The R2D2 Deep Neural Network Series for Scalable Non-Cartesian Magnetic Resonance Imaging

Yiwei Chen[1], Amir Aghabiglou[1], Shijie Chen[1], Motahare Torki[1], Chao Tang[1,2], Ruud B. van Heeswijk[3], Yves Wiaux[1] *
[1]Institute of Sensors, Signals and Systems, Heriot-Watt University Edinburgh, United Kingdom
[2]EPCC, University of Edinburgh, United Kingdom
[3]Department of Diagnostic Imaging and Interventional Radiology, Lausanne University and University Hospital, Switzerland
Email: *y.wiaux@hw.ac.uk

arXiv:2503.09559v2 [eess.IV] 13 Mar 2025

***Abstract*—We introduce the R2D2 Deep Neural Network (DNN) series paradigm for fast and scalable image reconstruction from highly-accelerated non-Cartesian k-space acquisitions in Magnetic Resonance Imaging (MRI). While unrolled DNN architectures provide a robust image formation approach via data-consistency layers, embedding non-uniform fast Fourier transform operators in a DNN can become impractical to train at large scale, *e.g.* in 2D MRI with a large number of coils, or for higher-dimensional imaging. Plug-and-play approaches that alternate a learned denoiser blind to the measurement setting with a data-consistency step are not affected by this limitation but their highly iterative nature implies slow reconstruction. To address this scalability challenge, we leverage the R2D2 paradigm that was recently introduced to enable ultra-fast reconstruction for large-scale Fourier imaging in radio astronomy. R2D2's reconstruction is formed as a series of residual images iteratively estimated as outputs of DNN modules taking the previous iteration's data residual as input. The method can be interpreted as a learned version of the Matching Pursuit algorithm. A series of R2D2 DNN modules were sequentially trained in a supervised manner on the fastMRI dataset and validated for 2D multi-coil MRI in simulation and on real data, targeting highly undersampled radial k-space sampling. Results suggest that a series with only few DNNs achieves superior reconstruction quality over its unrolled incarnation R2D2-Net (whose training is also much less scalable), and over the state-of-the-art diffusion-based "Decomposed Diffusion Sampler" approach (also characterised by a slower reconstruction process).**

***Index Terms*—non-Cartesian MRI, scalability, image reconstruction, deep learning.**

I. INTRODUCTION

COMPUTATIONAL imaging holds substantial scientific attention due to its broad spectrum of applications in disciplines such as astronomy [1] and medicine [2]. Among these, MRI stands out as a representative technique, enabling high-precision reconstruction of anatomical structures and organs, thereby serving an indispensable tool in clinical diagnostics [3]. By leveraging the interaction between magnetic fields and radio waves, MRI acquires signals in the spatial frequency domain, known as k-space, using multiple receivers (coils) focused on different regions. To accelerate the data acquisition process, k-space measurements are usually undersampled [4], which makes the inverse problem highly ill-posed. Optimization-based reconstruction algorithms enable injecting handcrafted regularization into the data, such as sparsity [5] and smoothness [6]. However, their highly iterative nature is computationally demanding, lacking scalability to the sheer data volumes from a large number of acquisition coils or leading-edge imaging techniques [7].

Recent advances in MRI methods based on Deep Neural Networks (DNNs) have shown significant advantages in recovering undersampled data from Cartesian sampling owing to their great modelling capabilities, including end-to-end DNNs. Pure DNNs [8], [9] learn mappings from the measurements in the k-space [10] or the back-projection of undersampled k-space data in the image domain [9], or a hybrid approach combining both [11]. However, they lack explicit incorporation of the underlying physics of the MRI acquisition process and fail to enforce data consistency to reconstruction, which limits their performance. To solve this, unrolled DNNs, such as PD-Net [12], mimic optimization algorithm iterations within their architecture, layer by layer, to enforce data consistency. Plug-and-Play (PnP) approaches [13] lie at the intersection of optimization theory and deep learning, integrating a learned denoiser agnostic to the specific measurement setting with a data-consistency step. They offer versatility and general applicability across diverse measurement setups, but their inherently highly-iterative nature also poses scalability challenges, particularly in large-scale scenarios, such as 2D imaging with numerous receive coils, or higher-dimensional imaging [7].

Reconstruction of undersampled non-Cartesian data is even more challenging, with scalability issues due to large-scale measurement operators based on the Non-Uniform Fast Fourier Transform (NUFFT). DNN models have also proven to be effective in non-Cartesian scenarios [14]. Here, pure DNNs are able to provide fast reconstruction from the back-projection image, but also lose robustness due to the lack of data consistency. For unrolled DNNs like NC-PDNet [14], as the measurement operator imposes significant computational and memory demands in training, they are impractical at large scale. Decomposed Diffusion Sampling (DDS) [15] has emerged as a scalable solution for non-Cartesian MRI. It combines diffusion models and advanced sampling strategies to enhance computation efficiency. In DDS, the denoisers are trained to remove noise and refine the intermediate outcomes, addressing the scalability challenge during the training phase. However, its highly iterative nature results in slow reconstruction. These scalability challenges, whether in training or reconstruction, highlight the pressing need for developing more efficient non-Cartesian MRI algorithms.

Recently, the "**R**esidual-to-**R**esidual **D**NN series for high-**D**ynamic range imaging" (R2D2) approach, a learned variant of the Matching Pursuit algorithm [16], has demonstrated a breakthrough in balancing reconstruction quality and computational efficiency in radio-interferometric imaging [17], [18] and a proof of concept in single-coil, real-valued non-Cartesian MRI [19]. R2D2 performs reconstruction as a sequence of residual image estimations. Each residual image is iteratively predicted by a DNN, which takes the previous image estimate and the associated back-protected data residual as input. This strategy avoids embedding the measurement operator directly into the network architecture, reducing computational overhead with respect to unrolled approaches. By externalizing data-consistency calculations, R2D2 significantly alleviates the computational burden associated with large-scale operators in training. Moreover, its number of iterations is significantly lower than the typical iteration count in PnP algorithms. R2D2 thus holds great potential for large-dimensional MRI, offering a promising direction for efficient and accurate imaging.

In this paper, we introduce R2D2 to 2D multi-coil non-Cartesian MRI. R2D2's flexibility allows for straightforward enhancement by updating the architecture of DNN modules. We propose employing the U-WDSR network [20] within R2D2 to achieve robust and accurate imaging. For benchmarking purposes, R2D2 is unrolled into an end-to-end network, named R2D2-Net, which incorporates NUFFT into the network structure, albeit with certain scalability limitations. We conducted an extensive evaluation of R2D2's performance on simulated datasets and real datasets, both using radial k-space sampling approaches. The results demonstrate that R2D2 outperforms State-of-the-Art (SOTA) methods in both reconstruction quality and efficiency.

II. PRELIMINARIES

A. Non-Cartesian MRI

Non-Cartesian MRI acquisitions offer advantages for advanced imaging applications [21]. Contemporary MRI scanners utilize multiple receiver coils positioned around a region of interest to acquire multiple MR signals from that region. Each coil possesses a distinct sensitivity map that determines its effectiveness at various imaging locations. In the non-Cartesian context, the measurement process reads as

$$\boldsymbol{y}_\ell = \boldsymbol{\Phi}_\ell \boldsymbol{x}^\star + \boldsymbol{n}_\ell \forall \ell = 1, \ldots, L \tag{1}$$

where $\boldsymbol{x}^\star \in \mathbb{C}^N$ denotes the 2D complex-valued unknown MR image of interest, $\boldsymbol{y}_\ell$ and $\boldsymbol{n}_\ell$ are the k-space measurement and Gaussian random noise with mean zero and standard deviation τ_ℓ of the ℓ-th coil, respectively. $\boldsymbol{\Phi}_\ell \triangleq \mathbf{F}_\Omega \mathbf{S}_\ell$ is the sampling operator based on the Non-Uniform Discrete Fourier Transform (NUDFT) $\mathbf{F}_\Omega$, with the k-space sampling location denoted by Ω and the diagonal matrix $\mathbf{S}_\ell \in \mathbb{C}^{N\times N}$ representing the ℓ-th coil sensitivity map. In practice, the approximation of $\mathbf{F}_\Omega$ is achieved via the NUFFT [22]. In the single-coil acquisition, $\mathbf{S}_1$ is the identity matrix $\mathbf{I}$. Generally, the coil sensitivity maps are assumed to satisfy the constraint

$$\sum_{\ell=1}^{L} \mathbf{S}_\ell^\dagger \mathbf{S}_\ell = \mathbf{I}, \tag{2}$$

which amounts to absorbing any deviation from this constraint into $\boldsymbol{x}^\star$. The back-projection of the ℓ-th coil data into the image domain reads as

$$\boldsymbol{x}_{\mathrm{b},\ell} = \mathbf{S}_\ell^\dagger \mathbf{F}_\Omega^\dagger \boldsymbol{y}_\ell \triangleq \boldsymbol{\Phi}_\ell^\dagger \boldsymbol{y}_\ell, \tag{3}$$

where $(\cdot)^\dagger$ is the adjoint operator. Combining coil images as

$$\boldsymbol{x}_{\mathrm{b}} = \kappa \sum_{\ell=1}^{L} \boldsymbol{x}_{\mathrm{b},\ell}, \tag{4}$$

we derive the normalized back-projected image $\boldsymbol{x}_{\mathrm{b}}$. In this context, $\kappa = 1/\max(\| \sum_{\ell=1}^{L} \boldsymbol{\Phi}_{\boldsymbol{\ell}}^\dagger \boldsymbol{\Phi}_{\boldsymbol{\ell}} \boldsymbol{\delta} \|_1)$ with $\| \cdot \|_1$ as the ℓ_1 norm, serves as the normalization constant. $\boldsymbol{\delta}$ represents the Dirac image, where the central element is $\sqrt{2}(1+i)/2$, with i as the imaginary unit, and all remaining elements are 0.

B. Density compensation

In contrast to Fast Fourier Transform (FFT), the NUFFT is not invertible. Thus, the application of the adjoint of NUFFT to k-space data frequently results in outcomes that diverge considerably from the sought image. For instance, in radial sampling, the densely sampled regions proximal to the centre of k-space tend to amass disproportionately large intensities, culminating in a back-projected image exhibiting unnaturally high values. To address this challenge, Density Compensation (DC), as introduced by [23], determines weighting factors in k-space that balance the contributions from various sample locations. This balance is achieved by iteratively applying both the interpolation matrix and its adjoint over iterations. After pre-calculating and storing the DC weights as the elements of the diagonal matrix $\mathbf{D}$, multiplying the k-space data by $\mathbf{D}$ before back-projection facilitates the generation of a more balanced and accurate back-projected image as

$$\boldsymbol{x}_{\mathrm{b}} = \kappa \sum_{\ell=1}^{L} \boldsymbol{\Phi}_\ell^\dagger \mathbf{D} \boldsymbol{y}_\ell. \tag{5}$$

C. Sensitivity estimation

Sensitivity estimation is designed to offer precise estimations of coil sensitivity maps for attaining high-quality MR images. In contrast to traditional methods like SENSE [24] and GRAPPAs [25], ESPIRiT [26] derives the sensitivity maps directly from the acquired k-space data, operating with minimal or no explicit calibration scans. This feature is especially advantageous for non-Cartesian sampling, where acquiring a separate calibration scan may be impractical.

In particular, ESPIRiT identifies eigenvectors with the largest eigenvalues, corresponding to the dominant coil sensitivity patterns. The estimated sensitivity maps are smooth and spatially consistent, reducing noise and artefacts compared to simple image-based sensitivity estimation, *e.g.* back-projected images from each coil divided by a combined image (a root-sum-of-squares). In this work, the sensitivities for both training and reconstruction are estimated by ESPIRiT.

D. SOTA reconstruction approaches

Optimization approaches have been extensively studied for image reconstruction from k-space data in MRI [27]. In this context, the image $\widehat{\boldsymbol{x}}$ is recovered by solving an optimization problem that balances data consistency with regularization: $\arg\min_{\widehat{\boldsymbol{x}}\in\mathbb{C}^N}\sum_{\ell=1}^{L}\frac{1}{2}\|\boldsymbol{y}_\ell-\boldsymbol{\Phi}_\ell\widehat{\boldsymbol{x}}\|_2^2+\mathbf{R}(\widehat{\boldsymbol{x}})$, where $\|\cdot\|_2$ is the ℓ_2 norm and $\mathbf{R}(\cdot)$ is a regularization term that acts as a prior. For example, $\mathbf{R}(\cdot)$ can be modelled as the ℓ_1 norm of the decomposition of $\boldsymbol{x}$ in an appropriate basis [28], or of its gradient magnitude [29]. These priors must be carefully designed and tailored to each specific inverse problem, requiring significant domain expertise and effort. Identifying an optimal configuration is not only time-consuming but also inherently challenging. At the intersection of optimization theory and deep learning, PnP algorithms [13] offer a compelling alternative. Instead of relying on handcrafted regularization operators, they train a denoising DNN to serve as a flexible and data-driven prior, which is then integrated into an iterative optimization algorithm, replacing the handcrafted regularization term. However, optimization and PnP algorithms often require numerous iterations to converge. For non-Cartesian k-space sampling, each iteration will involve NUFFT operations, making the process resource-demanding and limiting scalability, especially for large-scale applications.

To address these limitations, pure DNNs such as U-Net [30] have been proposed to enable direct image-to-image mapping for recovering images from back-projected images. They offer ultrafast reconstruction, but sacrifice robustness, including interpretability and generalizability, as they fail to explicitly enforce data consistency. Recently, unrolled networks, inspired by iterative optimization algorithms, enable the model to learn data consistency from the undersampled data and measurement operators. A notable example for non-Cartesian imaging is NC-PDNet [14], which unrolls the primal-dual algorithm and incorporates DC into its framework. However, in multi-coil MRI scenarios, where computationally intensive NUFFT are involved, high computational cost and substantial memory requirements present significant challenges during training. A strong competitor here is DDS [15], which eliminates the need to incorporate NUFFT into the training phase, requiring only the training of a denoising DNN for the diffusion process. To address large-scale inverse problems, DDS employs an efficient diffusion sampling strategy based on Krylov subspace methods, easing the computational burden of the reconstruction process. Nevertheless, it still involves hundreds of denoising steps that rely on NUFFT, which continues to hinder scalability in reconstruction.

III. R2D2 ALGORITHM

A. Algorithmic structure

The R2D2 algorithm require a series of I DNN modules represented as $\{\mathbf{G}_{\widehat{\boldsymbol{\theta}}^{(i)}}\}_{1\leq i\leq I}$, each with its own set of learnable parameters $\{\widehat{\boldsymbol{\theta}}^{(i)}\}_{1\leq i\leq I}$. Each DNN takes two input images: the previous image estimate $\boldsymbol{x}^{(i-1)}$ and its back-projected data residual

$$\boldsymbol{r}^{(i-1)}=\boldsymbol{x}_{\mathrm{b}}-\kappa\mathbf{P}\boldsymbol{x}^{(i-1)},\tag{6}$$

with $\mathbf{P}\triangleq\sum_{\ell=1}^{L}\boldsymbol{\Phi}_\ell^\dagger\mathbf{D}\boldsymbol{\Phi}_\ell$, which measures data consistency. The current image estimate is updated based on the output of the corresponding DNN module as:

$$\boldsymbol{x}^{(i)}=\boldsymbol{x}^{(i-1)}+\mathbf{G}_{\widehat{\boldsymbol{\theta}}^{(i)}}(\boldsymbol{r}^{(i-1)},\boldsymbol{x}^{(i-1)}).\tag{7}$$

The image estimate and back-projected data residual are initialized as $\boldsymbol{x}^{(0)}=\mathbf{0}$ and $\boldsymbol{r}^{(0)}=\boldsymbol{x}_{\mathrm{b}}$. This iterative structure allows for the progressive improvement of the resolution and structure details. An illustration of the R2D2 algorithm is shown in Figure 1, demonstrating how the learned residuals evolve over iterations, with early iterations capturing smooth structures and later iterations refining finer details and faint structures. The final reconstruction $\widehat{\boldsymbol{x}}$ can be expressed as

$$\widehat{\boldsymbol{x}}\triangleq\boldsymbol{x}^{(I)}=\sum_{i=1}^{I}\mathbf{G}_{\widehat{\boldsymbol{\theta}}^{(i)}}(\boldsymbol{r}^{(i-1)},\boldsymbol{x}^{(i-1)}).\tag{8}$$

R2D2 features a hybrid structure that combines aspects of learned matching pursuit. Unlike PnP algorithms and DDS, the DNN modules in R2D2 are designed to learn high-level features, rather than being trained solely as denoisers. In contrast to unrolled DNN modules [31], [14], R2D2 externalizes the back-projected data residual calculation from the network structure, which helps to alleviate the computational burden of NUFFT during training and improves scalability.

B. DNN series training

To minimize discrepancies between the estimated and Ground-Truth (GT) images, the networks are trained sequentially. For the i-th network, the loss function is defined as:

$$\min_{\widehat{\boldsymbol{\theta}}^{(i)}}\frac{1}{K}\sum_{k=1}^{K}\|\boldsymbol{x}_k^\star-\boldsymbol{x}_k^{(i-1)}+\mathbf{G}_{\widehat{\boldsymbol{\theta}}^{(i)}}(\boldsymbol{r}_k^{(i-1)},\boldsymbol{x}_k^{(i-1)})\|_1,\tag{9}$$

where K is the number of training samples. The output of each trained DNN is employed to iteratively update the image estimates, as specified in (7), and the corresponding data residual as per (6). These updated image pairs are subsequently included in the training dataset for the next DNN in the sequence. The learnable parameters of the first network are randomly initialized following a Gaussian distribution, and the parameters of the i-th network ($i\geq 2$) are initialized based on the parameter estimates from the previously trained network.

C. Number of DNN modules and series convergence

Formally, the number of DNN modules I is an algorithmic parameter, and an appropriate procedure is required to determine its optimal value. In [20], a convergence criterion is introduced whereby the reconstruction process stops when the data residual is compatible with noise rather than simply using all available DNN modules. This not only increases the reconstruction efficiency by reducing its computational cost and effectively renders R2D2 parameter-free, but also refines

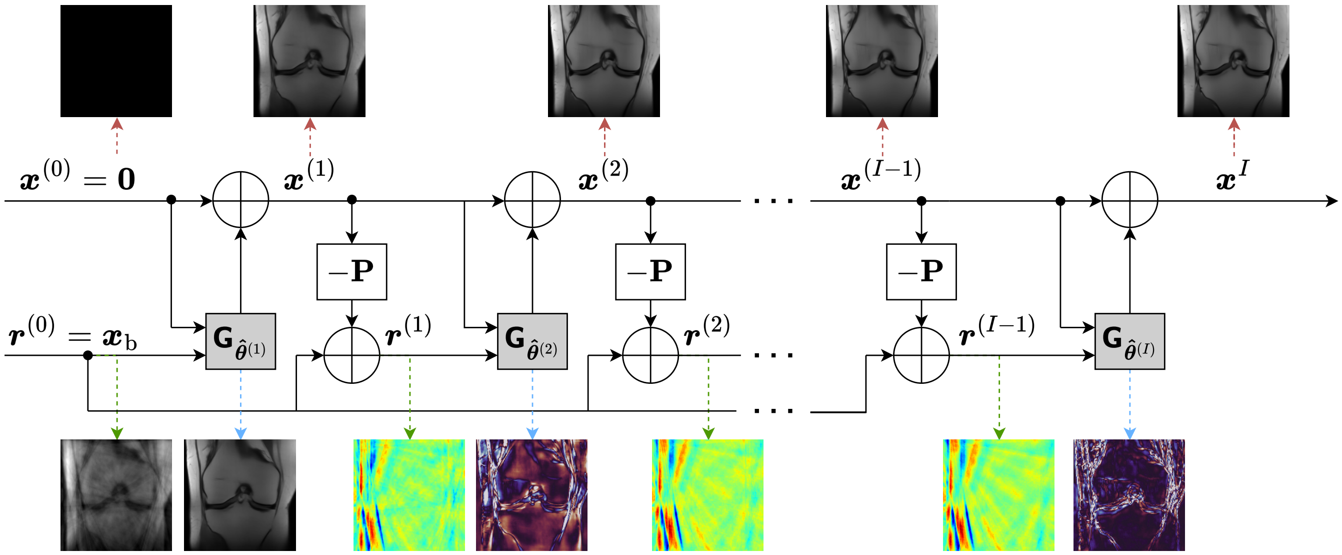

$\boldsymbol{x}^{(0)} = \mathbf{0}$
$\boldsymbol{x}^{(1)}$
$\boldsymbol{x}^{(2)}$
$\boldsymbol{x}^{(I-1)}$
$\boldsymbol{x}^{I}$
$-\mathbf{P}$
$-\mathbf{P}$
$-\mathbf{P}$
$\boldsymbol{r}^{(0)} = \boldsymbol{x}_{\mathrm{b}}$
$\mathbf{G}_{\widehat{\boldsymbol{\theta}}^{(1)}}$
$\boldsymbol{r}^{(1)}$
$\mathbf{G}_{\widehat{\boldsymbol{\theta}}^{(2)}}$
$\boldsymbol{r}^{(2)}$
$\boldsymbol{r}^{(I-1)}$
$\mathbf{G}_{\widehat{\boldsymbol{\theta}}^{(I)}}$

Fig. 1. Illustration of the R2D2 algorithm. The image iterates $\boldsymbol{x}_{i-1}$ and corresponding back-projected data residuals $\boldsymbol{r}_{i-1}$ are provided as inputs to R2D2 DNN modules. The outputs of the R2D2 DNN modules are used to update the subsequent image iterates. The progression of image iterates and back-projected data residuals is represented by dashed red and green arrows respectively, while the sequence of learned residual images is represented by dashed blue arrows.

training by pruning out the image pairs for which optimal data consistency is reached before training the next DNN.

This evolution is not implemented here. Instead, the sequential training process is assumed to be terminated when the evaluation metrics for the reconstruction quality computed on a validation dataset reach stabilization. The I DNN modules trained are then available for reconstruction.

D. Normalization

To mitigate generalization issues caused by discrepancies in pixel value ranges between the training dataset and the target images (*e.g.* test images), normalization techniques must be applied. We implement iteration-specific normalization during both the training and DNN inference phases. On the one hand, during the training of the DNN module at any iteration $i > 1$, the input dataset is normalized by dividing each image triplet $(\boldsymbol{x}_k^\star, \boldsymbol{x}_k^{(i-1)}, \boldsymbol{r}_k^{(i-1)})$ by $\alpha_k^{(i-1)}$, which is the mean value of the previous image magnitude estimate $\boldsymbol{x}_k^{(i-1)}$. The first DNN module is also trained using a normalized dataset, where each image pair $(\boldsymbol{x}_k^\star, \boldsymbol{x}_{\mathrm{b},k})$ is divided by $\alpha_k^{(0)}$, the mean pixel value of the back-projected image $\boldsymbol{x}_{\mathrm{b},k}$. On the other hand, at the reconstruction stage (see (7) and (8)), all DNN modules are applied to normalized inputs, with their outputs denormalized accordingly. Specifically, $\mathbf{G} \mapsto \alpha\mathbf{G}(./\alpha)$, where $\alpha \in \mathbb{R}_0^+$ is an iteration-specific normalization factor derived using the same procedure as in the training phase.

E. R2D2-Net

Formally, an unrolled incarnation of R2D2, named R2D2-Net [18], [19], can be developed for benchmarking purposes by unrolling the iterative R2D2 approach itself, with a predetermined number of internal iterations I. It transforms the iterative algorithm into an end-to-end DNN, where each iteration is a learnable subnetwork within the DNN. A notable drawback is the high computational demand during training due to the inclusion of NUFFT operations like NC-PDNet, making it less scalable and efficient compared to R2D2.

1) Forward process: Each subnetwork $\mathbf{G}_{\widehat{\boldsymbol{\theta}}^{(i)}}$ with $1 \leq i \leq I$ takes two images: the previous output $\boldsymbol{x}^{(i-1)}$ and its associated back-projected data residual $\boldsymbol{r}^{(i-1)}$ as the inputs. The final reconstruction $\widehat{\boldsymbol{x}}$ is the same as R2D2 in (8).

2) Training: In R2D2-Net, subnetworks function as reconstruction modules and are trained simultaneously rather than sequentially in R2D2. All learnable subnetworks are thus optimized jointly with respect to the following loss function:

$$\min_{\widehat{\boldsymbol{\theta}}^{(1)},\ldots,\widehat{\boldsymbol{\theta}}^{(I)}} \frac{1}{K}\sum_{k=1}^{K}\|\boldsymbol{x}_k^\star - \sum_{i=1}^{I}(\mathbf{G}_{\widehat{\boldsymbol{\theta}}^{(i)}}(\boldsymbol{r}_k^{(i-1)}, \boldsymbol{x}_k^{(i-1)}))\|_1. \qquad (10)$$

3) Normalization: At the training stage, R2D2-Net is trained using a normalized dataset, where each image pair $(\boldsymbol{x}_k^\star, \boldsymbol{x}_{\mathrm{b},k})$ is divided by $\alpha_k^{(0)}$. Within the subnetworks, the intermediate variables $\boldsymbol{x}_k^{(i)}$ and the back-projected data residuals $\boldsymbol{r}_k^{(i)}$ with $1 < i < I$ remain unnormalized. Instead, we rely on end-to-end training to learn the appropriate internal normalization. At the inference stage, R2D2-Net is applied to normalized inputs, with its outputs denormalized accordingly.

IV. TRAINING DETAILS

A. Network architectures

1) U-Net: Following from R2D2 developments in radio astronomy, the first DNN architecture utilized within R2D2 is U-Net [8], which comprises both compression and expansion sections. It employs 4 pooling layers using 2D average-pooling (stride 2) for downsampling, while upsampling is achieved via 2D transposed convolutions. The final output is generated through a 1×1 convolution layer, producing a single-channel output. To preserve detailed information from earlier layers, skip connections are utilized to bind relevant layers in the compression and expansion phases. For convolutions, we prefer real-valued over complex-valued ones, in line with findings that real-valued convolutions have a higher capacity

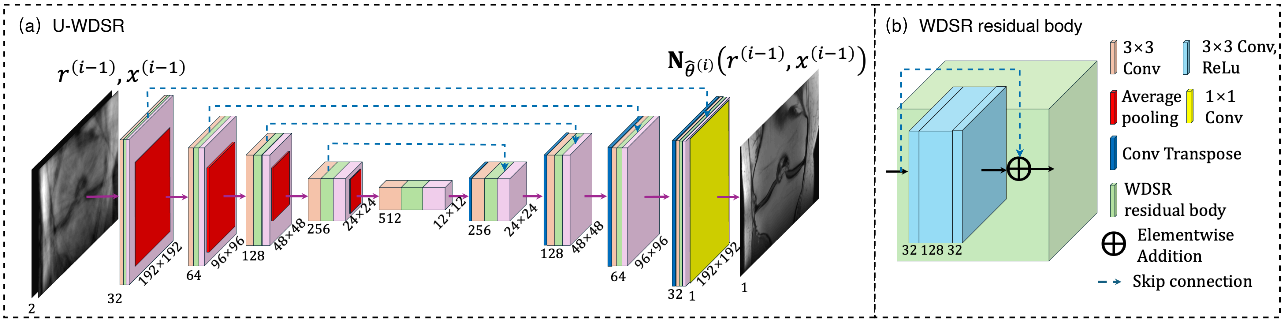


Fig. 2. Illustration of the U-WDSR model. (a) illustrates the U-WDSR architecture, and (b) depicts its WDSR layer. The WDSR residual body (in green boxes) is interlaced with the convolutional layers of the U-Net. WDSR consists of 16 consecutive residual blocks. At each stage, the spatial size of feature maps is indicated at the lower centre of each box. The number of channels is indicated at the outer edge of each box.

dealing with complex values [14], achieved by separating real and imaginary parts into distinct channels.

2) U-WDSR: U-WDSR is an advanced DNN architecture proposed in [20] as an alternative integration for the R2D2 algorithm. It combines the WDSR residual body architecture, originally designed for image and video super-resolution by [32], while retaining key structural components of U-Net, including the contracting and expanding paths and skip connections, to enhance performance. As illustrated in Figure 2, the U-WDSR architecture integrates the WDSR residual body as blocks interleaved with conventional convolutional layers of matching channel widths. Extended to 16 blocks, the WDSR residual body retains its original design from [32], incorporating features such as wide activation, dense connections, and low-rank convolution for enhanced accuracy and efficiency. ReLU activation is applied exclusively within the wide features of the residual blocks, maximizing information flow while ensuring non-linearity without excessive computational cost.

We note that a minor variation from [20] was implemented, consisting in deactivating weight initialization in the WDSR residual body to facilitate convergence.

B. GT dataset

A crucial aspect of training and validating our methods is a large dataset of high-quality GT images that encompass a diverse range of features and Dynamic Ranges (DRs), while remaining free from noise and artifacts. Therefore, we utilize complex-valued images back-projected from fully sampled MR data in the FastMRI knee dataset [30]. To enhance image quality and control noise levels, the raw images are preprocessed using a denoising DNN [33], applied separately to the real and imaginary components. Notably, the images with an original size of 320×320 are resized into 192×192 to match the resolution of the real data targeted by Section VI-A. The images are normalized to have intensity values ranging from 0 to 1. An example of the denoised GT images is shown in Figure 3(a). After preprocessing, the images are split into training and validation GT datasets, comprising 24915 and 4233 images, respectively.

The DR of the GT image is defined as the ratio between the maximum and faintest feature intensities. Since the GT images are normalized, the DR is the reciprocal of the faintest intensity, denoted by σ. For robust training, we aim to control the DR distribution of the dataset. The notion of faintest intensity is not well defined and the nominal DR of an image can be defined by fixing σ to a small percentile of the intensity distribution. Fixing this percentile to 6% yields a nominal DR distribution across the dataset roughly spanning from 1 to 10^3 and following a Gaussian-like distribution. While seemingly arbitrary in the definition of the GT images, this nominal σ value will be key to set the noise level on the k-space measurements when creating the back-projected image associated with each image of the training and validation datasets (see Section IV-C2).

C. Data generation details

1) Sampling: 2D Radial sampling typically begins at the periphery of k-space, traverses through the centre, and extends to the opposite end. Let (r, θ) represent the polar coordinates in the k-space. A radial spoke can be expressed as (r, θ) with $r \in [-\pi, \pi]$, $\theta \geq 0$, where r is the radial distance from the k-space centre and θ is the angular coordinate (angle from a reference axis). The polar coordinates can be converted to Cartesian coordinates (k_x, k_y) by

$$k_x = r\cos(\theta), \quad k_y = r\sin(\theta). \tag{11}$$

Let n_p denote the index of sampling points on a spoke, and $N_\mathrm{p} \in \mathbb{Z}_+$ represents the total number of sampling points per spoke. The number of spokes is denoted by $N_\mathrm{s} \in \mathbb{Z}_+$, and n_s represents the index of each spoke. For the n_s-th radial spoke, the radial distance of the n_p-th sampling point is defined as

$$r^{(n_\mathrm{s}, n_\mathrm{p})} = n_\mathrm{p}\Delta r - \pi, \quad n_p = 0, 1, 2, \ldots, N_p - 1, \tag{12}$$

where the interval is given by $\Delta r \triangleq 2\pi/(N_\mathrm{p} - 1)$. Typically, $N_\mathrm{p} = \sqrt{N}$ for a square MR image $\boldsymbol{x}^\star$ with $\sqrt{N} \times \sqrt{N}$ pixels in radial MRI. Thus, the coordinates of the sampling points $(k_x^{(n_\mathrm{s}, n_\mathrm{p})}, k_y^{(n_\mathrm{s}, n_\mathrm{p})})$ on the n_s-th radial spoke are

$$\begin{aligned} k_x^{(n_\mathrm{s}, n_\mathrm{p})} &= r^{(n_\mathrm{s}, n_\mathrm{p})}\cos(\theta_{n_\mathrm{s}}), \\ k_y^{(n_\mathrm{s}, n_\mathrm{p})} &= r^{(n_\mathrm{s}, n_\mathrm{p})}\sin(\theta_{n_\mathrm{s}}), \end{aligned} \tag{13}$$

where the angular position is given by $\theta_{n_\mathrm{s}} = n_\mathrm{s}\theta_\mathrm{g}$. Here, $\theta_\mathrm{g} = 68.25°$ corresponds to the small golden angle [21], which is employed as the angular increment to achieve an efficient and relatively uniform angular distribution of spokes.

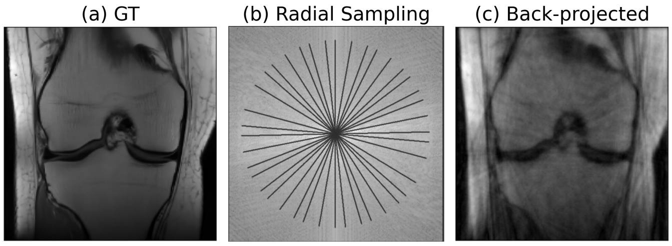

(a) GT (b) Radial Sampling (c) Back-projected

Fig. 3. Illustration of the non-Cartesian MRI problem. Panel (a) displays an example GT image. Panel (b) illustrates the radial sampling pattern in the spatial Fourier domain, superimposed to the Fourier transform of the GT image. Panel (c) shows the corresponding back-projected image.

The number of spokes determines the sparsity of k-space sampling. Fewer spokes lead to sparser sampling, which accelerates data acquisition but also introduces a more challenging inverse problem for image reconstruction. Following [14], the Acceleration Factor (AF) for 2D radial sampling is defined based on N_s as $\mathrm{AF} = \sqrt{N}/N_\mathrm{s}$. In previous studies, AF was set to be a constant, *e.g.* 4 or 8. To develop general models capable of adapting to a range of AFs, the number of spokes is treated as a variable during model training. A 2D sampling trajectory example is shown in Figure 3(b) with $\mathrm{AF} = 8$.

2) Training and validation datasets: We randomly select N_s in $\{10, \ldots, 79, 80\}$ that corresponds to AFs ranging from 19.2 to 2.6, and L in $\{8, \ldots, 31, 32\}$, to generate trajectories in k-space. Then the k-space measurements are generated using these radial trajectories and GT images based on (1) with random Gaussian noise in k-space. The noise level is set so that the target DR of reconstruction equates the DR of the GT image, denoted σ^{-1}. The DR of reconstruction is defined as the reciprocal of the noise level back-projected in the image domain. Following [34], we model the relationship between the additive noise in k-space and its back-projection into the image domain as: $\tau_\ell = \sigma\sqrt{2\mathrm{L}_\ell^2/\mathrm{L}_\ell^{'}}$, where L_ℓ and $\mathrm{L}_\ell^{'}$ are the spectral norm of the measurement operator of the ℓ-th coil when the DC weighting is applied once and twice, respectively. Readers can refer to [34] for technical details. For each GT image, based on a randomly selected radial sampling pattern, we generate one back-projected image (*e.g.* Figure 3 (c)), creating a pair of samples used for supervised learning. Thus, the total numbers of inverse problems for training and validation are 24915 and 4233, respectively.

V. VALIDATION ON SIMULATED DATA

A. Experimental setup

We randomly selected 50 GT images from the validation dataset. The testing followed the same simulation procedure as training and validation described in Section IV-C, with the number of spokes N_s in $\{12, 16, 24, 32, 48, 64\}$, corresponding to 6 AFs: $\{16, 12, 8, 6, 4, 3\}$, and the number of coils fixed to 16. For each GT image, one inverse problem is created with each of the 6 AFs, resulting in a total of 300 problems.

B. Benchmark algorithms

R2D2 performance is evaluated in comparison with the following benchmarking methods: (i) U-Net and U-WDSR – DNNs directly recover images from back-projected images without data consistency; (ii) NC-PDNet – unrolled DNN tailored for non-Cartesian MRI; (iii) DDS – diffusion model tailored for large-scale non-Cartesian MRI; (iv) R2D2(U-Net) and R2D2(U-WDSR) – R2D2 algorithms respectively utilizing U-Net and U-WDSR as their DNN architectures; (v) R2D2-Net(U-Net) and R2D2-Net(U-WDSR) – unrolled versions of R2D2(U-Net) and R2D2(U-WDSR).

C. Evaluation metrics

The Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) [35] are employed to evaluate imaging quality. The PSNR compares the maximum possible pixel value to the mean square error between the magnitude of the GT image $|\boldsymbol{x}^\star|$ and reconstructed image $|\widehat{\boldsymbol{x}}|$, with higher PSNR values signifying superior quality. The formula is

$$\mathrm{PSNR}(|\boldsymbol{x}^\star|, |\widehat{\boldsymbol{x}}|) = 10 \cdot \log_{10}\left(\frac{NM^2}{\||\boldsymbol{x}^\star| - |\widehat{\boldsymbol{x}}|\|_2^2}\right), \tag{14}$$

where M is the maximum possible pixel value of the GT image. The SSIM calculates image similarity as

$$\mathrm{SSIM}(|\boldsymbol{x}^\star|, |\widehat{\boldsymbol{x}}|) = \frac{(2\mu^\star\widehat{\mu} + c_1)(2\sigma_{\boldsymbol{c}} + c_2)}{((\mu^\star)^2 + \widehat{\mu}^2 + c_1)((\sigma^\star)^2 + \widehat{\sigma}^2 + c_2)}, \tag{15}$$

where $\mu^\star$, $\widehat{\mu}$ are the mean pixel value of $|\boldsymbol{x}^\star|$ and $|\widehat{\boldsymbol{x}}|$, respectively, $(\sigma^\star)^2$ and $\widehat{\sigma}^2$ are their variances, and $\sigma_{\boldsymbol{c}}$ is their covariance. c_1 and c_2 are stabilizing constants. SSIM values range from -1 to 1, with 1 indicating perfect similarity.

D. Implementation details

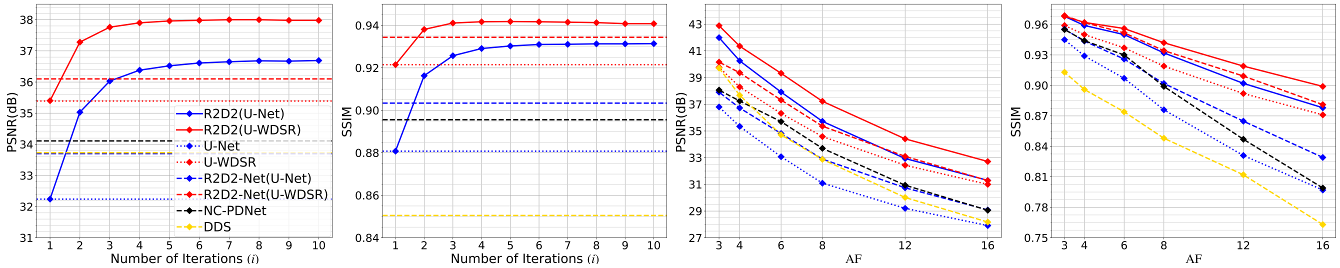
In DNN implementations, following [14], instead of using complex-valued convolutions, we employ vanilla convolutions within the network, where the real and imaginary parts of complex-valued images are treated as two separate channels.

The U-Net, implemented either individually or within the R2D2 framework, employs an initial convolution layer consisting of 64 channels. Besides, the U-WDSR, implemented either individually or within the R2D2 framework, employs an initial convolution layer consisting of 32 channels.

In the R2D2 implementation, sensitivity maps are derived using the ESPIRiT algorithm. Our tests indicate that the estimation of magnitude is more accurate than that of phase, resulting in a back-projected image with a more reliable magnitude than phase. Consequently, we calculate the data residual using the magnitude instead of the original residual defined in (6) to achieve the optimal residual achievable with ESPIRiT. The residual for the i-th ($i > 1$) iteration reads as

$$\boldsymbol{r}^{(i-1)} = |\boldsymbol{x}_\mathrm{b}| - |\kappa\mathbf{P}\boldsymbol{x}^{(i-1)}|, \tag{16}$$

where $|\cdot| \triangleq \sqrt{\Re(\cdot)^2 + \Im(\cdot)^2}$ with $\Re(\cdot)$ and $\Im(\cdot)$ represent the real and imaginary components, respectively. For the first DNN module, $\boldsymbol{r}^{(0)} = \boldsymbol{x}_\mathrm{b}$ and $\boldsymbol{x}^{(0)} = \mathbf{0}$. Therefore, the first DNN module in R2D2 takes a 3-channel input containing a zero real-valued image and the complex-valued back-projected image, while the following DNN modules take a 3-channel input containing the previously reconstructed complex-valued image and the back-projected data residual magnitude.



(a) PSNR vs. iterations (b) SSIM vs. iterations (c) PSNR vs. AF (d) SSIM vs. AF

Fig. 4. Quantitative reconstruction results from simulated experiments. The PSNR and SSIM of R2D2(U-Net) and R2D2(U-WDSR) (solid lines) are reported against the number of R2D2 iterations (panels (a) and (b); reported values are averages over 300 inverse problems across AFs), and the AF value (panels (c) and (d); reported values are averages over 50 inverse problems). Benchmark methods are represented via horizontal lines, with U-Net and U-WDSR represented by dotted lines, and R2D2-Net(U-Net), R2D2-Net(U-WDSR), NC-PDNet, and DDS represented by dashed lines.

With regards to the number of DNN modules, the sequential training process was terminated at $I = 10$ for both R2D2(U-Net) and R2D2(U-WDSR), two iterations beyond the point where the average PSNR and SSIM reach stabilization over the validation dataset. This provides a larger number of DNNs than needed *on average*, with the number of iterations $i \leq I$ needed for reconstruction estimated *a posteriori* in each experiment for the specific inverse problem(s) under scrutiny.

In the R2D2-Net implementations, we use 64 channels for the initial convolution layer of U-Net in R2D2-Net(U-Net), while limiting it to 16 channels for that of U-WDSR in R2D2-Net(U-WDSR) due to GPU memory limitations. The network takes a 4-channel input containing a zero complex-valued image and the complex-valued back-projected image. Each subnetwork takes a 4-channel input containing the complex-valued image and the complex-valued back-projected residual. The numbers of subnetworks for R2D2(U-Net) and R2D2(U-WDSR) are set to 5 and 3, respectively. This choice of parameters was made to optimise performance.

In the NC-PDNet implementation, the buffer strategy inherent in the original paper [14] is retained with a buffer size of 5, and a 3-layer CNN with a number of convolution filters as 32 is employed as the architecture for both the primal and dual subnetworks. The number of iterations is set to 10.

For DDS, we use 100 sampling steps with the stochasticity parameter as 0.85 following [36]. The number of conjugate gradient steps is 5, and the regularization parameter of DDS is fine-tuned through grid search to optimize the reconstruction quality, with respect to the noise level of the target MR image.

All algorithms were executed on a single NVIDIA Tesla V100-SXM2-16GB GPU in the UK national Tier-2 high-performance computing system Cirrus, using the deep learning library PyTorch [37]. The batch sizes for the training of the R2D2 modules, NC-PDNet, and R2D2-Nets were 4, 4, and 1, respectively. The above models were trained for 100 epochs using the Adam optimizer [38] with a learning rate of 0.0001. For DDS, we followed [39] and trained the model for 1 million iterations with the AdamW optimizer [40] and a learning rate of 0.0001. The NUFFT implementation is TorchKbNufft [41].

E. Quantitative results

Figure 4 (a-b) compares the PSNR and SSIM values of R2D2(U-Net) and R2D2(U-WDSR) against the number of iterations, in comparison to benchmark algorithms. All values are averages over the 300 test inverse problems. R2D2(U-Net) can be estimated to converge at iteration $i = 8$, when both average PSNR and SSIM values saturate, while R2D2(U-WDSR) shows higher efficiency, converging at iteration $i = 6$. Final PSNR and SSIM values are also reported with standard deviations in Table I. Overall, both R2D2 models deliver significant PSNR and SSIM improvements compared to their respective DNNs, U-Net and U-WDSR. R2D2(U-Net) falls short of R2D2(U-WDSR), indicating that the DNN structure significantly impacts the overall performance. R2D2-Net(U-Net) and R2D2-Net(U-WDSR) provide disappointing performance, with PSNR and SSIM stabilising between those of the first and second iterations of R2D2. NC-PDNet and DDS provide significantly lower quality than R2D2(U-Net), R2D2(U-WDSR), and R2D2-Net(U-WDSR).

Figure 4 (c-d) compares performance across the AFs. All values are averages over the 50 test inverse problems for each AF. A general decline in PSNR and SSIM is observed for all methods as AF increases. R2D2(U-WDSR) achieves the highest values across all AFs, demonstrating exceptional robustness. It is followed by R2D2(U-Net), R2D2-Net(U-WDSR), and U-WDSR. U-Net, R2D2-Net(U-Net), DDS, and NC-PDNet exhibit a steadier decline as AF increases.

Based on Figure 4 (c), Table II provides a comparison of the extra acceleration ratios ($\mathrm{AF}_{\mathrm{R2D2}}/\mathrm{AF}_{\mathrm{Algo}}$), where $\mathrm{AF}_{\mathrm{R2D2}}$ is the AF of R2D2(U-WDSR) and $\mathrm{AF}_{\mathrm{Algo}}$ is the AF of another algorithm at the same imaging quality with respect to PSNR. R2D2(U-WDSR) demonstrates extra acceleration ratios higher than 1 in comparison to all other algorithms, from U-Net (2.52-2.81), to U-WDSR (1.45-1.8), R2D2(U-Net) (1.24-1.27), NC-PDNet (1.67-1.71), DDS (1.88-2.06), R2D2-Net(U-Net) (1.86-2.05), and R2D2-Net(U-WDSR) (1.25-1.45).

F. Visual performance

Figures 5 and 6 present the visual results of various comparison methods for $\mathrm{AF} = 16$ and $\mathrm{AF} = 4$, respectively. This enables evaluating the changes in reconstruction quality with increasing AFs. $\mathrm{AF} = 4$ results also provide a per-method reference for the $\mathrm{AF} = 16$ results and the reader is invited to flip between Figures 5 and 6 for a comparison at a glance. In each figure, the first and fourth rows display the reconstructed images (with values of PSNR (dB) reported in

TABLE I

Quantitative reconstruction results, reconstruction times, and scalability indicators. The first column lists the algorithm names. The second and third columns report the PSNR and SSIM values (with standard deviations) averaged over 300 test inverse problems across AF values; these values match those in Figure 4(a-b), with the metric computed at iteration $i = 8$ for R2D2(U-Net) and iteration $i = 6$ for R2D2(U-WDSR). The best and second-best results are highlighted in bold and underlined, respectively. Columns four to seven detail the reconstruction times, broken down into model loading ($t_{\text{loa.}}$), inference ($t_{\text{inf.}}$) and residual calculation ($t_{\text{res.}}$); the total time ($t_{\text{tot.}}$) is given in the seventh column. The eighth column lists the number of parameters for each DNN architecture. Finally, scalability indicators categorize the algorithms with regards to their training scalability (column nine) or reconstruction scalability (column ten).

Algorithm	PSNR (dB)	SSIM	$t_{\text{loa.}}$(s)	$t_{\text{inf.}}$(s)	$t_{\text{res.}}$(s)	$t_{\text{tot.}}$(s)	Par. (M)	Tra. Scal.	Rec. Scal.
U-Net	32.23 ± 4.58	0.88 ± 0.08	0.08 ± 0.02	0.004 ± 0.006	–	0.09 ± 0.02	31.1	✓	✓
U-WDSR	35.39 ± 4.69	0.92 ± 0.06	0.08 ± 0.02	0.029 ± 0.015	–	0.10 ± 0.03	20.1	✓	✓
NC-PDNet	34.11 ± 5.01	0.90 ± 0.08	0.03 ± 0.01	0.284 ± 0.012	–	0.32 ± 0.01	0.16	✗	✓
DDS	33.73 ± 6.80	0.85 ± 0.13	0.67 ± 0.27	82.32 ± 6.43	–	82.39 ± 6.62	83.7	✓	✗
R2D2-Net(U-Net)	33.70 ± 4.57	0.90 ± 0.07	0.08 ± 0.02	0.195 ± 0.010	–	0.27 ± 0.02	155.5	✗	✓
R2D2-Net(U-WDSR)	36.10 ± 4.90	$\underline{0.93 \pm 0.06}$	0.08 ± 0.02	0.235 ± 0.018	–	0.32 ± 0.02	40.3	✗	✓
R2D2(U-Net)	$\underline{36.51 \pm 5.74}$	$\underline{0.93 \pm 0.06}$	0.29 ± 0.04	0.023 ± 0.006	0.51 ± 0.05	0.82 ± 0.09	248.8	✓	✓
R2D2(U-WDSR)	$\mathbf{37.90 \pm 5.67}$	$\mathbf{0.94 \pm 0.05}$	0.29 ± 0.03	0.184 ± 0.006	0.55 ± 0.08	1.08 ± 0.23	120.6	✓	✓

TABLE II

Extra acceleration ratios for R2D2 (U-WDSR) compared to benchmark algorithms in terms of PSNR.

Algorithm\PSNR (dB)	33	36	39
U-Net	2.52	2.81	-
U-WDSR	1.45	1.55	1.8
R2D2(U-Net)	1.25	1.27	1.24
NC-PDNet	1.67	1.71	-
DDS	1.97	1.88	2.06
R2D2-Net(U-Net)	1.86	2.05	-
R2D2-Net(U-WDSR)	1.25	1.33	1.45

the bottom left of each panel), while the second and fifth rows show the corresponding Regions Of Interests (ROIs). Dotted ellipses highlight areas with subtle differences between the reconstructed, GT, and back-projected images. The third and sixth rows provide error maps relative to the GT image. The images displayed and reported PSNR values are at iteration $i = 8$ for R2D2(U-Net) and $i = 6$ for R2D2(U-WDSR).

We firstly focus on $\mathrm{AF} = 16$. The back-projected image shows significant blurring, noticeable artifacts, and substantial errors. Methods like U-Net, DDS, NC-PDNet, and R2D2-Net(U-Net) exhibit improvements in structural detail over the back-projected image but still suffer from notable reconstruction errors. The U-WDSR architecture provides a noticeable jump in visual quality over U-Net, similar to R2D2(U-Net) and R2D2-Net(U-WDSR). R2D2(U-WDSR) provides the best performance, effectively preserving fine structural details and minimizing reconstruction errors. These results are in line with the quantitative analysis (see Section V-E).

All algorithms exhibit significant improvements at $\mathrm{AF} = 4$, but the general conclusions drawn at $\mathrm{AF} = 16$ remain. We note that DDS produces a noisy output, with a PSNR not superior to that of the back-projected image. Interestingly, R2D2(U-WDSR) exhibits the highest stability across accelerations, further confirming the results of the quantitative analysis.

G. Scalability

Table I presents the breakdown of reconstruction times across methods. Reported values are averages and standard deviations over the 300 test inverse problems across AFs. DDS exhibits the longest image reconstruction time, its highly iterative nature (100 steps) limiting reconstruction scalability. Pure DNNs demonstrate the lowest image reconstruction time but at the cost of limited performance. The R2D2 methods achieve high reconstruction quality while maintaining fast reconstruction thanks to their low iteration count, *i.e.* 8 iterations for R2D2(U-Net) against 6 for R2D2(U-WDSR. We highlight that the reported model loading times ($t_{\text{loa.}}$), inference times ($t_{\text{inf.}}$), residual calculation times ($t_{\text{res.}}$), and the total time ($t_{\text{tot.}}$) are the cumulative values across all iterations. Notably, the image reconstruction process of the R2D2 methods spends a significant amount of time on model loading, which can be reduced by processing multiple test samples in parallel.

Beyond image reconstruction scalability, training scalability is also a key consideration, with training time serving as a crucial metric. Here, training time refers to the total duration from the start to the completion of training. For the unrolled network, this represents the total time required to train the end-to-end model, while for R2D2, it includes the cumulative time for training all DNN modules and computing all the data residuals. Figure 7 illustrates the relationship between the number of coils and the training time (in hours) for three methods: R2D2(U-Net), NC-PDNet, and R2D2-Net(U-Net). As the number of coils increases, the training time for all models grows linearly. However, the rate of increase differs significantly among the models. The R2D2(U-Net) model exhibits the most efficient performance, achieving the lowest training times across all coil numbers and displaying a much shallower slope in the linear trend. In contrast, both NC-PDNet and R2D2-Net(U-Net) show steeper slopes, with training times increasing more rapidly as the number of coils grows. The efficiency of R2D2 stems from its ability to bypass back-propagation through the measurement operator, significantly reducing computational overhead. We note that the same conclusion holds for U-WDSR-based models.

For ease of comparison, training and reconstruction scalability indicators are reported for all methods in Table I. Only U-Net, U-WDSR, R2D2(U-Net), and R2D2(U-WDSR) demonstrate both training and image reconstruction scalability.

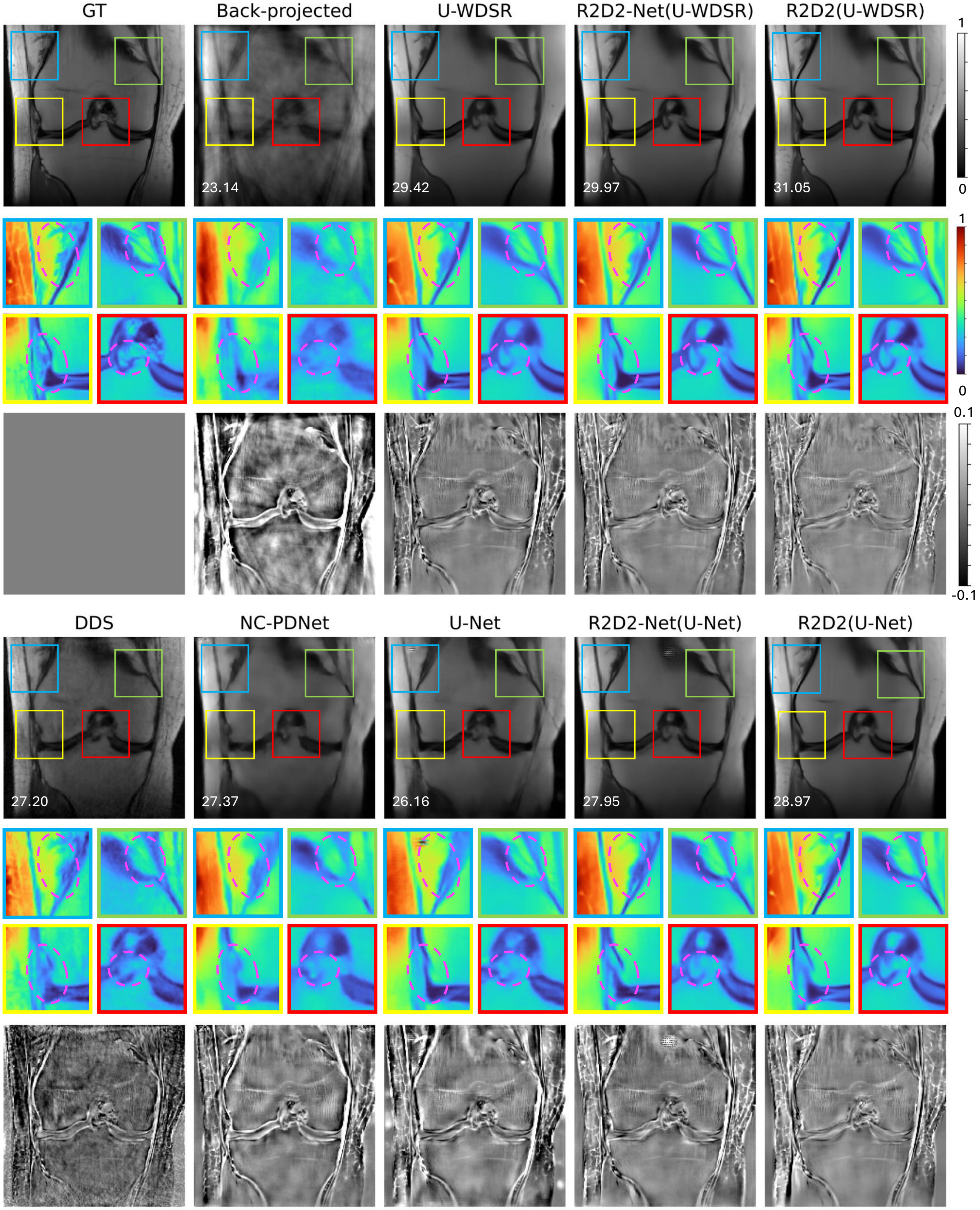

GT
Back-projected
U-WDSR
R2D2-Net(U-WDSR)
R2D2(U-WDSR)
23.14
29.42
29.97
31.05
1
0
1
0
0.1
-0.1
DDS
NC-PDNet
U-Net
R2D2-Net(U-Net)
R2D2(U-Net)
27.20
27.37
26.16
27.95
28.97

Fig. 5. Visual reconstruction results for one of test inverse problems of the simulated experiments, with $\mathrm{AF} = 16$ and 16 coils (flip with Figure 6 for $\mathrm{AF} = 4$ comparison at a glance). The first and fourth rows show the GT image and estimated images of different methods. The second and fifth rows show the selected ROI marked by boxes in the images. Dotted ellipses in these ROIs highlight key areas with subtle differences between the estimated images. The third and sixth rows present the error images between the GT and estimated images. Values of PSNR (dB) are reported in the bottom left of the estimated images.

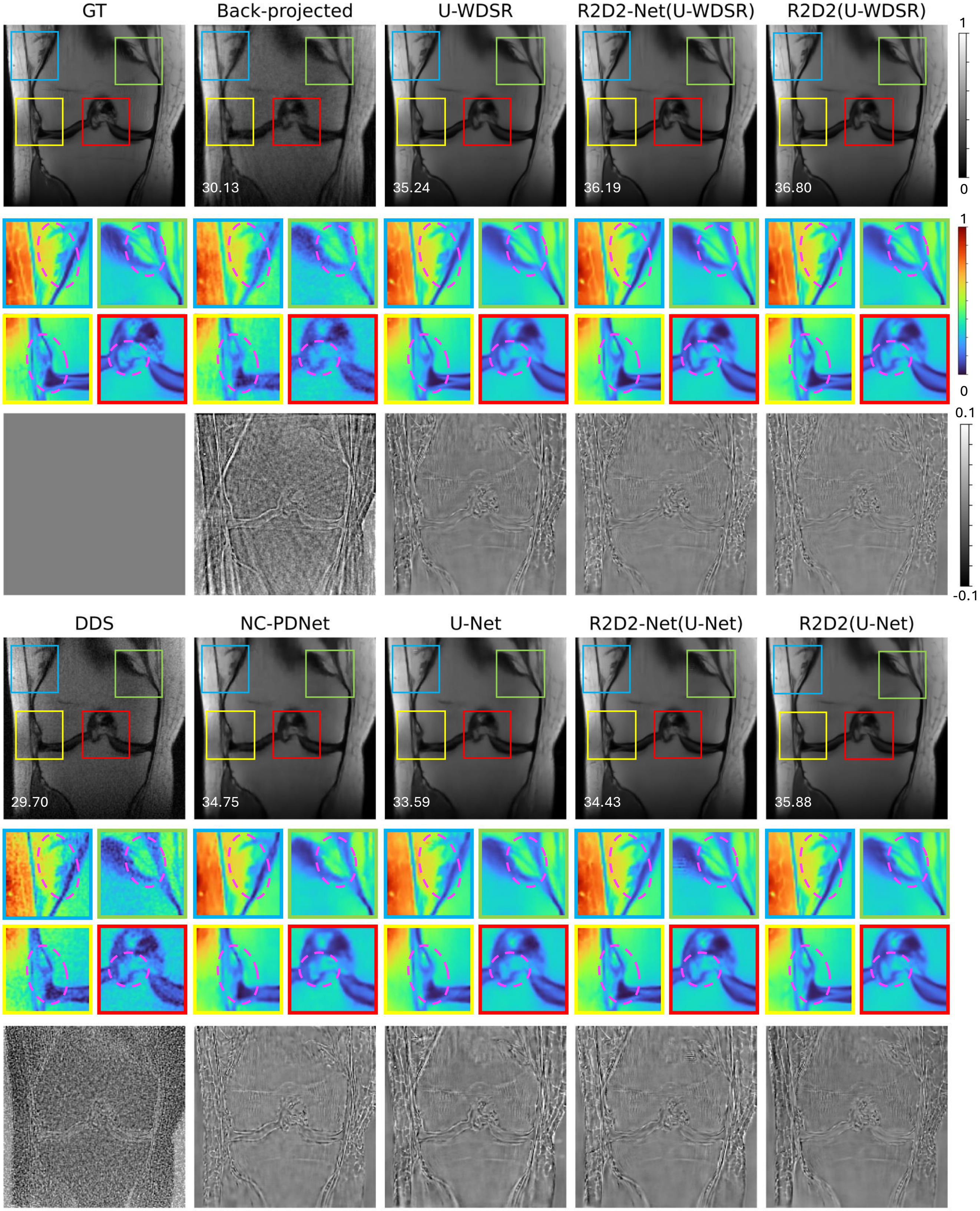

GT
Back-projected
U-WDSR
R2D2-Net(U-WDSR)
R2D2(U-WDSR)
30.13
35.24
36.19
36.80
1
0
1
0
0.1
-0.1
DDS
NC-PDNet
U-Net
R2D2-Net(U-Net)
R2D2(U-Net)
29.70
34.75
33.59
34.43
35.88

Fig. 6. Visual reconstruction results for one of test inverse problems of the simulated experiments, with $\mathrm{AF} = 4$ and 16 coils (flip with Figure 5 for $\mathrm{AF} = 16$ comparison at a glance). The first and fourth rows show the GT image and estimated images of different methods. The second and fifth rows show the selected ROI marked by boxes in the images. Dotted ellipses in these ROIs highlight key areas with subtle differences between the estimated images. The third and sixth rows present the error images between the GT and estimated images, with the same scaling as in Figure 5. Values of PSNR (dB) are reported in the bottom left of the estimated images.

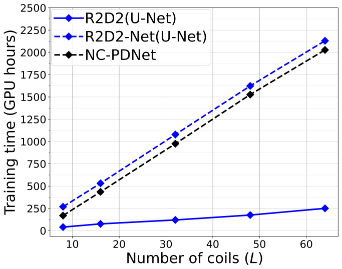
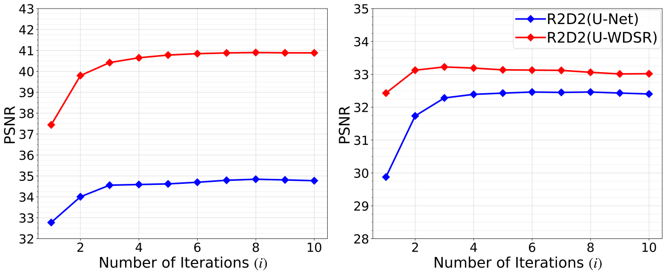


Fig. 7. Training scalability comparison. Training times are reported against the number of coils for R2D2(U-Net) (solid line), NC-PDNet and R2D2-Net(U-Net) (dashed lines).



(a) DR $= 100$ (b) DR $= 3.57$

Fig. 8. DR-specific quantitative reconstruction results from simulated experiments. The progression of the PSNR of R2D2(U-Net) and R2D2(U-WDSR) over iterations is reported for 1 of the 300 test inverse problems, with high DR and at AF $= 8$ (left), or low DR and at AF $= 8$ (right).

Among them, R2D2(U-WDSR) achieves the best performance.

H. *Impact of DR*

To gain deeper insight into the reconstruction performance in different DR regimes, we compared R2D2(U-Net) and R2D2(U-WDSR) on randomly selected test inverse problems with specific DR values. Figure 8 illustrates the PSNR values at AF $= 8$ as a function of the number of R2D2 iterations, both for a test inverse problem with DR= 3.57 (*i.e.* high noise) and another with DR=100 (*i.e.* low noise). The results reveal that the performance gap between R2D2(U-Net) and R2D2(U-WDSR) is smaller at low DR, as well as the improvement from U-WDSR to R2D2(U-WDSR). Additionally, the convergence is faster at low DR, requiring only 3 iterations for R2D2(U-WDSR) and 6 iterations for R2D2(U-Net). This observation corroborates the expectation that higher noise reduces the amount of useful information contained in the residual images.

VI. VALIDATION ON REAL DATA

A. *Experimental setup*

Data were acquired on the right knee of a healthy volunteer (F, 166cm, 56kg) in a 3T clinical scanner (PrismaFit, Siemens Healthineers, Erlangen, Germany) with a dedicated 15-element knee coil array. 2D golden-angle radial data were acquired with a balanced Steady-State Free Precession (bSSFP) pulse sequence with the following parameters: echo time TE = 1.58 ms, repetition time TR = 3.32 ms, flip angle 30-75°, orientation pseudo-coronal, slice thickness 9 mm, field of view $192 \times 192\ \mathrm{mm}^2$, 384 points per radial spoke of which the central 192 points were kept, and effective pixel size $1 \times 1\ \mathrm{mm}^2$. A total of 192 spokes were acquired per image and used for reference image reconstruction. All human studies were approved by the local ethics committee (CER-VD approval 2021-00697), and written informed consent was given by the participant.

B. *Visual and quantitative performance*

Figure 9 illustrates the back-projected images with AF $= 1, 4, 8$ and reconstructions with all R2D2 and benchmark methods at AF $= 4$ and AF $= 8$. For the reconstruction of R2D2 methods, we use the estimated images at iteration $i = 5$ of R2D2(U-Net) and iteration $i = 3$ of R2D2(U-WDSR) as no improvement results from using further iterations. Notably, while provided for reference, the back-projected image at AF $= 1$ is not a true GT image, since it is not properly reconstructed but only back-projected, affected by an imperfect calibration setup. In fact, it exhibits noticeable artefacts originating in the radial k-space sampling pattern. We also note that the DR for this sample is only ~ 8, which limits the achievable visual quality across reconstruction methods. In particular, at DR $= 8$, improvements of R2D2 across iterations are *a priori* expected to be subtle (see Section V-H).

Overall, NC-PDNet fails to reconstruct the knee image properly, both at AF $= 4$ and AF $= 8$. This is mainly due to the domain shift problem [42], where the U-Net-based calibration module (unique to NC-PDNet across all methods compared) struggles to generate accurate sensitivity maps for the real data. DDS shows noticeable blurring and loss of texture detail, with a particular degradation at AF $= 8$. R2D2-Net(U-WDSR) introduces strong "vertical-stripe" artefacts, and brightness issues at AF $= 8$. R2D2-Net(U-Net) mitigates these issues, but at the expense of an increased blurriness. U-Net provides blurred reconstructions and significant loss of structures, issues that R2D2(U-Net) strongly mitigates. While the U-WDSR architecture provides much sharper reconstructions than U-Net, without noticeable loss of structure, some "vertical-line" artefacts do appear, particularly at AF $= 8$. R2D2(U-WDSR) delivers the best reconstruction overall, in terms of sharpness, recovery of faint structure, absence of obvious artefacts, and stability across accelerations. It provides the reconstruction of the cartilage (blue arrows) and menisci area (red ellipses) that is most faithful to the back-projected image at AF $= 1$.

Finally, in terms of quantitative performance, we evaluate data fidelity via the Residual Data Ratio, defined as the ratio of back-projected residual to the back-projected image: RDR $= \|\boldsymbol{r}\|_2 / \|\boldsymbol{x}_{\mathrm{b}}\|_2$. RDR plots in Figure 10 reveal that R2D2(U-WDSR) achieves the best data fidelity. Additionally, the performance of R2D2(U-Net) improves until iteration $i = 5$, reducing to $i = 3$ and R2D2(U-WDSR). As expected at low DR, the performance gain is small, particularly for R2D2(U-WDSR) which starts from a lower value at $i = 1$.

VII. CONCLUSION

In this study, we introduced the novel R2D2 paradigm, which successfully leverages a series of DNN modules in a "Matching Pursuit" flavour, to address the image reconstruction and scalability issues in non-Cartesian MRI. Each network in the series iteratively refines the reconstruction by leveraging the back-projected data residual and the previous estimate to enhance image quality. Two variations of

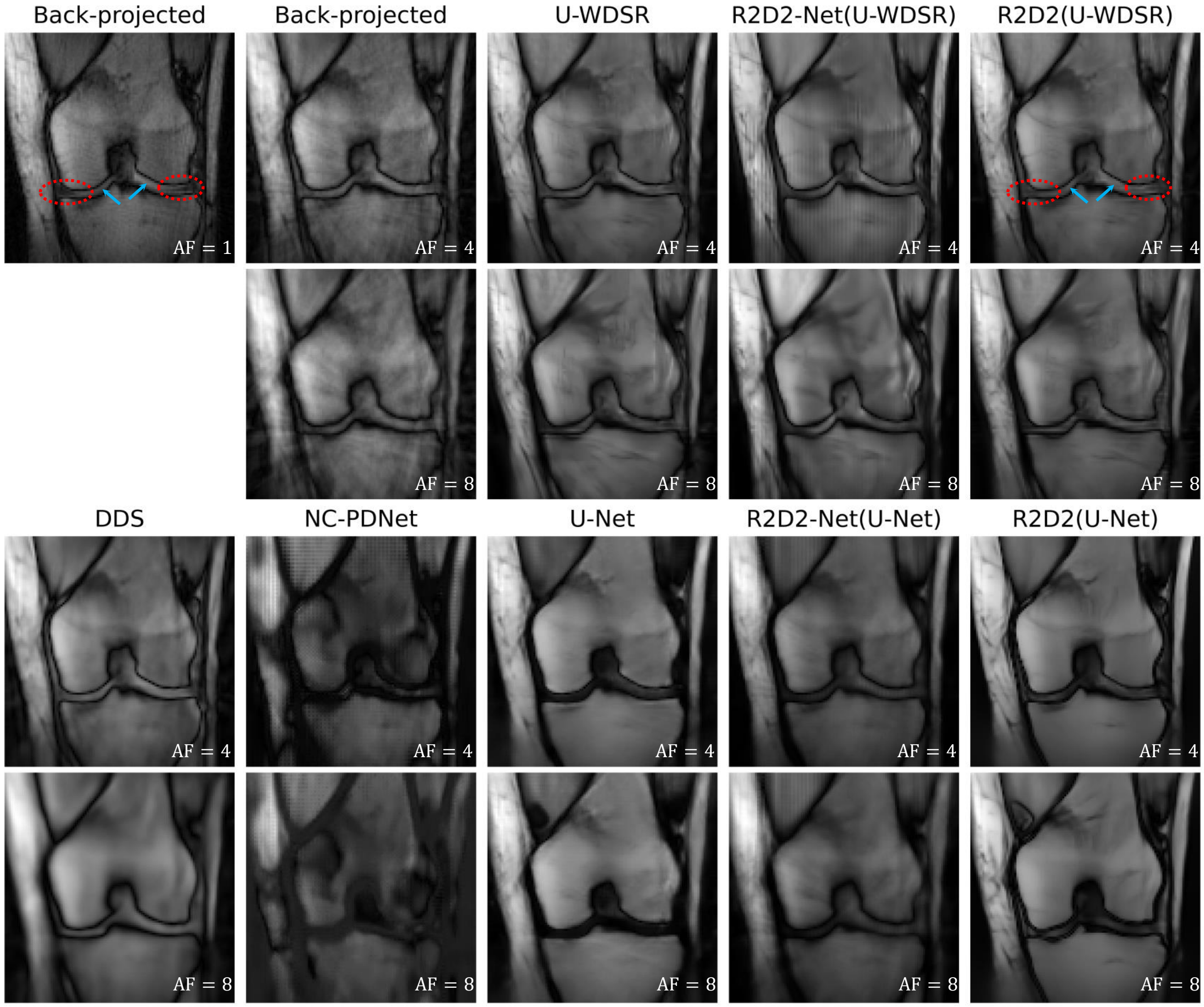
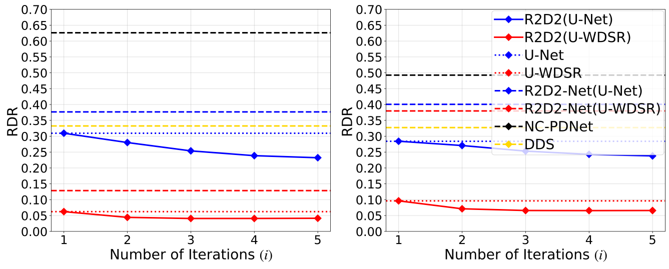


Fig. 9. Visual reconstruction results at $\mathrm{AF} = 4$ and $\mathrm{AF} = 8$ for the 15-coil real data experiment. The back-projected image with 192 sampling spokes (denoted as $\mathrm{AF} = 1$) is reported for reference on the top left panel. The following panels show the back-projected images as well as reconstructions with R2D2(U-Net) at iteration $i = 5$, R2D2(U-WDSR) at iteration $i = 3$, and benchmark algorithms, both at $\mathrm{AF} = 4$ (48 spokes) and $\mathrm{AF} = 8$ (24 spokes). Blue arrows and red ellipses in the back-projected image at $\mathrm{AF} = 1$ and reconstruction by R2D2(U-WDSR) at $\mathrm{AF} = 4$ indicate the cartilage and menisci, respectively. For improved visualization, the images are center-cropped to 60% of their original size.



(a) $\mathrm{AF} = 4$

(b) $\mathrm{AF} = 8$

Fig. 10. Quantitative evaluation of data fidelity for the 15-coil real data experiment: (a) $\mathrm{AF} = 4$; (b) $\mathrm{AF} = 8$. The RDR of R2D2(U-Net) and R2D2(U-WDSR) (solid lines) are reported against the number of R2D2 iterations. Benchmark methods are represented via horizontal lines, with U-Net and U-WDSR represented by dotted lines, and R2D2-Net(U-Net), R2D2-Net(U-WDSR), NC-PDNet, and DDS represented by dashed lines.

R2D2 were proposed based on distinct architectures for their DNN modules: R2D2(U-Net) and R2D2(U-WDSR). Their unrolled counterparts R2D2-Net(U-Net) and R2D2-Net(U-WDSR) were introduced as benchmarks.

Extensive simulations on complex-valued MR images with 2D multi-coil radial sampling sequences demonstrate that the R2D2 methods outperform SOTA methods across all AFs, including R2D2-Net, U-Net, U-WDSR, DDS and NC-PDNet, both quantitatively and visually. Moreover, R2D2 significantly reduces training time compared to unrolled networks like NC-PDNet and R2D2-Net, and it reconstructs images faster than iterative methods such as DDS, underscoring its scalability. Experiments on real data further confirm that R2D2 consistently delivers the best reconstructions, while benchmark algorithms show more noticeable errors.

To further explore R2D2's potential in non-Cartesian MRI, future research will include: (i) investigating more sophisticated DNN architectures as the core module in R2D2 to enhance reconstruction performance; (ii) developing efficient and accurate sensitivity estimation techniques to boost image quality; (iii) validating R2D2's practical utility for 3D and 4D

MRI, addressing scalability and computational challenges.

ACKNOWLEDGEMENTS

The authors thank JC Ye and HJ Chung for insightful discussion on DDS. YC, AA, MT, and YW were supported by UKRI (grants EP/T028270/1, ST/W000970/1, EP/S023291/1). RBvH was supported by the SNSF (grant CRSII5_202276). This work utilised the Cirrus HPC Service funded by the University of Edinburgh and UKRI (grant EP/P020267/1). The authors thank Adrian Jackson for related support.

DATA AND CODE AVAILABILITY

Codes will be made available as part of a future release of the BASPLib code library on GitHub. BASPLib is developed and maintained by the Biomedical and Astronomical Signal Processing Laboratory (BASP).

REFERENCES

[1] Y. Wiaux, L. Jacques, G. Puy, A. M. Scaife, and P. Vandergheynst, "Compressed sensing imaging techniques for radio interferometry," *Mon. Not. R. Astron. Soc.*, vol. 395, no. 3, pp. 1733–1742, 2009.

[2] G. H. Glover, "Overview of functional magnetic resonance imaging," *Neurosurg. Clin.*, vol. 22, no. 2, pp. 133–139, 2011.

[3] Z.-P. Liang and P. C. Lauterbur, *Principles of magnetic resonance imaging*. SPIE Optical Engineering Press Bellingham, 2000.

[4] X. Qu, D. Guo, B. Ning, Y. Hou, Y. Lin, S. Cai, and Z. Chen, "Undersampled MRI reconstruction with patch-based directional wavelets," *Magn. Reson. Imaging*, vol. 30, no. 7, pp. 964–977, 2012.

[5] M. Lustig, D. Donoho, and J. M. Pauly, "Sparse MRI: The application of compressed sensing for rapid MR imaging," *Magn. Reson. Med.*, vol. 58, no. 6, pp. 1182–1195, 2007.

[6] S. Poddar and M. Jacob, "Dynamic MRI using smoothness regularization on manifolds (SToRM)," *IEEE Trans. Med. Imaging*, vol. 35, no. 4, pp. 1106–1115, 2015.

[7] Z. Stankovic, B. D. Allen, J. Garcia, K. B. Jarvis, and M. Markl, "4D flow imaging with MRI," *Cardiovasc. Diagn. Ther.*, vol. 4, no. 2, p. 173, 2014.

[8] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. MICCAI*. Springer, Oct. 2015, pp. Part III 18, 234–241.

[9] C. M. Hyun, H. P. Kim, S. M. Lee, S. Lee, and J. K. Seo, "Deep learning for undersampled MRI reconstruction," *Phys. Med. Biol.*, vol. 63, no. 13, p. 135007, 2018.

[10] Y. Han, L. Sunwoo, and J. C. Ye, "k-space deep learning for accelerated MRI," *IEEE Trans. Med. Imaging*, vol. 39, no. 2, pp. 377–386, 2019.

[11] T. Eo, Y. Jun, T. Kim, J. Jang, H.-J. Lee, and D. Hwang, "KIKI-net: cross-domain convolutional neural networks for reconstructing undersampled magnetic resonance images," *Magn. Reson. Med.*, vol. 80, no. 5, pp. 2188–2201, 2018.

[12] J. Adler and O. Öktem, "Learned primal-dual reconstruction," *IEEE Trans. Med. Imaging*, vol. 37, no. 6, pp. 1322–1332, 2018.

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

[14] Z. Ramzi, G. Chaithya, J.-L. Starck, and P. Ciuciu, "NC-PDNet: A density-compensated unrolled network for 2D and 3D non-Cartesian MRI reconstruction," *IEEE Trans. Med. Imaging*, vol. 41, no. 7, pp. 1625–1638, 2022.

[15] H. Chung, S. Lee, and J. C. Ye, "Decomposed diffusion sampler for accelerating large-scale inverse problems," in *Proc. ICLR*, May 2024.

[16] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, 1993.

[17] A. Aghabiglou, M. Terris, A. Jackson, and Y. Wiaux, "Deep network series for large-scale high-dynamic range imaging," in *Proc. IEEE ICASSP*. IEEE, Jun. 2023, pp. 1–5.

[18] A. Aghabiglou, C. San Chu, A. Dabbech, and Y. Wiaux, "The R2D2 deep neural network series paradigm for fast precision imaging in radio astronomy," *Astrophys. J. Suppl. Ser.*, vol. 273, no. 1, p. 3, 2024.

[19] Y. Chen, C. Tang, A. Aghabiglou, C. S. Chu, and Y. Wiaux, "Scalable non-cartesian magnetic resonance imaging with R2D2," in *Proc. EUSIPCO*, Aug. 2024, pp. 1511–1515.

[20] A. Aghabiglou, C. S. Chu, C. Tang, A. Dabbech, and Y. Wiaux, "Towards a robust R2D2 paradigm for radio-interferometric imaging: revisiting DNN training and architecture," *arXiv preprint arXiv:2503.02554*, 2025.

[21] K. L. Wright, J. I. Hamilton, M. A. Griswold, V. Gulani, and N. Seiberlich, "Non-cartesian parallel imaging reconstruction," *J. Magn. Reson. Imaging*, vol. 40, no. 5, pp. 1022–1040, 2014.

[22] J. A. Fessler and B. P. Sutton, "Nonuniform fast Fourier transforms using min-max interpolation," *IEEE Trans. Signal Process.*, vol. 51, no. 2, pp. 560–574, 2003.

[23] J. G. Pipe and P. Menon, "Sampling density compensation in MRI: rationale and an iterative numerical solution," *Magn. Reson. Med.*, vol. 41, no. 1, pp. 179–186, 1999.

[24] K. P. Pruessmann, M. Weiger, M. B. Scheidegger, and P. Boesiger, "SENSE: sensitivity encoding for fast MRI," *Magn. Reson. Med.*, vol. 42, no. 5, pp. 952–962, 1999.

[25] M. A. Griswold, P. M. Jakob, R. M. Heidemann, M. Nittka, V. Jellus, J. Wang, B. Kiefer, and A. Haase, "Generalized autocalibrating partially parallel acquisitions (GRAPPA)," *Magn. Reson. Med.*, vol. 47, no. 6, pp. 1202–1210, 2002.

[26] M. Uecker, P. Lai, M. J. Murphy, P. Virtue, M. Elad, J. M. Pauly, S. S. Vasanawala, and M. Lustig, "ESPIRiT—an eigenvalue approach to autocalibrating parallel MRI: where SENSE meets GRAPPA," *Magn. Reson. Med.*, vol. 71, no. 3, pp. 990–1001, 2014.

[27] F. Knoll, K. Hammernik, C. Zhang, S. Moeller, T. Pock, D. K. Sodickson, and M. Akcakaya, "Deep-learning methods for parallel magnetic resonance imaging reconstruction: A survey of the current approaches, trends, and issues," *IEEE Signal Process. Mag.*, vol. 37, no. 1, pp. 128–140, 2020.

[28] M. Guerquin-Kern, M. Haberlin, K. P. Pruessmann, and M. Unser, "A fast wavelet-based reconstruction method for magnetic resonance imaging," *IEEE Trans. Med. Imaging*, vol. 30, no. 9, pp. 1649–1660, 2011.

[29] K. T. Block, M. Uecker, and J. Frahm, "Undersampled radial MRI with multiple coils. iterative image reconstruction using a total variation constraint," *Magn. Reson. Med.*, vol. 57, no. 6, pp. 1086–1098, 2007.

[30] J. Zbontar, F. Knoll, A. Sriram, T. Murrell, Z. Huang, M. J. Muckley, A. Defazio, R. Stern, P. Johnson, M. Bruno *et al.*, "fastMRI: An open dataset and benchmarks for accelerated MRI," *arXiv preprint arXiv:1811.08839*, 2018.

[31] A. Sriram, J. Zbontar, T. Murrell, A. Defazio, C. L. Zitnick, N. Yakubova, F. Knoll, and P. Johnson, "End-to-end variational networks for accelerated MRI reconstruction," in *Proc. MICCAI*. Springer, Oct. 2020, pp. Part II 23, 64–73.

[32] J. Yu, Y. Fan, J. Yang, N. Xu, Z. Wang, X. Wang, and T. Huang, "Wide activation for efficient and accurate image super-resolution," *arXiv preprint arXiv:1808.08718*, 2018.

[33] K. Zhang, Y. Li, J. Liang, J. Cao, Y. Zhang, H. Tang, D.-P. Fan, R. Timofte, and L. V. Gool, "Practical blind image denoising via Swin-Conv-UNet and data synthesis," *Mach. Intell. Res.*, vol. 20, no. 6, pp. 822–836, 2023.

[34] A. G. Wilber, A. Dabbech, A. Jackson, and Y. Wiaux, "Scalable precision wide-field imaging in radio interferometry: I. uSARA validated on ASKAP data," *Mon. Not. R. Astron. Soc.*, vol. 522, no. 4, pp. 5558–5575, 2023.

[35] A. Hore and D. Ziou, "Image quality metrics: PSNR vs. SSIM," in *Proc. ICPR*. IEEE, Aug. 2010, pp. 2366–2369.

[36] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *Proc. ICLR*, May 2021.

[37] A. Paszke, S. Gross, F. Massa, Lerer *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Proc. NIPS*, pp. 8024–8035, Dec. 2019.

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

[39] P. Dhariwal and A. Nichol, "Diffusion models beat GANs on image synthesis," *Proc. NIPS*, vol. 34, pp. 8780–8794, Dec. 2021.

[40] I. Loshchilov, F. Hutter *et al.*, "Fixing weight decay regularization in adam," *arXiv preprint arXiv:1711.05101*, vol. 5, 2017.

[41] M. J. Muckley, R. Stern, T. Murrell, and F. Knoll, "TorchKbNufft: A high-level, hardware-agnostic non-uniform fast Fourier transform," in *ISMRM Workshop*, Jan. 2020.

[42] K. Zhou, Z. Liu, Y. Qiao, T. Xiang, and C. C. Loy, "Domain generalization: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 4396–4415, 2022.