Spatiotemporal Context-Informed Unrolled Complex Reconstruction Networks for 4D Flow MRI

Alex Macintyre[1,2][0009-0003-0000-2882], Lingceng Ma[1][0009-0008-5620-5597], and Anthony G. Christodoulou[1, 2][0000-0002-9334-8684]

[1] Department of Radiological Sciences, David Geffen School of Medicine at UCLA, Los Angeles, United States of America
[2] Physics and Biology in Medicine Graduate Program, David Geffen School of Medicine at UCLA, Los Angeles, California
achristodoulou@mednet.ucla.edu

Abstract. 4D Flow MRI provides quantitative cardiac phase resolved hemodynamic assessment, however its clinical use remains limited due to long scan times. In this work, we train an unrolled deep learning network on a large dataset for 4D Flow MRI reconstruction. Our approach leverages complex convolutional blocks for improved phase reconstruction fidelity. Our models use feature-sharing across the velocity encode, cardiac phase, and slice dimension and use ACS-free sensitivity map estimation networks yielding impressive reconstruction results up to 50× acceleration factor.

Keywords: MRI Reconstruction, 4D Flow MRI, Deep Learning for Inverse Problems

1 Introduction

4D Flow MRI quantitatively measures cardiac-phase-resolved blood-flow velocity vector fields. It has clinical applications in congenital heart disease, valvular disease, and cardiovascular surgical planning and follow-up [1]. However, widespread adoption remains limited by long scan times [2] arising from high dimensionality cardiac-phase-resolved volumetric coverage for three velocity-encoding directions and a background reference encode.

Scan time can be reduced through k-space undersampling followed by parallel imaging [3, 4], compressed sensing [5, 6], or deep-learning reconstruction [7, 8]. In recent years, unrolled deep-learning networks have achieved the highest reconstruction fidelity across multiple cardiac MRI modalities in the CMRxRecon benchmarks [9, 10].

CMRxFlow differs from prior CMRxRecon challenges in three important ways. It focuses exclusively on 4D Flow MRI, evaluates quantitative velocity fidelity from image phase in addition to magnitude-based image quality, and includes acceleration factors up to 50× without a fully sampled autocalibration signal (ACS) region, compared with up to 24× with ACS in prior challenges. Beyond these acquisition demands, 4D Flow MRI is itself distinctive in both where its clinical information resides and how its data are structured: velocity information is encoded in image phase rather than

magnitude, and the data are five-dimensional, spanning three spatial dimensions, cardiac phase, and velocity encoding direction.

Most deep-learning reconstruction networks represent complex-valued images as separate real and imaginary components processed by real-valued convolutions, which may limit performance in this natively complex application. Complex convolutional networks have shown improved phase fidelity [11], making them particularly relevant at high acceleration factors, where small phase errors directly produce velocity errors.

4D Flow MRI contains substantial multidimensional redundancy. The same anatomy is imaged for the background reference and three velocity encodes, and across multiple cardiac phases. Fluid-flow physics connects these five dimensions through conservation of mass and momentum, as described by the Navier-Stokes equations. Prior networks have exploited subsets of this redundancy by jointly reconstructing all four encodes with feature sharing [12] or learning sparse temporal representations from temporally incoherent sampling [13]. Although partial-volume effects, eddy currents, residual acceleration, and higher-order terms may produce departures from Navier-Stokes-consistent flow, the governing equations still constrain the solution space. However, scarce fully sampled data and the curse of dimensionality make fully five-dimensional networks difficult to train. The large data available in this new CMRxFlow challenge creates a new opportunity for architectures that exchange information across all dimensions.

We therefore develop an unrolled reconstruction model architecture tailored to 4D Flow MRI. It jointly processes all velocity encodes and cardiac phases with feature sharing across these dimensions and incorporates adjacent spatial slices to promote flow-physics-consistent solutions. These operations connect velocity-encoding, temporal, and spatial information while retaining limited receptive fields to avoid the even larger data requirements of a fully global five-dimensional network. To preserve phase fidelity, the model uses complex convolutions rather than real-valued convolutions applied to split real and imaginary inputs. Because joint sensitivity-map estimation can improve reconstruction [14], each cascade also includes a smaller, similarly constructed network that progressively refines cardiac-phase-resolved sensitivity maps using the initial and current estimates and an image residual.

We train and evaluate this model architecture on the multi-center CMRxFlow2026 dataset and show strong reconstruction results up to 50× k-space undersampling factors.

2 Methods

2.1 Unrolled Reconstruction

In undersampled MRI reconstruction, we seek to find an image that minimizes a cost function consisting of data consistency and a regularization term:

$$\hat{x} = \arg\min_{x} \|Ax - y\|_2^2 + R(x) \tag{1}$$

where the forward encoding operator $A = MFS$, consists of receive coil sensitivity maps S, Fourier transform F, and the k-space undersampling mask M.

In unrolled deep learning reconstruction, the inverse problem is solved iteratively; at each iteration a network learns an approximate proximal operator for the regularization term $R(x)$ and the output goes through a data consistency step.

While the CMRxFlow2026 dataset provided pre-computed sensitivity maps, it has been shown that jointly estimating sensitivity maps can help improve reconstruction fidelity [11]. Prior CMRxRecon series winners [9, 10] performed sensitivity map estimation using a network that took in the fully sampled ACS to derive sensitivity map corrections.

In lieu of the ACS, at each cascade our model's coil sensitivity map corrector ($CSMC$) takes in the provided sensitivity maps (S_0), the last cascade's estimated sensitivity maps (S_{i-1}), the previous cascade's image reference encode magnitude at each time point ($\left|x_{i-1}^{ref}\right|$), a coil-wise sensitivity data consistency gradient from the last cascade ($g_{i-1,c}$), and a conditioning vector (c) based on k-space undersampling factor and encoded velocity polarity:

$$S_i = S_{i-1} + CSMC\left(S_0, S_{i-1}, \left|x_{i-1}^{ref}\right|, g_{i-1,c}, c\right) \tag{2}$$

where g_{i-1} is defined as:

$$g_{i-1,c} = \left(x_{i-1}^{ref}\right)^* \odot F^{-1}[M\left(FS_{i-1}\, x_{i-1}^{ref} - y_c^{ref}\right)] \tag{3}$$

In this expression, multiplying $\left(x_{i-1}^{ref}\right)^*$ removes the current image estimation's phase component, so the sensitivity gradient is phase-referenced to the current image. This helps decouple the CSMC model which aims to learn the coil sensitivity phase from the image network which aims to learn the flow-dependent phase.

While the provided sensitivity maps were constant over all cardiac phases, the CSMC predicts a residual update per cardiac phase, which can enhance performance as sensitivity can change due to B field fluctuations from cardiac motion [15].

The CSMC was configured to output 10 coil map correction terms per cardiac phase, which was the number of coils for most scans in the TaskR1R2 dataset. For scans without 10 coils, S_0 was used at every unroll.

Following sensitivity map estimation at each iteration, the image network strives to learn the noise which is subtracted from the input image to get the predicted image:

$$a_i,\ h_{i+1} = UNet(x_i, A_i^H A_i x_i, A_0^H y,\ h_i, m, c) \tag{4}$$

$$x_{pred,i} = x_i - a_i \tag{5}$$

Like PromptMR+ [10], our image correction UNet takes as input, not just the current image but the normal of the image estimate, the initial zero-filled image, a learned hidden state h_i (where h_0 is the zero-filled image), and a conditioning vector c. Additionally, our network takes in the provided blood flow segmentation mask, m.

From $x_{pred,i}$, the next iteration's image estimate comes from a data consistency step with the current image estimate and the measured data. At each cascade, the data consistency weighting term, w_i, is learned, conditioned on the acceleration factor:

$$x_{i+1} = E_i^H\left[(1-M)E_i x_{pred,i} + M\left(w_i E_i x_{i,pred} + (1-w_i)y\right)\right] \quad (6)$$

Where E_i is the fully sampled multi-coil encoding operator; $E_i = FS_i$.

2.2 Model Architecture

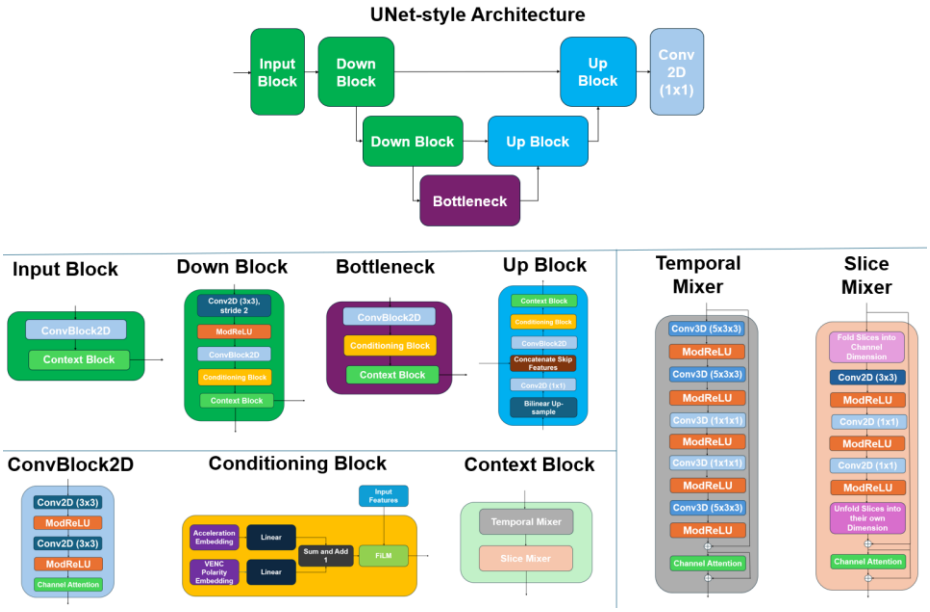

Fig. 1. UNet-style architecture and its components. This architecture is used at every cascade by both the CSMC and image networks.

At each cascade, unique parameters are learned for the coil sensitivity and image networks. Both these networks are UNet-inspired [16] that consist of an input block, 2 Down blocks, a bottleneck, 2 Up blocks and an output convolution. The network accepts 3 image slices across all 4 velocity encodes and timepoints. Outside of Context Blocks, image slices at each timepoint are processed independently, but the velocity encodes are processed together as separate channels. To account for intensity difference between scans, the initial image estimate is divided by a normalization factor to have mean magnitude 1. The output images are multiplied by this normalization factor to revert the scaling.

Both the sensitivity and image networks learn complex convolutions, with the phase-preserving modReLU [17] activation function:

$$\text{modReLU}(z) = \begin{cases} (|z| + b)\dfrac{z}{|z|}, & \text{if } |z| + b \geq 0 \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

Inputs to the Down blocks first go through a stride 2, 2D in-plane convolutions to spatially down sample and increase feature count. Then the down-sampled output is passed

through a ConvBlock2D, which consists of two 2D convolutional layers with ModReLU activation and channel attention. From the context vector (which encodes the acceleration factor and velocity polarity), the complex outputs are transformed by a learned, real Feature-wise Linear Modulation (FiLM) [18] layer. The last component of the Down block is the Context block.

While previous layers only acted on a single slice at one timepoint, the Context block modulates features through sharing features first across the cardiac phase dimension, and then the spatial dimension. Temporal feature sharing is performed through 2 3D complex convolutional layers across the in-plane slice direction and cardiac phase dimension. Along the cardiac phase dimension, the arrays are circularly padded to mimic the cyclical nature of the cardiac cycle. In between these convolutional layers, there is pointwise channel mixing. The output of the temporal block residually scales the input features to the temporal block. The slice mixer block also learns a residual scaling to the input block. The input features first go through a 2D in-plane convolutional layer, and then the slice dimension is folded into the channel dimension. Pointwise channel mixing is used to share information between the features at every slice. The output is then reshaped to restore the slice dimension; these features residually scale the input features to the slice mixer. Real residual channel attention is used after both the temporal and slice mixers.

The bottleneck inputs are through a ConvBlock2D, FiLM layer, and Context block. The input to the Up block is first bilinearly up-sampled, expanding the spatial size of the features. After a pointwise convolution, the features from the Down block at the respective feature spatial size are concatenated, which are processed through a ConvBlock2D, FiLM layer, and context block. After the 2 Up blocks, a pointwise convolution yields the network output.

2.3 Model Training

Training was conducted using a single Nvidia H100 GPU. Three models were trained, a large model used for Tasks R1, an intermediate-sized model for Tasks S1 and S2, and a lightweight model used for fast inference on Task R2. All models were trained on 118 raw k-space scans from the CMRxFlow2026 TaskR1R2 training data set. 20 scans from this dataset were withheld for in-training validation. The large and intermediate models were trained using a combined complex L2 loss across the entire image plus two additional loss terms computed over the blood flow segmentation mask, m:

$$Loss = \frac{\|\hat{x}-x\|_2^2}{|\hat{x}|^2+\varepsilon} + \lambda_1 \frac{\sum_{i=1,2,3} m|\hat{x}|^2(\Delta\varphi_i)^2}{(\pi^2 \sum_{i=1,2,3} m|\hat{x}|^2)+\varepsilon} + \lambda_2 \frac{\sum m(|\hat{x}|-|x|)^2}{\sum m|\hat{x}|^2+\varepsilon} \quad (8)$$

The second loss term is a magnitude-weighted phase difference loss. Here, φ_i is defined as the phase difference between the phase of the reference encode and the i-th velocity encode. $\Delta\varphi_i$ is the difference between the phase difference in the predicted images and the reference images. Since MRI noise is modeled as additive white Gaussian noise [19], at lower magnitudes, the expected phase difference error is higher. The second loss term was designed so low magnitude phase error would not dominate the overall

loss. The third loss term is an image magnitude L2 term in the mask. The large and intermediate models were trained with $\lambda_1 = 10$ and $\lambda_2 = 1$.

The small model was trained only using complex L2 loss, with double weighting inside the blood flow segmentation mask.

The intermediate model for Tasks S1 and S2 consisted of CSMC networks with 16 base features and the image network with 48 base features at each cascade. The model was trained in stages: first a 15 cascade image network with one slice (i.e. no slice context) was pre-trained. Then the CSMC networks were added and trained. Interior cascades were duplicated [9] to bring the cascade count to 28 and finally Slice Mixer blocks were added and 3 slices were used as input. The intermediate model has 85 million total parameters. Over these training stages, the learn rate was dropped from 5×10^{-4} to 1×10^{-5}. The AdamW optimizer was used with small weight decay (0.005). The large model for Task R1 was built off the intermediate model, duplicating the inner-most cascades, first to bring the model to 36 cascades, then finetuning and duplicating the new inner-most cascades to bring the final model to 45 cascades. The learn rate was 1×10^{-5} for these training stages using AdamW optimizer with 0.005 weight decay. The final 45 cascade large model has 136 million parameters.

For fast reconstructions on consumer-grade GPUs, a smaller model was trained for Task R2. This model omitted the CSMC networks and only had 2 cascades of image networks with 16 base features, totaling 144 thousand parameters. For faster inference, this model did not have Slice Mixer blocks and took in one slice only. This model was trained for 300 epochs with 5×10^{-4} initial learn rate and cosine learn-rate annealing using an AdamW optimizer with 0.005 weight decay.

3 Experiments and Results

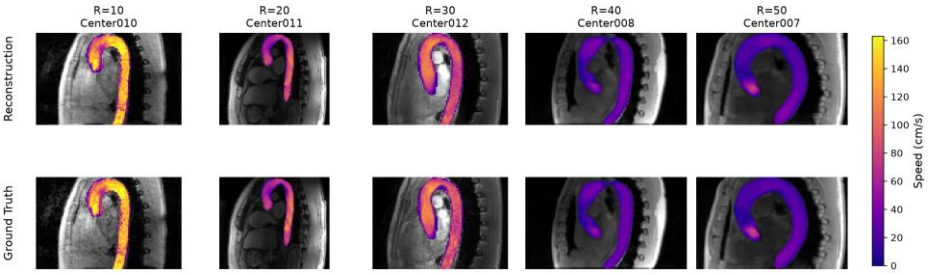

Fig. 2. The large model outputs at acceleration factors 10-50 on scans withheld from the training set. Reference velocity encoding magnitude was plotted, overlaid with calculated blood flow speed. No post-processing was performed.

Table 1. TaskR1 - Large Model In-Domain - Validation Results

	SSIM	NRMSE	RelErr	AngErr
Center 7	0.9704	0.0316	0.2196	20.78
Center 8	0.9919	0.0131	0.1868	17.64
Center 10	0.9398	0.0519	0.2971	27.68
Center 11	0.9667	0.0467	0.3179	32.06
Center 12	0.9755	0.0259	0.2811	23.47
Overall	0.9741	0.0286	0.2282	21.37

Table 2. TaskR2 - Small Model In-Domain - Validation Results

	SSIM	NRMSE	RelErr	AngErr
Center 7	0.9431	0.0504	0.3329	30.05
Center 8	0.9728	0.0284	0.3353	31.02
Center 10	0.9062	0.0686	0.3583	33.04
Center 11	0.9301	0.0729	0.4045	39.01
Center 12	0.9372	0.0425	0.3980	30.86
Overall	0.9469	0.0467	0.3457	31.11

Table 3. TaskS1 - Generalization to New Center and Diseases -Validation Results

	SSIM	NRMSE	RelErr	AngErr
Center 6	0.9867	0.0397	0.2989	22.04

Table 4. TaskS2 – Generalization to New Anatomy - Validation Results

	SSIM	NRMSE	RelErr	AngErr
Carotid	0.9983	0.0122	0.1773	18.26
Cerebrovascular	0.9933	0.0201	0.3347	30.71
Portal Vein	0.9873	0.0415	0.3764	32.66
Renal Artery	0.9881	0.0341	0.3330	26.15
Overall	0.9918	0.0270	0.3054	26.95

Figure 2 shows strong qualitative reconstruction performance, even at high acceleration rates. Tables 1-4 show quantitative results on the official validation set for Tasks R1, R2, S1, and S2 respectively. The evaluation metrics were calculated only within the flow segmentation mask. SSIM and NRMSE were calculated from the magnitude images, while Relative Error (RelErr, a speed error metric) and Angular Error (AngErr) were calculated from the calculated velocity field maps.

The large model was used for Tasks R1. Task R1 evaluated its performance on scans from the same center and anatomy (Aorta) as the training set. The intermediate-sized model was used for Task S1 and S2 inference. Task S1 consisted of the same anatomy, but new diseases at a center not seen in the training set. Task S2 consisted of 4D Flow MRI scans from different anatomies.

The smaller model was evaluated on Task R2, with the goal of providing faster reconstruction. While the smaller model did not perform as well as the larger model, it still outperformed the baseline FlowVN [8] on 30 out of the 32 scans in the official TaskR1R2 validation set. The smaller model was able to process a scan in under 10 seconds, while the intermediate model took about 6 minutes per scan and the large model took about 10 minutes per scan on a Nvidia L40S.

4 Discussion and Conclusion

In this work, we present unrolled deep learning models for 4D Flow MRI reconstruction. Using complex convolutions and feature sharing across the spatial, velocity encode, and cardiac phase dimensions our model can reconstruct scans with high qualitative and quantitative performance at up to a 50× undersampling rate. This was enabled using a large, multi-center training corpus.

Some limitations of the current design are that this CSMC implementation was specifically for 10-coil input and therefore was not applied to scans with other coil configurations; this should be generalized for future use. Only 1 of 40 scans in the Task R1R2 and 3 of 40 scans in the Task S2 did not have 10 coils, but all the scans in the Task S1 validation set did not have 10 coils. Furthermore, these models use a flow-region

segmentation mask as an input, which in practice would require reconstruction and segmentation from a single cardiac phase before model inference; this may hinder clinical translation. The evaluation study did not include component-wise ablations, so the individual contributions of the proposed architectural and loss-function components cannot be isolated from the present results. Additionally, our model sees a performance drop in quantitative velocity fidelity metrics when the model is evaluated on scans from another center and other anatomies. This presents a key challenge for the clinical implementation of deep learning 4D flow reconstruction algorithms, as models trained on a healthy cohort may not generalize to congenital heart disease anatomies. To help alleviate these concerns, future work will explore training-data free approaches [20] and further methods to impose Navier-Stokes physics constraints.

Acknowledgments. This research was partially funded by the UCLA Asrican Sophie and Jack Award (A.M.). We thank the CMRxFlow2026 organizers for providing the k-space datasets and the UCLA Center for Computer Vision and Imaging Biomarkers (CVIB), Department of Radiological Sciences, for providing computational resources.

Disclosure of Interests. The authors have no relevant financial interests to disclose.

References

1. Azarine, A., Garçon, P., Stansal, A., Canepa, N., Angelopoulos, G., Silvera, S., Sidi, D., Marteau, V., Zins, M.: Four-dimensional Flow MRI: Principles and Cardiovascular Applications. RadioGraphics. 39, 632–648 (2019). https://doi.org/10.1148/rg.2019180091.
2. Markl, M., Schnell, S., Barker, A.J.: 4D Flow Imaging: Current Status to Future Clinical Applications. Curr Cardiol Rep. 16, 481 (2014). https://doi.org/10.1007/s11886-014-0481-8.
3. Pruessmann, K.P., Weiger, M., Scheidegger, M.B., Boesiger, P.: SENSE: Sensitivity encoding for fast MRI. Magnetic Resonance in Medicine. 42, 952–962 (1999). https://doi.org/10.1002/(SICI)1522-2594(199911)42:5<952::AID-MRM16>3.0.CO;2-S.
4. Griswold, M.A., Jakob, P.M., Heidemann, R.M., Nittka, M., Jellus, V., Wang, J., Kiefer, B., Haase, A.: Generalized autocalibrating partially parallel acquisitions (GRAPPA). Magn Reson Med. 47, 1202–1210 (2002). https://doi.org/10.1002/mrm.10171.
5. Lustig, M., Donoho, D.L., Santos, J.M., Pauly, J.M.: Compressed Sensing MRI. IEEE Signal Processing Magazine. 25, 72–82 (2008). https://doi.org/10.1109/MSP.2007.914728.
6. Gottwald, L.M., Peper, E.S., Zhang, Q., Coolen, B.F., Strijkers, G.J., Nederveen, A.J., van Ooij, P.: Pseudo-spiral sampling and compressed sensing reconstruction provides flexibility of temporal resolution in accelerated aortic 4D flow MRI: A comparison with k-t principal component analysis. NMR in Biomedicine. 33, e4255 (2020). https://doi.org/10.1002/nbm.4255.

7. Knoll, F., Hammernik, K., Zhang, C., Moeller, S., Pock, T., Sodickson, D.K., Akcakaya, M.: Deep-Learning Methods for Parallel Magnetic Resonance Imaging Reconstruction: A Survey of the Current Approaches, Trends, and Issues. IEEE Signal Processing Magazine. 37, 128–140 (2020). https://doi.org/10.1109/MSP.2019.2950640.
8. Vishnevskiy, V., Walheim, J., Kozerke, S.: Deep variational network for rapid 4D flow MRI reconstruction. Nat Mach Intell. 2, 228–235 (2020). https://doi.org/10.1038/s42256-020-0165-6.
9. Wang, P., Guo, P., Chai, K., Zhou, J., Xu, D., Jiang, S.: SDUM: A Scalable Deep Unrolled Model for Universal MRI Reconstruction, https://arxiv.org/abs/2512.17137v2, last accessed 2026/07/30.
10. Xin, B., Ye, M., Axel, L., Metaxas, D.N.: Rethinking Deep Unrolled Model for Accelerated MRI Reconstruction. In: Leonardis, A., Ricci, E., Roth, S., Russakovsky, O., Sattler, T., and Varol, G. (eds.) Computer Vision – ECCV 2024. pp. 164–181. Springer Nature Switzerland, Cham (2025). https://doi.org/10.1007/978-3-031-73226-3_10.
11. Cole, E., Cheng, J., Pauly, J., Vasanawala, S.: Analysis of deep complex-valued convolutional neural networks for MRI reconstruction and phase-focused applications. Magnetic Resonance in Medicine. 86, 1093–1109 (2021). https://doi.org/10.1002/mrm.28733.
12. Jacobs, L., Piccirelli, M., Vishnevskiy, V., Kozerke, S.: FlowMRI-Net: A generalizable self-supervised 4D flow MRI reconstruction network. Journal of Cardiovascular Magnetic Resonance. 27, 101913 (2025). https://doi.org/10.1016/j.jocmr.2025.101913.
13. Sun, A., Lu, H., Zhao, B.: Accelerated 4D Flow MRI with Low-Rank Modeling and A Deep Generative Prior. Presented at the 2024 ISMRM & ISMRT Annual Meeting, Toronto, ON, Canada. https://doi.org/10.58530/2024/4799.
14. Arvinte, M., Vishwanath, S., Tewfik, A.H., Tamir, J.I.: Deep J-Sense: Accelerated MRI Reconstruction via Unrolled Alternating Optimization. In: de Bruijne, M., Cattin, P.C., Cotin, S., Padoy, N., Speidel, S., Zheng, Y., and Essert, C. (eds.) Medical Image Computing and Computer Assisted Intervention – MICCAI 2021. pp. 350–360. Springer International Publishing, Cham (2021). https://doi.org/10.1007/978-3-030-87231-1_34.
15. Zhang, S., Joseph, A.A., Voit, D., Schaetz, S., Merboldt, K.-D., Unterberg-Buchwald, C., Hennemuth, A., Lotz, J., Frahm, J.: Real-time magnetic resonance imaging of cardiac function and flow—recent progress. Quant Imaging Med Surg. 4, 313–329 (2014). https://doi.org/10.3978/j.issn.2223-4292.2014.06.03.
16. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab, N., Hornegger, J., Wells, W.M., and Frangi, A.F. (eds.) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. pp. 234–241. Springer International Publishing, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28.
17. Arjovsky, M., Shah, A., Bengio, Y.: Unitary Evolution Recurrent Neural Networks. In: Proceedings of The 33rd International Conference on Machine Learning. pp. 1120–1128. PMLR (2016).

18. Perez, E., Strub, F., Vries, H. de, Dumoulin, V., Courville, A.: FiLM: Visual Reasoning with a General Conditioning Layer. Proceedings of the AAAI Conference on Artificial Intelligence. 32, (2018). https://doi.org/10.1609/aaai.v32i1.11671.
19. Wiest-Daesslé, N., Prima, S., Coupé, P., Morrissey, S.P., Barillot, C.: Rician Noise Removal by Non-Local Means Filtering for Low Signal-to-Noise Ratio MRI: Applications to DT-MRI. In: Metaxas, D., Axel, L., Fichtinger, G., and Székely, G. (eds.) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2008. pp. 171–179. Springer, Berlin, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85990-1_21.
20. Yaman, B., Hosseini, S.A.H., Akcakaya, M.: Zero-Shot Self-Supervised Learning for MRI Reconstruction. . In: *10th International Conference on Learning Representations (ICLR 2022)*. (2022).