Received: 23 June 2021 | Revised: 13 October 2021 | Accepted: 20 October 2021

DOI: 10.1002/mrm.29073

TECHNICAL NOTE

Magnetic Resonance in Medicine

Learning a preconditioner to accelerate compressed sensing reconstructions in MRI

Kirsten Koolstra[1] | Rob Remis[2]

[1]Division of Image Processing, Department of Radiology, Leiden University Medical Center, Leiden, The Netherlands

[2]Circuits & Systems Group, Electrical Engineering, Mathematics and Computer Science Faculty, Delft University of Technology, Delft, The Netherlands

Correspondence
Kirsten Koolstra, Division of Image Processing, Radiology, Leiden University Medical Center, Albinusdreef 2, 2333 ZA, Leiden, The Netherlands.
Email: K.Koolstra@lumc.nl

Funding information
Nederlandse Organisatie voor Wetenschappelijk Onderzoek, Grant/Award Number: HTSM 17104

Purpose: To learn a preconditioner that accelerates parallel imaging (PI) and compressed sensing (CS) reconstructions.

Methods: A convolutional neural network (CNN) with residual connections was used to train a preconditioning operator. Training and validation data were simulated using 50% brain images and 50% white Gaussian noise images. Each multichannel training example contains a simulated sampling mask, complex coil sensitivity maps, and two regularization parameter maps. The trained model was integrated in the preconditioned conjugate gradient (PCG) method as part of the split Bregman CS method. The acceleration performance was compared with that of a circulant PI-CS preconditioner for varying undersampling factors, number of coil elements and anatomies.

Results: The learned preconditioner reduces the number of PCG iterations by a factor of 4, yielding a similar acceleration as an efficient circulant preconditioner. The method generalizes well to different sampling schemes, coil configurations and anatomies.

Conclusion: It is possible to learn adaptable preconditioners for PI and CS reconstructions that meet the performance of state-of-the-art preconditioners. Further acceleration could be achieved by optimizing the network architecture and the training set. Such a preconditioner could also be integrated in fully learned reconstruction methods to accelerate the training process of unrolled networks.

KEYWORDS
compressed sensing, deep learning, MR image reconstruction, parallel imaging, preconditioning, split Bregman

1 | INTRODUCTION

Compressed sensing (CS) allows for a reduction in scan time at the cost of longer reconstruction times, which are a consequence of the non-linear formulation of the CS minimization problem: it uses ℓ_1-norm terms that promote sparsity of the image in certain transform domains. Examples of such sparsifying transformations are the total variation operator, wavelet transformations or a combination of these.[1–5] Preconditioning has been proposed in the past to accelerate CS reconstructions.[6–9] This technique can reduce the number of iterations that are needed to converge to the optimal solution.[10] Designing an efficient preconditioner, however, is not straightforward. It is often difficult to approximate the inverse operation of the signal model of interest in a computationally inexpensive manner.[11] For CS this is especially the case when multiple coil sensitivity maps are taken into account, which add an additional level of complexity to the structure of the system matrix. In Ref. [8], this problem was addressed by approximating the CS system matrix by a block circulant matrix with circulant blocks, resulting in a speed up factor of 5 when using the conjugate gradient (CG) method in combination with the split Bregman (SB) reconstruction framework. Furthermore, Ong et al. managed to construct a diagonal k-space preconditioner for the primal dual hybrid gradient method that supports non-Cartesian trajectories.[9] Although these preconditioners have already shown to successfully reduce the number of iterations needed for CS reconstructions, the approximations made in their design process often limit the acceleration factor that can be achieved.

In the past decade, deep learning reconstruction approaches have also gained popularity. Once the model has been trained, these approaches are often inherently fast.[12–16] An additional advantage is that regularization parameters and regularization functions of the CS formulation can be learned during the training process,[17] potentially allowing larger undersampling factors. On the other hand, training a neural network requires tuning many other machine learning hyperparameters instead. While deep learning reconstructions have shown great reconstruction performance over the past years,[18] a difficulty of this class of approaches is the risk of the trained network not generalizing well to unseen cases or scans with different SNR and sampling patterns,[19,20] and the lack of understanding and detecting corresponding image artifacts. For this reason, most recent deep learning approaches integrate prior (physics) knowledge into the training process, in this way constraining the solution space.[17,18,21,22] The unrolled type of networks tend to be relatively large, since they follow the general structure of model-based reconstruction algorithms, such that each network block represents one iteration of the CS problem. These type of approaches could therefore also benefit from a preconditioner that reduces the number of network blocks and hence simplifies the network structure.

In this work, we integrate deep learning techniques into a physics-driven model-based reconstruction framework.[23] We explore the feasibility of learning a preconditioner using a convolutional neural network (CNN) to accelerate classical CS reconstructions. Such a preconditioner will support reconstruction acceleration without introducing uncertainty to the reconstruction quality. We learn the action of a preconditioner on a vector, such that the evaluation of the learned preconditioner will be fast and requires little memory. We first analyze the training performance of the preconditioning model on simulated test data. We then demonstrate the acceleration performance of the learned preconditioner when integrated in CG as part of the SB reconstruction framework.[24] The final preconditioned reconstruction algorithm is tested for multiple anatomies, coil configurations and undersampling factors, and results are compared to that obtained with the circulant preconditioner designed in Ref. [8].

2 | METHODS

2.1 | Compressed sensing

The 2D compressed sensing problem can be written as

$$\hat{\mathbf{x}} = \text{argmin}_{\mathbf{x}} \left\{ \sum_{i=1}^{N_c} ||RFS_i\mathbf{x}-\mathbf{y}_i||_2^2 + \frac{\lambda}{2}\left(||D_x\mathbf{x}||_1 + ||D_y\mathbf{x}||_1\right) + \frac{\gamma}{2}||W\mathbf{x}||_1 \right\}. \tag{1}$$

In this formulation, $\mathbf{x} \in \mathbb{C}^{N\times1}$ is the unknown image, $\mathbf{y}_i \in \mathbb{C}^{N\times1}$ is the acquired k-space data for coil i with $S_i \in \mathbb{C}^{N\times N}$ the corresponding coil sensitivity map, $R \in \mathbb{R}^{N\times N}$ is the sampling mask and $F \in \mathbb{C}^{N\times N}$ is the uniform 2D Fourier transform. The finite difference operators, $D_x, D_y \in \mathbb{R}^{N\times N}$, and the wavelet transform, $W \in \mathbb{R}^{N\times N}$, are used as sparsifying operations that promote sparsity of the unknown image solution in its transformed domain. The regularization parameters λ and γ determine the balance between the data consistency term and the regularization terms. Such a nonlinear problem can be solved iteratively using SB.[24] This is a reconstruction framework in which the ℓ_2 norm terms are decoupled from the ℓ_1 norm terms, such that each iteration of SB solves a linear system of equations followed by Bregman parameter updates. In this algorithm, the most time-consuming step is repeatedly solving the linear system of equations

$$A\mathbf{x} = \mathbf{b} \tag{2}$$

in which

$$A = \sum_{i=1}^{N_c} S_i^H F^H R^H RFS_i + \lambda\left(D_x^H D_x + D_y^H D_y\right) + \gamma W^H W \tag{3}$$

and

$$\mathbf{b} = \sum_{i=1}^{N_c} S_i^H F^H R^H \mathbf{y}_i + \lambda \left(D_x^H (\mathbf{d}_x^k - \mathbf{b}_x^k) + D_y^H \left(\mathbf{d}_y^k - \mathbf{b}_y^k \right) \right) + \gamma W^H \left(\mathbf{d}_w^k - \mathbf{b}_w^k \right). \tag{4}$$

The Bregman parameters $\mathbf{b}_x^k, \mathbf{b}_y^k, \mathbf{b}_w^k$ and the auxiliary parameters $\mathbf{d}_x^k, \mathbf{d}_y^k, \mathbf{d}_w^k$ in Bregman iteration k are introduced by the Bregman scheme.

2.2 | Preconditioner

Solving the linear system of equations in Equation (2) is done iteratively and can be accelerated with the help of a preconditioner. An efficient preconditioner M^{-1} reduces the number of iterations that are needed to solve the linear system. A reduction in iterations is typically achieved when the following holds:

$$M^{-1} A \mathbf{x} = M^{-1} \mathbf{b} \approx \mathbf{x}. \tag{5}$$

The reduction in iterations needs to outweigh the additional computational costs that are introduced by incorporating the preconditioner into the iterative scheme. In this case, the total computation time of solving the preconditioned system is shorter than that of solving the original system. Since the preconditioned numerical scheme solves the same model as the original numerical scheme, shorter computation times are obtained without affecting the accuracy of the numerical solution.

In this work, we train a network that learns such a preconditioner M^{-1} from a simulated training set. Specifically, we train a network that learns the inverse operation of Equation (2), $M^{-1} \approx A^{-1}$, such that Equation (5) is satisfied. Instead of learning M^{-1} in the form of a full matrix, we design the model such that the network learns the action of the preconditioner M^{-1} on an arbitrary vector $\mathbf{b}$. This makes using the learned preconditioner computationally inexpensive (~ 0.03 s per evaluation). Moreover, little memory storage is required.

2.3 | Simulation of training examples

The training set was simulated in MATLAB (Mathworks Inc, Natick, MA) and constructed from an equal amount of white Gaussian complex noise images and complex brain images. For the complex brain images, magnitude brain images (70 MP-RAGE and 63 T_2-weighted 3D brain images) were downloaded from the Human Connectome Project[25] (https://ida.loni.usc.edu/login.jsp) and transformed into transverse (MPRAGE:40, T_2-weighted:65) and sagittal (MPRAGE:36, T_2-weighted:66) slices. White Gaussian noise was added to the background of the images. A Gaussian-shaped phase was randomly added to each image, after which the resulting brain image set was augmented by rotating each image four times in steps of 90°, resulting in 53 160 images. These were used to simulate the actual training examples, for which a wide variety of undersampling factors, sampling masks, coil sensitivity maps and regularization parameters were taken into account. Since computing A^{-1} for each C_i, R, λ, γ combination is computationally expensive, we compute the training examples using only the forward operator A, according to $A\mathbf{x}^{[k]} = \mathbf{b}^{[k]}$. After this, we swap the operator's input and output before feeding them to the network: $\mathbf{b}^{[k]}$ is used as the network's input image, while $\mathbf{x}^{[k]}$ is used as the label image. In this way, the network will learn the operator A^{-1}, even though the training data were simulated using the forward operator A. Note that both $\mathbf{x}^{[k]}$ and $\mathbf{b}^{[k]}$ represent image-domain vectors. Each $\left(\mathbf{b}^{[k]}, \mathbf{x}^{[k]}\right)$ pair was simulated with an integer undersampling factor ranging between 1 and 4 and Gaussian-shaped complex coil sensitivity maps for a number of coil elements ranging between 1 and 16. The parameters λ and γ were randomly selected from the interval (0,5) and transformed into uniform regularization maps. This range was based on empirical tuning of the regularization parameters in SB reconstructions without using the learned network as preconditioner. The input images $\mathbf{b}^{[k]}$ were normalized to have a maximum magnitude value of 1 and the label images $\mathbf{x}^{[k]}$ were accordingly scaled. The images $(\mathbf{b}^{[k]}, \mathbf{x}^{[k]})$ and the coil sensitivity maps were split into real and imaginary components, and the resulting matrices were stacked (Figure 1A). If the number of coil elements was smaller than 16, zero-valued maps were added to the set of coil sensitivity maps until the number of coil sensitivity maps was equal to 16. This yielded a 37 channel $(\mathbf{b}{:}\,2, R{:}\,1, C{:}\,32, \lambda{:}\,1, \gamma{:}\,1)$ network input. Slices for one MP-RAGE and one T_2-weighted volunteer were selected to create a model validation set.

2.4 | Model and training

To train the preconditioner, we used a residual neural network with three residual blocks containing two layers each,[26] schematically shown in Figure 1B. A 2D convolution (3 × 3 kernels, 128 features) was performed at every layer. ReLu activation functions were chosen for the hidden layers, whereas the tanh activation function was chosen for the output layer to enable positive as well as negative matrix element predictions: magnitude

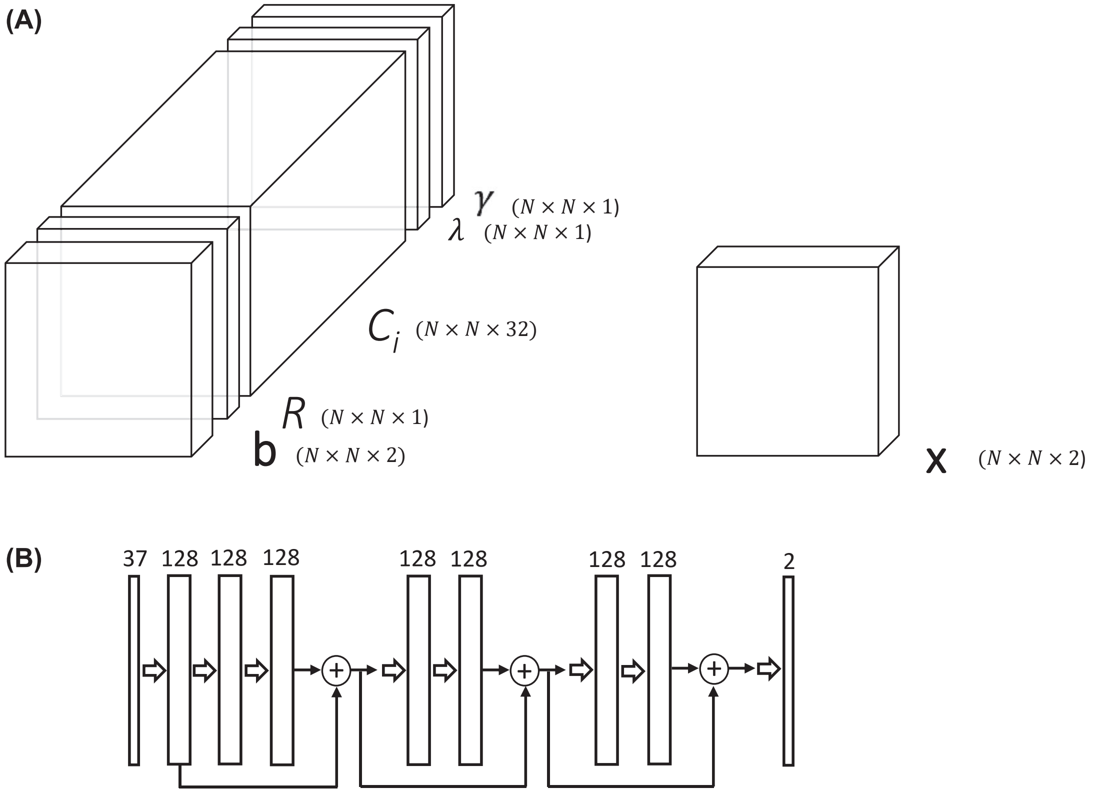

FIGURE 1 Model input and output and network architecture. A, Each 37-channel input contains a complex input image (**b**), a real sampling mask (R), 16 complex coil sensitivity maps and real regularization masks (λ and γ). The 2-channel output contains its corresponding complex image (**x**). Note that all complex matrices are first split into real and imaginary components before stacking them in the network's input and output volumes. B, A convolutional neural network with residual connections is used for training. Each residual block contains two layers.[26] A 2D convolution (3 × 3 kernels, 128 features) was performed at every layer with ReLu (hidden layers) and tanh (output layer) activation functions

values on the interval (0, 1) imply real and imaginary component values on the interval $(-1, 1)$. We minimized the mean absolute error over 50 epochs using the Adam optimizer with an initial learning rate of $5 \cdot 10^{-4}$, a batch size of 16 and drop out with a probability of 0.25. Training was performed in Tensorflow with a 24 GB Quadro RTX 6000 gpu, resulting in a training time of 2 days and 20 hr.

2.5 | MR data acquisition

Experiments were performed in three healthy volunteers after giving informed consent. The Leiden University Medical Center Committee for Medical Ethics approved the experiment. Undersampled and fully sampled scans were acquired on an Ingenia 3T dual transmit MR system (Philips, Best, The Netherlands), equipped with a 15-channel head coil and a 16-channel knee coil (informed consent obtained). The following data were acquired:

brain T_2-weighted turbo spin-echo (TSE): FOV = 230×230 mm^2; in-plane resolution 0.90×0.90 mm^2; 4 mm slice thickness; 1 slice; TE/TR/TSE factor = 80 ms/3000 ms/16; refocusing angle = 120°; WFS = 2.5 pixels; scan time = 00:30 min ($R = 2$).

brain fluid-attenuated inversion recovery (FLAIR): FOV = 240×224 mm^2; in-plane resolution 1.0×1.0 mm^2; 4 mm slice thickness; 1 slice; TE/TR/TSE factor = 120 ms/9000 ms/24; IR delay = 2500 ms; refocusing angle = 110°; WFS = 2.7 pixels; scan time = 01:30 min ($R = 2$).

knee gradient echo (FFE): FOV = 160×160 mm^2; in-plane resolution 1.25×1.25 mm^2; 3 mm slice thickness; 32 slices; TE/TR = 10 ms/455 ms; FA = 90°; WFS = 1.4 pixels; scan time = 1:01 min ($R = 1$), retrospectively undersampled to $R = 3$.

brain T_1-weighted inversion recovery turbo spin-echo (IR TSE): FOV = 230×230 mm^2; in-plane resolution 0.9×0.9 mm^2; 4 mm slice thickness; 24 slices; TE/TR/TSE factor = 20 ms/2000 ms/8; IR delay = 800 ms; refocussing angle = 120°; WFS = 2.6 pixels; scan time = 5:50 min ($R = 1$), retrospectively undersampled to $R = 4$.

2.6 | Image reconstruction

Reconstruction of the images was performed in Python (Python Software Foundation, Beaverton, OR) and run on a Linux machine with Intel Xe 6234 CPU and 256 GB internal memory. For each data set, the complex coil sensitivity maps were estimated from the center of k-space using ESPIRiT[27] and masked outside the object. SB was used as the ℓ_1-norm minimization algorithm to solve the compressed sensing problem, following the implementation described in Ref. [8]. The number of inner Bregman iterations was set to 1. The tolerance of PCG was set to 10^{-2}. The regularization parameters were empirically tuned. Reconstructions were performed with and without the learned preconditioner, and compared to the circulant preconditioner designed in Ref. [8]. For reconstructions with the learned preconditioner, the trained TensorFlow model was imported once at the beginning of the reconstruction pipeline and used for inference in every PCG iteration of SB. The methods were compared for three matrix sizes (128 × 128, 256 × 256 and 240 × 224), three different undersampling factors ($R = 2$, $R = 3$ and $R = 4$), two different anatomies and coil configurations (brain and knee) and three different MR contrasts (TSE, FFE and FLAIR).

3 | RESULTS

Figure 2A shows the loss function during training for the training data (red) and the validation data (blue). Figure 2B,C shows that the performance of the trained network is not dependent on the undersampling factor and on the number of coil elements for the validation data. Figure 2D, however, shows that the validation error is smallest for large regularization parameters, for which the inverse linear system is best conditioned.

Figure 3 shows the network's input (R, λ, γ, coil sensitivities, and image) and output for (A) a brain and (B) a noise example from the validation set. The network's prediction

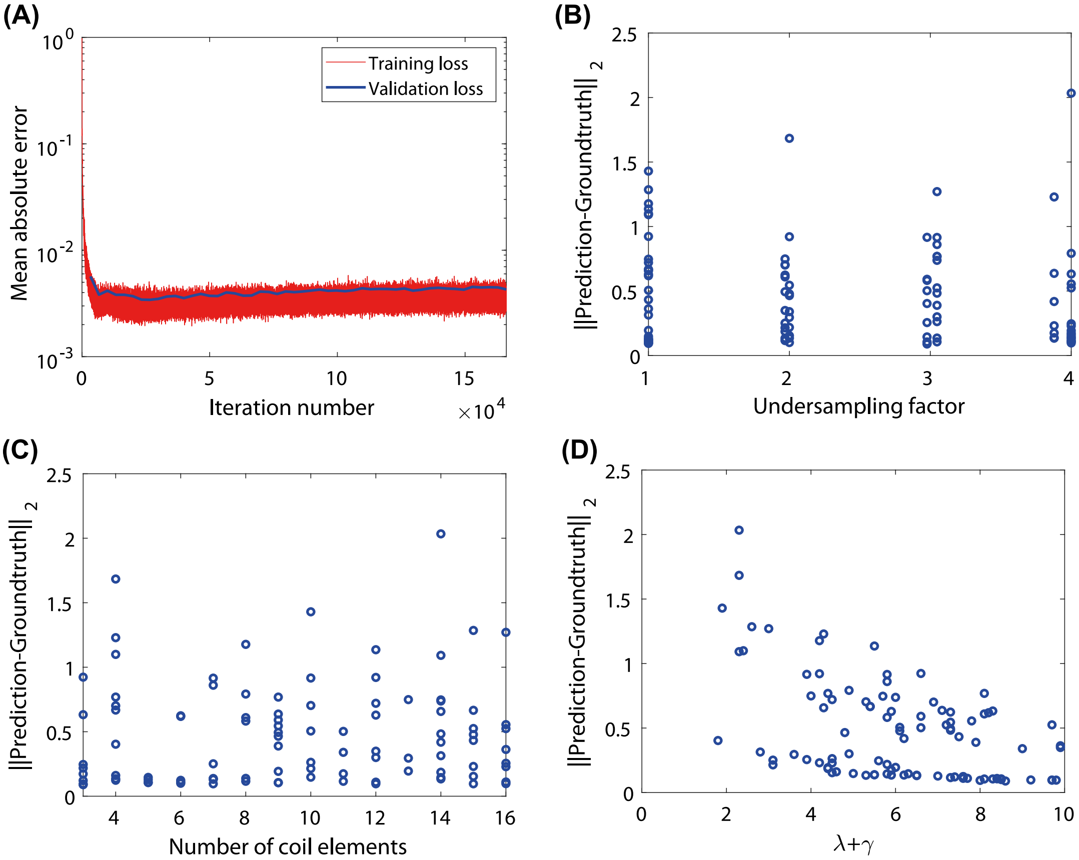

FIGURE 2 Training and validation performance. A, The mean absolute error (MAE) loss function during training for training data (red) and validation data (blue) over 50 epoch plotted on a log scale. B, C, The validation error after 50 epoch does not show large dependence on the undersampling factor and the number of coil elements. D, The validation error after 50 epoch is smallest for large regularization factors, for which the inverse operation is best defined

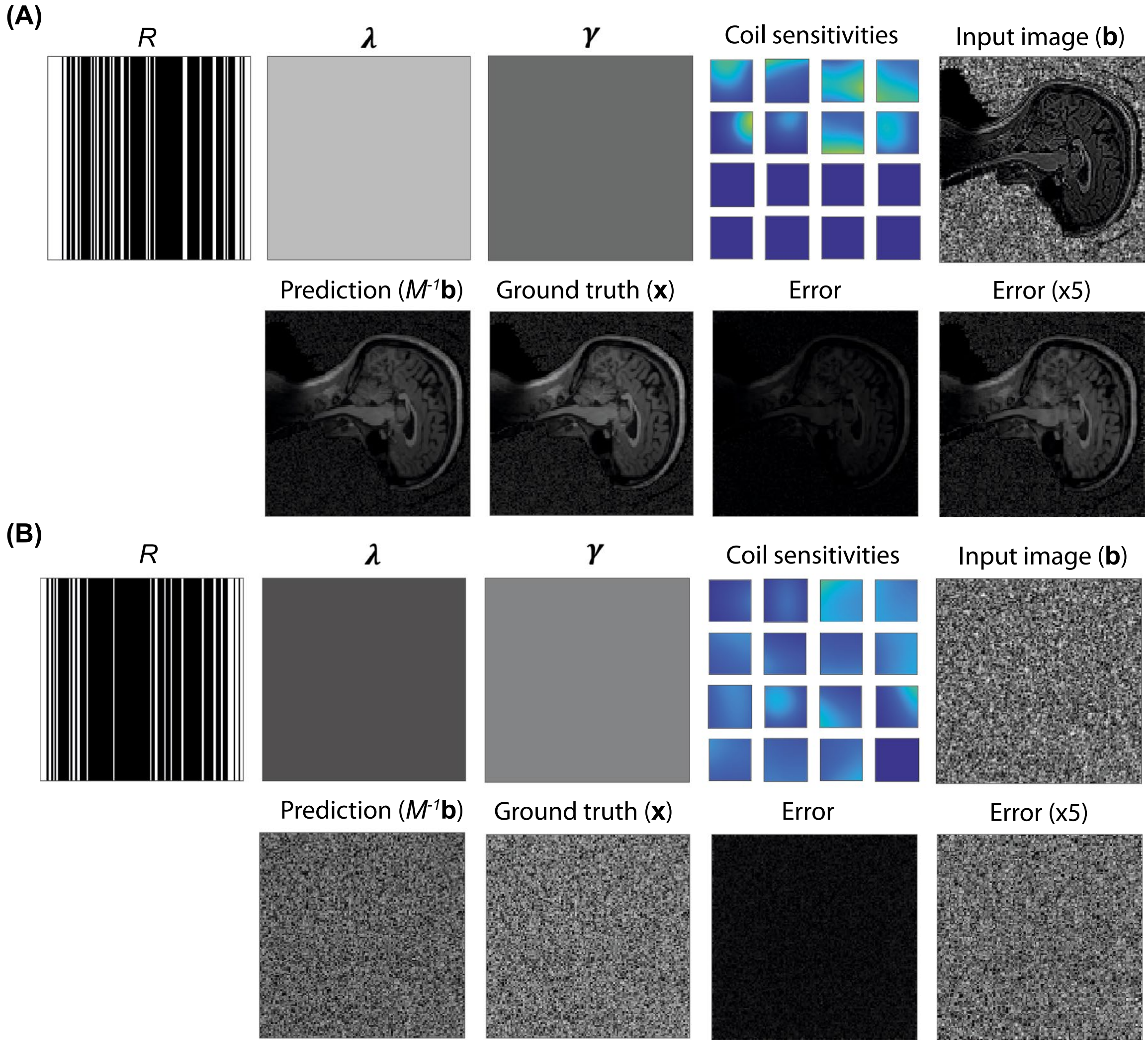

FIGURE 3 The network's prediction performance for two validation examples. The network's prediction, $M^{-1}\mathbf{b}$, is close to the ground truth, $\mathbf{x}$, which is confirmed by the small error values both for the brain case (A) and for the noise case (B). The brain example was simulated with an 8-channel receive coil and an undersampling factor of 3, whereas the noise example was simulated with a 15-channel receive coil and an undersampling factor of 4. A Fourier shift was applied to the sampling mask as a preprocessing step before training. The brain images in (A) were rotated by the data augmentation step

$M^{-1}\mathbf{b}$ of the operation $A^{-1}\mathbf{b}$ is close to the ground truth image $\mathbf{x}$ for both cases. This is confirmed by the low absolute errors (normalized ℓ_1-norm <0.18) in the error map.

The reconstruction quality and the convergence behavior with and without the learned preconditioner integrated in SB is shown in Figure 4 for an IR TSE brain scan ($R = 4$, matrix size 256 × 256, 15 coil elements) and an FFE knee scan ($R = 3$, matrix size 128 × 128, 16 coil elements). The learned preconditioner results in an acceleration factor of 4.0 in CG compared to the reconstruction without preconditioner. This is in the same range as the acceleration factor of 4.3 obtained with the circulant preconditioner. The reconstruction error plotted as a function of time for the entire SB algorithm shows that the learned preconditioner is slightly more expensive compared to the circulant preconditioner in terms of computing $M^{-1}\mathbf{b}$ in each CG iteration.

Similar results are presented in Figure 5 for a prospectively undersampled FLAIR scan ($R = 2$, matrix size 240 × 224, 15 coil elements) and a TSE scan ($R = 2$, matrix size 256 × 256, 13 coil elements) in the brain, which show the ability for the learned preconditioner to slightly outperform the circulant preconditioner. Figure 5B also shows that the model trained on both brain and noise images results in a much better preconditioning performance than a model trained on either noise or brain images.

4 | DISCUSSION

The results in this paper showed that it is possible to learn a preconditioner for CS and PI reconstructions using a relatively small CNN. The model was designed such that

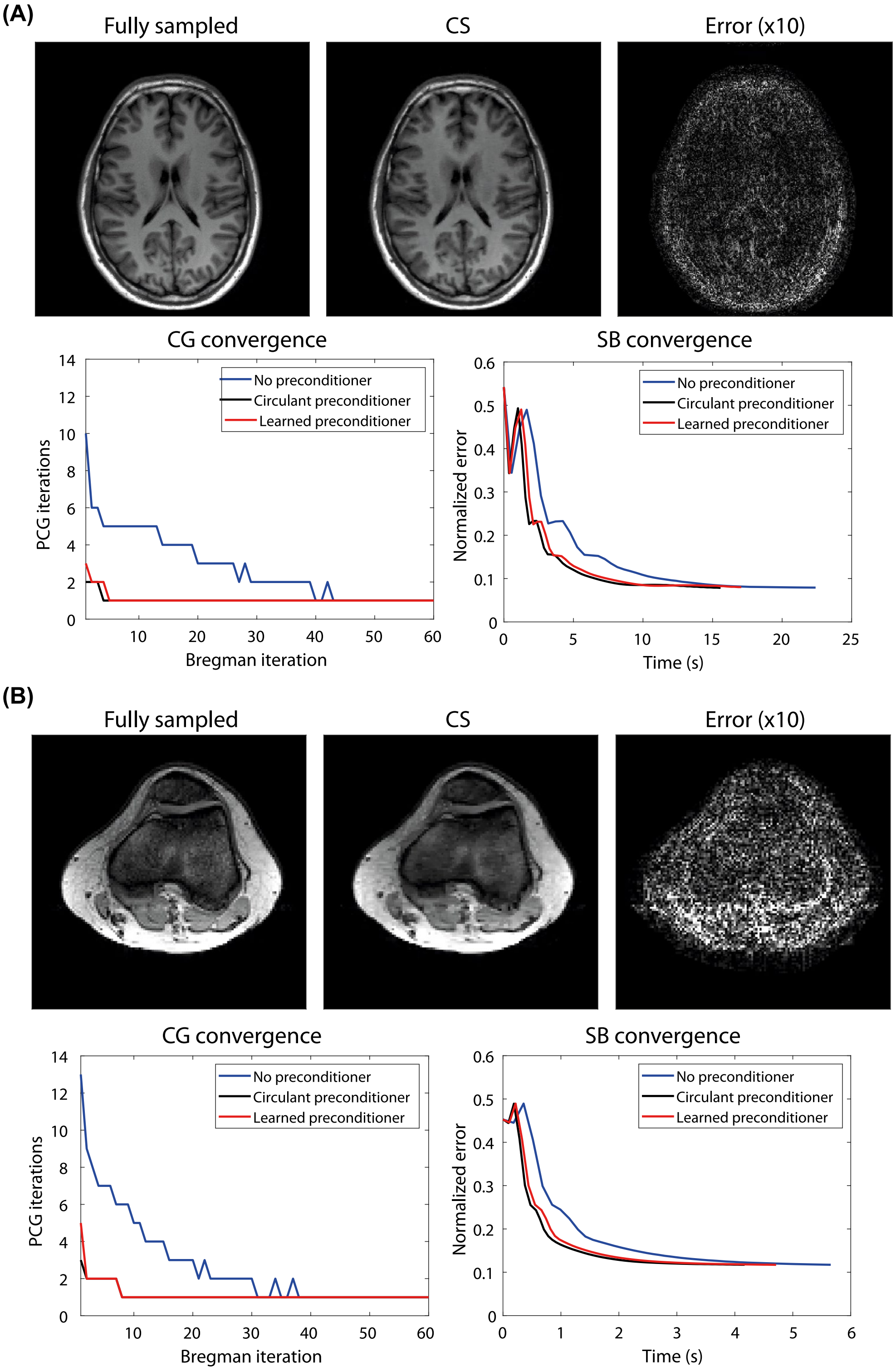

FIGURE 4 Accelerated SB reconstructions of retrospectively undersampled data using the learned and the circulant preconditioner. A, The CS reconstruction of an IR TSE brain scan ($R = 4$, 15 coil elements) is close to the fully sampled image, which is confirmed by the low values in the error map (magnified 10 times). The learned preconditioner reduces the number of PCG iterations by a factor of 4.0 over 20 Bregman iterations, which is in the same order as the acceleration factor of 4.3 obtained with the circulant preconditioner. The final acceleration for the entire SB algorithm is slightly larger for the circulant preconditioner than for the learned preconditioner because of the slightly more efficient computation of $M^{-1}\mathbf{b}$ for the circulant preconditioner compared to the learned preconditioner. B, Similar results are obtained for an FFE scan in the knee, acquired with an undersampling factor of 3 and 16 coil elements. The regularization parameters were set to (A): $\lambda = 4, \gamma = 2$ and (B): $\lambda = 4, \gamma = 1$. The number of outer/inner Bregman iterations was set to 60/1. The reconstruction time with the learned preconditioner was (A): 16.9 s and (B): 4.7 s

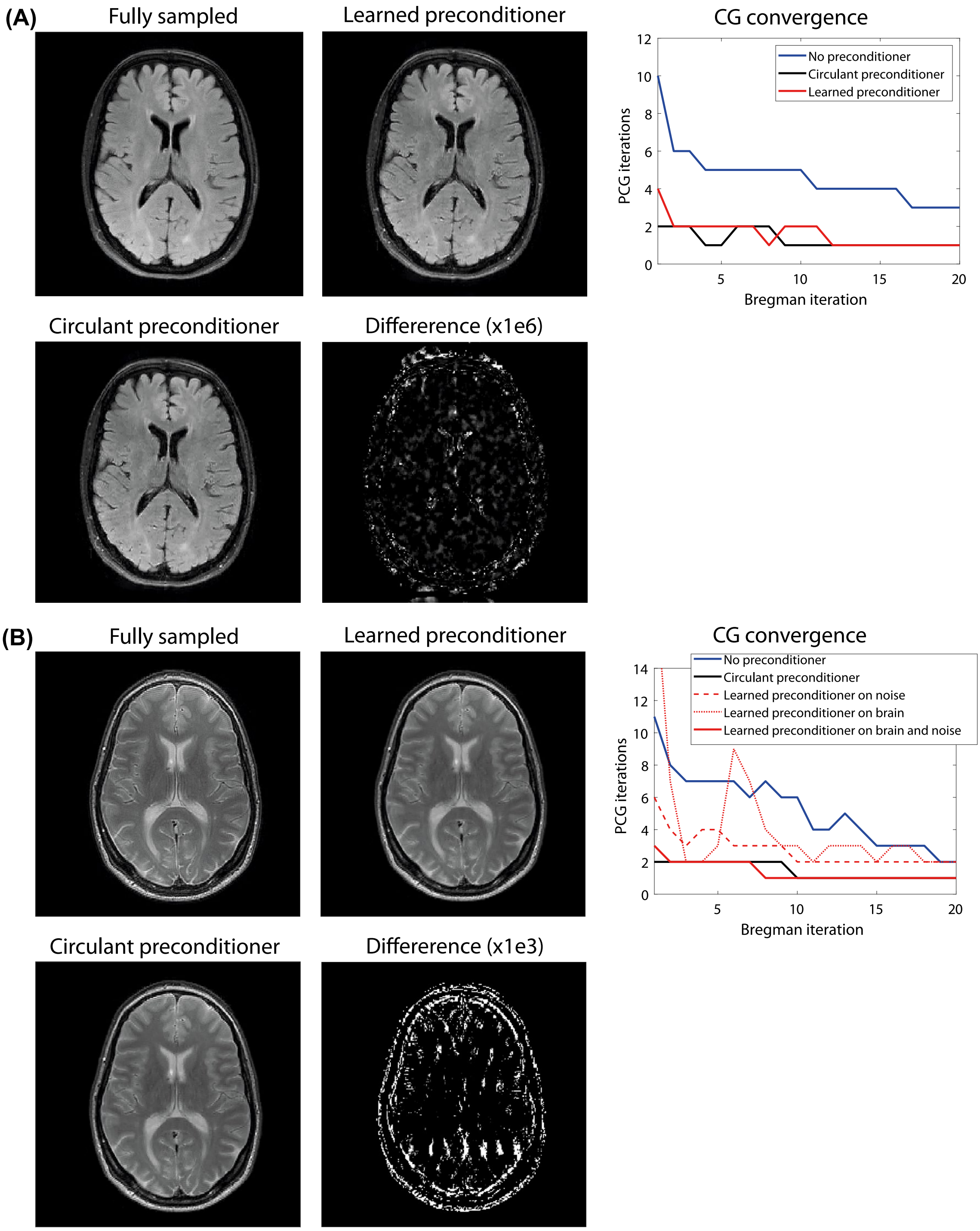

FIGURE 5 Accelerated SB reconstructions of prospectively undersampled data using the learned and the circulant preconditioner. A, The CS reconstructions of a FLAIR brain scan ($R = 2$, 15 coil elements, matrix size of 240×224) are close to the fully sampled image. Note that the relative difference (magnified 1e6 times) between de-reconstructed images obtained with the learned and the circulant preconditioner are neglible, since the CG tolerance level was fixed for all reconstructions. The learned preconditioner reduces the number of PCG iterations, yielding a similar performance as the circulant preconditioner. B, Similar results are obtained for a TSE brain scan in the brain, acquired with an undersampling factor of 2, 13 coil elements and a matrix size of 256×256. These results show that the performance of the preconditioner is much more stable when the preconditioner is trained on noise images only compared to when the preconditioner is trained on brain images only. This suggests that the distribution of the brain images from the human connectome project database is different than that of the acquired test images. This could be due to a difference in SNR level, image contrast or image phase, for example. Training the preconditioner on both brain and noise images in equal amount results in the best preconditioning performance. The regularization parameters were set to (A): $\lambda = 3, \gamma = 2$ and (B): $\lambda = 4, \gamma = 1$. The number of outer/inner Bregman iterations was set to 20/1. The reconstruction time with the learned preconditioner was (A): 5.1 s and (B): 5.6 s

the learned preconditioner operates on a vector, which makes integrating the preconditioner in a numerical scheme computationally inexpensive and memory efficient compared to learning a full preconditioning matrix. The learned preconditioner reduced the number of PCG iterations by a factor of approximately 4, hereby meeting the performance of an efficient circulant preconditioner.[8] The training data were simulated for a wide range of sampling masks, coil sensitivity maps and regularization parameters, such that the same trained model can be used to reconstruct images acquired with different coil configurations and of different anatomies. This provides us with a flexible preconditioning framework, in which the learned preconditioner is not restricted to a certain structure of the system matrix, as is the case for circulant preconditioners.

For this proof-of-principle study, a relatively small CNN was chosen as network architecture, on the one hand to support accelerated reconstructions of varying matrix size and resolution and on the other hand to limit the inference time. With this simple network architecture, the learned preconditioner was able to meet and sometimes slightly improve upon the performance of the efficient circulant preconditioner. Network, model and training set optimization is expected to increase the acceleration factor further, although these design choices are not trivial. Supporting Information Figure S1 shows, for example, that using a deeper network does not directly lead to a better preconditioning performance. Furthermore, by learning the preconditioner's action on a vector instead of learning the preconditioning matrix itself, the preconditioning performance becomes dependent on the input vector. To minimize this effect, the input vectors should be as general as possible, with as little coherent structures between training examples as possible. Therefore, combining noise and brain images in the training set results in a better preconditioning performance than using only noise or brain images. Further research is needed to investigate which image types complement the current brain and noise images best for optimal generalizability of the preconditioner. Finally, the Adam optimizer has shown good performance in combination with the current network architecture,[26] but it is worth investigating whether other optimizers can achieve a higher preconditioning performance for the test data in this application.

The current method has demonstrated the feasibility of learning a preconditioner for 2D CS reconstructions. To provide the deep learning model with as much information about the underlying MR physics as possible, coil sensitivity maps and the sampling mask were fed to the network along with the input image and regularization parameters, yielding a 37-channel network input for each training example. Following the same procedure for 3D reconstructions would increase the dimensionality of the network input further due to the volumetric nature of the coil sensitivity maps. Such a large network input may hinder efficient learning of the inverse operation of the SB system matrix. Coil compression and grouped convolutions could be used to overcome this limitation and coil sensitivity map-independent preconditioners should be investigated.

This work focussed on preconditioning as a means to speed up classical model-based reconstructions, while fully learned reconstructions are often inherently fast. The advantage of accelerating reconstructions via preconditioning compared to using fully learned reconstructions is that preconditioners do not affect the reconstruction quality, whereas the stability of fully learned reconstructions still needs refining.[20] An ill-designed preconditioner could, in a worst-case scenario, either result in a slower convergence behavior than reconstruction without the preconditioner. We cannot guarantee that the learned network leads to (fast) convergence of CG in all cases, but a nonconverging reconstruction is easily and automatically detected by monitoring the CG residual. In our work we observed fast convergence behavior for all studied cases. However, besides having short reconstruction times, the large degree of freedom in fully learned reconstructions can also help to better exploit image features such as sparsity,[17] possibly allowing reconstructions from higher undersampling factors. Systems of similar structure as the one described in Equations (2)–(4) are often encountered in such variational networks, which could therefore also benefit from the preconditioning technique described in this work. A preconditioner could potentially be learned simultaneously with the reconstruction model, similar to how ADMM-Net learns the inverse operation corresponding to the system matrix.[28] Integration of learned preconditioners in variational networks might reduce the number of network blocks (representing CS iterations) needed for convergence and hence accelerate the learned reconstruction process further. This could be particularly relevant for applications where real-time reconstructions are needed.

In conclusion, it is possible to learn a preconditioner for CS and PI reconstructions using a relatively small CNN. Such an approach can potentially help to further accelerate CS reconstructions compared to existing preconditioners. Future research is needed to optimize the efficiency of the learned preconditioner.

ACKNOWLEDGEMENTS

This publication is part of the project HTSM (with project number 17104) which is financed by the Dutch Research

Council (NWO). We would like to thank the Division of Image Processing from the Leiden University Medical Center for providing computing recourses. We thank Jeroen van Gemert for inspiring brainstorming sessions.

ORCID

Kirsten Koolstra https://orcid.org/0000-0002-7873-1511

REFERENCES

1. Liang D, Liu B, Wang J, Ying L. Accelerating SENSE using compressed sensing. *Magn Reson Med*. 2009;62:1574-1584.
2. Chandarana H, Feng L, Block TK, et al. Free-breathing contrast-enhanced multiphase MRI of the liver using a combination of compressed sensing, parallel imaging, and golden-angle radial sampling. *Invest Radiol*. 2013;48:10-16.
3. 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.
4. Vasanawala S, Murphy M, Alley M, et al. Practical parallel imaging compressed sensing MRI: summary of two years of experience in accelerating body MRI of pediatric patients. In *2011 IEEE international Symposium on Biomedical Imaging: From Nano to Macro*. 2011:1039-1043.
5. Murphy M, Alley M, Demmel J, Keutzer K, Vasanawala S, Lustig M. Fast l1-SPIRiT compressed sensing parallel imaging MRI: scalable parallel implementation and clinically feasible runtime. *IEEE Trans Med Imaging*. 2012;31:1250-1262.
6. Chen C, Li Y, Axel L, Huang J. Real time dynamic MRI with dynamic total variation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*. 2014;17:138-145.
7. Xu Z, Wang S, Li Y, Zhu F, Huang J. PRIM: an efficient preconditioning iterative reweighted least squares method for parallel brain MRI reconstruction. *Neuroinformatics*. 2018;16:425-430.
8. Koolstra K, van Gemert J, Börnert P, Webb A, Remis R. Accelerating compressed sensing in parallel imaging reconstructions using an efficient circulant preconditioner for cartesian trajectories. *Magn Reson Med*. 2019;81:670-685.
9. Ong F, Uecker M, Lustig M. Accelerating non-cartesian MRI reconstruction convergence using k-space preconditioning. *IEEE Trans Med Imaging*. 2020;39:1646-1654.
10. Boyd S, Vandenberghe L. *Convex Optimization*. Cambridge University Press; 2004.
11. Cauley SF, Xi Y, Bilgic B, et al. Fast reconstruction for multichannel compressed sensing using a hierarchically semiseparable solver. *Magn Reson Med*. 2015;73:1034-1040.
12. Wang S, Su Z, Ying L, et al. Accelerating magnetic resonance imaging via deep learning. In *13th IEEE International Symposium on Biomedical Imaging (ISBI 2016)*. 2016:514–517.
13. Lee D, Yoo J, Ye JC. Deep residual learning for compressed sensing MRI. In *14th IEEE International Symposium on Biomedical Imaging (ISBI 2017)*. 2017:15–18.
14. Wang G, Ye JC, Mueller K, Fessler JA. Image reconstruction is a new frontier of machine learning. *IEEE Trans Med Imaging*. 2018;37:1289-1296.
15. Zhu B, Liu JZ, Cauley SF, Rosen BR, Rosen MS. Image reconstruction by domain-transform manifold learning. *Nature*. 2018;555:487-492.
16. Lee D, Yoo J, Tak S, Ye JC. Deep residual learning for accelerated MRI using magnitude and phase networks. *IEEE Trans Biomed Eng*. 2018;65:1985-1995.
17. Hammernik K, Klatzer T, Kobler E, et al. Learning a variational network for reconstruction of accelerated MRI data. *Magn Reson Med*. 2018;79:3055-3071.
18. Pezzotti N. An adaptive intelligence algorithm for undersampled knee MRI reconstruction. *IEEE Access*. 2020;8:204825-204838.
19. 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.
20. Antun V, Renna F, Poon C, Adcock B, Hansen A. On instabilities of deep learning in image reconstruction and the potential costs of AI. *Proc Nat Acad Sci*. 2020;117:30088-300950095.
21. Wang H, Cheng J, Jia S, et al. Accelerating MR imaging via deep Chambolle-Pock network. *41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. 2019:6818-6821.
22. Cheng J, Wang H, Ying L, Liang D. Model learning: primal dual networks for fast MR imaging. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer; 2019:21-29.
23. Bilgic B, Chatnuntawech I, Manhard MK, et al. Highly accelerated multishot echo planar imaging through synergistic machine learning and joint reconstruction. *Magn Reson Med*. 2019;82:1343-1358.
24. Goldstein T, Osher S. The split Bregman method for L1-regularized problems. *SIAM J Imaging Sci*. 2009;2:323-343.
25. Van Essen DC, Smith SM, Barch DM, et al. The WU-Minn human connectome project: an overview. *NeuroImage*. 2013;80:62-79.
26. Zeng DY, Shaikh J, Holmes S, et al. Deep residual network for off-resonance artifact correction with application to pediatric body MRA with 3D cones. *Magn Reson Med*. 2019;82:1398-1411.
27. 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.
28. Yang Y, Sun J, Li H, Xu Z. Deep ADMM-Net for compressive sensing MRI. *Proc 30th Int Neural Inf Process Syst*. 2016;29:10-18.

SUPPORTING INFORMATION

Additional supporting information may be found in the online version of the article at the publisher's website.

FIGURE S1 Comparison of the preconditioning performance for different number of network layers and network blocks. During training of the network, the model was stored for each of the 50 epoch. Afterwards, the brain TSE image (see Fig. 5B) was reconstructed 50 times (x-axis), each time using the stored model from a different training epoch as preconditioner. The y-axis represents the outer Bregman iteration number. The colors represent the number of CG iterations needed to reach the desired tolerance level in each outer Bregman iteration. (A) The network used in this work contains 3 blocks with 2 layers each. The total number of CG iterations decreases as the

epoch number increases. The convergence behavior does not vary much after 40 epoch. (B) Using a larger number of network blocks (5 blocks, 2 layers each) results in a similar convergence behavior for some of the epoch, but shows unstable behavior for larger epoch numbers. This is an indication that the network starts to overfit after approximately 40 epoch. (C) When a larger number of network layers is used in each block (3 blocks, 4 layers each) this unstable behavior is not observed, but the required number of CG iterations is still larger than for the small network used in (A). Note that the maximum number of CG iterations was set to 20 to detect nonconvergent behavior for the early epoch

How to cite this article: Koolstra K, Remis R. Learning a preconditioner to accelerate compressed sensing reconstructions in MRI. *Magn Reson Med*. 2022;87:2063–2073. doi:10.1002/mrm.29073