallel MR imaging with deep convolutional neural network. In Proceedings of the 25th Annual Meeting of ISMRM, Honolulu, HI, 2017. p. 0642.
20. Lee D, Yoo J, Ye JC. Compressed sensing and parallel MRI using deep residual learning. In Proceedings of the 25th Annual Meeting of ISMRM, Honolulu, HI, 2017. p. 0641.
21. Kwon K, Kim D, Park H. A parallel MR imaging method using multilayer perceptron. *Med Phys*. 2017;44:6209–6224.
22. Yang Y, Sun J, Li H, Xu Z. Deep ADMM-net for compressive sensing MRI. In: Lee DD, Sugiyama M, Luxburg UV, Guyon I, Garnett R, eds. *Advances in Neural Information Processing Systems 29 (NIPS 2016)*, Barcelona: Curran Associates; 2016.
23. Zhu B, Liu JZ, Rosen BR, Rosen MS. Neural network MR image reconstruction with AUTOMAP: automated transform by manifold approximation. In Proceedings of the 25th Annual Meeting of ISMRM, Honolulu, HI, 2017. p. 0640.
24. Zhu BO, Liu JZ, Cauley SF, Rosen BR, Rosen MS. Image reconstruction by domain transform manifold learning. *Nature*. 2018; 555:487–492.
25. Hyun CM, Kim HP, Lee SM, Lee S, Seo JK. Deep learning for undersampled MRI reconstruction. *Phys Med Biol*. 2018;63:135007.
26. Quan TM, Nguyen-Duc T, Jeong W-K. Compressed sensing MRI reconstruction with cyclic loss in generative adversarial networks. *IEEE Trans Med Imaging*. 2018;37:1488–1497.
27. Akçakaya M, Moeller S, Weingärtner S, Ugurbil K. Scan-specific Robust Artificial-neural-networks for k-space Interpolation (RAKI): database-free deep learning reconstruction for fast imaging. In Proceedings of the 26th Annual Meeting of ISMRM, Paris, France, 2018. p. 0576.
28. Pawar K, Chen Z, Shah N, Egan G. ReconNet: a deep learning framework for transforming image reconstruction into pixel classification. In Proceedings of the 26th Annual Meeting of ISMRM, Paris, France, 2018. p. 0574.
29. Fu Z, Mandava S, Keerthivasan M, Martin D, Altbach M, Bilgin A. A multi-scale deep ResNet for radial MR parameter mapping. In Proceedings of the 26th Annual Meeting of ISMRM, Paris, France, 2018. p. 0568.
30. Chang P, Liu M, Chow D, et al. Inception-CS: deep learning for sparse MR reconstruction in glioma patients. In Proceedings of the 26th Annual Meeting of ISMRM, Paris, France, 2018. p. 0571.
31. Zhu B, Bilgic B, Liao C, Rosen B, Rosen M. Deep learning MR reconstruction with Automated Transform by Manifold Approximation (AUTOMAP) in real-world acquisitions with imperfect training. In Proceedings of the 26th Annual Meeting of ISMRM, Paris, France, 2018. p. 0572.
32. Lee J, Ye JC. Deep convolutional framelet neural network for reference-free EPI ghost correction. In Proceedings of the 26th Annual Meeting of ISMRM, Paris, France, 2018. p. 0575.
33. Moeller S, Weingärtner S, Ugurbil K, Akcakaya M. Application of a scan-specific deep learning reconstruction to multiband/SMS imaging. In Proceedings of the 26th Annual Meeting of ISMRM, Paris, France, 2018. p. 0577.
34. Chen F, Taviani V, Malkiel I, et al. Variable-density single-shot fast spin-echo MRI with deep learning reconstruction by using variational networks. *Radiology*. 2018;289:366–373.
35. Cheng JY, Chen F, Alley MT, Pauly JM, Vasanawala SS. Highly scalable image reconstruction using deep neural networks with bandpass filtering. 2018;arXiv:1805.03300:[cs.CV].
36. Mardani M, Gong E, Cheng J, et al. Deep generative adversarial neural networks for compressed sensing (GANCS) automates MRI. In Proceedings of the 26th Annual Meeting of ISMRM, Paris, France, 2018. p. 0573.

37. Hammernik K, Knoll F, Sodickson DK, Pock T. Learning a variational model for compressed sensing MRI reconstruction. In Proceedings of the 24th Annual Meeting of ISMRM, Singapore, 2016. p. 1088.
38. Mardani M, Gong E, Cheng JY, et al. Deep generative adversarial neural networks for compressive sensing MRI. *IEEE Trans Med Imaging*. 2019;38:167–179.
39. Hammernik K, Klatzer T, Kobler E, et al. Learning a Variational Network for Reconstruction of Accelerated MRI Data. *Magn Reson Med*. 2017;79:3055–3071.
40. Knoll F, Hammernik K, Kobler E, Pock T, Recht MP, Sodickson DK. Assessment of the generalization of learned image reconstruction and the potential for transfer learning. *Magn Reson Med*. 2019;81:116–128.
41. Cheng JY, Mardani M, Alley MT, Pauly JM, Vasanawala SS. DeepSPIRiT: generalized parallel imaging using deep convolutional neural networks. In Proceedings of the 26th Annual Meeting of ISMRM, Paris, France, 2018. p. 0570.
42. Krizhevsky A, Sutskever I, Hinton GE. ImageNet Classification with Deep Convolutional Neural Networks. In: Pereira F, Burges CJC, Bottou L, Weinberger KQ, editors. *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc.; 2012. pp. 1097–1105.
43. Donahue J, Jia Y, Vinyals O, et al. DeCAF: a deep convolutional activation feature for generic visual recognition. Xing EP, Jebara T, eds. In Proceedings of the 31st International Conference on Machine Learning, PMLR. Bejing, China. 2014;32:647–655.
44. Yosinski J, Clune J, Bengio Y, Lipson H. How transferable are features in deep neural networks? In: Ghahramani Z, Welling M, Cortes C, Lawrence ND, Weinberger KQ, eds. *Advances in Neural Information Processing Systems 27 (NIPS 2014)*. Red Hook, NY: Curran Associates; 2014:3320–3328.
45. Lustig M, Pauly JM. SPIRiT: iterative self-consistent parallel imaging reconstruction from arbitrary k-space. *Magn Reson Med*. 2010;64:457–471.
46. Dar SUH, Çukur T. Transfer learning for reconstruction of accelerated MRI acquisitions via neural networks. In Proceedings of the 26th Annual Meeting of ISMRM, Paris, France, 2018. p. 0569.
47. Shahdloo M, Ilicak E, Tofighi M, Saritas EU, Cetin AE, Cukur T. Projection onto epigraph sets for rapid self-tuning compressed sensing MRI. *IEEE Trans Med Imaging*. 2018;38:1677–1689.
48. Bauschke HH, Borwein JM. On projection algorithms for solving convex feasibility problems. *SIAM Rev*. 1996;38:367–426.
49. Uecker M, Lai P, Murphy MJ, et al. ESPIRiT-an eigenvalue approach to autocalibrating parallel MRI: where SENSE meets GRAPPA. *Magn Reson Med*. 2014;71:990–1001.
50. Bullitt E, Zeng D, Gerig G, et al. Vessel tortuosity and brain tumor malignancy: a blinded study. *Acad. Radiol*. 2005;12:1232–1240.
51. Russakovsky O, Deng J, Su H, et al. ImageNet large scale visual recognition challenge. *Int J Comput Vis*. 2015;115:211–252.
52. Zhang T, Pauly JM, Vasanawala SS, Lustig M. Coil compression for accelerated imaging with Cartesian sampling. *Magn Reson Med*. 2013;69:571–582.
53. Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. *Nature*. 1986;323:533–536.
54. Kingma DP, Ba J. Adam: a method for stochastic optimization. In Proceedings of 3rd International Conference on Learning Representations, San Diego, CA. 2015. arXiv:1412.6980 [cs.LG]. doi: http://doi.acm.org.ezproxy.lib.ucf.edu/10.1145/1830483.1830503.
55. Goodfellow I, Bengio Y, Courville A. *Deep Learning*. Cambridge, MA: MIT Press; 2016.
56. Abadi M, Barham P, Chen J, et al. TensorFlow: large-scale machine learning on heterogenous distributed systems. In the Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16), Savannah, GA, 2016. p. 265–284.
57. Domingos P. A few useful things to know about machine learning. *Commun ACM*. 2012;55:78–87.
58. Burton GJ, Moorhead IR. Color and spatial structure in natural scenes. *Appl Opt*. 1987;26:157–170.
59. Goodfellow I.NIPS 2016 Tutorial: Generative Adversarial Networks (GANs). 2016. arXiv:1701.00160 [cs.LG].
60. Isola P, Zhu J-Y, Zhou T, Efros AA. Image-to-image translation with conditional adversarial networks. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017. p. 1125–1134.
61. Ouyang C, Schlemper J, Biffi C, et al. Generalising deep learning MRI reconstruction across different domains. 2019; arXiv:1902.10815 [cs.CV].
62. Griswold MA, Jakob PM, Heidemann RM, et al. Generalized autocalibrating partially parallel acquisitions (GRAPPA). *Magn Reson Med*. 2002;47:1202–1210.

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section.

FIGURE S1 Demonstration of (A) convolutional neural network (CNN), (B) calibration consistency (CC) and (C) data consistency (DC) blocks given a multi-coil image x as an input. (A) The CNN block first combines undersampled multi-coil images using coil-sensitivity maps A, estimated via ESPIRiT. Real and imaginary parts of the coil-combined image are then reconstructed using two separate networks. The outputs of the real and imaginary networks are joined to form a complex image, which is then back projected onto individual coils again using the coil sensitivity maps. (B) The CC block transforms the input image into Fourier domain, applies the interpolation operator on multi-coil k-space data, and converts the image back into image domain. (C) The DC block performs a weighted combination of samples recovered by the previous block (CNN or CC) and the originally-acquired samples

FIGURE S2 Representative synthetic complex multi-coil natural images. Complex multi-coil natural images were simulated from magnitude images in ImageNet (see Methods for details). Magnitude and phase of two simulated multi-coil natural images (A and B) are shown along with their reference magnitude images

FIGURE S3 Percentage change in validation error as a function of number of epochs for T_2 to T_1 domain transfer at acceleration factor R = 4. Results are shown for sequential training of individual CNN blocks (A-E), end-to-end training of the complete network (F) and fine-tuning of the complete network (G). Initial CNN block training was performed on 2000 T_2-weighted images, end-to-end training was performed on

100 T_2-weighted images, and fine-tuning was performed on 20 T_1-weighted images

FIGURE S4 Percentage change in validation error as a function of number of epochs for ImageNet to T_1 domain transfer at acceleration factor R = 4. Results are shown for learning rates (lr) equal to (A) 10^{-4}, (B) 10^{-5} and (C) 10^{-6}. Initial CNN block training was performed on 2000 ImageNet images, end-to-end training was performed on 100 ImageNet images, and fine-tuning was performed on 20 T_1-weighted images. Learning rate equal to 10^{-5} facilitates convergence while preventing undesirable oscillations in the validation error. A learning rate of 10^{-5} ensures both stable fine-tuning and faster convergence. In contrast, a higher learning rate of 10^{-4} leads to oscillatory behavior in validation error, potentially suggesting overfitting to fine-tuning samples. While fine-tuning is relatively stable for a lower learning rate of 10^{-6}, network convergence is noticeably slower

FIGURE S5 Representative reconstructions of a T_2-weighted acquisition at acceleration factor R = 4. Reconstructions were performed via the Zero-filled Fourier method (ZF), and ImageNet-trained, T_2-trained, and T_1-trained networks. (A) Reconstructed images and error maps for raw networks (see colorbar). (B) Reconstructed images and error maps for fine-tuned networks. The fully-sampled reference image is also shown. Network training was performed on a training dataset of 2000 images and fine-tuned on a sample of 20 T_2-weighted images. Following fine-tuning, ImageNet-trained and T_1-trained networks yield reconstructions of highly similar quality to the T_2-trained network

FIGURE S6 Reconstructions of a T_2-weighted acquisition with R = 4 via ZF, conventional compressed-sensing (CS), and ImageNet-trained, T_1-trained and T_2-trained networks along with the fully-sampled reference image. Error maps for each reconstruction are shown below (see colorbar). Networks were trained on 2000 images and fine-tuned on 20 images acquired with the test contrast. The domain-transferred networks maintain nearly identical performance to the networks trained directly in the testing domain. Furthermore, the domain-transferred networks reconstructions outperform conventional CS in terms of image sharpness and residual aliasing artifacts

FIGURE S7 Reconstruction performance was evaluated for undersampled T_2-weighted acquisitions. Average PSNR values across T_2-weighted validation images were measured for the T_2-trained network (trained on 4k images and fine-tuned on 100 images), ImageNet-trained networks (trained on 500, 1000, 2000, or 4000 images), and T_1-trained network (trained on 4000 images). Results are plotted as a function of number of fine-tuning samples for acceleration factors (A) R = 4, (B) R = 6, (C) R = 8, and (D) R = 10. As the number of fine-tuning samples increases, the PSNR differences decay gradually to a negligible level. Domain-transferred networks trained on fewer samples require more fine-tuning samples to yield similar performance consistently across R

FIGURE S8 Number of fine-tuning samples required for the PSNR values for ImageNet-trained networks to converge. Average PSNR values across T_2-weighted validation images were measured for the ImageNet-trained networks trained on (A) 500, (B) 1000, (C) 2000, and (D) 4000 images. Convergence was taken as the number of fine-tuning samples where the percentage change in PSNR by incrementing N_{tune} fell below 0.05% of the average PSNR for the T_2-trained network (see Supporting Information Figure S7). Domain-transferred networks trained on fewer samples require more fine-tuning samples for the PSNR values to converge. Furthermore, at higher values of R, more fine-tuning samples are required for convergence

FIGURE S9 Reconstruction performance was evaluated for undersampled multi-coil T_2-weighted acquisitions. Average PSNR values across T_2-weighted validation images were measured for the T_2-trained network (trained and fine-tuned on 360 images), and ImageNet-trained network trained on 2000 images. Results are plotted as a function of number of fine-tuning samples for acceleration factors (A) R = 4, (B) R = 6, (C) R = 8, and (D) R = 10. Without fine-tuning, the T_2-trained network outperforms the domain-transferred network. As the number of fine-tuning samples increases, the PSNR differences decay gradually to a negligible level

FIGURE S10 Number of fine-tuning samples required for the PSNR values for ImageNet-trained networks (trained on multi-coil complex images) to converge. Average PSNR values across T_2-weighted validation images were measured for the ImageNet-trained network trained on 2000 images. Convergence was taken as the number of fine-tuning samples where the percentage change in PSNR by incrementing N_{tune} fell below 0.05% of the average PSNR for the T_2-trained network (see Supporting Information Figure S9). At higher values of R, more fine-tuning samples are required for convergence

FIGURE S11 Representative reconstructions of a multi-coil T_2-weighted acquisition at acceleration factor R = 10. Reconstructions were performed via ZF, ImageNet-trained and T_2-trained networks, and SPIRiT (top row). Corresponding error maps are also shown (see colorbar; bottom row) along with the fully-sampled reference (top row). Network training was performed on a training dataset of 2000 images and fine-tuned on a sample of 20 T_2-weighted images. The ImageNet-trained network maintains similar performance to the T_2-trained network trained directly on the images from the test domain. Furthermore, the domain-transferred network outperforms conventional SPIRiT in terms of residual aliasing artifacts

FIGURE S12 Percentage change in network weights as a function of network depth for multi-coil ImageNet to (A) T_1 and (B) T_2 domain transfer averaged across acceleration factors (R = 4-10). Red dots correspond to the percentage change, and blue dashed lines correspond to a linear least squares fit to the percentage change. Overall, the percentage change in weights is higher for earlier versus later layers of

the network. For ImageNet to T_1 domain transfer, percentage change varies from 2.27% to 0.56%, and for ImageNet to T_2 domain transfer percentage change varies from 3.28% to 0.47%. Note that the layer number ranges from 1 to 25. This is because the CNN architecture used in this study consists of 5 cascades and each cascade consists of 5 layers

FIGURE S13 Reconstruction performance was evaluated for undersampled single-coil complex T_1-weighted acquisitions. Average PSNR values across T_1-weighted validation images were measured for the T_1-trained network (trained and fine-tuned on 360 images), and ImageNet-trained network trained on 2000 images. Results are plotted as a function of number of fine-tuning samples for acceleration factors (A) R = 4, (B) R = 6, (C) R = 8, and (D) R = 10. Without fine-tuning, the T_1-trained network outperforms the domain-transferred network. As the number of fine-tuning samples increases, the PSNR differences decay gradually to a negligible level

FIGURE S14 Number of fine-tuning samples required for the PSNR values for ImageNet-trained networks (trained on single-coil complex images) to converge. Average PSNR values across T_1-weighted validation images were measured for the ImageNet-trained network trained on 2000 images. Convergence was taken as the number of fine-tuning samples where the percentage change in PSNR by incrementing N_{tune} fell below 0.05% of the average PSNR for the T_1-trained network (see Supporting Information Figure S13)

FIGURE S15 Reconstruction performance was evaluated for undersampled single-coil complex T_2-weighted acquisitions. Average PSNR values across T_2-weighted validation images were measured for the T_2-trained network (trained and fine-tuned on 360 images), and ImageNet-trained network trained on 2000 images. Results are plotted as a function of number of fine-tuning samples for acceleration factors (A) R = 4, (B) R = 6, (C) R = 8, and (D) R = 10. Without fine-tuning, the T_2-trained network outperforms the domain-transferred network. As the number of fine-tuning samples increases, the PSNR differences decay gradually to a negligible level

FIGURE S16 Number of fine-tuning samples required for the PSNR values for ImageNet-trained networks (trained on single-coil complex images) to converge. Average PSNR values across T_2-weighted validation images were measured for the ImageNet-trained network trained on 2000 images. Convergence was taken as the number of fine-tuning samples where the percentage change in PSNR by incrementing N_{tune} fell below 0.05% of the average PSNR for the T_2-trained network (see Supporting Information Figure S15)

TABLE S1 Reconstruction quality for single-coil magnitude T_1-weighted images undersampled at R = 4, 6, 8, 10. Reconstructions were performed via ImageNet-trained, T_1-trained, T_2-trained and limited networks, as well as conventional CS. PSNR and SSIM values are reported as mean ± standard deviation across test images. Results are shown for raw networks trained on 2000 training images (raw), and fine-tuned networks tuned with tens of T_1-weighted images (tuned)

TABLE S2 Reconstruction quality for single-coil magnitude T_2-weighted images undersampled at R = 4, 6, 8, 10. Reconstructions were performed via ImageNet-trained, T_1-trained, T_2-trained and limited networks, as well as conventional CS. PSNR and SSIM values are reported as mean ± standard deviation across test images. Results are shown for raw networks trained on 2000 training images (raw), and fine-tuned networks tuned with tens of T_2-weighted images (tuned)

TABLE S3 Reconstruction quality for single-coil magnitude T_1-weighted images undersampled at R = 4, 6, 8, 10. Reconstructions were performed via ImageNet-trained, T_1-trained, T_2-trained and limited networks, as well as conventional CS. PSNR and SSIM values are reported as mean ± standard deviation across test images. Results are shown for raw networks trained on 2000 training images (raw), and fine-tuned networks tuned with 100 T_1-weighted images (tuned)

TABLE S4 Reconstruction quality for single-coil magnitude T_2-weighted images undersampled at R = 4, 6, 8, 10. Reconstructions were performed via ImageNet-trained, T_1-trained, T_2-trained and limited networks, as well as conventional CS. PSNR and SSIM values are reported as mean ± standard deviation across test images. Results are shown for raw networks trained on 2000 training images (raw), and fine-tuned networks tuned with 100 T_2-weighted images (tuned)

TABLE S5 Reconstruction quality for multi-coil complex T_2-weighted images undersampled at R = 4, 6, 8, 10. Reconstructions were performed via ImageNet-trained and T_1-trained networks as well as SPIRiT. PSNR and SSIM values are reported as mean ± standard deviation across test images. Results are shown for raw networks trained on 2000 training images (raw), and fine-tuned networks tuned with tens of T_2-weighted images (tuned)

TABLE S6 Reconstruction quality for single-coil complex T_1-weighted images undersampled at R = 4, 6, 8, 10. Reconstructions were performed via ImageNet-trained and T_1-trained networks. PSNR and SSIM values are reported as mean ± standard deviation across test images. Results are shown for raw networks trained on 2000 training images (raw), and fine-tuned networks tuned with tens of T_2-weighted images (tuned)

TABLE S7 Reconstruction quality for single-coil complex T_2-weighted images undersampled at R = 4, 6, 8, 10. Reconstructions were performed via ImageNet-trained and T_2-trained networks. PSNR and SSIM values are reported as mean ± standard deviation across test images. Results are shown for raw networks trained on 2000 training images (raw), and fine-tuned networks tuned with tens of T_2-weighted images (tuned)

How to cite this article: Dar SUH, Özbey M, Çatlı AB, Çukur T. A Transfer-Learning Approach for Accelerated MRI Using Deep Neural Networks. *Magn Reson Med*. 2020;84:663–685. https://doi.org/10.1002/mrm.28148