

Learned Tensor Low-CP-Rank and Bloch Response Manifold Priors for Non-Cartesian MRF Reconstruction

Peng Li, *Student Member, IEEE*, Yue Hu, *Member, IEEE*

Abstract— Magnetic resonance fingerprinting (MRF) can rapidly perform simultaneous imaging of multiple tissue parameters. However, the rapid acquisition schemes used in MRF inevitably introduce aliasing artifacts in the recovered tissue fingerprints, reducing the accuracy of the predicted parameter maps. Current regularized reconstruction methods are based on iterative procedures which are usually time-consuming. In addition, most of the current deep learning-based methods for MRF often lack interpretability owing to the black-box nature, and most deep learning-based methods are not applicable for non-Cartesian scenarios, which limits the practical applications. In this paper, we propose a joint reconstruction model incorporating MRF-physics prior and the data correlation constraint for non-Cartesian MRF reconstruction. To avoid time-consuming iterative procedures, we unroll the reconstruction model into a deep neural network. Specifically, we propose a learned CANDECOMP/PARAFAC (CP) decomposition module to exploit the tensor low-rank priors of high-dimensional MRF data, which avoids computationally burdensome singular value decomposition. Inspired by the MRF-physics, we also propose a Bloch response manifold module to learn the mapping between reconstructed MRF data and the multiple parameter maps. Numerical experiments show that the proposed network can reconstruct high-quality MRF data and multiple parameter maps within significantly reduced computational time.

Index Terms— MRF, Tensor Low-rank, CP Decomposition, Unrolled Networks, Bloch Response Manifold, non-Cartesian

I. INTRODUCTION

MAGNETIC resonance fingerprinting (MRF) is a quantitative MRI framework proposed by Ma *et al.* [1], which can rapidly perform simultaneous imaging of multiple tissue parameters such as spin-lattice relaxation time (T_1), and spin-spin relaxation time (T_2). Pseudo-randomized pulse sequences are used in MRF to acquire data, leading to unique signal evolution from different tissues, known as tissue fingerprints. The multiple quantitative tissue parameters are then obtained by mapping each tissue fingerprint to a precomputed dictionary, which contains theoretical fingerprints of a set of possible tissues. Rapid acquisition schemes are widely used in MRF to accelerate data acquisition, but they inevitably introduce

This work is supported by China NSFC 61871159 and Natural Science Foundation of Heilongjiang YQ2021F005.

Y. Hu and P. Li are with the School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin, China (e-mail: huyue@hit.edu.cn).

aliasing artifacts in the recovered tissue fingerprints [2]. Tissue fingerprints contaminated with aliasing artifacts and noise will subsequently reduce the estimation accuracy and precision of tissue parameters.

To improve the accuracy of the estimated parameter maps, many model-based reconstruction methods have been proposed to overcome undersampling artifacts. Davis *et al.* [3] proposed to apply Bloch response manifold projection in a compressed sensing framework to improve MRF reconstruction. Zhao *et al.* [4] proposed a maximum likelihood formalism to estimate multiple parameter maps directly from highly undersampled data. Mazor *et al.* [5] developed a subspace-constrained low-rank projection method (FLOR), based on the fact that the MRF signal can be sparsely represented in the generated dictionary domain. Yang *et al.* [6] proposed a randomized SVD-based method to estimate the low-dimensional dictionary subspace directly, which can effectively reduce both the computational time and memory demand required for the dictionary construction. Zhao *et al.* [7] proposed a constrained imaging method based on low-rank and subspace modeling to improve the accuracy and speed of MRF. Then, they [8] extended this method by introducing a low-rank tensor model, which can mitigate the information loss caused by the matrix preprocessing step in the low-rank reconstruction method. Bustin *et al.* [9] proposed a high-dimensionality undersampled patch-based method that enables efficient reconstruction of multi-channel multi-contrast MR images with high quality, even when acquired in a short time. Cruz *et al.* [10] developed a sparse and locally low-rank regularized reconstruction method, enabling shorter scan times or increased spatial resolution. In our previous work, we also proposed a structured low-rank matrix completion and subspace projection framework (SL-SP) [11] to recover MRF data from its highly undersampled measurements, resulting in improved reconstruction performance. Although the methods mentioned above have shown promising results in improving reconstruction quality, their computational performance can often be suboptimal due to the large number of iterations required. This issue is particularly pronounced in non-Cartesian scenarios that require computationally expensive non-uniform fast Fourier transform (NUFFT) operations, leading to longer reconstruction times and higher computational costs.

Recently, deep learning-based methods have been introduced in MRF to improve the speed and/or accuracy of parameter map estimation. Cohen *et al.* [12] proposed a 4-

layer fully connected neural network to perform signal-to-parameter mapping, replacing the memory-consuming dictionary and time-consuming dictionary matching. Oksuz et al. [13] proposed a recurrent neural network to perform MRF map reconstruction, exploiting the time-dependent information of tissue fingerprints. Fang *et al.* [14] proposed a two-step deep learning model (SCQ) to learn the mapping from the signals to the tissue parameters, enabling accurate parametric reconstructions under quadruple accelerated acquisition. Soyak *et al.* [15] proposed a neural network consisting of a channel-wise attention module and a fully convolutional network, and the strategy of overlapping patches for patch-level multi-parameter estimation was adopted to effectively reduce the error of parameter reconstruction. However, most of the existing approaches are less interpretable since they reconstruct parameter maps using existing network structures. In addition, most deep learning-based methods cannot directly reconstruct MRF data from non-Cartesian undersampled measurements. In recent years, deep learning-based image reconstruction using non-Cartesian k -space trajectories has received increasing research attention [16]–[18] due to its potential to improve image quality and reduce acquisition time. However, it should be noted that deep learning-based image reconstruction using Cartesian sampling patterns still predominates in research, particularly in MRF. Therefore, to fully explore the potential of non-Cartesian k -space trajectories in MRF, it is essential to conduct further studies to investigate their integration with deep learning-based image reconstruction methods.

More recently, an emerging technique called deep unrolling [19]–[21] has been introduced in MR image processing. Schlemper *et al.* [22] proposed a framework for reconstructing dynamic sequences of 2-D cardiac magnetic resonance images from undersampled data using a deep cascade of convolutional neural networks (CNNs) to accelerate the data acquisition process. Ke *et al.* [23] proposed a learned low-rank prior and introduced it into a model-based unrolling deep network architecture in dynamic MR imaging to obtain improved reconstruction results. Chen *et al.* [24] reconstructed data and parameter maps from non-Cartesian samples based on a deep unrolling network. Deep unrolling-based methods have the interpretability of model-based iterative methods while being able to exploit the powerful learning capabilities of deep neural networks. By providing a concrete and systematic connection between model-based algorithms and deep neural networks, deep unrolling-based methods are promising to build efficient and interpretable neural networks [19]. However, despite the potential advantages of deep unfolded networks for MRF data reconstruction, their applicability has been limited by the challenges associated with the high dimensionality of MRF data and the computational complexity of the resulting reconstruction problem.

Tensor low-rank priors [25] have been introduced in high-dimensional data reconstruction as powerful alternatives to traditional low-rank regularized image recovery methods, as tensors can better describe multi-linear latent structures beyond the pairwise interactions captured by matrices [26]. Particularly, the tensor low-rank priors [27]–[29] are constructed via tensor decomposition, including the CANDE-

COMP/PARAFAC (CP) [30], the Tucker decomposition [31], etc. The CP decomposition schemes decompose a high-dimensional tensor into a sum of component rank-one tensors constructed by the vector outer product, and the CP rank is defined as the minimum number of required component tensors. The Tucker decomposition decomposes a tensor into a product of a core tensor and a number of factor matrices, and the Tucker rank is a vector composed of the ranks of unfolding matrices of the target tensor along different dimensions. However, most tensor-based methods constrain tensor rank based on higher-order singular value decomposition (SVD) [27], [32], [33], which significantly increases the computational overhead under high-dimensional MRF data.

In this paper, we propose to exploit the learned low-CP-rank [25] and the Bloch response manifold [3] priors for non-Cartesian MRF reconstruction in an unrolling manner. Specifically, we propose a joint reconstruction model incorporating MRF-physics prior and the data correlation prior for non-Cartesian MRF reconstruction. To avoid time-consuming iterative procedures, we unroll the reconstruction model into a deep neural network to learn the tensor low-rank priors and the Bloch response manifold priors, termed TLR-BM-Net. To avoid computationally burdensome singular value decomposition, we propose a learned CP decomposition module to exploit the tensor low-rank priors of high-dimensional MRF data. Inspired by the MRF-physics, we also propose a Bloch response manifold module to learn the mapping between reconstructed MRF data and the multiple parameter maps. The proposed network can simultaneously reconstruct MRF data and multi-parameter maps from non-Cartesian undersampled measurements. Numerical experiments show that the proposed network can improve the reconstruction quality of MRF data and multiple parameter maps within significantly reduced computational time.

II. PROBLEM FORMULATION

The undersampled data acquisition of MRF images can be modeled as:

$$\mathbf{b} = \mathcal{A}\mathcal{X} + \mathbf{n} \quad (1)$$

where $\mathbf{b} \in \mathbb{C}^{N_s \times L \times N_c}$ is the acquired k -space measurements with the number of coils being N_s , N_c is the number of k -space samples collected by each coil in each frame, and L is the number of time frames. $\mathcal{X} \in \mathbb{C}^{N_x \times N_y \times L}$ denotes the MRF data, where N_x and N_y are the dimensions of each image frame. $\mathbf{n} \in \mathbb{C}^{N_s \times L \times N_c}$ is the noise matrix of all coils. $\mathcal{A} = \mathbf{F}_u \mathbf{C}$ denotes a linear operator that considers the coil sensitivities \mathbf{C} and the undersampled Fourier transform \mathbf{F}_u . According to the MRF imaging mechanism, the magnetization response at any voxel of \mathcal{X} can be written as a parametric nonlinear mapping as:

$$\mathcal{X}_{i,j,:} = \rho_{i,j} \mathbf{B}(\eta_{i,j}; \theta_L) \quad (2)$$

where $\rho_{i,j} \in \mathbb{R}_+$ represents the proton density (PD) of the corresponding voxel. $\eta_{i,j} = [\mathbf{T}_1, \mathbf{T}_2] \in \mathcal{M}$ represents a row vector composed of different parameters, and $\mathcal{M} \in \mathbb{R}^2$ denotes the set of feasible values for $\eta_{i,j}$. θ_L is the parameter vector of the excitation pulse with length L , including the repetition

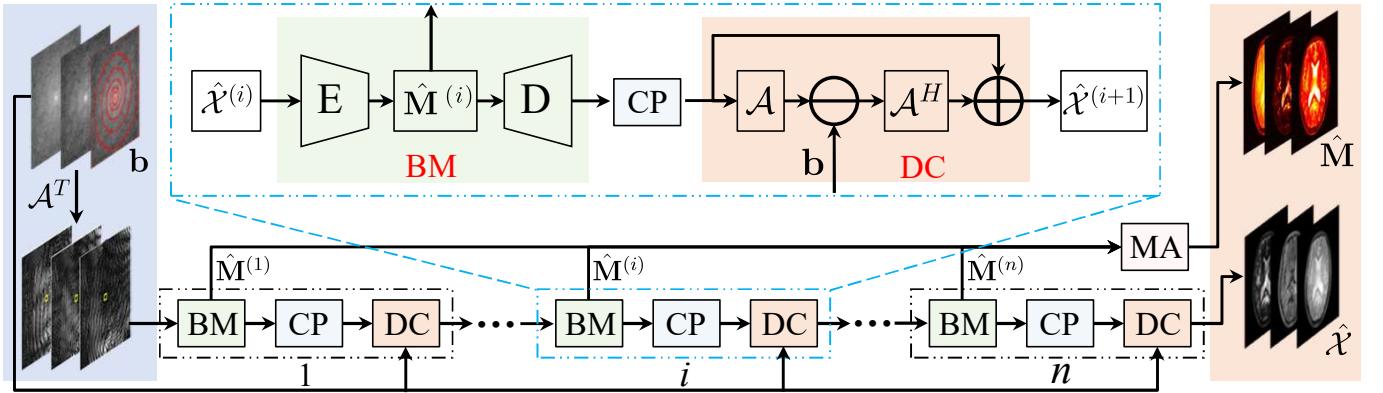


Fig. 1. Overview of the proposed TLR-BM-Net. The data consistency module (DC), the Bloch response manifold module (BM), and the learned CP decomposition module (CP) are constructed according to the three procedures in (11)-(13). The parameter map aggregation module (MA) at the end of the network aggregates all the parameter maps estimated by BM modules into the final parameter map. n represents the cascade depth of the network.

time (TR), echo time (TE), and the flip angle (FA). $\mathbf{B}(\cdot) : \mathcal{M} \rightarrow \mathbb{C}^{1 \times L}$ denotes a smoothing mapping induced by the Bloch equation dynamics. The Bloch response manifold [3] can be formulated as:

$$\mathcal{B} = \mathbf{B}(\mathcal{M}; \theta_L) \quad (3)$$

where \mathcal{B} denotes the Bloch response manifold. The precomputed dictionary in MRF is a discretized Bloch response manifold [3], [34], which can be modeled as:

$$\mathbf{D} = [\mathbf{d}_{m,:}], \quad \mathbf{d}_{m,:} = \mathbf{B}(\eta_m, \theta) \in \mathbb{C}^{1 \times L} \quad (4)$$

where \mathbf{D} denotes the constructed dictionary whose entries represent the theoretical response signal evolution for all possible tissues. $\{\eta_m, m = 1, \dots, M\}$ represents a set of discrete samples taken from the parameter space. The parameter look-up table (LUT) records the correspondence between parameters and dictionary entries. Therefore, the estimation of multiple parameters from the acquired signal evolution [1] can be expressed as:

$$k^i = \arg \max_k \frac{|\langle \mathbf{d}_{k,:}, \mathcal{X}_{i,:} \rangle|}{\|\mathbf{d}_{k,:}\|_2^2} \quad (5)$$

$$\hat{\eta}_i = \text{LUT}[k^i], \quad \hat{\rho}_i = \max \left\{ \frac{\text{real} \langle \mathbf{d}_{k^i,:}, \mathcal{X}_{i,:} \rangle}{\|\mathbf{d}_{k^i,:}\|_2^2}, 0 \right\} \quad (6)$$

where k^i represents the index of the best matching entry in the dictionary with the corresponding signal, $\hat{\eta}_i$ and $\hat{\rho}_i$ denote estimated parameters vector and estimated proton density, respectively. Thus, the multiple parameter maps \mathbf{M} can be obtained by matching each acquired signal $\mathcal{X}_{i,:}$ with dictionary entries according to (5) and (6). In order to effectively reduce the scan time, rapid acquisition schemes are widely used in (1) to accelerate data acquisition, but they also inevitably introduce aliasing artifacts in the recovered MRF images $\hat{\mathcal{X}}$, thereby reducing the parameters estimation accuracy in (5) and (6). In addition, the size of the dictionary \mathbf{D} grows exponentially as the number of tissue parameters M included is increased, which may result in prohibitively large dictionaries that require extensive computational resources to acquire multiple parameter maps.

III. METHODS

Based on the assumption that the resonance $\mathcal{X}_{i,:}$ of each voxel can be projected onto the Bloch response manifold \mathcal{B} , we can formulate the MRF reconstruction problem as the following convex optimization problem:

$$\min_{\mathcal{X}} \mathcal{J}(\mathcal{X}), \quad s.t. \quad \mathbf{b} = \mathcal{A}\mathcal{X}, \quad \mathcal{X} \in \mathcal{B} \quad (7)$$

where $\mathcal{J}(\mathcal{X})$ denotes the tensor low-rank regularization term. The above-regularized minimization problem can be rewritten in its Lagrangian form:

$$\min_{\mathcal{X} \in \mathcal{B}} \|\mathcal{A}\mathcal{X} - \mathbf{b}\|_F^2 + \lambda \mathcal{J}(\mathcal{X}) \quad (8)$$

where λ is a tunable regularization parameter. The first term is the data consistency term, which ensures that the k -space of the reconstructed image is consistent with the actual measurements \mathbf{b} . We adopt the incremental gradient-proximal method [35] to solve (8) iteratively. At the i -th iteration, it involves solving the following two subproblems:

$$\mathbf{Z}^{(i)} = \mathcal{P}_{\mathcal{B}}(\mathcal{X}^{(i)}) - \mu \mathcal{A}^H (\mathcal{A}\mathcal{X}^{(i)} - \mathbf{b}) \quad (9)$$

$$\mathcal{X}^{(i+1)} = \arg \min_{\mathcal{X} \in \mathcal{B}} \lambda \mathcal{J}(\mathcal{X}) + \frac{1}{2\mu} \|\mathcal{X} - \mathbf{Z}^{(i)}\|_F^2 \quad (10)$$

where $\mathcal{P}_{\mathcal{B}}(\cdot)$ denotes the projection operator of the Bloch response manifold \mathcal{B} , and μ is a positive step size. \mathbf{Z} represents an auxiliary variable. By combining gradient (9) and proximal (10) iterations, the sequences $\{\mathbf{Z}^{(i)}\}$ and $\{\mathcal{X}^{(i)}\}$ are constrained to lie on the Bloch response manifold \mathcal{B} , which is computationally expensive. To address this problem, we relax the constraint of $\mathcal{X} \in \mathcal{B}$ by the proximal iteration to improve the computational efficiency [35], which leads to the following iterative steps:

$$\mathcal{Z}^{(i)} = \mathcal{X}^{(i)} - \mu \mathcal{A}^H (\mathcal{A}\mathcal{X}^{(i)} - \mathbf{b}) \quad (11)$$

$$\hat{\mathbf{Z}}^{(i)} = \mathcal{P}_{\mathcal{B}}(\mathbf{Z}^{(i)}) \quad (12)$$

$$\mathcal{X}^{(i+1)} = \arg \min_{\mathcal{X} \in \mathbb{C}} \lambda \mathcal{J}(\mathcal{X}) + \frac{1}{2\mu} \|\mathcal{X} - \hat{\mathbf{Z}}^{(i)}\|_F^2 \quad (13)$$

Generally, the above iterative procedures take a relatively long computational time to obtain high-quality results. Besides, the selection of the regularization parameter is time-consuming and tedious. To address these problems, we unroll

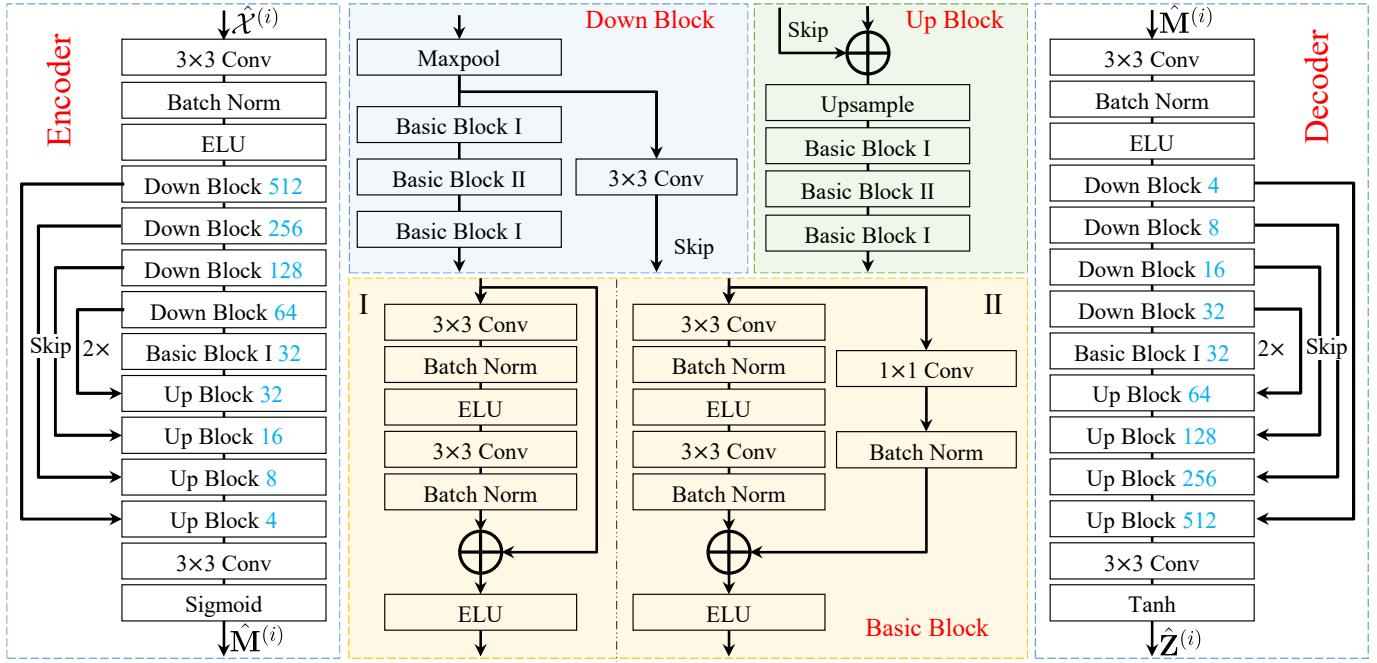


Fig. 2. Illustration of the proposed learned Bloch response manifold module (BM), which consists of a pair of encoder and decoder networks. The encoder and decoder have a symmetric network structure with short residual connections and skip-long connections, making full use of multi-scale feature information. The numbers with blue font represent the channel numbers associated with the input data of the corresponding blocks.

the above iterations into a deep neural network, as shown in Fig.1.

The proposed TLR-BM-Net is defined over the three iterative procedures in (11)-(13). The three procedures in (11)-(13) correspond to the three modules in the TLR-BM-Net, which are named the data consistency module (DC), the Bloch response manifold module (BM), and the learned CP decomposition module (CP). The BM module learns the mapping between reconstructed MRF data and the estimated parameter maps, which improves the reconstruction quality by exploiting the Bloch response manifold prior. The CP module is proposed to enforce the low-rank constraint of the high dimensional MRF tensor while avoiding computationally burdensome singular value decomposition. In addition, we introduce a parameter map aggregation module (MA) at the end of the network to aggregate all the parameter maps estimated by BM modules into the final parameter map. The detailed descriptions of the four modules are as follows.

A. Data Consistency Module (DC)

The data consistency module is constructed according to (11), which is used to constrain the reconstructed data to have the same sampling measurements as the distortion-free data. To enable the application of the DC module in non-Cartesian scenarios, we adopt a Non-Uniform Fast Fourier Transform (NUFFT) framework in Pytorch [36]. This NUFFT framework¹ enables efficient forward and backpropagation of the DC module, which is essential for training deep neural networks. Generally, data entering the convolutional layers in deep neural networks need to have normalized (or nearly normalized) values to avoid numerical problems [18]. Therefore,

according to [18], [37], [38], we calculate coil sensitivities for multi-coil sampled data and perform density compensation to obtain the properly weighted measurements, thereby avoiding numerical instability problems and enhancing the stability of network training. The step size μ in the DC module is a learnable parameter that can be adaptively adjusted during training to optimize the performance of the network.

B. Bloch Response Manifold Module (BM)

The projection operator $\mathcal{P}_B(\cdot)$ in (12) can be realized by separately projecting the individual voxel fingerprint sequence $\mathcal{X}_{i,j,:}$ onto the Bloch response manifold \mathcal{B} [39] [40] according to (5) and (6). However, projecting the acquired tissue fingerprints one by one is computationally inefficient, especially when dealing with numerous fingerprint signals. In addition, the calculation of the gradient required for backpropagation also greatly increases the computational complexity. Specifically, the parallel calculation of 10,000 fingerprint signals requires approximately 11.4G of GPU memory, while 48,400 MRF fingerprints need to be calculated in one mapping. This high computational complexity makes the conventional method impractical. To this end, we propose a Bloch resonance manifold module (BM) to efficiently learn the mapping between the reconstructed MRF data and the Bloch response manifold. The proposed BM module consists of a pair of encoder and decoder networks. The encoder learns to map the reconstructed MRF data \mathcal{X} to multiple parameter maps $\mathbf{M} \in \mathcal{M}$, while the decoder reconstructs MRF data from multiple parameter maps according to the Bloch response manifold, which can be modeled as:

$$\hat{\mathbf{M}}^{(i)} = f_E(\hat{\mathcal{X}}^{(i)}) \quad (14)$$

$$\hat{\mathbf{Z}}^{(i)} = f_D(\hat{\mathbf{M}}^{(i)}) \quad (15)$$

¹<https://github.com/mmuckley/torchkbnufft>

where f_E and f_D denote the encoder and decoder networks, respectively. The encoder and decoder have a symmetric network structure with short residual connections and skip-long connections, making full use of multi-scale feature information, as shown in Fig.2. The numbers with blue font in the figure represent the channel numbers associated with the input data of the corresponding blocks.

C. Learned CP Decomposition Module (CP)

Subproblem (13) is usually solved with the singular value thresholding (SVT) algorithm in most of the regularized and learning-based methods. However, the singular value decomposition (SVD) involved in SVT algorithms is computationally expensive for large-scale MRF data. Furthermore, if the difference between any two singular values is close to zero, the gradient will be numerically unstable¹. To address this problem, we propose a learned CP decomposition module to explore the tensor low-rank priors of MRF data. The MRF data \mathcal{X} is a 3-order tensor, which can be factorized into a sum of component rank-one tensors [25]:

$$\mathcal{X} \approx \sum_{r=1}^R \alpha_r \cdot \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r \quad (16)$$

where $\mathbf{a}_r \in \mathbb{R}^L$, $\mathbf{b}_r \in \mathbb{R}^{N_x}$, $\mathbf{c}_r \in \mathbb{R}^{N_y}$ are normalized rank-one basis vectors, α_r is the scalar weight, \circ denotes vector outer product, and R is the predefined CP rank.

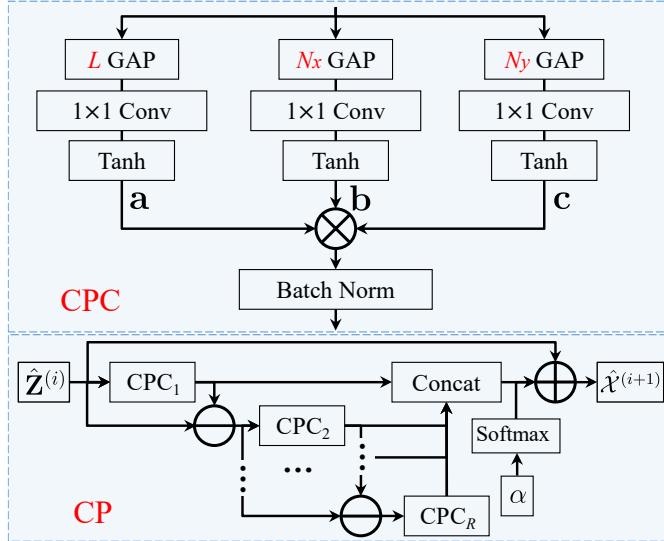


Fig. 3. Illustration of the learned CP decomposition module (CP). The proposed CP module first learns the discriminative rank-one tensor components by several CP component modules (CPC) and then aggregates them into a low-rank tensor.

According to the CP decomposition in (16), the proposed CP module first learns the discriminative rank-one tensor components by several CP component modules (CPC), and then aggregates them into a low-rank tensor, as shown in Fig.3. Note that the number of the CPC modules is the CP rank R . Specifically, the CPC module first applies *global average pooling (GAP)-Conv-Tanh* operation on each dimension of the

MRF data to generate rank-one basis vector \mathbf{a}_r , \mathbf{b}_r , \mathbf{c}_r . Then, the rank-one tensor \mathbf{O} is generated by applying the vector outer product and *Batch Normalization*:

$$\mathbf{O} = f_{bn}(\mathbf{a} \circ \mathbf{b} \circ \mathbf{c}) \quad (17)$$

where f_{bn} denotes the batch normalization layer. Generating rank-one tensors from high-dimensional data is a nonlinear operation. Each *GAP* layer is utilized to reduce the dimension of the input data along its corresponding dimensions (i.e., N_x , N_y , and L), extracting essential features. Then, the *Conv-Tanh* layer plays a crucial role in learning the nonlinear mapping between the reduced-dimension features and the corresponding rank-one tensor. By combining the *GAP* and *Conv-Tanh* layers, the proposed CPC modules can effectively generate rank-one tensors from complex and high-dimensional MRF data.

Inspired by [41], we introduce residual learning in the CP module so that the information of different frequencies can be better captured. Specifically, the first CPC module generates a rank-one tensor from its input: $\mathbf{O}_1 = f_{CPC_1}(\mathbf{Z}^{(i)})$. The second CPC module then generates a rank-one tensor from the residual component: $\mathbf{O}_2 = f_{CPC_2}(\mathbf{Z}^{(i)} - \mathbf{O}_1)$, which captures the remaining information that is not captured by \mathbf{O}_1 . This process is repeated for each CPC module, generating a rank-one tensor from the remaining residual components in turn. Given the predefined hyperparameter CP rank R , we can obtain R rank-one tensor in order: $\{\mathbf{O}_1, \dots, \mathbf{O}_R\}$. By extracting the rank-one tensors hierarchically, the proposed scheme guarantees an efficient representation of low-rank tensors with a minimum number of CPC modules. We further adopt an additional long residual connection between input and output, enabling CPC modules to focus on capturing the main components of the input. It should be noted that the hyperparameter CP rank R does not indicate the actual CP rank of the output. Instead, it just represents the number of CPC modules in our proposed CP module.

In addition, the scalar weight vector $\alpha = \{\alpha_1, \dots, \alpha_R\}$ is learned during training and shared across different inputs, efficiently integrating rank-one tensors with hierarchical relationships into the low-rank tensor. The shared weight scheme can improve computational efficiency and is also beneficial to enhance the generalization performance of the model.

D. Map Aggregation Module (MA)

We add a parameter map aggregation module (MA) at the end of the network to aggregate all the parameter maps ($\hat{\mathbf{M}}^{(i)}$, $i = 1, \dots, n$) estimated by different BM modules into the final parameter map $\hat{\mathbf{M}}$, which can be modeled as:

$$[\mathbf{W}^{(i)}] = f_{MA}([\hat{\mathbf{M}}^{(i)}]) \quad (18)$$

$$\hat{\mathbf{M}} = \sum_{i=1}^n \mathbf{W}^{(i)} \cdot \hat{\mathbf{M}}^{(i)} \quad (19)$$

where $\mathbf{W}^{(i)}$ denotes the learned normalization weights matrix for the i -th parameter map $\hat{\mathbf{M}}^{(i)}$, \cdot is the matrix element-wise multiplication, and f_{MA} represents the MA module. The proposed MA module is designed to learn normalization weights from the concatenated multiple parameter maps,

¹<https://pytorch.org/docs/stable/generated/torch.linalg.svd.html?highlight=svd#torch.linalg.svd>

which are then used to compute the weighted sum of all the parameter maps. The MA module can make full use of the estimated information at different stages of the network, and the detailed structure is shown in Fig.4. In addition, the MA module establishes a long connection between the output parameter maps and each BM module, which is beneficial for backpropagation and avoiding gradient vanishing.

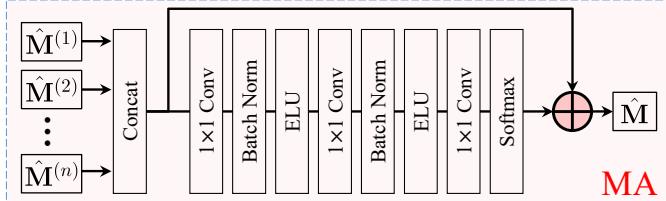


Fig. 4. Illustration of the map aggregation module (MA). The proposed MA module learns normalization weights from all the parameter maps ($\hat{M}^{(i)}$, $i = 1, \dots, n$) and then computes a weighted sum to obtain the final parameter map \hat{M} .

E. Implementation Details

The proposed scheme was developed based on the Pytorch deep-learning framework and implemented on a Linux workstation with an Intel Xeon CPU and an Nvidia Quadro GV100 GPU. The learning rate was initially set to 3e-4 and exponentially decayed for every epoch. The learnable step size μ in each DC module was initially set to 1e-3. The network was trained for 50 epochs. We set the cascade depth n of the network to 4 and the number of CPC modules R to 5 empirically. Exponential Linear Unit (ELU) [42] was chosen as the nonlinear activation function to speed up the learning process of the proposed network. The proposed network was trained by Adam optimizer [43] with parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 1e - 8$. Our code is available at <https://github.com/bigponglee/TLR-BM-MRF>.

IV. EXPERIMENTAL RESULTS

A. Datasets

The research study was conducted on both simulated data and *in vivo* data.

1) *In vivo* data: The *in vivo* data were acquired on a 3T Siemens Prisma scanner with a 32-channel head coil from 8 healthy volunteers across 10 slices each. Each volunteer was scanned using 1000-length fast imaging with steady-state precession (FISP) pulse sequence with the TR and FA patterns as shown in Fig.5. The relevant imaging parameters included $\text{FOV} = 220 \times 220 \text{ mm}^2$, and slice thickness = 5 mm. 36 spiral trajectories (see Fig.5: c) were used to acquire 2880 samples in each frame, resulting in an undersampling ratio of $\sim 6\%$. As demonstrated in [11], high-quality parameter maps could be reconstructed using the SL-SP method and 1000-length MRF data with an accuracy close to the ground truth. Thus, the multiple parameter maps reconstructed using the SL-SP method were considered the ground truth for quantitative comparisons.

2) *Simulation data*: The simulation data was generated by using the parameter maps (T_1 , T_2) obtained from Brainweb, including 150 sets each from the normal and multiple sclerosis lesion brain databases [44]. The Bloch simulations were

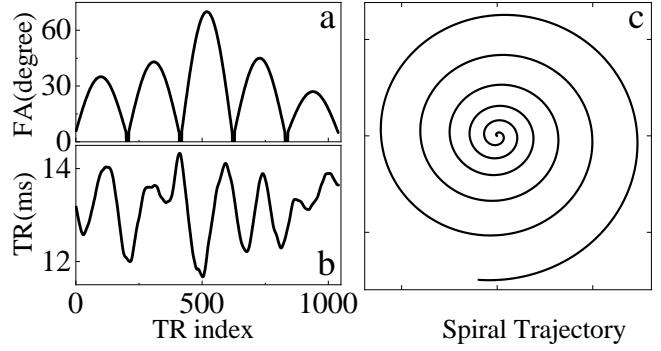


Fig. 5. a and b are the flip angles and repetition time patterns of the FISP sequence. c shows the spiral undersampling trajectory used in one repetition time.

performed to generate the multi-frame MRF data by using the FISP pulse sequence shown in Fig.5. The same sampling strategy as used in the *in vivo* data experiment was adopted to generate the simulation dataset.

$\sim 10\%$ of the simulation dataset were used for testing and the remaining were used for training. For *in vivo* data, all 10 slices from one randomly selected volunteer were used for testing and the rest were used for training. In addition, based on our previous study [11], we found that 1) the reconstruction performance of the methods improved with the increase of data length since there is more available known information; 2) when the data length exceeds 500, satisfactory results could be obtained by appropriate algorithm. Therefore, in order to compare the performance of different algorithms in extreme scenarios, we performed experiments at a sequence length of 500.

B. Training Strategy

The proposed TLR-BM-Net can simultaneously reconstruct MRF data and multi-parameter maps from non-Cartesian undersampled measurements. However, due to the excessively long acquisition time required for fully sampled data, the acquired *in vivo* data lacked distortion-free multi-frame MRF data as a ground truth. The absence of ground truth MRF data increased the difficulty of network training, especially for the BM module, which needed to learn the mapping between the constructed MRF data and the Bloch response manifold. Thus, we proposed to pre-train the BM module using the simulated data. The proposed TLR-BM-Net was then trained end-to-end. We adopted the MSE loss \mathcal{L} as the evaluation criterion in all experiments.

The simulation data with full ground truth MRF data \mathcal{X} and parameter map M pairs were used to pre-train the BM module based on the following loss function:

$$\text{Loss}_{BM} = \mathcal{L}(\hat{\mathcal{X}}, \mathcal{X}) + \mathcal{L}(M, \hat{M}) \quad (20)$$

where $\hat{\mathcal{X}}$ and \hat{M} were the reconstructed MRF data and parameter maps, respectively. \mathcal{X} and M were the ground truth MRF data and parameter maps, respectively. In addition, the pre-train scheme was beneficial to ensure that the output of the BM modules lies in the Bloch response manifold.

Subsequently, the entire TLR-BM-Net was trained end-to-end using only the parameter maps as ground truth. The corresponding loss function can be defined as:

$$\text{Loss} = \mathcal{L}(\mathcal{A}\hat{\mathcal{X}}, \mathbf{b}) + \mathcal{L}(\mathbf{M}, \hat{\mathbf{M}}) \quad (21)$$

where the first item was used to enforce data consistency.

The high-dimensional MRF data reconstruction led to a large number of parameters in the network, which may reduce the performance of training and inference. To alleviate this problem, we utilized the automatic mixed precision (AMP) [45] training scheme that adaptively performs network operations on half-precision 16-bit floating point (FP16) to improve the performance while maintaining accuracy. A gradient scaling strategy [46] was also employed to improve the convergence of networks with FP16 gradients by minimizing gradient underflow.

C. Evaluation Indexes

We adopted two performance evaluation indexes to quantitatively evaluate the performance of the proposed method, including the peak signal-to-noise ratio (PSNR) and the normalized mean square error (NMSE). Specifically, we used the peak signal-to-noise ratio (PSNR) to measure the quality of the reconstructed MRF data, which can be defined as:

$$\text{PSNR} = 10 \log_{10} \left(\frac{1}{\frac{1}{N} \sum_{i=1}^N (\hat{x}_i - x_i)^2} \right) \quad (22)$$

where $N = N_x \times N_y \times L$ was the number of pixels in the MRF data, x_i and \hat{x}_i denoted specific values at the i -th location in the ground truth and reconstructed MRF data, respectively. The normalized mean square error (NMSE) was used to measure the quality of the reconstructed parameter maps $\hat{\mathbf{M}}$, which can be defined as:

$$\text{NMSE} = \frac{\|\hat{m}_i - m_i\|^2}{\|m_i\|^2} \quad (23)$$

where m_i and \hat{m}_i denoted the ground truth and reconstructed parameters, respectively.

D. Reconstruction Performance

1) Effect of cascade depths: The proposed TLR-BM-Net was constructed by unrolling the reconstruction model into a deep neural network. The basic modules of the network were built according to the iterative steps of the reconstruction model. Thus, in general, the performance of an unrolled network increased as the cascade depth of the network increases [22], [23]. In this experiment, we explored the degree of performance improvement obtained by increasing the cascade depth of the network. In the experiment, we fixed the other experimental conditions but varied the cascade depth n of the network, i.e., $n \in \{2, 3, 4, 5, 6\}$.

Fig. 6 shows the quantitative comparison results under different cascade depths n , including the mean PSNR of the reconstructed MRF data $\hat{\mathcal{X}}$ and the average NMSE error of the reconstructed parameter maps. Experimental results show that the performance of the proposed TLR-BM-Net gets improved as the cascade depth n increases, while the performance gain

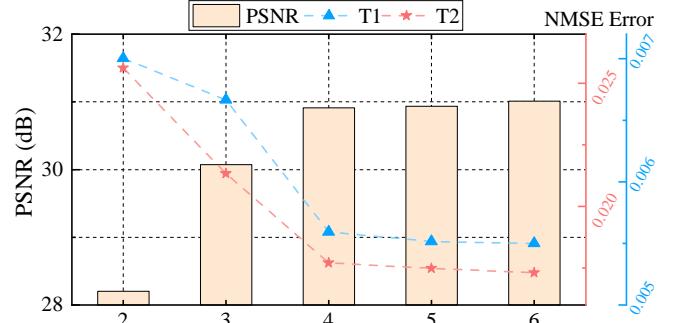


Fig. 6. Quantitative comparison results under different cascade depths n . The horizontal axis represents the cascade depth n of the network. The histogram represents the mean PSNR of the reconstructed MRF data $\hat{\mathcal{X}}$ in the test dataset. The blue and red curves represent the average NMSE error of the T1 and T2 parameter maps reconstructed using the test dataset, respectively.

gradually saturates when the cascade depth exceeds 4. When the cascade depth is greater than 4, the number of parameters in the network increase greatly, requiring more training epochs to achieve convergence. Table I shows the processing time for training and testing of the proposed TLR-BM-Net with different cascade depths n . We observe that the processing time for training and testing of the proposed TLR-BM-Net increases approximately linearly with the increase of cascade depth n . Compared with the increase in the training time, the testing time remains acceptable (less than 10 sec) when a larger cascade depth is used. Therefore, we set the cascade depth n to 4 in the experiments after a trade-off between the performance and the efficiency of the proposed scheme.

TABLE I
COMPARISON OF THE PROCESSING TIME FOR TRAINING AND TESTING
OF THE PROPOSED TLR-BM-NET UNDER DIFFERENT CASCADE
DEPTHES n .

n	2	3	4	5	6
Training time (h)	47.5	64.1	80.3	96.9	117.1
Avg-Processing time (s)	2.87	4.27	5.88	7.08	8.81

2) Effect of the number of CPC modules: The proposed learned CP module was designed based on the CP decomposition. The predefined CP rank R determines the number of CPC modules, which plays an important role in the reconstruction

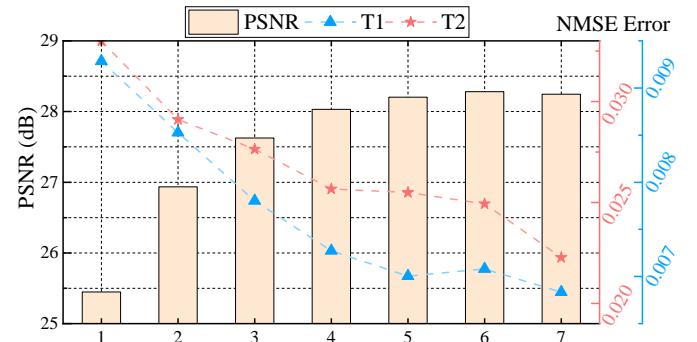


Fig. 7. Quantitative comparison results under the different number of CPC modules R of the learned CP module. The horizontal axis represents the number of CPC modules R . The histogram represents the mean PSNR of the reconstructed MRF data $\hat{\mathcal{X}}$ in the test dataset. The blue and red curves represent the average NMSE error of the T1 and T2 parameter maps reconstructed using the test dataset, respectively.

TABLE II

QUANTITATIVE COMPARISON OF DIFFERENT METHODS ON THE TEST DATASET, INCLUDING SEPARATE EVALUATION RESULTS FOR SIMULATION AND *in vivo* DATA (MEAN \pm STANDARD DEVIATION). NON-PARAMETRIC STATISTICAL TEST (KRUSKAL-WALLIS TEST) INDICATES STATISTICALLY SIGNIFICANT PERFORMANCE DIFFERENCES BETWEEN OUR PROPOSED METHOD AND OTHER EXISTING METHODS BASED ON MULTIPLE EVALUATION METRICS.

	Methods	FLOR	SCQ	LRSM	SL-SP	BM	TLR-BM
Simulation	PSNR	24.85 \pm 1.85	-	26.07 \pm 1.73	28.71 \pm 1.62	28.40 \pm 1.53	30.91\pm1.51
	T1 NMSE	0.0383 \pm 0.0097	0.0392 \pm 0.0089	0.0217 \pm 0.0045	0.0083 \pm 0.0033	0.0049 \pm 0.0023	0.0030\pm0.0014
	T2 NMSE	0.0769 \pm 0.0287	0.0685 \pm 0.0283	0.0632 \pm 0.0235	0.0232 \pm 0.0159	0.0192 \pm 0.0124	0.0098\pm0.0071
<i>in vivo</i>	T1 NMSE	0.0610 \pm 0.0387	0.1158 \pm 0.0378	0.0539 \pm 0.0251	0.0195 \pm 0.0171	0.0304 \pm 0.0120	0.0196\pm0.0070
	T2 NMSE	0.4611 \pm 0.1279	0.4066 \pm 0.1807	0.444 \pm 0.107	0.4109 \pm 0.0579	0.1117 \pm 0.0373	0.0605\pm0.0248
KW test VS TLRBM (<i>p</i> -value)		1.51E-06	2.90E-43	5.74E-04	2.30E-03	1.92E-02	-
Avg Processing time		50.13 min	1.42 s	61.73 min	59.55 min	5.79 s	5.88 s

performance. In this experiment, we explored the effect of the number of CPC modules on the network performance. We fixed the other experimental conditions but varied the number of CPC modules of the learned CP module, i.e., $R \in \{1, 2, 3, 4, 5, 6, 7\}$. For the sake of computational complexity, the cascade depth n of the network was set to 2 in this experiment.

Fig.7 shows the quantitative comparison results under different numbers of CPC modules of the learned CP module. We observe that a small number of CPC modules ($R < 4$) makes the CP module unable to fully capture the structural priors of MRF data and leads to poor performance. As the number of CPC modules increases, the CP module is able to capture more structural priors of MRF data, gradually improving the reconstruction performance. However, the increase of the number of CPC modules is accompanied by additional memory overhead. The experimental results in Fig.7 demonstrate that when $R > 5$, the performance improvement brought by improving the number of CPC modules has tended to saturation. Therefore, we set the number of CPC modules R to 5 in our experiments to obtain significantly improved performance with acceptable computational overhead.

3) Comparison with the state-of-the-art methods: We compared the performance of the proposed TLR-BM method with three state-of-the-art methods: FLOR [5], SCQ [14] and SL-SP [11]. We have tuned the parameters of all experiments based on the default parameters provided by the authors to ensure optimal performance for each scenario. In addition, in order to better evaluate the performance of the proposed learned CP module, we also conducted ablation experiments to evaluate the performance of the network without using the CP module, which is abbreviated as BM. The cascade depth n of the proposed network was set to 4 and the CP rank R was set to 5 in this experiment.

In Fig.8, we plot the comparison results of the proposed method with the methods under comparison using data simulated by a healthy brain model. The SCQ method can achieve accurate parameter reconstruction under quadruple accelerated acquisition [14], but its performance is limited when a high acceleration factor is applied (only 6% sampled measurements are used) due to the lack of utilization of MRF data and Bloch manifold priors. We observe that the proposed TLR-BM-Net reconstructs the most accurate parameter maps among all methods. Fig.9 shows the comparison results using data simulated by a brain model with multiple sclerosis (MS). The

lesion area was zoomed-in for clear observation. Experimental results show that FLOR, SCQ and BM methods cannot accurately recover smaller lesion areas. The proposed TLR-BM-Net can accurately recover the detailed texture of the parameter map and reconstruct the lesion area. To better demonstrate the effectiveness of our proposed method in reconstructing high-dimensional MRF data, we present a visual comparison of image results from different frames in MRF data reconstructed by different methods, as shown in Fig.10. We compared the reconstruction results of different methods on various frames of the MRF data. The results show that our method consistently produces higher accuracy parameter maps and better quality MRF data than other existing methods. Additionally, we observe that the parameter maps obtained by MRF can provide more detailed tissue information than the data frames themselves. Fig.11 shows the parameter maps reconstructed by different methods from one set of *in vivo* data in the test dataset. The proposed method reconstructs parameter maps most accurately and significantly improves the reconstruction accuracy of the T2 parameter map.

TABLE III
QUANTIZATION ERROR OF T1 AND T2 ON DIFFERENT TISSUES,
INCLUDING GRAY MATTER (GM), WHITE MATTER (WM),
CEREBROSPINAL FLUID (CSF), AND FAT (MEAN \pm STANDARD
DEVIATION, UNIT: %)

	Subject	FLOR	SCQ	SL-SP	LRSM	BM	TLR-BM
T1	GM	1.02 \pm 0.10	2.27 \pm 0.24	0.79 \pm 0.09	0.81 \pm 0.09	0.82 \pm 0.11	0.65\pm0.08
	WM	2.28 \pm 0.09	1.54 \pm 0.18	1.12 \pm 0.07	1.29 \pm 0.07	1.09\pm0.14	1.10 \pm 0.13
	CSF	1.88 \pm 0.07	0.97 \pm 0.16	0.68 \pm 0.06	0.95 \pm 0.06	0.21 \pm 0.04	0.13\pm0.02
	FAT	3.91 \pm 0.08	2.05 \pm 0.18	2.52 \pm 0.07	2.58 \pm 0.07	1.48 \pm 0.16	1.30\pm0.15
T2	GM	1.69 \pm 0.13	2.18 \pm 0.28	1.21 \pm 0.11	1.45 \pm 0.11	1.30 \pm 0.11	1.03\pm0.09
	WM	1.68 \pm 0.08	2.17 \pm 0.15	1.35 \pm 0.06	1.39 \pm 0.07	2.12 \pm 0.10	1.12\pm0.11
	CSF	2.52 \pm 0.10	1.71 \pm 0.21	1.23 \pm 0.09	1.41 \pm 0.09	0.80 \pm 0.10	0.49\pm0.07
	FAT	1.71 \pm 0.07	4.3 \pm 0.12	1.42 \pm 0.07	1.47 \pm 0.062	1.56 \pm 0.09	1.40\pm0.10

Table.II shows quantitative comparison results of different methods on the test dataset. To comprehensively evaluate the generalization ability of our proposed method on both simulation and *in vivo* data, we report the quantitative results separately for each dataset in the test dataset. Additionally, to test the consistency of performance differences between our proposed method and other existing methods, we conducted a non-parametric statistical test (Kruskal-Wallis test [47]) using multiple evaluation metrics. The statistical results reveal that the p-values of our proposed method and other methods are much less than 0.005, which indicates that the differences in performance between the methods are statistically significant. To further show the accuracy of the proposed approach, we

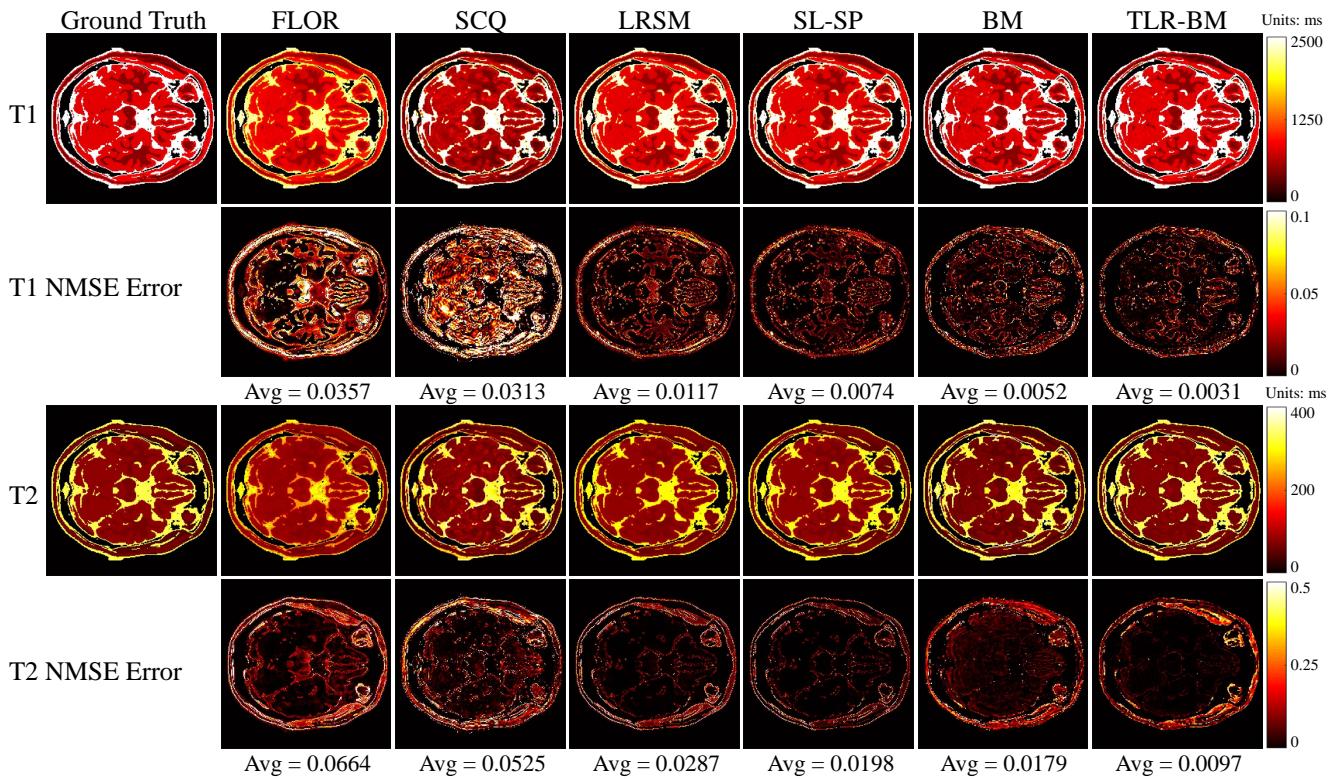


Fig. 8. Illustration of the comparison results using data simulated by a healthy brain model in the test dataset. From the left to the right columns are the ground truth maps, estimated maps by the FLOR, SCQ, LRSM, SL-SP, BM, and the proposed TLR-BM-Net. The first and third rows show the reconstructed T1 and T2 parameter maps, respectively, and the second and fourth rows show the corresponding NMSE error maps.

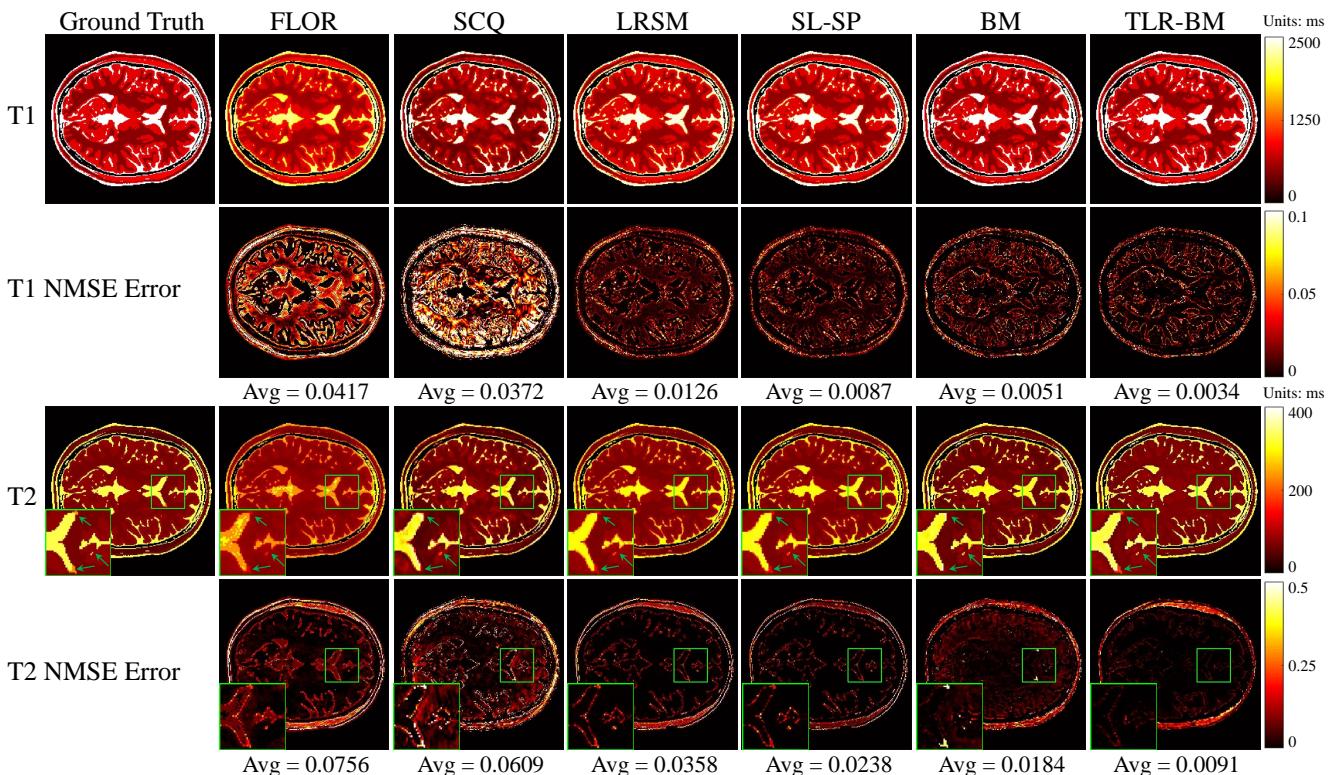


Fig. 9. Illustration of the comparison results using data simulated by a brain model with multiple sclerosis (MS) in the test dataset. From the left to the right columns are the ground truth maps, estimated maps by the FLOR, SCQ, LRSM, SL-SP, BM, and the proposed TLR-BM-Net. The first and third rows show the reconstructed T1 and T2 parameter maps, respectively, and the second and fourth rows show the corresponding NMSE error maps. The lesion area is zoomed-in for clear observation.

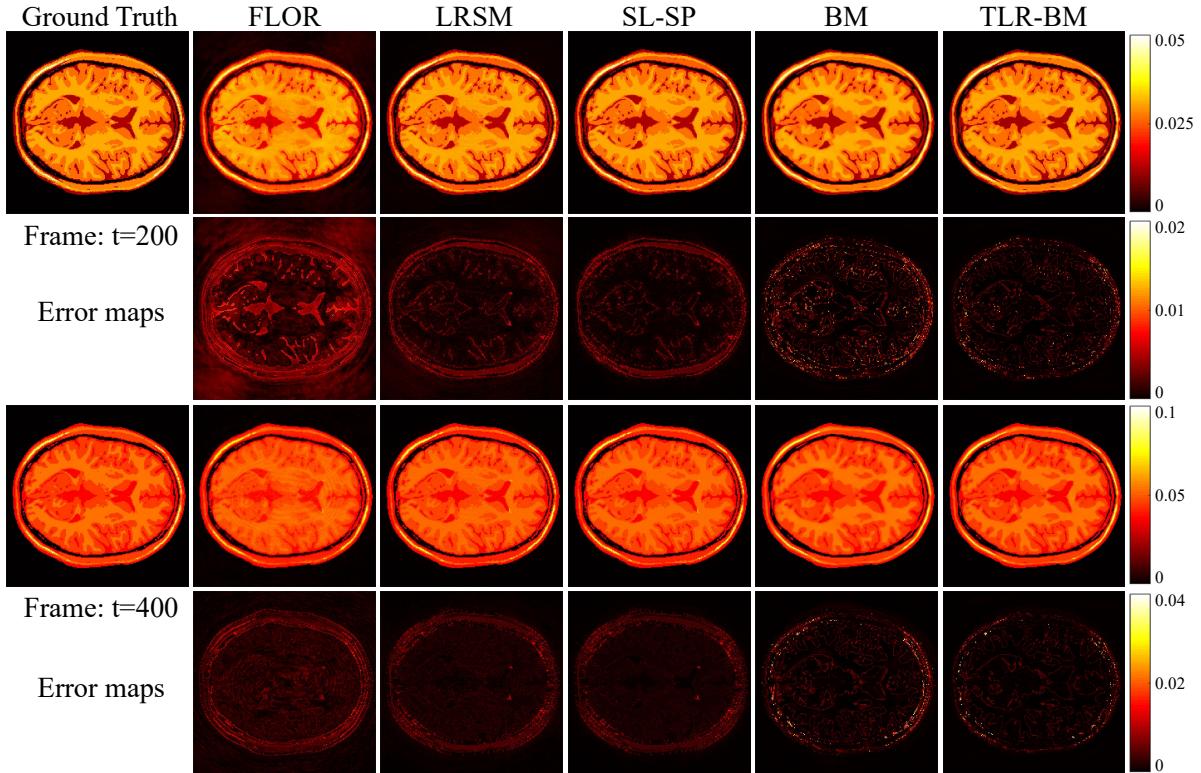


Fig. 10. Comparison of reconstructed MRF data for time frame 200 and time frame 400, respectively. The first to the last column represents the ground truth data frames and the corresponding estimated data frames obtained by FLOR, LRSM, SL-SP, BM, and our proposed TLR-BM-Net, respectively. The first and third rows display the 200-th and 400-th frames of the reconstructed MRF data, respectively, and the second and fourth rows depict the corresponding NMSE error maps.

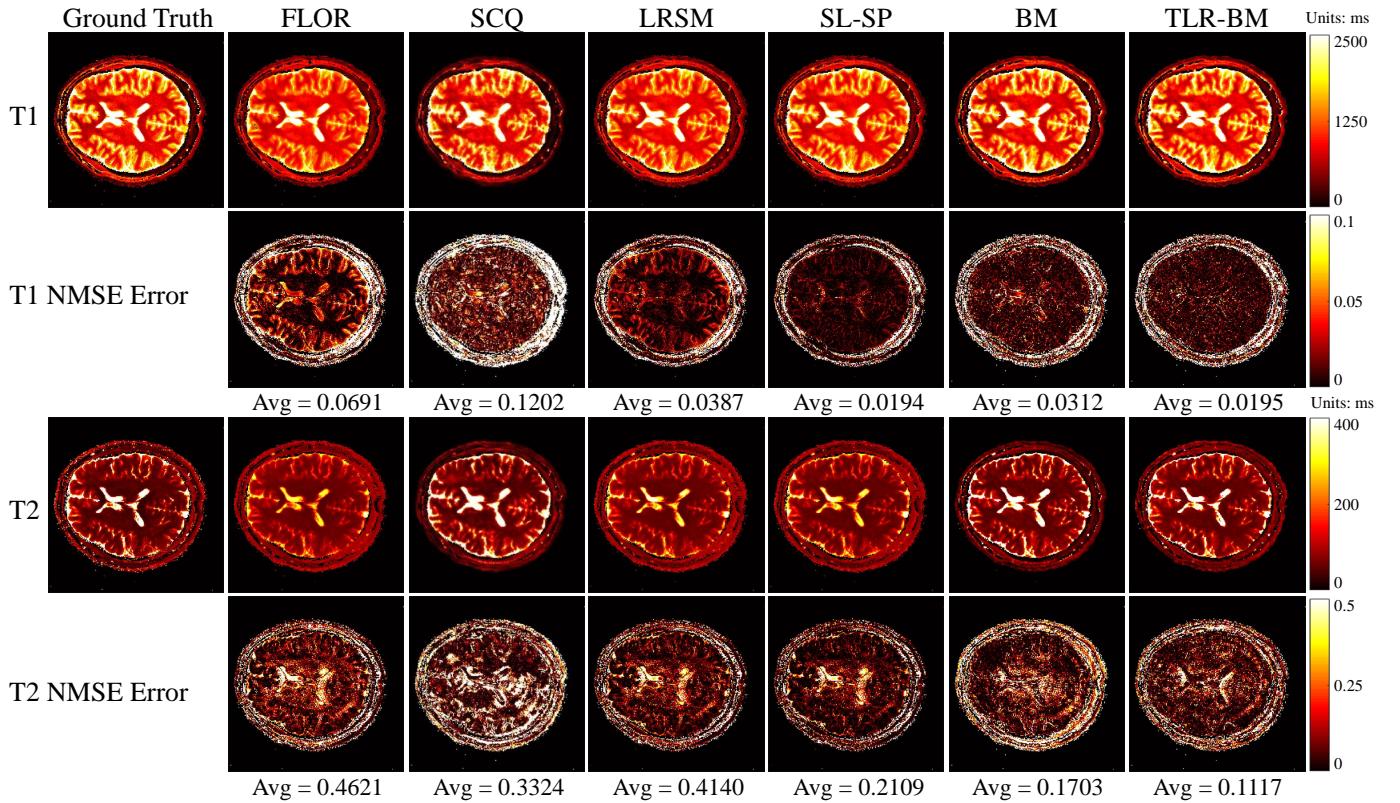


Fig. 11. Illustration of the comparison results using a set of *in vivo* data in the test dataset. From the left to the right columns are the ground truth maps, estimated maps by the FLOR, SCQ, LRSM, SL-SP, BM, and the proposed TLR-BM-Net. The first and third rows show the reconstructed T1 and T2 parameter maps, respectively, and the second and fourth rows show the corresponding NMSE error maps.

also present quantitative error measurements of estimated T1 and T2 values for various tissues in the simulation dataset in Table.III, including gray matter (GM), white matter (WM), cerebrospinal fluid (CSF), and fat. Note that the labels of various tissues were provided by Brainweb [44]. Compared with traditional end-to-end network methods (SCQ), our proposed method has comparable processing performance. Meanwhile, the unrolled network based on the tensor low-rank priors and the Bloch response manifold priors can make full use of prior knowledge and greatly improve the accuracy of the estimated parameter maps. On the other hand, compared with the traditional iterative methods (FLOR and SL-SP), our proposed method can significantly reduce the reconstruction time with improved reconstruction performance. In addition, ablation experiments also show that the proposed CP module plays an important role in capturing the structural prior information of MRF data, which can lead to significantly improved reconstruction performance.

4) Cross Validation: To further evaluate the generalization ability of the proposed TLR-BM-Net, we conducted a 4-fold cross-validation experiment. Specifically, the dataset was randomly divided into 4 subsets, and each subset was used as the test dataset once. The remaining 3 subsets were used as the training dataset. Each subset consisted of randomly selected 75 simulation data and 20 *in vivo* data slices from two randomly selected volunteers, providing a diverse range of data for training and testing. The 4-fold cross-validation experiment was repeated four times, and the average performance is reported in Table IV. The results show that the proposed method exhibits consistent performance across different datasets, further demonstrating its generalization ability. Although the performance on simulated data is slightly better than that on real data, the level of volatility is acceptable. The robustness of the proposed TLR-BM-Net is confirmed by its consistent performance across different datasets.

TABLE IV

PERFORMANCE COMPARISON OF THE PROPOSED TLR-BM-NET USING 4-FOLD CROSS-VALIDATION ON THE DATASET (MEAN \pm STANDARD DEVIATION).

Fold	Simulation			<i>in vivo</i>	
	PSNR	T1 NMSE	T2 NMSE	T1 NMSE	T2 NMSE
1	31.33 \pm 1.49	0.0038 \pm 0.0004	0.0110 \pm 0.0035	0.0201 \pm 0.0108	0.0961 \pm 0.06855
2	30.33 \pm 1.38	0.0028 \pm 0.0005	0.0077 \pm 0.0025	0.0219 \pm 0.0100	0.0531 \pm 0.0182
3	30.12 \pm 1.16	0.0037 \pm 0.0006	0.0098 \pm 0.0022	0.0179 \pm 0.0065	0.0806 \pm 0.0447
4	30.89 \pm 1.32	0.0030 \pm 0.0006	0.0120 \pm 0.0101	0.0218 \pm 0.0135	0.0627 \pm 0.0213
Avg	30.67	0.0033	0.0101	0.0204	0.0731

V. DISCUSSION

A. Analysis of the BM Module

As the dictionary \mathbf{D} is constructed using a discrete and finite set of tissue parameters, an approximation is made when projecting the MRF data onto the cone of the Bloch response manifold according to (5) and (6). In other words, the accuracy of the estimated Bloch manifold is positively related to the number of tissue parameters in the dictionary. However, it is difficult to estimate a high accuracy Bloch manifold using the matching procedure in (5) and (6) due to the high computational overhead. In contrast, the powerful learning ability of neural networks has the potential to estimate

tissue parameters in a continuous parameter space [12], which means that the proposed BM module can approximate the Bloch response manifold on a nearly continuous basis. Thus, the proposed BM module can fully mine data features and exploit the Bloch response manifold prior, improving the reconstruction quality.

B. CP Module VS CP Decomposition

Our proposed CP module is built upon CP decomposition, but we have extended the low-rank tensor representation capability of CP decomposition and made it more suitable for neural networks. Traditional CP decomposition may require hundreds of rank-one component tensors [8] to achieve adequate low-rank tensor representations. This can lead to significant computational overhead when building CP modules directly based on CP decomposition. Moreover, using hundreds of CPC modules to equally extract features from input may lead to redundancy. To address these issues, our proposed CP module generates rank-one tensors at different hierarchical levels, allowing for efficient representation of low-rank tensors using a minimum number of CPC modules. We further adopted an additional long residual connection between the input and the output, allowing CPC modules to focus on capturing the main components of the input without necessarily needing to learn a complete low-rank tensor representation. The specially designed CP module allows for effectively low-rank tensor representation with a few CPC modules.

C. CP VS Tucker Decomposition

The CP and Tucker decompositions are two widely used methods for representing high-dimensional data with low-rank tensors. The CP decomposition decomposes a tensor into a sum of rank-one tensors constructed by the vector outer product, while the Tucker decomposition decomposes a tensor into a core tensor and a number of factor matrices. Although the Tucker decomposition can more finely utilize the low-rank priors of different dimensions of the data, it involves a higher computational overhead due to the need for matrix factorization. Furthermore, the Tucker rank is a vector composed of the ranks of unfolding matrices of the target tensor along different dimensions, which increases the complexity of directly applying Tucker decomposition into a neural network. In contrast, the CP rank is defined as the minimum number of required component tensors, making it simpler and more suitable for efficient implementation in a neural network. Moreover, the CP decomposition can be easily extended to the residual learning scheme, which accelerates the training process and reduces computational overhead. Therefore, we believe that the CP decomposition is a more appropriate choice for our low-rank tensor model due to its simplicity, interpretability, and ease of implementation in a neural network. However, future work can explore more efficient methods to use Tucker decomposition to further improve the reconstruction performance.

D. the CP-imposed Spatial Structure

To evaluate the contribution of the proposed CP module, we have conducted an ablation experiment and compared the

performance of three networks: 1) the proposed TLR-BM-Net with the CP module; 2) BM network without the CP module; and 3) 2D-CNN-BM network replacing the CP modules with 2D-CNN modules. The ablation experiments are used to show the added value of the temporal correlation captured by the CP module instead of the spatial structure alone. There are different choices of spatial transform sparsity [48], [49], including TV, wavelet, etc. Choosing one of the types might be a limited and unfair comparison. To address this concern, we chose to learn the optimal spatial transform by CNN, which is applied in many recently published research papers [23]. Each network was trained and tested under identical experimental conditions as in Section IV-D. The quantitative results of these experiments are presented in Table V. We find that the proposed network obtains the best performance compared with networks 2) and 3), which indicates that the temporal correlation constraint provided by the CP module is significant in improving the overall performance of the proposed method.

TABLE V

QUANTITATIVE COMPARISON OF THE PROPOSED CP MODULE WITH ONLY USING SPATIAL STRUCTURE (MEAN \pm STANDARD DEVIATION).

	Methods	BM	2D-CNN-BM	TLR-BM
Simulation	PSNR	29.06 \pm 1.53	29.93 \pm 1.77	31.05\pm1.51
	T1 NMSE	0.0049 \pm 0.0023	0.0038 \pm 0.0028	0.0030\pm0.0014
	T2 NMSE	0.0192 \pm 0.0124	0.0188 \pm 0.0163	0.0098\pm0.0071
<i>in vivo</i>	T1 NMSE	0.0304 \pm 0.0120	0.0287 \pm 0.0131	0.0196\pm0.0070
	T2 NMSE	0.1117 \pm 0.0373	0.1095 \pm 0.0361	0.0605\pm0.0248

E. Limitations and Future Directions

Although the proposed TLR-BM-Net has shown promising results in reconstructing MRF data and multi-parameter maps from non-Cartesian undersampled measurements, it still has some limitations that need to be addressed in future research. One of the main limitations is the high demand for GPU memory during training due to the high dimensionality of the MRF data and the high overheads of the NUFFT operations. We have attempted to use technologies such as automatic mixed precision to reduce memory requirements. However, high memory resources are still required to realize the training and testing of the network. This limitation can hinder the generalizability of the proposed method for larger datasets and more complex applications. Another problem is that the training of the network takes a long time due to the time-consuming NUFFT operation. Although the BM module pre-training strategy can reduce training time to some extent, the overall training time is still longer than that of the network with Cartesian sampling. One possible solution is to explore network compression techniques or improve the network architecture to reduce its computational and memory overhead while maintaining its performance. Another potential direction for future research could be to investigate the generalizability and robustness of the proposed method across different MRI acquisition protocols and hardware platforms, as well as its clinical applicability for various disease diagnoses and treatment planning.

VI. CONCLUSION

In this paper, we proposed a deep unrolled network for non-Cartesian MRF reconstruction by unrolling the MRF reconstruction model regularized by the tensor low-rank and the Bloch resonance manifold priors. We proposed a learned CP decomposition module to explore the tensor low-rank priors and a Bloch response manifold module to learn the mapping between the reconstructed MRF data and the Bloch response manifold. The proposed network can simultaneously reconstruct high-quality MRF data and multiparameter maps from non-Cartesian undersampled measurements within significantly reduced computational time. However, reconstructing data directly from non-Cartesian samples requires NUFFT operation, which leads to a long network training time and additional memory overhead. To alleviate this problem, we utilize the automatic mixed precision (AMP) training scheme and gradient scaling strategy to improve the computational efficiency while maintaining parameter reconstruction accuracy.

REFERENCES

- [1] D. Ma, V. Gulani, N. Seiberlich, K. Liu, J. L. Sunshine *et al.*, "Magnetic resonance fingerprinting," *Nature*, vol. 495, no. 7440, pp. 187–192, 2013.
- [2] C. Tippareddy, W. Zhao, J. L. Sunshine, M. Griswold, D. Ma *et al.*, "Magnetic resonance fingerprinting: an overview," *European Journal of Nuclear Medicine and Molecular Imaging*, pp. 1–12, 2021.
- [3] M. Davies, G. Puy, P. Vandergheynst, and Y. Wiaux, "A compressed sensing framework for magnetic resonance fingerprinting," *Siam journal on imaging sciences*, vol. 7, no. 4, pp. 2623–2656, 2014.
- [4] B. Zhao, K. Setsompop, H. Ye, S. F. Cauley, and L. L. Wald, "Maximum likelihood reconstruction for magnetic resonance fingerprinting," *IEEE transactions on medical imaging*, vol. 35, no. 8, pp. 1812–1823, 2016.
- [5] G. Mazor, L. Weizman, A. Tal, and Y. C. Eldar, "Low-rank magnetic resonance fingerprinting," *Medical physics*, vol. 45, no. 9, pp. 4066–4084, 2018.
- [6] M. Yang, D. Ma, Y. Jiang, J. Hamilton, N. Seiberlich *et al.*, "Low rank approximation methods for MR fingerprinting with large scale dictionaries," *Magnetic resonance in medicine*, vol. 79, no. 4, pp. 2392–2400, 2018.
- [7] B. Zhao, K. Setsompop, E. Adalsteinsson, B. Gagoski, H. Ye *et al.*, "Improved magnetic resonance fingerprinting reconstruction with low-rank and subspace modeling," *Magnetic resonance in medicine*, vol. 79, no. 2, pp. 933–942, 2018.
- [8] B. Zhao, K. Setsompop, D. Salat, and L. L. Wald, "Further Development of Subspace Imaging to Magnetic Resonance Fingerprinting: A Low-rank Tensor Approach," in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE, 2020, pp. 1662–1666.
- [9] A. Bustin, G. Lima da Cruz, O. Jaubert, K. Lopez, R. M. Botnar *et al.*, "High-dimensionality undersampled patch-based reconstruction (HD-PROST) for accelerated multi-contrast MRI," *Magnetic resonance in medicine*, vol. 81, no. 6, pp. 3705–3719, 2019.
- [10] G. Lima da Cruz, A. Bustin, O. Jaubert, T. Schneider, R. M. Botnar *et al.*, "Sparsity and locally low rank regularization for MR fingerprinting," *Magnetic resonance in medicine*, vol. 81, no. 6, pp. 3530–3543, 2019.
- [11] Y. Hu, P. Li, H. Chen, L. Zou, and H. Wang, "High-Quality MR Fingerprinting Reconstruction Using Structured Low-Rank Matrix Completion and Subspace Projection," *IEEE Transactions on Medical Imaging*, vol. 41, no. 5, pp. 1150–1164, 2021.
- [12] O. Cohen, B. Zhu, and M. S. Rosen, "MR fingerprinting deep reconstruction network (DRONE)," *Magnetic resonance in medicine*, vol. 80, no. 3, pp. 885–894, 2018.
- [13] I. Oksuz, G. Cruz, J. Clough, A. Bustin, N. Fuin *et al.*, "Magnetic resonance fingerprinting using recurrent neural networks," in *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*. IEEE, 2019, pp. 1537–1540.
- [14] Z. Fang, Y. Chen, M. Liu, L. Xiang, Q. Zhang *et al.*, "Deep learning for fast and spatially constrained tissue quantification from highly accelerated data in magnetic resonance fingerprinting," *IEEE transactions on medical imaging*, vol. 38, no. 10, pp. 2364–2374, 2019.

- [15] R. Soyak, E. Navruz, E. O. Ersoy, G. Cruz, C. Prieto *et al.*, “Channel Attention Networks for Robust MR Fingerprint Matching,” *IEEE Transactions on Biomedical Engineering*, vol. 69, no. 4, pp. 1398–1405, 2021.
- [16] H. K. Aggarwal, M. P. Mani, and M. Jacob, “MoDL: Model-based deep learning architecture for inverse problems,” *IEEE transactions on medical imaging*, vol. 38, no. 2, pp. 394–405, 2018.
- [17] Z. Chen, Y. Chen, Y. Xie, D. Li, and A. G. Christodoulou, “Data-Consistent non-Cartesian deep subspace learning for efficient dynamic MR image reconstruction,” in *2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI)*. IEEE, 2022, pp. 1–5.
- [18] 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 Transactions on Medical Imaging*, 2022.
- [19] V. Monga, Y. Li, and Y. C. Eldar, “Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing,” *IEEE Signal Processing Magazine*, vol. 38, no. 2, pp. 18–44, 2021.
- [20] K. Gregor and Y. LeCun, “Learning fast approximations of sparse coding,” in *Proceedings of the 27th international conference on international conference on machine learning*, 2010, pp. 399–406.
- [21] S. Wang, T. Xiao, Q. Liu, and H. Zheng, “Deep learning for fast MR imaging: a review for learning reconstruction from incomplete k-space data,” *Biomedical Signal Processing and Control*, vol. 68, p. 102579, 2021.
- [22] J. Schlemper, J. Caballero, J. V. Hajnal, A. N. Price, and D. Rueckert, “A Deep Cascade of Convolutional Neural Networks for Dynamic MR Image Reconstruction,” *IEEE Transactions on Medical Imaging*, vol. 37, no. 2, pp. 491–503, 2018.
- [23] Z. Ke, W. Huang, Z.-X. Cui, J. Cheng, S. Jia *et al.*, “Learned low-rank priors in dynamic MR imaging,” *IEEE Transactions on Medical Imaging*, vol. 40, no. 12, pp. 3698–3710, 2021.
- [24] D. Chen, M. E. Davies, and M. Golbabaei, “Deep Unrolling for Magnetic Resonance Fingerprinting,” in *2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI)*. IEEE, 2022, pp. 1–4.
- [25] T. G. Kolda and B. W. Bader, “Tensor decompositions and applications,” *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.
- [26] B. Yaman, S. Weingärtner, N. Kargas, N. D. Sidiropoulos, and M. Akçakaya, “Low-rank tensor models for improved multidimensional MRI: Application to dynamic cardiac t_1 mapping,” *IEEE transactions on computational imaging*, vol. 6, pp. 194–207, 2019.
- [27] J. He, Q. Liu, A. G. Christodoulou, C. Ma, F. Lam *et al.*, “Accelerated high-dimensional MR imaging with sparse sampling using low-rank tensors,” *IEEE transactions on medical imaging*, vol. 35, no. 9, pp. 2119–2129, 2016.
- [28] C. Ma, B. Clifford, Y. Liu, Y. Gu, F. Lam *et al.*, “High-resolution dynamic 31P-MRSI using a low-rank tensor model,” *Magnetic resonance in medicine*, vol. 78, no. 2, pp. 419–428, 2017.
- [29] J.-L. Wang, T.-Z. Huang, X.-L. Zhao, T.-X. Jiang, and M. K. Ng, “Multi-dimensional visual data completion via low-rank tensor representation under coupled transform,” *IEEE Transactions on Image Processing*, vol. 30, pp. 3581–3596, 2021.
- [30] H. Kiers, “Towards a standardized notation and terminology in multiway analysis,” *Journal of Chemometrics*, vol. 14, no. 3, pp. 105–122, 2000.
- [31] L. Tucker, “Some mathematical notes on three-mode factor analysis,” *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966.
- [32] A. G. Christodoulou, J. L. Shaw, C. Nguyen, Q. Yang, Y. Xie *et al.*, “Magnetic resonance multitasking for motion-resolved quantitative cardiovascular imaging,” *Nature biomedical engineering*, vol. 2, no. 4, pp. 215–226, 2018.
- [33] J. L. Shaw, Q. Yang, Z. Zhou, Z. Deng, C. Nguyen *et al.*, “Free-breathing, non-ECG, continuous myocardial T1 mapping with cardiovascular magnetic resonance multitasking,” *Magnetic resonance in medicine*, vol. 81, no. 4, pp. 2450–2463, 2019.
- [34] D. F. McGivney, R. Boyacioglu, Y. Jiang, M. E. Poorman, N. Seiberlich *et al.*, “Magnetic resonance fingerprinting review part 2: Technique and directions,” *Journal of Magnetic Resonance Imaging*, vol. 51, no. 4, pp. 993–1007, 2020.
- [35] S. Sra, S. Nowozin, and S. J. Wright, *Optimization for machine learning*. Mit Press, 2012.
- [36] M. J. Muckley, R. Stern, T. Murrell, and F. Knoll, “TorchKbNufft: a high-level, hardware-agnostic non-uniform fast Fourier transform,” in *ISMRM Workshop on Data Sampling & Image Reconstruction*, 2020.
- [37] D. O. Walsh, A. F. Gmitro, and M. W. Marcellin, “Adaptive reconstruction of phased array MR imagery,” *Magnetic Resonance in Medicine*, vol. 43, no. 5, pp. 682–690, 2000.
- [38] J. G. Pipe and P. Menon, “Sampling density compensation in MRI: rationale and an iterative numerical solution,” *Magnetic Resonance in Medicine*, vol. 41, no. 1, pp. 179–186, 1999.
- [39] F. Liu, L. Feng, and R. Kijowski, “MANTIS: model-augmented neural network with incoherent k-space sampling for efficient MR parameter mapping,” *Magnetic resonance in medicine*, vol. 82, no. 1, pp. 174–188, 2019.
- [40] J. I. Hamilton, “A self-supervised deep learning reconstruction for shortening the breathhold and acquisition window in cardiac magnetic resonance fingerprinting,” *Frontiers in Cardiovascular Medicine*, p. 1665, 2022.
- [41] S. Zhang, L. Wang, L. Zhang, and H. Huang, “Learning tensor low-rank prior for hyperspectral image reconstruction,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 006–12 015.
- [42] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, “Fast and accurate deep network learning by exponential linear units (elus),” *arXiv preprint arXiv:1511.07289*, 2015.
- [43] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [44] D. L. Collins, A. P. Zijdenbos, V. Kollokian, J. G. Sled, N. J. Kabani *et al.*, “Design and construction of a realistic digital brain phantom,” *IEEE transactions on medical imaging*, vol. 17, no. 3, pp. 463–468, 1998.
- [45] P. Micikevicius, S. Narang, and *et al.*, “Mixed precision training,” *arXiv preprint arXiv:1710.03740*, 2017.
- [46] N. Mellempudi, S. Srinivasan, D. Das, and B. Kaul, “Mixed precision training with 8-bit floating point,” *arXiv preprint arXiv:1905.12334*, 2019.
- [47] P. E. McKnight and J. Najab, “Kruskal-wallis test,” *The corsini encyclopedia of psychology*, pp. 1–1, 2010.
- [48] S. G. Lingala, Y. Hu, E. DiBella, and M. Jacob, “Accelerated dynamic MRI exploiting sparsity and low-rank structure: kt SLR,” *IEEE transactions on medical imaging*, vol. 30, no. 5, pp. 1042–1054, 2011.
- [49] B. Zhao, J. P. Halder, A. G. Christodoulou, and Z.-P. Liang, “Image reconstruction from highly undersampled (k, t)-space data with joint partial separability and sparsity constraints,” *IEEE transactions on medical imaging*, vol. 31, no. 9, pp. 1809–1820, 2012.