

# A Complex Quasi-Newton Proximal Method for Image Reconstruction in Compressed Sensing MRI

Tao Hong, Luis Hernandez-Garcia, and Jeffrey A. Fessler, *Fellow, IEEE*

**Abstract**—Model-based methods are widely used for reconstruction in compressed sensing (CS) magnetic resonance imaging (MRI), using regularizers to describe the images of interest. The reconstruction process is equivalent to solving a composite optimization problem. Accelerated proximal methods (APMs) are very popular approaches for such problems. This paper proposes a complex quasi-Newton proximal method (CQNPM) for the wavelet and total variation based CS MRI reconstruction. Compared with APMs, CQNPM requires fewer iterations to converge but needs to compute a more challenging proximal mapping called weighted proximal mapping (WPM). To make CQNPM more practical, we propose efficient methods to solve the related WPM. Numerical experiments on reconstructing non-Cartesian MRI data demonstrate the effectiveness and efficiency of CQNPM.

**Index Terms**—Compressed sensing, magnetic resonance imaging (MRI), non-Cartesian trajectory, sparsity, wavelets, total variation, second-order.

## I. INTRODUCTION

MAGNETIC resonance imaging (MRI) scanners acquire samples of the Fourier transform (known as k-space data) of the image of interest. However, MRI is slow since the speed of acquiring k-space data is limited by many constraints, e.g., hardware, physics, and physiology etc. Improving the acquisition speed is crucial for many MRI applications. Lustig et al. [1] proposed a technique called compressed sensing (CS) MRI that improves the imaging speed significantly. CS MRI allows one to get an image of interest from undersampling data by solving the following composite optimization problem:

$$\mathbf{x}^* = \arg \min_{\mathbf{x} \in \mathbb{C}^N} \underbrace{\frac{1}{2} \|\mathbf{Ax} - \mathbf{y}\|_2^2}_{f(\mathbf{x})} + \lambda h(\mathbf{x}), \quad (1)$$

where  $\mathbf{A} \in \mathbb{C}^{ML \times N}$  denotes the forward model describing a mapping from the latent image  $\mathbf{x}$  to the acquired k-space data  $\mathbf{y} \in \mathbb{C}^{ML}$ ,  $h(\mathbf{x})$  is the regularizer that provides some prior assumptions about  $\mathbf{x}$ ,  $L \geq 1$  denotes the number of coils, and  $\lambda > 0$  is a tradeoff parameter to balance  $f(\mathbf{x})$  and  $h(\mathbf{x})$ . We note that  $\mathbf{A}$  consists of  $L$  different submatrices  $\mathbf{A}_l = \Phi \mathbf{F} \mathbf{S}_l \in \mathbb{C}^{M \times N}$  for  $l = 1, 2, \dots, L$ , where  $\Phi$  denotes the downsampling mask,  $\mathbf{F}$  represents the nonuniform fast Fourier transform that depends on the sampling trajectory, and  $\mathbf{S}_l$  is a

T. Hong and L. Hernandez-Garcia are with the Department of Radiology, University of Michigan, Ann Arbor, MI 48109, USA (Email: {tahong, hernan}@umich.edu).

J. Fessler is with the Department of Electrical and Computer Engineering, University of Michigan, Ann Arbor, MI 48109, USA (Email: fessler@umich.edu).

diagonal matrix involving the sensitivity map for the  $l$ th coil which differs for each scan.

Sparsity plays a key role in the success of CS MRI. In general, MR images are not sparse but they can be sparsely represented under some transforms, e.g., total variation (TV) [1], wavelets [2], and transform-learning [3] etc. Recently, more advanced priors or frameworks were introduced for CS MRI reconstruction, such as low-rank [4], plug and play [5, 6], model-based deep learning [7], score-based generative models [8], to name a few. Although deep learning based reconstruction methods have shown better performance than classical priors like TV and wavelet when trained with sufficient data, Gu et al. [9] recently found that suitably trained wavelet regularizers can also achieve comparable performance, demonstrating the power of the classical regularizers. Following Lustig et al. work in [1], we consider both wavelet and TV regularizers for CS MRI reconstruction, i.e., we address the following composite minimization problem for image reconstruction in CS MRI:

$$\mathbf{x}^* = \arg \min_{\mathbf{x} \in \mathbb{C}^N} \frac{1}{2} \|\mathbf{Ax} - \mathbf{y}\|_2^2 + \lambda [\alpha \|\mathbf{T}\mathbf{x}\|_1 + (1 - \alpha) \text{TV}(\mathbf{x})], \quad (2)$$

where  $\mathbf{T}$  and  $\|\cdot\|_1$  denote a general wavelet transform and  $\ell_1$  norm,  $\text{TV}(\cdot)$  represents the TV function (see definition in Section II-B), and  $\alpha \in [0, 1]$  is used to balance the wavelet and TV terms. For  $\alpha = 1$  (respectively,  $\alpha = 0$ ), (2) becomes the wavelet (respectively, TV) based CS MRI reconstruction. Since  $\ell_1$  and TV functions are nonsmooth, accelerated proximal methods (APMs) [10], which have the optimal convergence rate  $O(1/k^2)$  where  $k$  is the number of iterations, are very popular algorithms for (2). In [11], Beck et al. proposed a fast iterative shrinkage-thresholding algorithm (FISTA) (a specific type of APMs) for wavelet-based image reconstruction and showed a closed-form solution for the related proximal mapping [10]. Beck et al. [12] extended FISTA to solve TV-based image reconstruction and suggested a fast dual gradient descent method to compute the proximal mapping. Primal-dual methods [13] are also appealing methods for composite problems. The work in [14] showed that primal-dual methods can also achieve the optimal convergence rate and showed their connection to proximal methods. However, primal-dual methods have to tune parameters that affect the practical convergence rates and such tuning is nontrivial. For a review of different variants of primal-dual methods, see [15]. For more optimization methods and the use of different regularizers for reconstruction in CS MRI, see [16].

Modern MR images are typically acquired using multiple receiver coils and non-Cartesian trajectories, resulting in an

expensive forward process from the image to the k-space domains and ill-conditioned or under-determined  $\mathbf{A}$  [16]. An ill-conditioned  $\mathbf{A}$  can lead to slow reconstruction [17]. To accelerate the recovery process, some preconditioning techniques have been introduced. In [18], Ong et al. proposed a diagonal matrix  $\tilde{\mathbf{D}}$  as a preconditioner such that they solved the following problem instead of (1):

$$\mathbf{x}^* = \arg \min_{\mathbf{x} \in \mathbb{C}^N} \frac{1}{2} \|\tilde{\mathbf{D}}^{\frac{1}{2}}(\mathbf{Ax} - \mathbf{y})\|_2^2 + \lambda h(\mathbf{x}). \quad (3)$$

Recently, Iyer et al. [17] developed more effective polynomial preconditioners than  $\tilde{\mathbf{D}}$ , based on Chebyshev polynomials. Although [17] showed promising results for practical reconstruction, adding such a preconditioner changes the incoherence of  $\mathbf{A}$ , which breaks the original theoretical guarantee. For  $\alpha \in (0, 1)$ , both wavelet and TV are used as regularizers. When we have two nonsmooth terms, the alternating direction method of multipliers (ADMM) [19] is one of the appealing approaches. However, ADMM only provides linear convergence rate  $O(1/k)$  [20] and the computation in each iteration is high because we need to solve a least square problem. In [21, 23], the authors proposed several preconditioning methods to solve the least square problem quickly, which reduces the computation time of the whole reconstruction significantly.

Similar to the quasi-Newton methods for smooth minimization problems [24], the authors in [25, 26] developed quasi-Newton proximal methods (QNPMs) that use second-order information for solving composite problems when  $\mathbf{x} \in \mathbb{R}^N$ . Compared with APMs, QNPMs need fewer iterations to converge which is appealing for problems when computing the gradient  $\nabla f(\mathbf{x})$  is expensive. Indeed, the authors in [27, 29] applied QNPMs to solving the RED model and the TV based inverse-scattering and X-ray reconstruction and observed faster convergence than APMs. However, QNPMs require computing a weighted proximal mapping (WPM), defined in [6], that needs more computation than computing proximal mapping in APMs. So often QNPMs are impractical for many real applications. To compute the WPM, Kadu et al. [28] applied primal-dual methods. Alternatively, Ge et al. [29] treated the WPM as a TV based image deblurring problem and computed the WPM with APMs [12]. Those methods require inner and outer (i.e., two layers) iterations to compute the WPM, making them inefficient. Similar to QNPMs, the variable metric operator splitting methods (VMOSMs) [30] introduce new metrics to accelerate the proximal methods. For a discussion of the differences between QNPMs and VMOSMs, see the prior work section in [26].

The primary contribution of this paper lies in two significant advancements. Firstly, we expand QNPMs to address (1) for complex  $\mathbf{x}$  (Recall that reconstructed MRI images are inherently complex [31] and in some applications the image phase itself is useful, e.g., high-field MRI [32] and quantitative susceptibility mapping [33]). This is achieved by introducing a symmetric rank-1 method in the complex plane to approximate the Hessian matrix of  $f(\mathbf{x})$ , which we called complex quasi-Newton proximal methods (CQNPMs). Secondly, we propose efficient approaches to compute the WPM. Notably, the computational needs of CQNPMs align

closely with the proximal mapping in APMs for wavelet and/or TV-based reconstructions. Our numerical experiments on wavelet and TV based CS MRI reconstruction show that CQNPMs converge faster than APMs in terms of iterations and CPU time, demonstrating the potential advantage of CQNPMs for practical applications.

The rest of this paper is organized as follows. Section II first defines some notation and then reviews the formulation of the discretized TV function and the definition of WPM. Section III derives our algorithm. Section IV reports numerical experiments on the wavelet and TV based CS MRI reconstruction. Section V presents some conclusions and future work.

## II. PRELIMINARIES

This section first defines some notation that simplifies the following discussion and then describes the discretized TV functions. Finally, we define the WPM that generalizes the well-known proximal mapping.

### A. Notation

- Denote by  $\mathbf{X} \in \mathbb{C}^{I \times J}$  the matrix form of  $\mathbf{x} \in \mathbb{C}^N$  with relation  $\mathbf{x} = \text{vec}(\mathbf{X})$  and  $\mathbf{X} = \text{mat}(\mathbf{x})$  where  $\text{vec}(\cdot)$  denotes a column-stacking operator and  $\text{mat}(\cdot)$  is an operator to reshape a vector to its matrix form.
- The  $(i, j)$ th (respectively,  $n$ th) element of a matrix  $\mathbf{X} \in \mathbb{C}^{I \times J}$  (respectively, vector  $\mathbf{x} \in \mathbb{C}^N$ ) is represented as  $\mathbf{X}_{i,j}$  (respectively,  $\mathbf{x}_n$ ).
- $\mathcal{P}_1$  denotes the set of matrix-pairs  $(\mathbf{P}, \mathbf{Q})$  where  $\mathbf{P} \in \mathbb{C}^{(I-1) \times J}$  and  $\mathbf{Q} \in \mathbb{C}^{I \times (J-1)}$  satisfy

$$\begin{aligned} |\mathbf{P}_{i,j}|^2 + |\mathbf{Q}_{i,j}|^2 &\leq 1, \quad i = 1, \dots, I-1, \quad j = 1, \dots, J-1, \\ |\mathbf{P}_{i,J}| &\leq 1, \quad i = 1, \dots, I-1, \\ |\mathbf{Q}_{i,j}| &\leq 1, \quad J = 1, \dots, J-1. \end{aligned}$$

- $\mathcal{P}_2$  is the set of matrix-pairs  $(\mathbf{P}, \mathbf{Q})$  where  $\mathbf{P} \in \mathbb{C}^{(I-1) \times J}$  and  $\mathbf{Q} \in \mathbb{C}^{I \times (J-1)}$  satisfy  $|\mathbf{P}_{i,j}| \leq 1$ ,  $|\mathbf{Q}_{i,j}| \leq 1, \forall i, j$ .
- $\mathcal{Z}$  is the set of vectors  $\mathbf{z} \in \mathbb{C}^N$  such that  $|\mathbf{z}_n| \leq 1, \forall n$ .
- $\mathcal{L} : \mathbb{C}^{(I-1) \times J} \times \mathbb{C}^{I \times (J-1)} \rightarrow \mathbb{C}^{I \times J}$  denotes a linear operator that satisfies

$$\mathcal{L}(\mathbf{P}, \mathbf{Q})_{i,j} = \mathbf{P}_{i,j} + \mathbf{Q}_{i,j} - \mathbf{P}_{i-1,j} - \mathbf{Q}_{i,j-1}, \quad \forall i, j,$$

where we assume that  $\mathbf{P}_{0,j} = \mathbf{P}_{I,j} = \mathbf{Q}_{i,0} = \mathbf{Q}_{i,J} = 0, \forall i, j$ .

- The adjoint operator of  $\mathcal{L} : \mathbb{C}^{I \times J} \rightarrow \mathbb{C}^{(I-1) \times J} \times \mathbb{C}^{I \times (J-1)}$  is

$$\mathcal{L}^T(\mathbf{X}) = (\mathbf{P}, \mathbf{Q}),$$

where  $\mathbf{P} \in \mathbb{C}^{(I-1) \times J}$  and  $\mathbf{Q} \in \mathbb{C}^{I \times (J-1)}$  are the matrix pairs that satisfy

$$\begin{aligned} \mathbf{P}_{i,j} &= \mathbf{X}_{i,j} - \mathbf{X}_{i+1,j}, \quad i = 1, \dots, I-1, \quad j = 1, \dots, J, \\ \mathbf{Q}_{i,j} &= \mathbf{X}_{i,j} - \mathbf{X}_{i,j+1}, \quad i = 1, \dots, I, \quad j = 1, \dots, J-1. \end{aligned}$$

### B. Discretized Total Variation

Assuming zero Neumann boundary conditions for an image  $\mathbf{X} \in \mathbb{C}^{I \times J}$ , i.e.,

$$\mathbf{X}_{I+1,j} - \mathbf{X}_{I,j} = 0, \quad \forall j \text{ and } \mathbf{X}_{i,J+1} - \mathbf{X}_{i,J} = 0, \quad \forall i,$$

the isotropic and anisotropic TV functions are defined as follows

$$\begin{aligned} \text{TV}_{\text{iso}}(\mathbf{X}) = & \sum_{i=1}^{I-1} \sum_{j=1}^{J-1} \sqrt{(\mathbf{X}_{i,j} - \mathbf{X}_{i+1,j})^2 + (\mathbf{X}_{i,j} - \mathbf{X}_{i,j+1})^2} \\ & + \sum_{i=1}^{I-1} |\mathbf{X}_{i,J} - \mathbf{X}_{i+1,J}| + \sum_{j=1}^{J-1} |\mathbf{X}_{I,j} - \mathbf{X}_{I,j+1}|, \end{aligned} \quad (4)$$

and

$$\begin{aligned} \text{TV}_{\ell_1}(\mathbf{X}) = & \sum_{i=1}^{I-1} \sum_{j=1}^{J-1} \left\{ |\mathbf{X}_{i,j} - \mathbf{X}_{i+1,j}| + |\mathbf{X}_{i,j} - \mathbf{X}_{i,j+1}| \right\} \\ & + \sum_{i=1}^{I-1} |\mathbf{X}_{i,J} - \mathbf{X}_{i+1,J}| + \sum_{j=1}^{J-1} |\mathbf{X}_{I,j} - \mathbf{X}_{I,j+1}|, \end{aligned} \quad (5)$$

respectively. Hereafter, we use  $\text{TV}(\mathbf{x})$  to represent either  $\text{TV}_{\text{iso}}(\mathbf{X})$  or  $\text{TV}_{\ell_1}(\mathbf{X})$ .

### C. Weighted Proximal Mapping

Given a proper closed convex function  $h(\mathbf{x})$  and a Hermitian positive definite matrix  $\mathbf{W} \succ 0 : \in \mathbb{C}^{N \times N}$ , the WPM associated to  $h$  is defined as

$$\text{prox}_h^{\mathbf{W}}(\mathbf{x}) = \arg \min_{\mathbf{u}} \left( h(\mathbf{u}) + \frac{1}{2} \|\mathbf{u} - \mathbf{x}\|_{\mathbf{W}}^2 \right), \quad (6)$$

where  $\|\cdot\|_{\mathbf{W}}$  denotes the  $\mathbf{W}$ -norm defined by  $\|\mathbf{q}\|_{\mathbf{W}} = \sqrt{\mathbf{q}^H \mathbf{W} \mathbf{q}}$ . Here  $\mathcal{H}$  denotes Hermitian transpose. Clearly, (6) simplifies to the proximal mapping for  $\mathbf{W} = \mathbf{I}_N$  where  $\mathbf{I}_N$  represents the identity matrix. Since  $h(\mathbf{u}) + \frac{1}{2} \|\mathbf{u} - \mathbf{x}\|_{\mathbf{W}}^2$  is strongly convex,  $\text{prox}_h^{\mathbf{W}}(\mathbf{x})$  exists and is unique for  $\mathbf{x} \in \text{dom} h$  so that the WPM is well defined.

## III. COMPLEX QUASI-NEWTON PROXIMAL METHODS

This section first describes a complex quasi-Newton proximal method (CQNPM) for solving (1) with regularizer  $h(\mathbf{x}) = \alpha \|\mathbf{T}\mathbf{x}\|_1 + (1 - \alpha)\text{TV}(\mathbf{x})$  and  $\alpha \in [0, 1]$ . Here, we consider  $\mathbf{T} \in \mathbb{C}^{N \times N}$  to be a wavelet transform. Then, we propose efficient methods to compute the related WPM. Moreover, to avoid applying wavelet transforms when computing the WPM for  $\alpha \in [0, 1]$ , we propose a partial smooth approach. Our numerical experiments show that such a partial smooth strategy recovers the desired images with less computation.

At  $k$ th iteration, CQNPM solves (7) for  $\mathbf{x}_{k+1}$ ,

$$\begin{aligned} \mathbf{x}_{k+1} = & \arg \min_{\mathbf{x} \in \mathbb{C}^N} f(\mathbf{x}_k) + \langle \nabla f(\mathbf{x}_k), \mathbf{x} - \mathbf{x}_k \rangle + \frac{1}{2a_k} \|\mathbf{x} - \mathbf{x}_k\|_{\mathbf{B}_k}^2 \\ & + \lambda h(\mathbf{x}) \\ = & \text{prox}_{a_k \lambda h}^{\mathbf{B}_k}(\mathbf{x}_k - a_k \mathbf{B}_k^{-1} \nabla_{\mathbf{x}} f(\mathbf{x}_k)), \end{aligned} \quad (7)$$

where  $a_k$  is the step-size and  $\mathbf{B}_k \in \mathbb{C}^{N \times N}$  is a Hermitian symmetric positive definite matrix. For clarity, we present the detailed steps of CQNPM in Algorithm 1. Note that Algorithm 1 would be identical to the proximal methods [10] if one chose  $\mathbf{B}_k = \mathbf{I}_N$ . In [30, 34], the authors suggested using a diagonal matrix  $\mathbf{B}_k$  for their application. However, building such a diagonal matrix is nontrivial and its effectiveness is problem dependent. In this paper, we choose  $\mathbf{B}_k$  to be a more

---

**Algorithm 1** Proposed complex quasi-Newton proximal method.

---

**Initialization:**  $\mathbf{x}_1$ .

**Iteration:**

- 1: **for**  $k = 1, 2, \dots$  **do**
  - 2: pick the step-size  $a_k$  and the weighting  $\mathbf{B}_k$ .
  - 3:  $\mathbf{x}_{k+1} \leftarrow \text{prox}_{a_k \lambda h}^{\mathbf{B}_k}(\mathbf{x}_k - a_k \mathbf{B}_k^{-1} \nabla_{\mathbf{x}} f(\mathbf{x}_k))$ .
  - 4: **end for**
- 

**Algorithm 2** SR1 updating.

**Initialization:**  $\gamma > 1$ ,  $\delta = 10^{-8}$ ,  $\Xi > 0$  a fixed real scalar,  $\mathbf{x}_k$ ,  $\mathbf{x}_{k-1}$ ,  $\nabla f(\mathbf{x}_k)$ , and  $\nabla f(\mathbf{x}_{k-1})$ .

- 1: **if**  $k = 1$  **then**
  - 2:  $\mathbf{B}_k \leftarrow \Xi \mathbf{I}$ .
  - 3: **else**
  - 4: Set  $\mathbf{s}_k \leftarrow \mathbf{x}_k - \mathbf{x}_{k-1}$  and  $\mathbf{m}_k \leftarrow \nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}_{k-1})$ .
  - 5: Compute  $\tau_k \leftarrow \gamma \frac{\|\mathbf{m}_k\|_2^2}{\langle \mathbf{s}_k, \mathbf{m}_k \rangle}$ . %  $\langle \mathbf{a}, \mathbf{b} \rangle = \mathbf{b}^H \mathbf{a}$
  - 6: **if**  $\tau < 0$  **then**
  - 7:  $\mathbf{B}_k \leftarrow \Xi \mathbf{I}$ .
  - 8: **else**
  - 9:  $\mathbf{H}_0 \leftarrow \tau_k \mathbf{I}$ .
  - 10:  $\mathbf{u}_k \leftarrow \mathbf{m}_k - \mathbf{H}_0 \mathbf{s}_k$ .
  - 11: **if**  $|\langle \mathbf{u}_k, \mathbf{s}_k \rangle| \leq \delta \|\mathbf{s}_k\|_2 \|\mathbf{u}_k\|_2$  **then**
  - 12:  $\mathbf{u}_k \leftarrow \mathbf{0}$ .
  - 13: **end if**
  - 14:  $\mathbf{B}_k \leftarrow \mathbf{H}_0 + \frac{\mathbf{u}_k \mathbf{u}_k^H}{\langle \mathbf{m}_k - \mathbf{H}_0 \mathbf{s}_k, \mathbf{s}_k \rangle}$ .
  - 15: **end if**
  - 16: **end if**
  - 17: **Return:**  $\mathbf{B}_k$
- 

accurate approximation of the Hessian of  $f(\mathbf{x})$ . Specifically, we select  $\mathbf{B}_k$  based on the Symmetric Rank-1 (SR1) method [24], a popular method used in quasi-Newton methods for approximating a Hessian matrix. Following the derivation of SR1 for real variables, we derive a complex plane SR1 that is similar to the real one. Algorithm 2 presents the implementation details of SR1 in the complex plane. We found that using  $\gamma > 1$  is crucial to ensure that  $\mathbf{B}_k$  is Hermitian positive definite in our setting because otherwise  $\langle \mathbf{m}_k - \mathbf{H}_0 \mathbf{s}_k, \mathbf{s}_k \rangle$  can become negative, causing  $\mathbf{B}_k$  to turn indefinite. In our numerical experiments, we found that a fixed  $\gamma > 1$  worked well.

### A. Compute Weighted Proximal Mapping

The dominant computation in Algorithm 1 is computing the WPM at Step 3 which could be as hard as solving (1) for a general  $\mathbf{B}_k$ . However, we find one can compute  $\text{prox}_{a_k \lambda h}^{\mathbf{B}_k}(\cdot)$  as easily as the case when  $\mathbf{B}_k = \mathbf{I}_N$  by using the structure of  $\mathbf{B}_k$ .

To compute the WPM  $\text{prox}_{\lambda h}^{\mathbf{B}_k}(\mathbf{v}_k)$  at  $k$ th iteration, we need to solve the following problem

$$\min_{\mathbf{x} \in \mathbb{C}^N} \|\mathbf{x} - \mathbf{v}_k\|_{\mathbf{B}_k}^2 + 2\bar{\lambda} [\alpha \|\mathbf{T}\mathbf{x}\|_1 + (1 - \alpha)\text{TV}(\mathbf{x})], \quad (8)$$

where  $\mathbf{v}_k = \mathbf{x}_k - a_k \mathbf{B}_k^{-1} \nabla_{\mathbf{x}} f(\mathbf{x}_k)$  and  $\bar{\lambda} = a_k \lambda$ . A difficulty of (8) is the nonsmoothness of  $\|\cdot\|_1$  and  $\text{TV}(\cdot)$ . To address this difficulty, we consider a dual approach for (8) that is similar to Chambolle's approach for TV-based image reconstruction

[35]. Our method only uses one inner iteration to compute the WPM, and the related gradient is computed easily. Proposition 1 describes the dual problem of (8) and the relation between the primal and dual optimal solutions.

**Proposition 1.** Let

$$(\mathbf{z}^*, \mathbf{P}^*, \mathbf{Q}^*) = \operatorname{argmin}_{\substack{\mathbf{z} \in \mathcal{Z} \\ (\mathbf{P}, \mathbf{Q}) \in \mathcal{P}}} \|\mathbf{w}_k(\mathbf{z}, \mathbf{P}, \mathbf{Q})\|_{\mathbf{B}_k}^2 \quad (9)$$

where  $\mathbf{w}_k(\mathbf{z}, \mathbf{P}, \mathbf{Q}) = \mathbf{v}_k - \bar{\lambda} \mathbf{B}_k^{-1} (\alpha \mathbf{T}^H \mathbf{z} + (1 - \alpha) \operatorname{vec}(\mathcal{L}(\mathbf{P}, \mathbf{Q})))$  and  $\mathcal{P} = \mathcal{P}_1$  or  $\mathcal{P}_2$  depending on which TV is used. Then the optimal solution of (8) is given by  $\mathbf{x}_{k+1} = \mathbf{w}_k(\mathbf{z}^*, \mathbf{P}^*, \mathbf{Q}^*)$ .

*Proof.* See Section A.  $\square$

Using Proposition 1, we can apply the FISTA [11, 36] to solve (9) for computing  $\operatorname{prox}_{\bar{\lambda} h}^{\mathbf{B}_k}$  since (9) is convex and continuously differentiable. Lemma 1 specifies the corresponding gradient and Lipschitz constant of (9).

**Lemma 1.** The gradient of (19) is

$$-2\bar{\lambda} \begin{bmatrix} \alpha \mathbf{T} \\ (1 - \alpha) \mathcal{L}^T \end{bmatrix} \mathbf{w}_k(\mathbf{z}, \mathbf{P}, \mathbf{Q}) \quad (10)$$

and the corresponding Lipschitz constant is

$$L_c = 2\sigma_{\min} \bar{\lambda}^2 (\alpha^2 \|\mathbf{T}\|^2 + 8(1 - \alpha)^2)$$

where  $\sigma_{\min}$  is the smallest eigenvalue of  $\mathbf{B}_k$ .

*Proof.* See Section B.  $\square$

According to the formulation of  $\mathbf{B}_k$  proposed in Algorithm 2, we can obtain  $\sigma_{\min}$  easily through

$$\sigma_{\min} = \begin{cases} \Xi & \text{if } \tau < 0, \\ \tau & \text{if } \langle \mathbf{m}_k - \mathbf{H}_0 \mathbf{s}_k, \mathbf{s}_k \rangle > 0, \\ \tau + \frac{\mathbf{u}^H \mathbf{u}}{\langle \mathbf{m}_k - \mathbf{H}_0 \mathbf{s}_k, \mathbf{s}_k \rangle} & \text{if } \langle \mathbf{m}_k - \mathbf{H}_0 \mathbf{s}_k, \mathbf{s}_k \rangle < 0. \end{cases}$$

The value of  $\|\mathbf{T}\|$  depends on the choice of wavelets which can be computed in advance, so the computational cost of obtaining the Lipschitz constant of (19) is cheap. For completeness, Algorithm 3 presents the implementation details of FISTA for solving (19). We terminate Algorithm 3 when it reaches a maximal number of iterations or a given accuracy tolerance. The initial value  $(\mathbf{z}_1, \mathbf{P}_1, \mathbf{Q}_1)$  in Algorithm 3 uses the final solution of the previous iteration.

**Remark 1.** Compared with APMs for addressing (1), the additional cost of CQNPM is applying  $\mathbf{B}_k^{-1}$  in computing  $\mathbf{v}_k$  and  $\mathbf{w}_k$  in Algorithms 1 and 3. This inversion can be computed cheaply through the Woodbury matrix identity. Moreover, computing the projectors  $\operatorname{Proj}_{\mathcal{Z}}(\cdot)$  and  $\operatorname{Proj}_{\mathcal{P}}(\cdot)$  is also cheap and identical to the one shown in [12], so we omit the details here. The step-size  $a_k$  in Algorithm 1 can be simply set to be 1.

<sup>1</sup>We note that  $\langle \mathbf{m}_k - \mathbf{H}_0 \mathbf{s}_k, \mathbf{s}_k \rangle$  is real in our setting, see Observation I.

---

### Algorithm 3 FISTA for solving (19).

---

**Initialization:**  $\mathbf{B}_k, \mathbf{v}_k, \bar{\lambda} > 0, \alpha \in [0, 1]$ , Lipschitz constant  $L_c$ , maximal iteration Max\_Iter, tolerance  $\epsilon > 0$ , and initial values  $\mathbf{z}_1, \mathbf{P}_1, \mathbf{Q}_1$ .

**Iteration:**

```

1:  $t_1 \leftarrow 1$ .
2:  $(\bar{\mathbf{z}}_1, \bar{\mathbf{P}}_1, \bar{\mathbf{Q}}_1) \leftarrow (\mathbf{z}_1, \mathbf{P}_1, \mathbf{Q}_1)$ .
3: for  $s = 1, 2, \dots, \text{Max\_Iter}$  do
4:   Compute  $\bar{\mathbf{w}} \leftarrow \mathbf{w}_k(\bar{\mathbf{z}}_s, \bar{\mathbf{P}}_s, \bar{\mathbf{Q}}_s)$ .
5:   if  $\alpha \neq 0$  then
6:      $\mathbf{z}_{s+1} \leftarrow \operatorname{Proj}_{\mathcal{Z}}(\bar{\mathbf{z}}_s + \frac{2\bar{\lambda}\alpha}{L_c} \mathbf{T} \bar{\mathbf{w}})$ .
7:   else
8:     Set  $\mathbf{z}_{s+1}$  empty.
9:   end if
10:  if  $\alpha \neq 1$  then
11:     $(\mathbf{P}_{s+1}, \mathbf{Q}_{s+1}) \leftarrow \operatorname{Proj}_{\mathcal{P}}((\bar{\mathbf{P}}_s, \bar{\mathbf{Q}}_s) + \frac{2\bar{\lambda}(1-\alpha)}{L_c} \mathcal{L}^T \bar{\mathbf{w}})$ .
12:  else
13:    Set  $\mathbf{P}_{s+1}$  and  $\mathbf{Q}_{s+1}$  empty.
14:  end if
15:  if  $\|(\mathbf{z}_{s+1} - \mathbf{z}_s, \mathbf{P}_{s+1} - \mathbf{P}_s, \mathbf{Q}_{s+1} - \mathbf{Q}_s)\| \leq \epsilon$  then
16:    break.
17:  end if
18:   $t_{s+1} \leftarrow \frac{1 + \sqrt{1 + 4t_s^2}}{2}$ .
19:   $(\bar{\mathbf{z}}_{s+1}, \bar{\mathbf{P}}_{s+1}, \bar{\mathbf{Q}}_{s+1}) \leftarrow \frac{t_{s+1} + t_s - 1}{t_{s+1}} (\mathbf{z}_{s+1}, \mathbf{P}_{s+1}, \mathbf{Q}_{s+1}) - \frac{t_s - 1}{t_{s+1}} (\mathbf{z}_s, \mathbf{P}_s, \mathbf{Q}_s)$ .
20:   $t_s \leftarrow t_{s+1}$ .
21: end for

```

---

### B. Compute the Weighted Proximal Mapping when $\alpha = 1$

For  $\alpha = 1$ , running Algorithm 3 to compute the WPM would be inefficient since we would have to apply wavelet transform many times at each outer iteration. However, if  $\mathbf{T}$  is left invertible that  $\mathbf{T}^H \mathbf{T} = \mathbf{I}_N$ , we can solve the following problem instead of (2) to avoid using Algorithm 3 to compute the WPM:

$$\bar{\mathbf{x}}^* = \operatorname{argmin}_{\mathbf{x} \in \mathbb{C}^N} \underbrace{\frac{1}{2} \|\mathbf{A} \mathbf{T}^H \bar{\mathbf{x}} - \mathbf{y}\|_2^2 + \lambda \|\bar{\mathbf{x}}\|_1}_{f(\bar{\mathbf{x}})}. \quad (11)$$

Then the recovered image is  $\mathbf{x}^* = \mathbf{T}^H \bar{\mathbf{x}}^*$ . Now the corresponding WPM becomes

$$\operatorname{prox}_{\bar{\lambda} \|\cdot\|_1}^{\mathbf{B}_k}(\mathbf{v}_k) = \operatorname{argmin}_{\bar{\mathbf{x}} \in \mathbb{C}^N} \|\bar{\mathbf{x}} - \mathbf{v}_k\|_{\mathbf{B}_k}^2 + 2\bar{\lambda} \|\bar{\mathbf{x}}\|_1. \quad (12)$$

Note that (2) and (11) represent the analysis-based and synthetic-based priors, respectively. For a detailed discussion of their relations and equivalence, see [37].

Let  $\mathbf{W} \in \mathbb{C}^{N \times N} := \mathbf{D} \pm \mathbf{u} \mathbf{u}^H$  where  $\mathbf{D} \in \mathbb{R}^{N \times N}$  is a diagonal matrix and  $\mathbf{u} \in \mathbb{C}^N$ . Becker et al. proposed the following theorem that relates  $\operatorname{prox}_{\lambda h}^{\mathbf{W}}(\mathbf{x})$  and  $\operatorname{prox}_{\lambda h}^{\mathbf{D}}(\mathbf{x})$ .

**Theorem 1** (Theorem 3.4, [26]<sup>2</sup>). Let  $\mathbf{W} = \mathbf{D} \pm \mathbf{u} \mathbf{u}^H$ . Then,

$$\operatorname{prox}_{\lambda h}^{\mathbf{W}}(\mathbf{x}) = \operatorname{prox}_{\lambda h}^{\mathbf{D}}(\mathbf{x} \mp \mathbf{D}^{-1} \mathbf{u} \beta^*),$$

<sup>2</sup>The theorem is proved in real plane but it is also valid in complex plane.

where  $\beta^* \in \mathbb{C}$  is the unique zero of the following nonlinear equation

$$\mathbb{J}(\beta) : \mathbf{u}^{\mathcal{H}} \left( \mathbf{x} - \text{prox}_{\lambda h}^D(\mathbf{x} \mp D^{-1} \mathbf{u} \beta) \right) + \beta.$$

Using the notation in Algorithm 2, we have the following observation:

**Observation I.**  $\tau$  and  $\langle \mathbf{m}_k - \mathbf{H}_0 \mathbf{s}_k, \mathbf{s}_k \rangle$  in Algorithm 2 are real.

*Proof.* Note that  $f(\mathbf{x}) = \frac{1}{2} \|\mathbf{Ax} - \mathbf{y}\|_2^2$ . Then we have  $\mathbf{m}_k = \mathbf{A}^{\mathcal{H}} \mathbf{As}_k$ , so  $\langle \mathbf{s}_k, \mathbf{m}_k \rangle$  is real.  $\square$

Since  $\langle \mathbf{m}_k - \mathbf{H}_0 \mathbf{s}_k, \mathbf{s}_k \rangle$  is real, we rewrite  $\mathbf{B}_k$  as

$$\mathbf{B}_k = \mathbf{H}_0 + \text{sgn}(\langle \mathbf{m}_k - \mathbf{H}_0 \mathbf{s}_k, \mathbf{s}_k \rangle) \tilde{\mathbf{u}}_k \tilde{\mathbf{u}}_k^{\mathcal{H}}, \quad (13)$$

where  $\tilde{\mathbf{u}}_k = \frac{\mathbf{u}_k}{\sqrt{\langle \mathbf{m}_k - \mathbf{H}_0 \mathbf{s}_k, \mathbf{s}_k \rangle}}$  and  $\text{sgn}(\cdot)$  denotes the sign function such that  $\mathbf{B}_k$  holds the same structure as  $\mathbf{W}$  in Theorem 1. So, instead of solving (12) directly, we first solve  $\mathbb{J}(\beta) = 0$  and then use Theorem 1 to obtain  $\text{prox}_{\lambda \|\cdot\|_1}^{\mathbf{B}_k}(\mathbf{v}_k)$ . In this paper, we solve  $\mathbb{J}(\beta) = 0$  using “SciPy” library in Python.

### C. Partial Smoothing

For  $\alpha \in (0, 1)$ , Algorithm 3 still requires applying many wavelet transforms, which can dominate the computational cost. An alternative way is to use the idea proposed in [38] where one partially smooths the objective and then applies Algorithm 1. For comparison purposes, we apply Algorithm 1 to the following problem

$$\min_{\mathbf{x} \in \mathbb{C}^N} \underbrace{\frac{1}{2} \|\mathbf{Ax} - \mathbf{y}\|_2^2}_{f(\mathbf{x})} + \underbrace{\lambda \alpha \cdot S^{\eta}(\|\mathbf{T}\mathbf{x}\|_1)}_{h(\mathbf{x})} + \underbrace{\lambda(1-\alpha) \text{TV}(\mathbf{x})}, \quad (14)$$

such that each outer iteration needs only two wavelet transforms. For the comparisons in this paper, we used  $S^{\eta}(\|\mathbf{x}\|_1) = \sum_{n=1}^N \sqrt{\mathbf{x}_n^2 + \eta}$  with  $\eta > 0$  so that  $f(\mathbf{x})$  in (14) is differentiable. Our numerical experiments compare the performance of such a partial smoothing approach to methods based on the original cost function for image reconstruction in CS MRI.<sup>3</sup>

## IV. NUMERICAL EXPERIMENTS

This section studies the performance of our algorithm for image reconstruction in CS MRI with non-Cartesian sampling trajectories. Specifically, we consider the radial and spiral trajectories. Moreover, we also study the robustness of our algorithm to the choice of  $\gamma$  and Max\_Iter in Algorithms 2 and 3 respectively. Similar to [1], we focus on wavelet and TV regularizers. We first present our experimental and algorithmic settings and then show our reconstruction results.

**Experimental Settings:** We took complex k-space data from the brain and knee training datasets (one each) in the NYU fastMRI dataset [39] to generate the simulated k-space data. We applied the ESPIRiT algorithm [40] to recover the complex images and then cropped the images to size  $256 \times 256$  to define the ground-truth images, with maximum magnitude scaled to

<sup>3</sup>One could instead partially smooth the TV regularizer. However, in our settings, we found that smoothing  $\|\mathbf{T}\mathbf{x}\|_1$  led to better quality than TV smoothing.

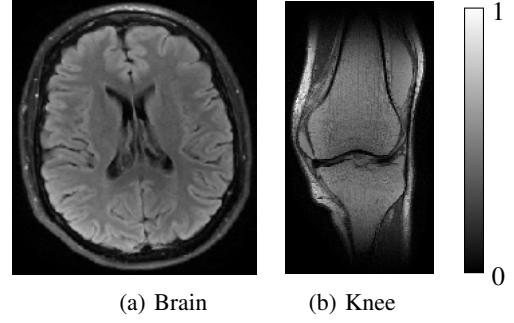


Fig. 1. The magnitude of the complex-valued ground truth images.

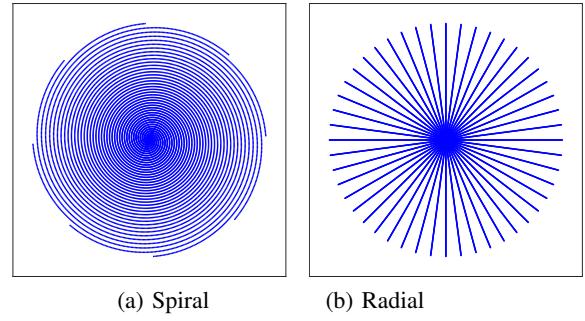


Fig. 2. The non-Cartesian MRI trajectories used in this paper.

one. Figure 1 shows the magnitude of the complex-valued ground-truth images. Following [17], we used 32 interleaves, 1688 readout points, and 12-coils (respectively, 96 radial projections, 512 readout points, and 12 coils) for the spiral (respectively, radial) trajectory to define the forward model  $\mathbf{A}$ . Figure 2 presents the used trajectories in this paper. For clarity, we plot only every 4th sample of the trajectories. Applying the used forward model to the ground truth image generated the noiseless multi-coil k-space data. We added complex i.i.d Gaussian noise with mean zero and variance  $10^{-2}$  to all coils to form the measurements,  $\mathbf{y}$ . The data input SNR was below 7dB. We also studied a higher data input SNR case of around 30dB. Our implementation used Python programming language with SigPy library [41]. The reconstructions ran on a workstation with 2.3GHz AMD EPYC 7402. Our code is available on <https://github.com/hongtao-argmin/CQNPMs-MRIReco>. The supplementary material provides additional experimental results and a comparison with a Plug-and-Play reconstruction method using BM3D and a deep denoiser [42]. **Algorithmic Settings:** For APM, we precomputed the Lipschitz constant for all experiments. For CQNPM, we set  $a_k = 1$  and  $\gamma = 1.7$ . Denote by S-APM (respectively, S-CQNPM) when APM (respectively, CQNPM) is used to solve (14). We chose the step-size in S-APM using a backtracking strategy [43]. Moreover, we also compared our method with primal-dual (PD) methods [44]. The tradeoff parameters  $\lambda$  and  $\alpha$  were chosen to reach the highest peak signal-to-noise ratio (PSNR) when running enough iterations of APM. We set  $\eta = 10^{-5}$  in our experiments. The maximal number of iterations and

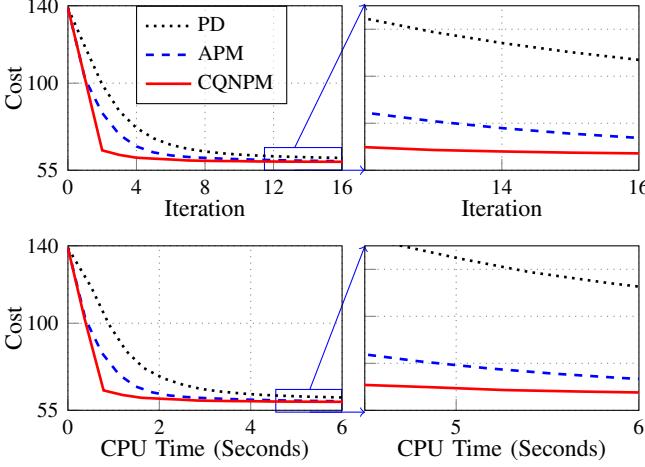


Fig. 3. Cost values versus iteration (top) and CPU time (bottom) of the brain image with regularizer  $h(\mathbf{x}) = \|\mathbf{T}\mathbf{x}\|_1$  and  $\lambda = 5 \times 10^{-4}$  for a left invertible wavelet transform  $\mathbf{T}$  with 5 levels. Acquisition: radial trajectory with 96 projections, 512 readout points, and 12 coils.

tolerance in Algorithm 3 are set to be 20 and  $10^{-6}$  for both CQNPM and APM.

#### A. Radial Acquisition MRI Reconstruction

Figures 3 and 4 show the performance of Algorithm 1 for the wavelet based reconstruction of the brain image and the comparison with APM [12] and PD [44]. Here, we used Theorem 1 to compute the WPM. Clearly, CQNPM converged faster than APM and PD in terms of iterations. Compared with the cost of computing the proximal mapping, the additional cost of computing WPM with our method is insignificant. Figures 3 and 4 show that the computational costs of CQNPM and APM per iteration are similar. The comparison of PSNR versus CPU time in Figure 4 also shows that CQNPM reached a higher PSNR with less CPU time, illustrating the fast convergence of CQNPM. The reconstructed images at 3, 10, 13, and 16th iteration illustrate that CQNPM yielded a clearer image than APM for the same number of iterations. Since PD led to a much lower PSNR than APM, we do not present the reconstructed images of PD. Similar observations apply to the knee image and the related results are provided in the supplemental material.

We also studied the performance of our algorithm when using both wavelet and TV regularizers. Here, we used Algorithm 3 to compute the proximal mapping and WPM. Since ADMM is a classical method for (2) with  $h(\mathbf{x}) = \alpha\|\mathbf{T}\mathbf{x}\|_1 + (1-\alpha)\text{TV}(\mathbf{x})$ , we include a comparison with ADMM. Moreover, we also studied the performance of the partial smoothing technique. Although PD does not require any inner iteration, unlike ADMM, APM, and our method, our method is still faster than PD in terms of iterations and CPU time.

Figures 5 and 6 present the results for the reconstruction of the brain image. CQNPM reduced the cost faster than APM in terms of iterations and CPU time. Although we solved (14) instead of (2) for the partial smoothing method, the cost is still

computed with (2). Surprisingly, in this setting, we see that, for the cost values versus iterations, S-APM (respectively, S-CQNPM) converged similar to APM (respectively, CQNPM) in terms of iterations. However, from the cost values versus CPU time plot, S-CQNPM converged faster than CQNPM, as expected since the partial smoothing method requires only two wavelet transforms per outer iteration. However, S-APM converged slower than APM in terms of CPU time because S-APM requires applying a line search to choose the step-size, increasing the computational cost. Although CQNPM/S-CQNPM require an iterative method to solve the WPM, Section IV-D demonstrated that the WPM can be solved inexactly, and the computation for solving the WPM is relatively inexpensive compared to executing  $\mathbf{Ax}$  in CS MRI reconstruction. Thus, CQNPM/S-CQNPM converged faster than ADMM/PD both in terms of iteration numbers and in CPU time. Note that ADMM requires solving a least-squares problem at each iteration, which involves applying  $\mathbf{Ax}$  multiple times, leading to significantly slower convergence in terms of CPU time.

The PSNR versus CPU time plot in Figure 6 also demonstrates the fast convergence of CQNPM and S-CQNPM. Compared with the previous experiments that only used a wavelet regularizer, we see an improved PSNR here, confirming the benefit of using both wavelet and TV regularizers. The reconstructed images at 3, 10, 13, and 16th iteration for each method<sup>4</sup> illustrate that the partial smoothing method works as well as the nonsmoothing one. In summary, the proposed method converged faster than other methods in terms of iterations and CPU time, and S-CQNPM is the best algorithm for (2) in this setting. We also tested our algorithm on the knee image and provided the results in the supplementary material.

#### B. Spiral Acquisition MRI Reconstruction

This part studies the reconstruction with spiral acquisition that used 32 interleaves, 1688 readout points, and 12 coils. Figures 9 and 10 show the results of the knee image with wavelet and TV regularizers. The trends are similar to the radial acquisition case. Note that CQNPM reduced the cost values faster than S-CQNPM in terms of iterations and CPU time in this setting. However, S-CQNPM reached a higher PSNR than CQNPM with same CPU time. We provided the reconstruction of the brain and knee images with wavelet regularizer and the brain image with wavelet and TV regularizers in the supplementary material.

#### C. The Choice of $\gamma$

We tried several different  $\gamma$  values to study how  $\gamma$  affects the convergence of CQNPM. We reconstructed the brain image with wavelet and TV regularizers and radial acquisition. Figure 7 presents the results that show that CQNPM is quite robust to different  $\gamma$  values, and  $\gamma = 1.7$  worked slightly better than the others. So we simply set  $\gamma = 1.7$  for all experiments.

<sup>4</sup>We do not show the reconstructed image of ADMM since it yielded a much lower PSNR than other methods.

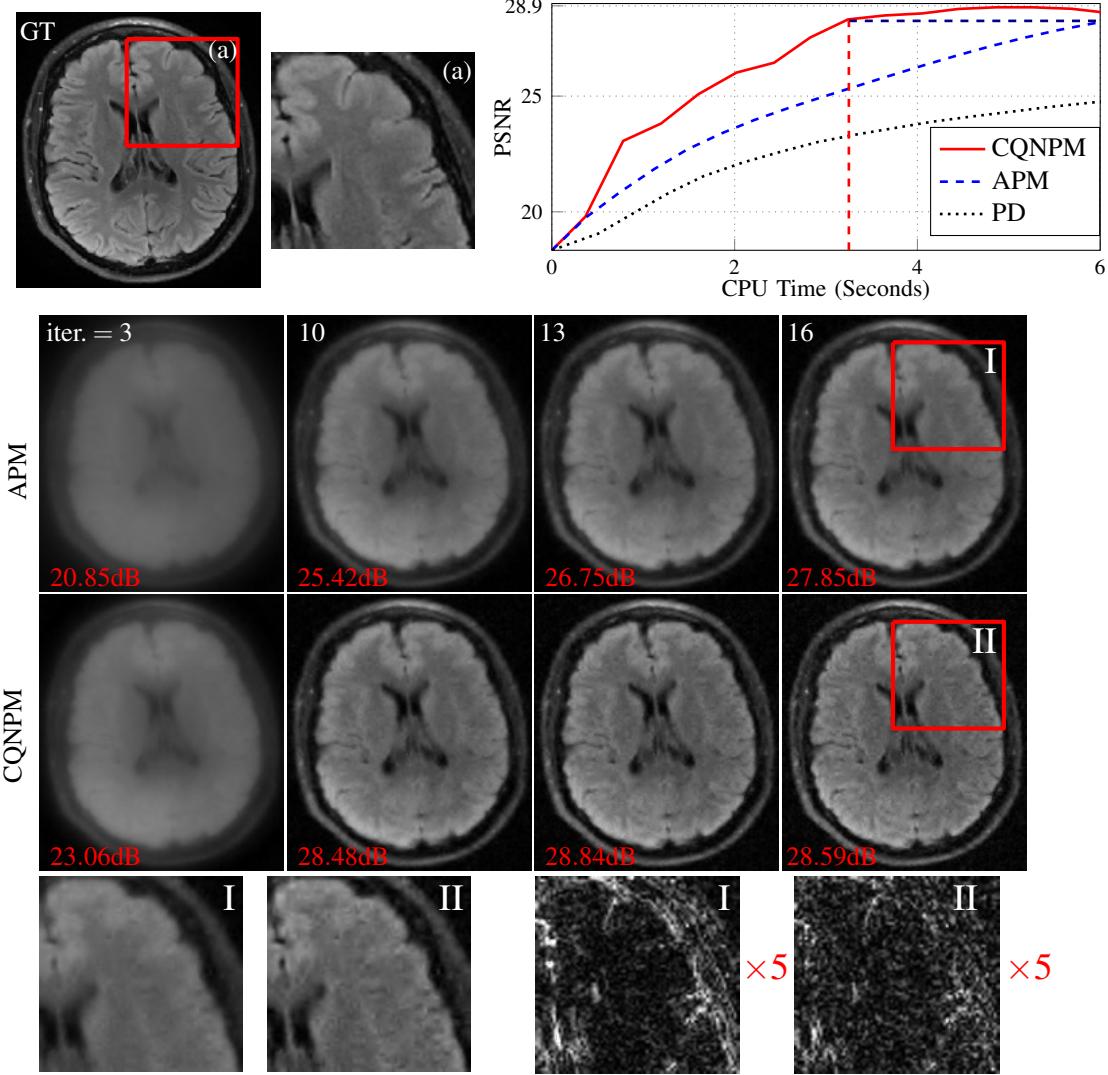


Fig. 4. First row: the ground truth image and PSNR values versus CPU time; second to third row: the reconstructed brain images at 3, 10, 13, and 16th iteration with Figure 3 setting; fourth row: the zoomed-in regions and the corresponding error maps ( $\times 5$ ) of the 16th iteration reconstructed images.

#### D. The Choice of Max\_Iter in Algorithm 3

Following the setting used in Section IV-C, we studied how the choice of Max\_Iter in Algorithm 3 affects the converge of CQNPM. Figure 8 presents the cost values versus iteration with different values of Max\_Iter. Clearly we see that CQNPM is quite robust to the choice of Max\_Iter. However, a small Max\_Iter (e.g., Max\_Iter= 10) can slightly increase the cost and Max\_Iter= 20,50 converged faster than other vaules. In our experiments, we found that Max\_Iter= 20 is sufficient.

#### E. Reconstruction with High Data Input SNR

This part studies the reconstruction for complex additive Gaussian noise with mean zero and lower variance  $4 \times 10^{-5}$ , yielding around 30dB data input SNR. Figure II displays the reconstructed results using spiral acquisition and  $h(\mathbf{x}) = \alpha \|\mathbf{Tx}\|_1 + (1 - \alpha)\text{TV}(\mathbf{x})$ . The reconstructed images are much clearer than those in the low data input SNR cases. Moreover,

the convergence trends of different algorithms are similar to those observed in low data input SNR reconstructions. The supplementary material provides the reconstructed results of the knee image that align with the observations made from the brain image presented here.

#### V. CONCLUSIONS AND FUTURE WORK

This paper proposes complex quasi-Newton proximal methods for solving (2) that led to faster convergence than APMs. By using the structure of  $\mathbf{B}_k$ , we develop efficient approaches for computing the WPM by considering wavelet and TV regularizers. Compared with computing the proximal mapping in APMs, i.e.,  $\mathbf{B}_k = \mathbf{I}_N$ , the increased computational cost in computing the WPM is insignificant, as illustrated by our comparisons in terms of CPU time. CQNPM is appealing for large-scale problems because CQNPM requires fewer iterations than APMs to converge, reducing the times of computing  $\nabla f(\mathbf{x})$  that it is expensive in large-scale settings. Interestingly, in our

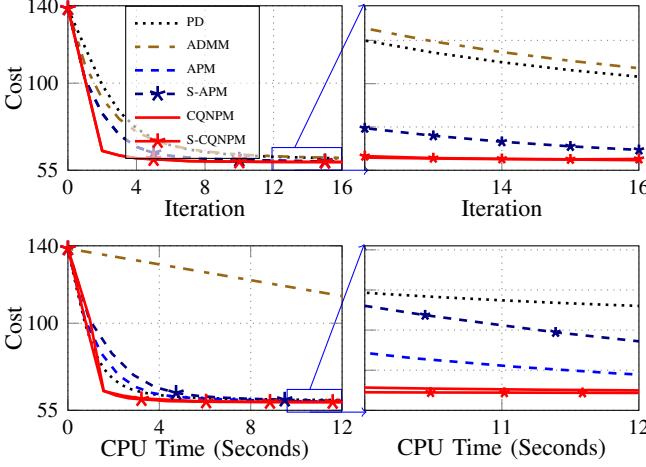


Fig. 5. Cost values versus iteration (top) and CPU time (bottom) of the brain image with regularizer  $h(\mathbf{x}) = \alpha\|\mathbf{Tx}\|_1 + (1-\alpha)\text{TV}(\mathbf{x})$  and same acquisition as Figure 3. The parameters were  $\lambda = 6 \times 10^{-4}$  and  $\alpha = \frac{1}{6}$ .

setting, we found the partial smoothing method worked pretty well when both wavelet and TV regularizers are used. So the partial smoothing approach may be a good method for solving problems with two nonsmooth terms. To adapt CQNPM to other regularizers, one must find an efficient approach to address the WPM for the chosen regularizer to preserve the computational efficiency.

Clearly,  $\mathbf{B}_k$  plays an important role in our algorithm and a more accurate  $\mathbf{B}_k$  can accelerate the convergence further. Since the Hessian matrix in CS MRI is known, i.e.,  $\mathbf{A}^H \mathbf{A}$ , we plan to learn a fixed weighting  $\mathbf{B}$  to approximate  $\mathbf{A}^H \mathbf{A}$  accurately for future work. However,  $\mathbf{B}$  must be easy to invert so that  $\mathbf{B}$  should have some special structures, e.g.,  $\mathbf{B} = \mathbf{D} \pm \mathbf{U} \mathbf{U}^H$ , and finding such a  $\mathbf{B}$  should be computationally cheap since  $\mathbf{A}$  is different from each acquisition because the sensitivity mapping is patient dependent. Moreover, with such a fixed  $\mathbf{B}$ , we can adopt the accelerated manner used in APMs for Algorithm I and obtain an even faster algorithm than the one presented here.

## VI. ACKNOWLEDGEMENTS

This work was funded by National Institutes of Health grant R01NS112233.

## APPENDIX A PROOF OF PROPOSITION I

Similar to [12] for the real case, one can prove the following relations for complex numbers  $x, y \in \mathbb{C}$

$$\begin{aligned} \sqrt{|x|^2 + |y|^2} &= \max_{p_1, p_2 \in \mathbb{C}} \{ \Re(p_1^* x + p_2^* y) : |p_1|^2 + |p_2|^2 \leq 1 \} \\ |x| &= \max_{p \in \mathbb{C}} \{ \Re(p^* x) : |p| \leq 1 \} \end{aligned}$$

where  $*$  denotes the conjugate operator and  $\Re(\cdot)$  represents an operator to take the real part. With these relations and the

definition of TV functions, we can rewrite  $\text{TV}(\mathbf{x})$  and  $\|\mathbf{Tx}\|_1$  as

$$\begin{aligned} \text{TV}(\mathbf{x}) &= \max_{(\mathbf{P}, \mathbf{Q}) \in \mathcal{P}} \Re \left\{ \text{vec}(\mathcal{L}(\mathbf{P}, \mathbf{Q}))^H \mathbf{x} \right\}, \\ \|\mathbf{Tx}\|_1 &= \max_{\mathbf{z} \in \mathcal{Z}} \Re \left\{ \mathbf{z}^H \mathbf{T} \mathbf{x} \right\}, \end{aligned}$$

where  $\mathcal{P} = \mathcal{P}_1$  (respectively,  $\mathcal{P}_2$ ) for  $\text{TV}_{\text{iso}}$  (respectively,  $\text{TV}_{\ell_1}$ ). Hence, we represent (8) as

$$\min_{\mathbf{x} \in \mathbb{C}^N} \max_{\substack{\mathbf{z} \in \mathcal{Z}, \\ (\mathbf{P}, \mathbf{Q}) \in \mathcal{P}}} \|\mathbf{x} - \mathbf{v}_k\|_{\mathbf{B}_k}^2 + 2\bar{\lambda}g(\mathbf{x}, \mathbf{z}, \mathbf{P}, \mathbf{Q}), \quad (15)$$

where

$$g(\mathbf{x}, \mathbf{z}, \mathbf{P}, \mathbf{Q}) = \Re \left\{ \alpha \langle \mathbf{T} \mathbf{x}, \mathbf{z} \rangle + (1-\alpha) \text{vec}(\mathcal{L}(\mathbf{P}, \mathbf{Q}))^H \mathbf{x} \right\}.$$

Reorganizing (15), we get

$$\min_{\mathbf{x} \in \mathbb{C}^N} \max_{\substack{\mathbf{z} \in \mathcal{Z}, \\ (\mathbf{P}, \mathbf{Q}) \in \mathcal{P}}} \|\mathbf{x} - \mathbf{w}_k(\mathbf{z}, \mathbf{P}, \mathbf{Q})\|_{\mathbf{B}_k}^2 - \|\mathbf{w}_k(\mathbf{z}, \mathbf{P}, \mathbf{Q})\|_{\mathbf{B}_k}^2, \quad (16)$$

where

$$\mathbf{w}_k(\mathbf{z}, \mathbf{P}, \mathbf{Q}) = \mathbf{v}_k - \bar{\lambda} \mathbf{B}_k^{-1} \left( \alpha \mathbf{T}^H \mathbf{z} + (1-\alpha) \text{vec}(\mathcal{L}(\mathbf{P}, \mathbf{Q})) \right).$$

Since (16) is convex in  $\mathbf{x}$  and concave in  $(\mathbf{z}, \mathbf{P}, \mathbf{Q})$ , we interchange the minimum and maximum and then get

$$\max_{\mathbf{z} \in \mathcal{Z}} \min_{\mathbf{x} \in \mathbb{C}^N} \|\mathbf{x} - \mathbf{w}_k(\mathbf{z}, \mathbf{P}, \mathbf{Q})\|_{\mathbf{B}_k}^2 - \|\mathbf{w}_k(\mathbf{z}, \mathbf{P}, \mathbf{Q})\|_{\mathbf{B}_k}^2. \quad (17)$$

Note that  $\mathbf{x}$  only appears in the first term of (17) so that the optimal solution of the minimum part is

$$\mathbf{x}^* = \mathbf{w}_k(\mathbf{z}, \mathbf{P}, \mathbf{Q}). \quad (18)$$

Substituting (18) into (17), we get the following dual problem that contains only unknown dual variables  $(\mathbf{z}, \mathbf{P}, \mathbf{Q})$

$$(\mathbf{z}^*, \mathbf{P}^*, \mathbf{Q}^*) = \underset{\substack{\mathbf{z} \in \mathcal{Z}, \\ (\mathbf{P}, \mathbf{Q}) \in \mathcal{P}}}{\operatorname{argmin}} \|\mathbf{w}_k(\mathbf{z}, \mathbf{P}, \mathbf{Q})\|_{\mathbf{B}_k}^2. \quad (19)$$

After solving (19), the primal variable update is  $\mathbf{x}_{k+1} = \mathbf{w}_k(\mathbf{z}^*, \mathbf{P}^*, \mathbf{Q}^*)$ . This completes the proof.

## APPENDIX B PROOF OF LEMMA I

Denote by  $h(\mathbf{z}, \mathbf{P}, \mathbf{Q}) \triangleq \|\mathbf{w}_k(\mathbf{z}, \mathbf{P}, \mathbf{Q})\|_{\mathbf{B}_k}^2$ . Applying the chain rule, we get

$$\nabla h(\mathbf{z}, \mathbf{P}, \mathbf{Q}) = -2\bar{\lambda} \left[ \frac{\alpha \mathbf{T}}{(1-\alpha) \mathcal{L}^T} \right] \mathbf{w}_k(\mathbf{z}, \mathbf{P}, \mathbf{Q}).$$

Now, we compute the Lipschitz constant of  $h(\mathbf{z}, \mathbf{P}, \mathbf{Q})$ . For every two pairs of  $(\mathbf{z}_1, \mathbf{P}_1, \mathbf{Q}_1)$  and  $(\mathbf{z}_2, \mathbf{P}_2, \mathbf{Q}_2)$ , we have

$$\begin{aligned} & \|\nabla h(\mathbf{z}_1, \mathbf{P}_1, \mathbf{Q}_1) - \nabla h(\mathbf{z}_2, \mathbf{P}_2, \mathbf{Q}_2)\| \\ &= 2\bar{\lambda}^2 \left\| \begin{bmatrix} \alpha \mathbf{T} \\ (1-\alpha) \mathcal{L}^T \end{bmatrix} \mathbf{B}_k^{-1} \begin{bmatrix} \alpha \mathbf{T}^H & (1-\alpha) \mathcal{L} \end{bmatrix} \right. \\ &\quad \left. [(\mathbf{z}_1, \mathbf{P}_1, \mathbf{Q}_1) - (\mathbf{z}_2, \mathbf{P}_2, \mathbf{Q}_2)] \right\| \\ &\leq 2\bar{\lambda}^2 \left\| \alpha^2 \mathbf{T}^H \mathbf{T} + (1-\alpha)^2 \mathcal{L}^T \mathcal{L} \right\| \left\| \mathbf{B}_k^{-1} \right\| \\ &\quad \left\| [(\mathbf{z}_1, \mathbf{P}_1, \mathbf{Q}_1) - (\mathbf{z}_2, \mathbf{P}_2, \mathbf{Q}_2)] \right\| \\ &\leq 2\bar{\lambda}^2 (\alpha^2 \|\mathbf{T}\|^2 + (1-\alpha)^2 \|\mathcal{L}\|^2) \sigma_{\min} \\ &\quad \left\| [(\mathbf{z}_1, \mathbf{P}_1, \mathbf{Q}_1) - (\mathbf{z}_2, \mathbf{P}_2, \mathbf{Q}_2)] \right\|, \end{aligned}$$

where  $\sigma_{\min}$  is the smallest eigenvalue of  $\mathbf{B}_k$ . With the proof of [12] Lemma 4.2], we know  $\|\mathcal{L}\| = \sqrt{8}$  such that the Lipschitz constant of  $h(\mathbf{z}, \mathbf{P}, \mathbf{Q})$  is  $L_c = 2\sigma_{\min}\bar{\lambda}^2(\alpha^2 \|\mathbf{T}\|^2 + 8(1-\alpha)^2)$ . This completes the proof.

## REFERENCES

- [1] M. Lustig, D. Donoho, and J. M. Pauly, “Sparse MRI: The application of compressed sensing for rapid MR imaging,” *Magnetic Resonance in Medicine*, vol. 58, no. 6, pp. 1182–1195, 2007.
- [2] M. Guerquin-Kern, M. Haberlin, K. P. Pruessmann, and M. Unser, “A fast wavelet-based reconstruction method for magnetic resonance imaging,” *IEEE Transactions on Medical Imaging*, vol. 30, no. 9, pp. 1649–1660, 2011.
- [3] S. Ravishankar and Y. Bresler, “MR image reconstruction from highly undersampled k-space data by dictionary learning,” *IEEE Transactions on Medical Imaging*, vol. 30, no. 5, pp. 1028–1041, 2011.
- [4] W. Dong, G. Shi, X. Li, Y. Ma, and F. Huang, “Compressive sensing via nonlocal low-rank regularization,” *IEEE Transactions on Image Processing*, vol. 23, no. 8, pp. 3618–3632, 2014.
- [5] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, “Plug-and-play priors for model based reconstruction,” in *IEEE Global Conference on Signal and Information Processing*. IEEE, 2013, pp. 945–948.
- [6] 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 Processing Magazine*, vol. 37, no. 1, pp. 105–116, 2020.
- [7] 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.
- [8] Y. Song, L. Shen, L. Xing, and S. Ermon, “Solving inverse problems in medical imaging with score-based generative models,” in *International Conference on Learning Representations*, 2021.
- [9] H. Gu, B. Yaman, S. Moeller, J. Ellermann, K. Ugurbil, and M. Akçakaya, “Revisiting  $\ell_1$ -wavelet compressed-sensing MRI in the era of deep learning,” *Proceedings of the National Academy of Sciences*, vol. 119, no. 33, p. e2201062119, 2022.
- [10] N. Parikh, S. Boyd *et al.*, “Proximal algorithms,” *Foundations and Trends® in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.
- [11] A. Beck and M. Teboulle, “A fast iterative shrinkage-thresholding algorithm for linear inverse problems,” *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.
- [12] ———, “Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems,” *IEEE Transactions on Image Processing*, vol. 18, no. 11, pp. 2419–2434, 2009.
- [13] A. Chambolle and T. Pock, “A first-order primal-dual algorithm for convex problems with applications to imaging,” *Journal of Mathematical Imaging and Vision*, vol. 40, pp. 120–145, 2011.
- [14] L. Condat, “A primal-dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms,” *Journal of Optimization Theory and Applications*, vol. 158, no. 2, pp. 460–479, 2013.
- [15] N. Komodakis and J.-C. Pesquet, “Playing with duality: An overview of recent primal-dual approaches for solving large-scale optimization problems,” *IEEE Signal Processing Magazine*, vol. 32, no. 6, pp. 31–54, 2015.
- [16] J. A. Fessler, “Optimization methods for magnetic resonance image reconstruction: Key models and optimization algorithms,” *IEEE Signal Processing Magazine*, vol. 37, no. 1, pp. 33–40, 2020.
- [17] S. S. Iyer, F. Ong, X. Cao, C. Liao, D. Luca, J. I. Tamir, and K. Setsompop, “Polynomial preconditioners for regularized linear inverse problems,” *arXiv preprint arXiv:2204.10252*, 2022.
- [18] F. Ong, M. Uecker, and M. Lustig, “Accelerating non-Cartesian MRI reconstruction convergence using k-space preconditioning,” *IEEE Transactions on Medical Imaging*, vol. 39, no. 5, pp. 1646–1654, 2019.
- [19] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [20] B. He and X. Yuan, “On the  $O(1/n)$  convergence rate of the Douglas–Rachford alternating direction method,” *SIAM Journal on Numerical Analysis*, vol. 50, no. 2, pp. 700–709, 2012.
- [21] S. Ramani and J. A. Fessler, “Parallel MR image reconstruction using augmented Lagrangian methods,” *IEEE Transactions on Medical Imaging*, vol. 30, no. 3, pp. 694–706, 2010.
- [22] D. S. Weller, S. Ramani, and J. A. Fessler, “Augmented Lagrangian with variable splitting for faster non-Cartesian  $\ell_1$ -SPIRiT MR image reconstruction,” *IEEE Transactions on Medical Imaging*, vol. 33, no. 2, pp. 351–361, 2013.
- [23] K. Koolstra, J. van Gemert, P. Börnert, A. Webb, and R. Remis, “Accelerating compressed sensing in parallel imaging reconstructions using an efficient circulant

- preconditioner for Cartesian trajectories,” *Magnetic Resonance in Medicine*, vol. 81, no. 1, pp. 670–685, 2019.
- [24] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, 2006.
- [25] J. D. Lee, Y. Sun, and M. A. Saunders, “Proximal Newton-type methods for minimizing composite functions,” *SIAM Journal on Optimization*, vol. 24, no. 3, pp. 1420–1443, 2014.
- [26] S. Becker, J. Fadili, and P. Ochs, “On quasi-Newton forward-backward splitting: Proximal calculus and convergence,” *SIAM Journal on Optimization*, vol. 29, no. 4, pp. 2445–2481, 2019.
- [27] T. Hong, I. Yavneh, and M. Zibulevsky, “Solving RED with weighted proximal methods,” *IEEE Signal Processing Letters*, vol. 27, pp. 501–505, 2020.
- [28] A. Kadu, H. Mansour, and P. T. Boufounos, “High-contrast reflection tomography with total-variation constraints,” *IEEE Transactions on Computational Imaging*, vol. 6, pp. 1523–1536, 2020.
- [29] T. Ge, U. Villa, U. S. Kamilov, and J. A. O’Sullivan, “Proximal Newton methods for X-ray imaging with non-smooth regularization,” *Electronic Imaging*, vol. 2020, no. 14, pp. 7–1, 2020.
- [30] E. Chouzenoux, J.-C. Pesquet, and A. Repetti, “Variable metric forward–backward algorithm for minimizing the sum of a differentiable function and a convex function,” *Journal of Optimization Theory and Applications*, vol. 162, no. 1, pp. 107–132, 2014.
- [31] J. A. Fessler, “Model-based image reconstruction for MRI,” *IEEE Signal Processing Magazine*, vol. 27, no. 4, pp. 81–9, Jul. 2010.
- [32] J. H. Duyn, P. van Gelderen, T.-Q. Li, J. A. de Zwart, A. P. Koretsky, and M. Fukunaga, “High-field MRI of brain cortical substructure based on signal phase,” *Proceedings of the National Academy of Sciences*, vol. 104, no. 28, pp. 11 796–11 801, 2007.
- [33] Y. Wang and T. Liu, “Quantitative susceptibility mapping (QSM): decoding MRI data for a tissue magnetic biomarker,” *Magnetic Resonance in Medicine*, vol. 73, no. 1, pp. 82–101, 2015.
- [34] S. Bonettini, I. Loris, F. Porta, and M. Prato, “Variable metric inexact line-search-based methods for nonsmooth optimization,” *SIAM Journal on Optimization*, vol. 26, no. 2, pp. 891–921, 2016.
- [35] A. Chambolle, “An algorithm for total variation minimization and applications,” *Journal of Mathematical Imaging and Vision*, vol. 20, no. 1, pp. 89–97, 2004.
- [36] Y. Nesterov, “A method for unconstrained convex minimization problem with the rate of convergence  $O(1/k^2)$ ,” in *Doklady an SSSR*, vol. 269, no. 3, 1983, pp. 543–547.
- [37] M. Elad, P. Milanfar, and R. Rubinstein, “Analysis versus synthesis in signal priors,” *Inverse problems*, vol. 23, no. 3, p. 947, 2007.
- [38] A. Beck and M. Teboulle, “Smoothing and first order methods: A unified framework,” *SIAM Journal on Optimization*, vol. 22, no. 2, pp. 557–580, 2012.
- [39] J. Zbontar, F. Knoll, A. Sriram, T. Murrell, Z. Huang, M. J. Muckley, A. Defazio, R. Stern, P. Johnsson, M. Bruno *et al.*, “fastMRI: An open dataset and benchmarks for accelerated MRI,” *arXiv preprint arXiv:1811.08839*, 2018.
- [40] 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,” *Magnetic Resonance in Medicine*, vol. 71, no. 3, pp. 990–1001, 2014.
- [41] F. Ong and M. Lustig, “SigPy: a python package for high performance iterative reconstruction,” in *Proceedings of the ISMRM 27th Annual Meeting, Montreal, Quebec, Canada*, vol. 4819, 2019.
- [42] K. Zhang, Y. Li, W. Zuo, L. Zhang, L. Van Gool, and R. Timofte, “Plug-and-play image restoration with deep denoiser prior,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 10, pp. 6360–6376, 2021.
- [43] A. Beck, *First-Order Methods in Optimization*. SIAM, 2017, vol. 25.
- [44] E. Y. Sidky, J. H. Jørgensen, and X. Pan, “Convex optimization problem prototyping for image reconstruction in computed tomography with the Chambolle–Pock algorithm,” *Physics in Medicine & Biology*, vol. 57, no. 10, p. 3065, 2012.

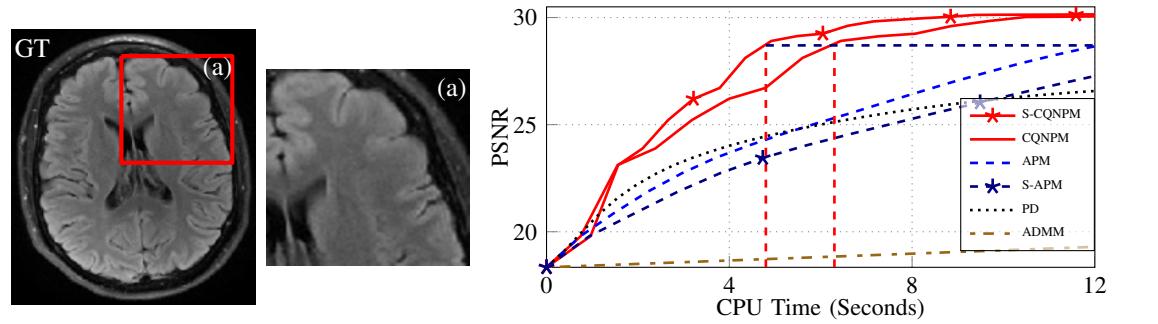


Fig. 6. First row: the ground truth image and PSNR values versus CPU time; second to sixth row: the reconstructed brain images at 3, 10, 13, and 16th iteration with Figure 5 setting and the zoomed-in regions of the 16th iteration reconstruction. We did not show the reconstructed image of ADMM since it yielded a much lower PSNR than other methods.

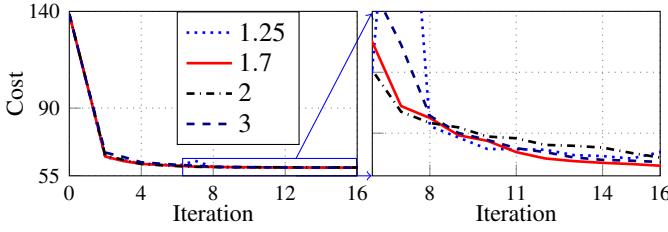


Fig. 7. Influence of  $\gamma$  on the convergence of CQNPM. Test on the brain image with wavelet and TV regularizers and radial acquisition. We set Max\_Iter= 20.

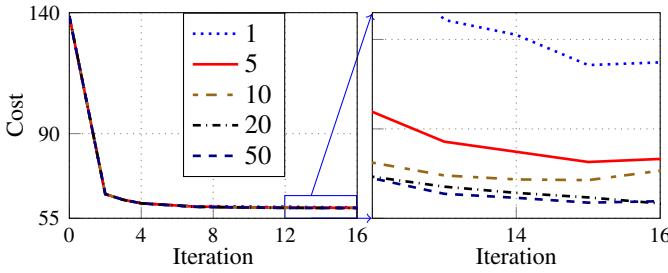


Fig. 8. Influence of Max\_Iter on the convergence of CQNPM. Test on the same problem as Figure 7 with  $\gamma = 1.7$ .

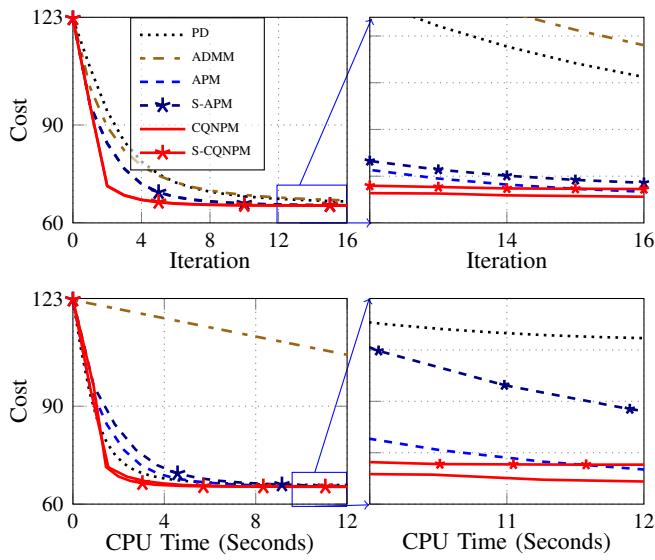


Fig. 9. Cost values versus iteration (top) and CPU time (bottom) of the knee image with same regularizer as Figure 5 but with spiral acquisition: 32 intervals, 1688 readout points, and 12 coils. The parameters  $\lambda$  and  $\alpha$  were  $10^{-3}$  and  $\frac{1}{2}$ .

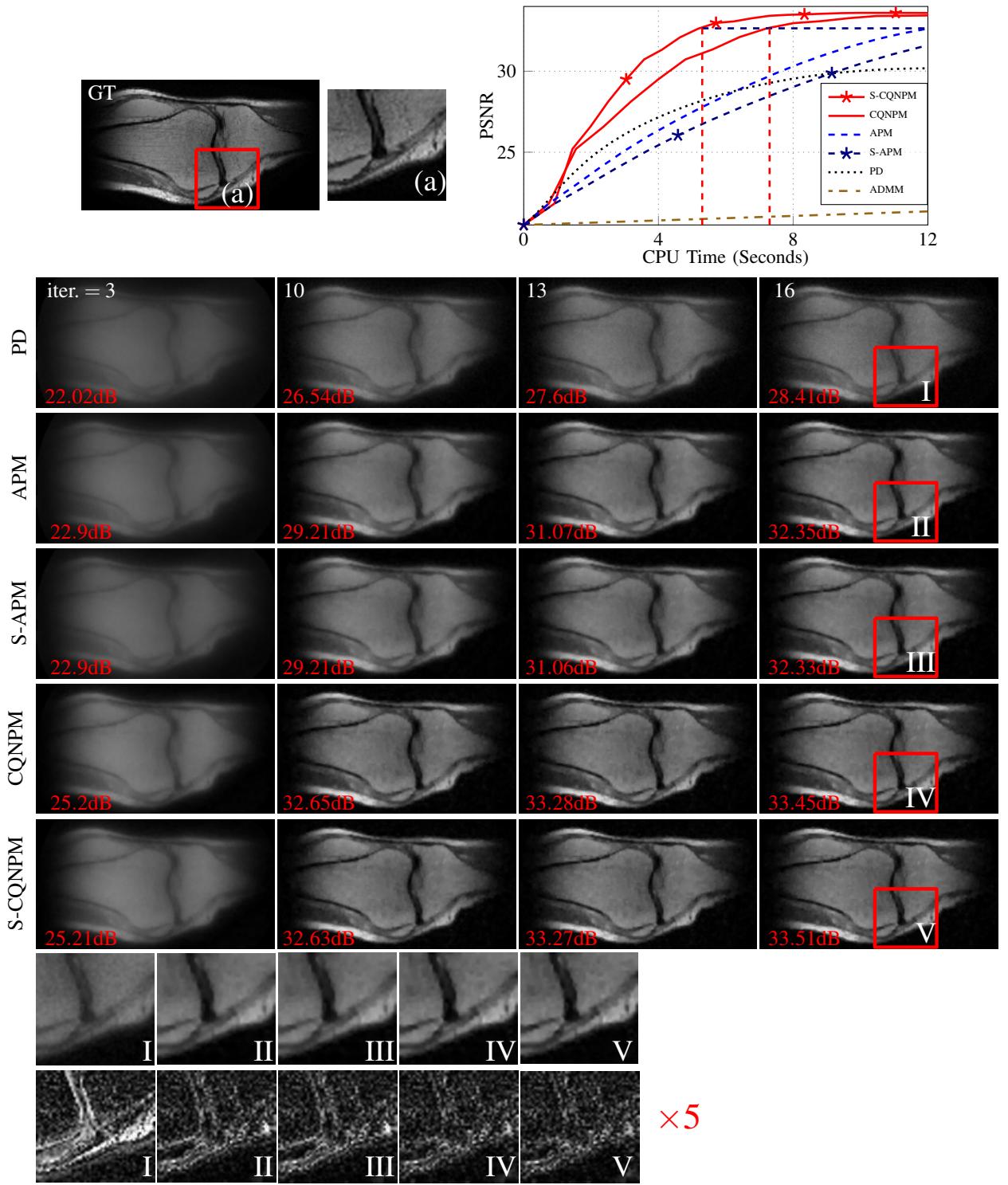


Fig. 10. First row: the ground truth image and PSNR values versus CPU time; second to sixth row: the reconstructed knee images at 3, 10, 13, and 16th iteration with Figure 9 setting. We did not show the reconstructed image of ADMM since it yielded a much lower PSNR than other methods. The seventh and eighth rows represent the zoomed-in regions and the corresponding error maps ( $\times 5$ ) of the 16th iteration reconstructed images with PD  $\rightarrow$  APM  $\rightarrow$  S-APM  $\rightarrow$  CQNPM  $\rightarrow$  S-CQNPM.

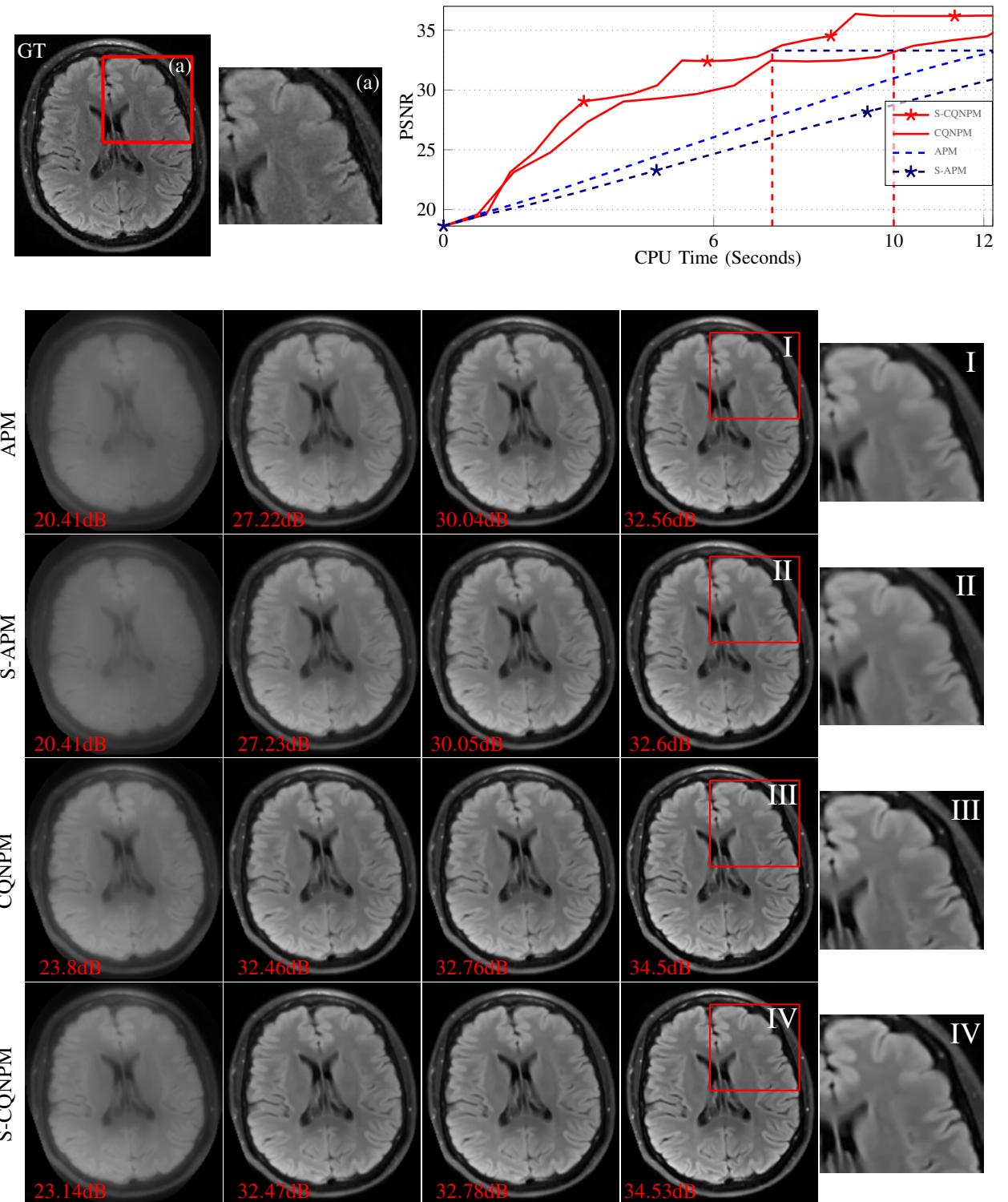


Fig. 11. First row: the ground truth image and PSNR values versus CPU time; second to sixth row: the reconstructed brain images at 3, 10, 13, and 16th iteration with spiral acquisition and  $h(\mathbf{x}) = \alpha\|\mathbf{Tx}\|_1 + (1 - \alpha)\text{TV}(\mathbf{x})$  and the zoomed-in regions of the 16th iteration reconstruction. The parameters were  $\lambda = 4 \times 10^{-5}$  and  $\alpha = \frac{3}{4}$ .

# Supplementary Material: A Complex Quasi-Newton Proximal Method for Image Reconstruction in Compressive Sensing MRI

Tao Hong, Luis Hernandez-Garcia, and Jeffrey A. Fessler *Fellow, IEEE*

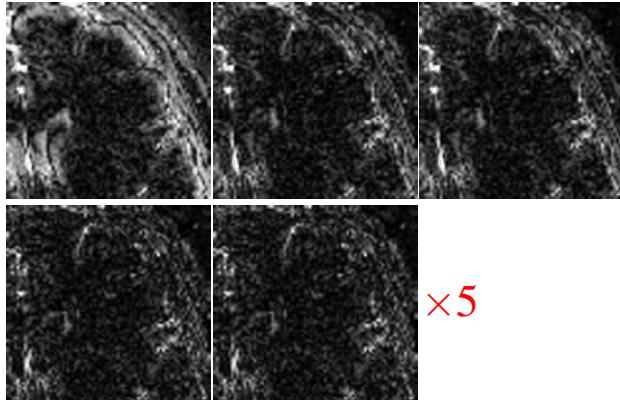


Fig. S.1. Error maps ( $\times 5$ ) of the zoomed-in regions of the 16th iteration reconstructed images in Fig. 6 of the main manuscript. From left to right, first row: PD  $\rightarrow$  APM  $\rightarrow$  S-APM; second row: CQNPM  $\rightarrow$  S-CQNPM.

## S.I. RADIAL ACQUISITION MRI RECONSTRUCTION

We provided the additional results with radial trajectory here. Figure S.1 describes the error maps of the zoom in regions of Fig. 6 in the main manuscript. Figures S.3 and S.4 present the results of using wavelet regularizer with different algorithms for the knee image. Clearly, we observe similar trends as we tested on the brain image.

Figures S.5 and S.6 display the results of using wavelet and TV regularizers for the knee image. Our method again outperforms other algorithms. Figure S.5 shows that APM and CQNPM reduced the cost values slightly faster than S-APM and S-CQNPM in terms of iterations and CPU time, which is slightly different from our test on brain image. However, it is unsurprising because S-APM and S-CQNPM solved the partial smoothing cost function, rather than original one, yet we still computed the original non-smooth cost.

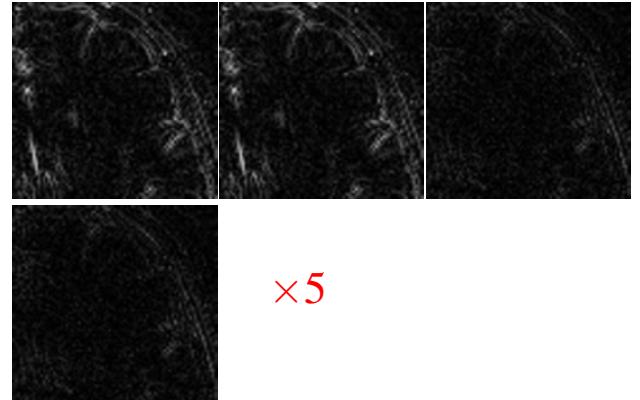


Fig. S.2. Error maps ( $\times 5$ ) of the zoomed-in in regions of the 16 reconstructed images in Fig. 11 of the manuscript. From left to right, first row: APM  $\rightarrow$  S-APM  $\rightarrow$  CQNPM; second row: S-CQNPM.

## S.II. SPIRAL ACQUISITION MRI RECONSTRUCTION

We provided the reconstruction with spiral acquisition that used 32 interleaves, 1688 readout points, and 12 coils.

Figure S.2 describes the error maps of the zoomed-in regions in Fig. 11 of the main manuscript. Figures S.7 to S.10 present the results of the brain and knee images with wavelet regularizer. Figures S.11 and S.12 show the results of the brain image with wavelet and TV regularizers. Figure S.13 describes the error maps of the zoom in regions of Figure S.12. Figure S.14 depicts the knee image with wavelet and TV regularizers on 30dB input SNR. Clearly, the trends are similar to the radial acquisition described in our paper.

## S.III. COMPARISON WITH PLUG-AND-PLAY

This section compares the reconstruction performance between wavelet and total variation regularizers, and Plug-and-Play (PnP) with a deep denoiser prior [1] and BM3D denoiser [2]. To obtain the deep denoiser prior, we employed

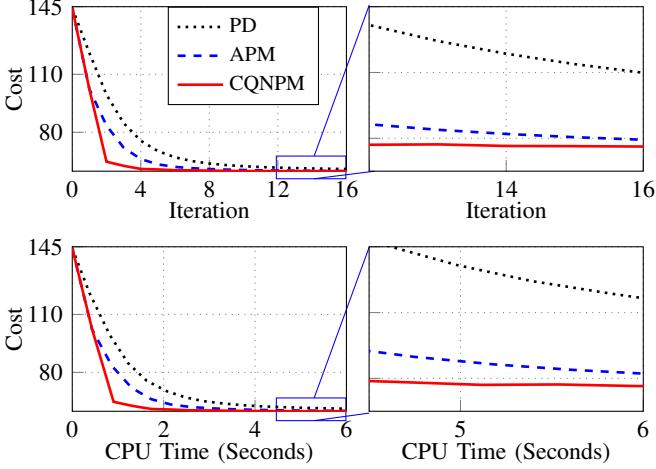


Fig. S.3. Cost values versus iteration (top) and CPU time (bottom) of the knee image with regularizer  $h(\mathbf{x}) = \|\mathbf{T}\mathbf{x}\|_1$  and  $\lambda = 5 \times 10^{-4}$  for a left invertible wavelet transform  $\mathbf{T}$  with 5 levels. Acquisition: radial trajectory with 96 projections, 512 readout points, and 12 coils. The parameter  $\lambda$  was  $10^{-3}$ .

the DnCNN [3] denoiser, training it on the brain dataset as described in [4]. We trained on three different noise variance  $\sigma = \{0.1, 1, 5\}$  yielding three different DnCNN denoisers. For testing, we selected one brain image from the test dataset and generated the k-space data using the spiral trajectory outlined in the manuscript. We then added complex i.i.d. Gaussian noise, resulting in a  $\sim 30$ dB data input SNR.

In the reconstruction process, we applied 35 iterations for both APM and CQNPM. For the PnP, we continued the iterations until the change in PSNR between two consecutive iterations was less than 0.01dB. For the PnP with the DnCNN denoiser, we chose the DnCNN trained with  $\sigma = 0.1$  as it yielded the best performance. Figure S.15 presents the reconstructed images that achieved the highest PSNR during the reconstruction process for each method. Here, PnP-BM3D and the wavelet and total variation regularizers led to higher SNR images than PnP with a deep denoiser prior. As described in [5], the deep denoiser may suffer from a noise scaling problem, which may explain its inferior SNR performance observed here. Further investigation of this issue is reserved for future work.

## REFERENCES

- [1] K. Zhang, Y. Li, W. Zuo, L. Zhang, L. Van Gool, and R. Timofte, “Plug-and-play image restoration with deep denoiser prior,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 10, pp. 6360–6376, 2021.
- [2] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising by sparse 3-d transform-domain collaborative filtering,” *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.
- [3] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, “Beyond a Gaussian denoiser: Residual learning of deep

- cnn for image denoising,” *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.
- [4] 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.
- [5] X. Xu, J. Liu, Y. Sun, B. Wohlberg, and U. S. Kamilov, “Boosting the performance of plug-and-play priors via denoiser scaling,” *arXiv preprint arXiv:2002.11546*, 2020.

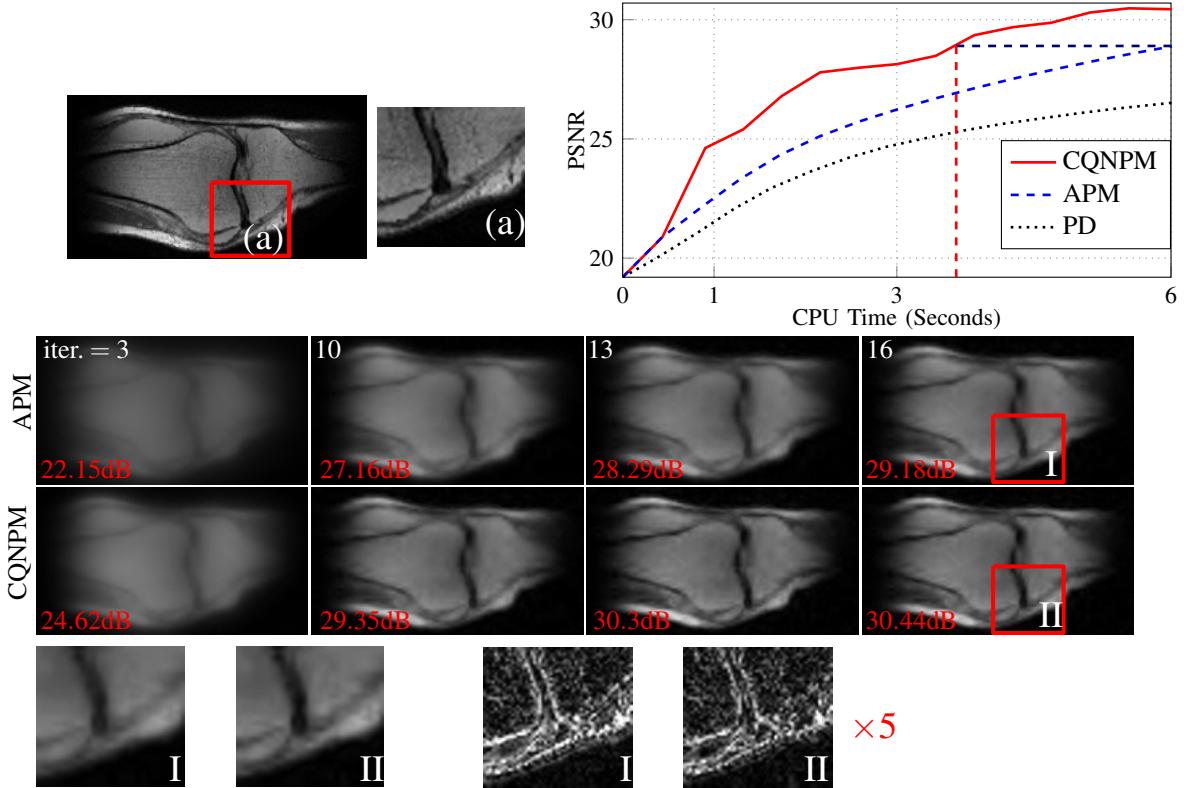


Fig. S.4. First row: the groundth image and PSNR values versus CPU time; second to third row: the reconstructed knee images at 3, 10, 13, and 16th iteration with Figure S.3 setting; fouth row: the zoomed-in regions and the corresponding error maps ( $\times 5$ ) of the 16th iteration reconstructed images.

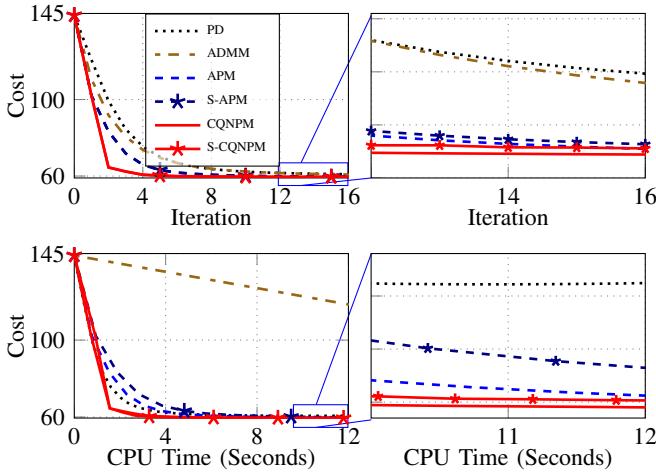


Fig. S.5. Cost values versus iteration (top) and CPU time (bottom) of the knee image with regularizer  $h(\mathbf{x}) = \alpha\|\mathbf{T}\mathbf{x}\|_1 + (1-\alpha)\text{TV}(\mathbf{x})$  and same acquisition as Figure S.3. The parameters  $\lambda$  and  $\alpha$  were  $10^{-3}$  and  $\frac{1}{2}$ .

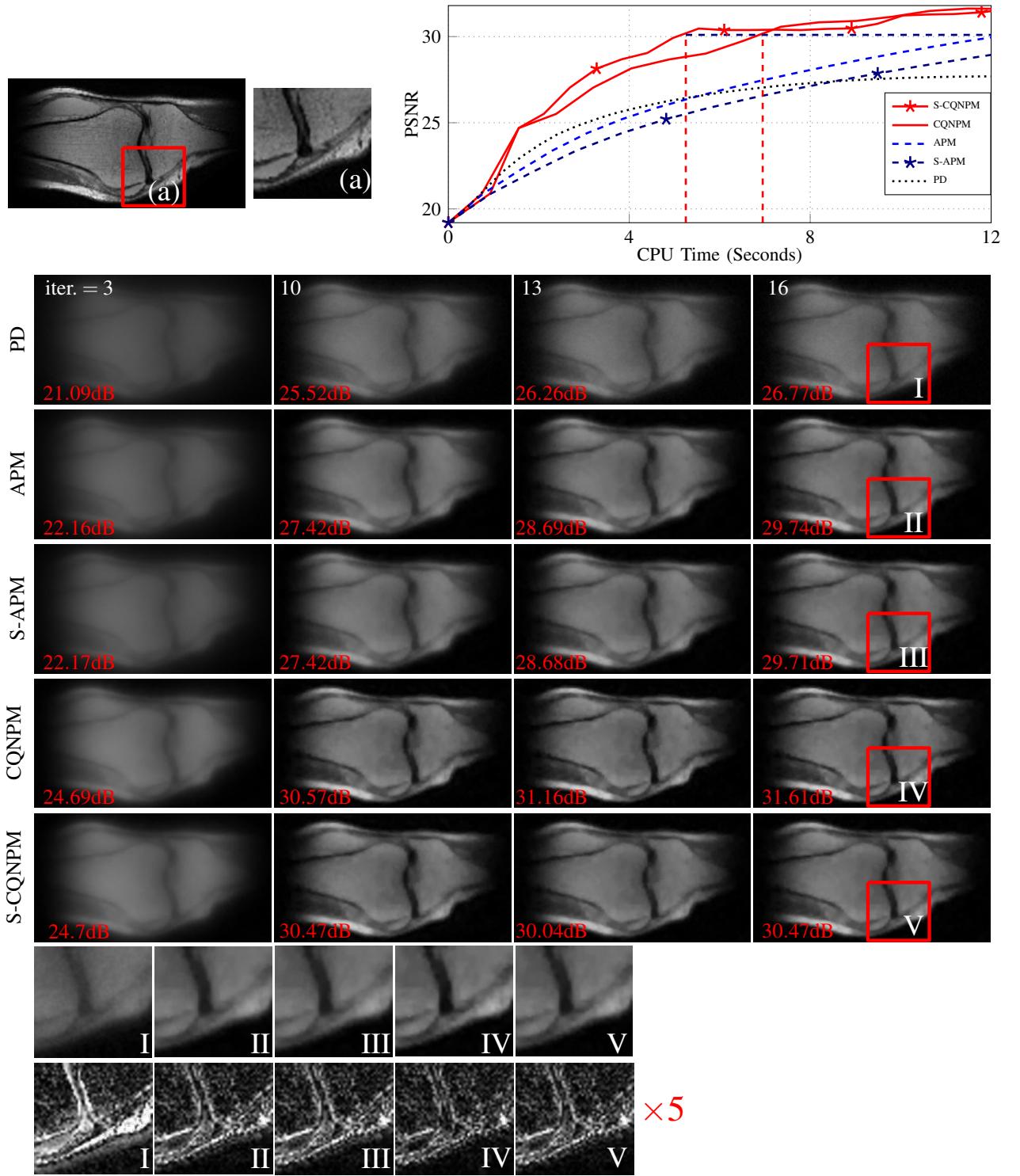


Fig. S.6. First row: the ground truth image and PSNR values versus CPU time; second to sixth row: the reconstructed knee images at 3, 10, 13, and 16th iteration with Figure S.5 setting. We did not show the reconstructed image of ADMM since it yielded a much lower PSNR than other methods. The seventh and eighth rows represent the zoomed-in regions and the corresponding error mapps ( $\times 5$ ) of the 16th iteration reconstructed images with PD  $\rightarrow$  APM  $\rightarrow$  S-APM  $\rightarrow$  CQNPM  $\rightarrow$  S-CQNPM.

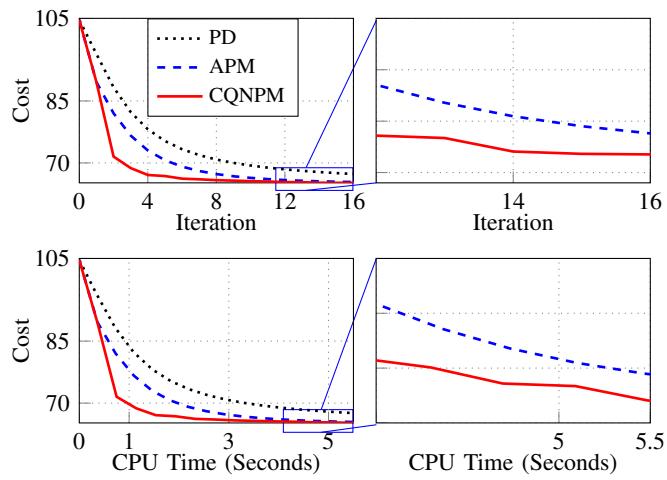


Fig. S.7. Cost values versus iteration (top) and CPU time (bottom) of the brain image with regularizer  $h(\mathbf{x}) = \|\mathbf{T}\mathbf{x}\|_1$  and  $\lambda = 10^{-3}$  for a left invertible wavelet transform  $\mathbf{T}$  with 5 levels. Acquisition: spiral trajectory with 32 intervals, 1688 readout points, and 12 coils.

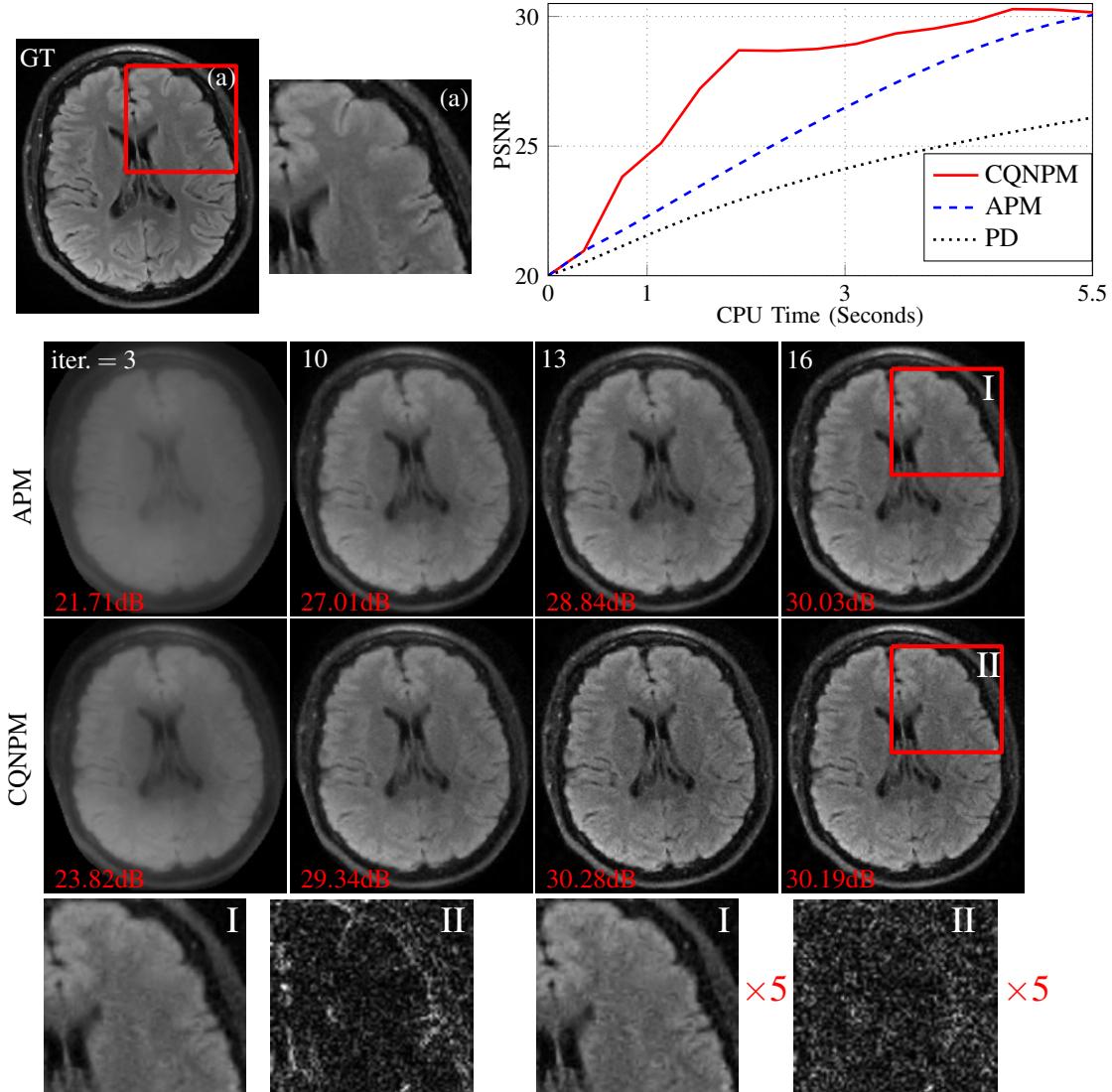


Fig. S.8. First row: the ground truth image and PSNR values versus CPU time; second to third row: the reconstructed brain images at 3, 10, 13, and 16th iteration by APM and CQNPM methods with Figure S.7 setting; fourth row: the zoomed-in regions and the corresponding error maps ( $\times 5$ ) of the 16th iteration reconstructed images.

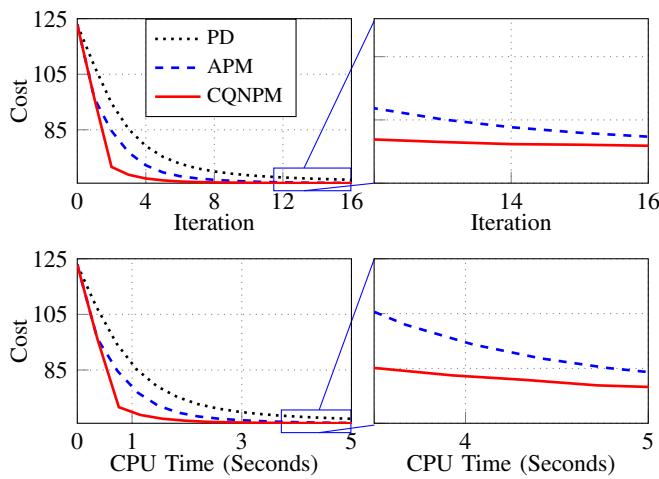


Fig. S.9. Cost values versus iteration (top) and CPU time (bottom) of the knee image with same regularizer and acquisition as Figure S.7. The parameter  $\lambda = 2 \times 10^{-3}$ .

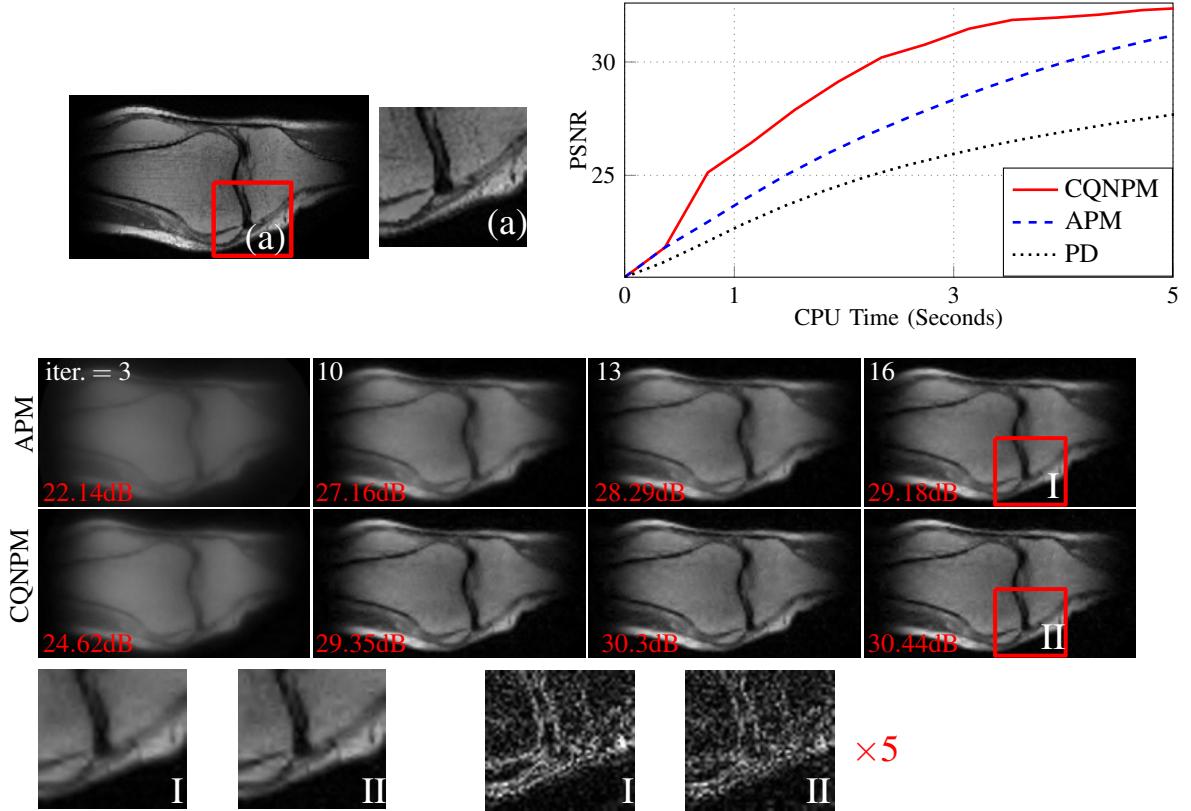


Fig. S.10. First row: the groundth image and PSNR values versus CPU time; second to third row: the reconstructed knee images at 3, 10, 13, and 16th iteration with Figure S.9 setting; fourth row: the zoomed-in regions and the corresponding error maps ( $\times 5$ ) of the 16th iteration reconstructed images.

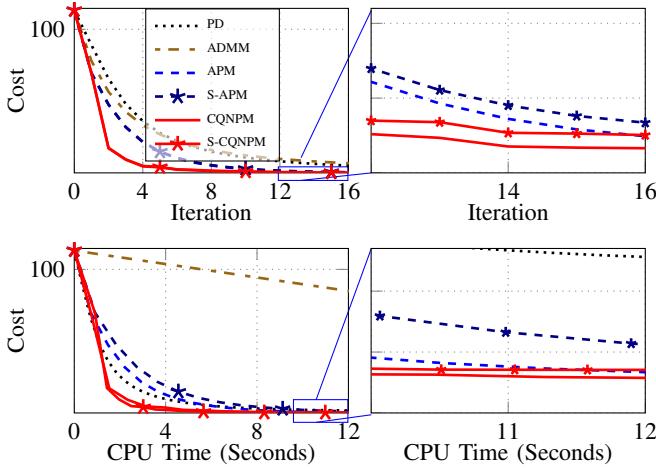


Fig. S.11. Cost values versus iteration (top) and CPU time (bottom) of the knee image with regularizer  $h(\mathbf{x}) = \alpha\|\mathbf{T}\mathbf{x}\|_1 + (1 - \alpha)\text{TV}(\mathbf{x})$  and same acquisition as Figure S.7. The parameters  $\lambda$  and  $\alpha$  were  $10^{-3}$  and  $\frac{1}{2}$ .

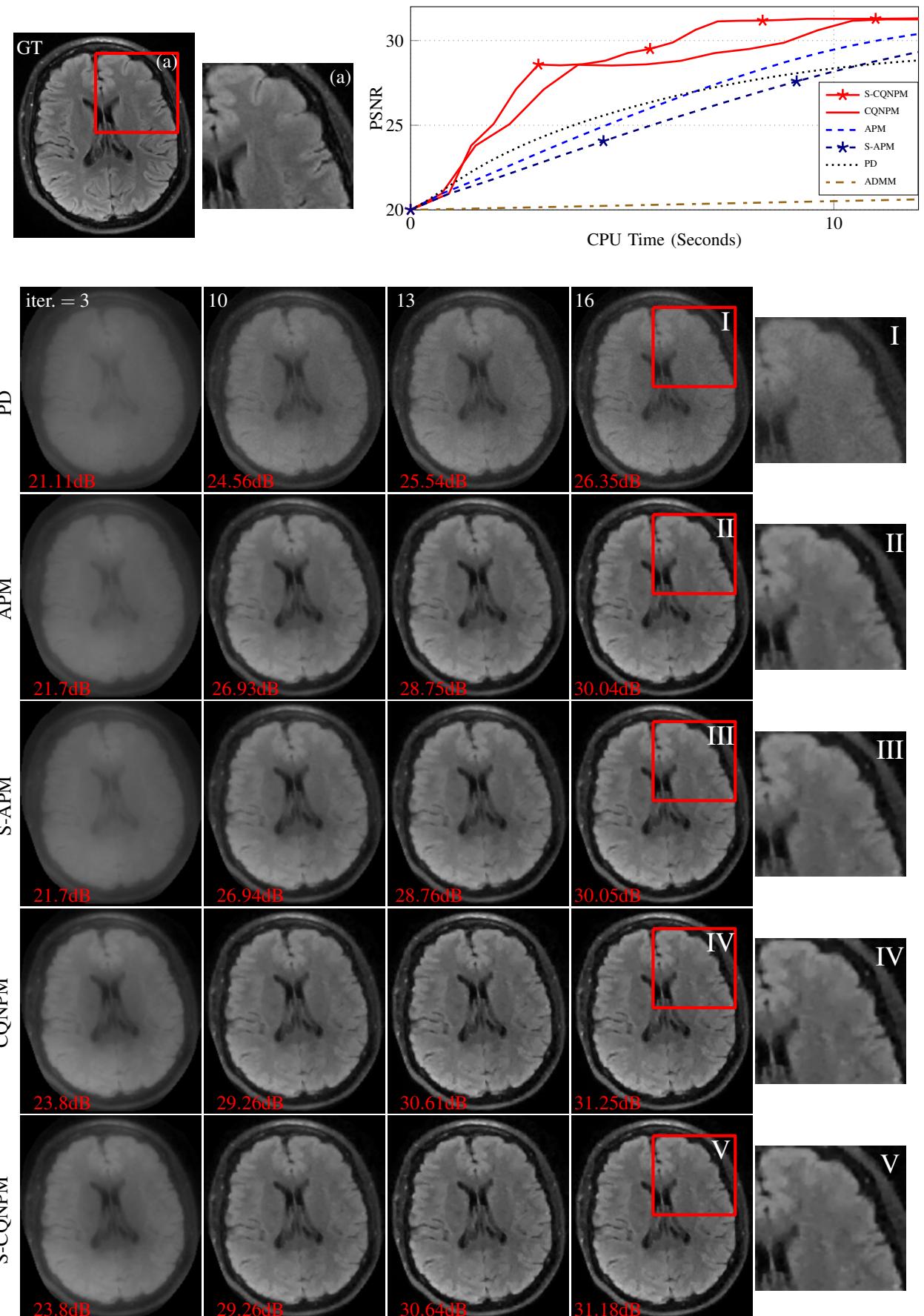


Fig. S.12. First row: the ground truth image and PSNR values versus CPU time; second to sixth row: the reconstructed brain images at 3, 10, 13, and 16th iteration with Figure S.11 setting and the zoomed-in regions of the 16th iteration reconstruction. We did not show the reconstructed image of ADMM since it yielded a much lower PSNR than other methods.

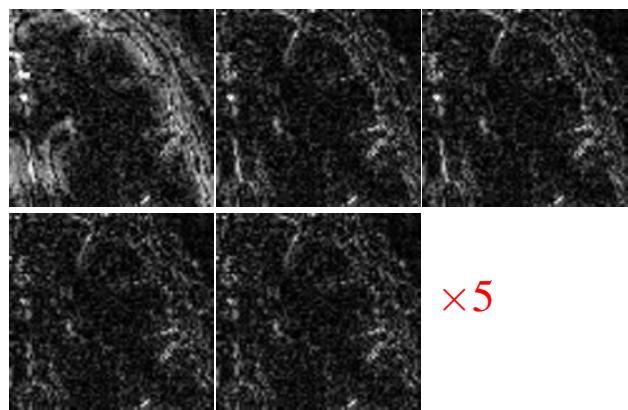


Fig. S.13. Error maps ( $\times 5$ ) of the zoomed-in regions of the 16 reconstructed images in Figure S.12. From left to right, first row: PD  $\rightarrow$  APM  $\rightarrow$  S-APM; second row: CQNPM  $\rightarrow$  S-CQNPM.

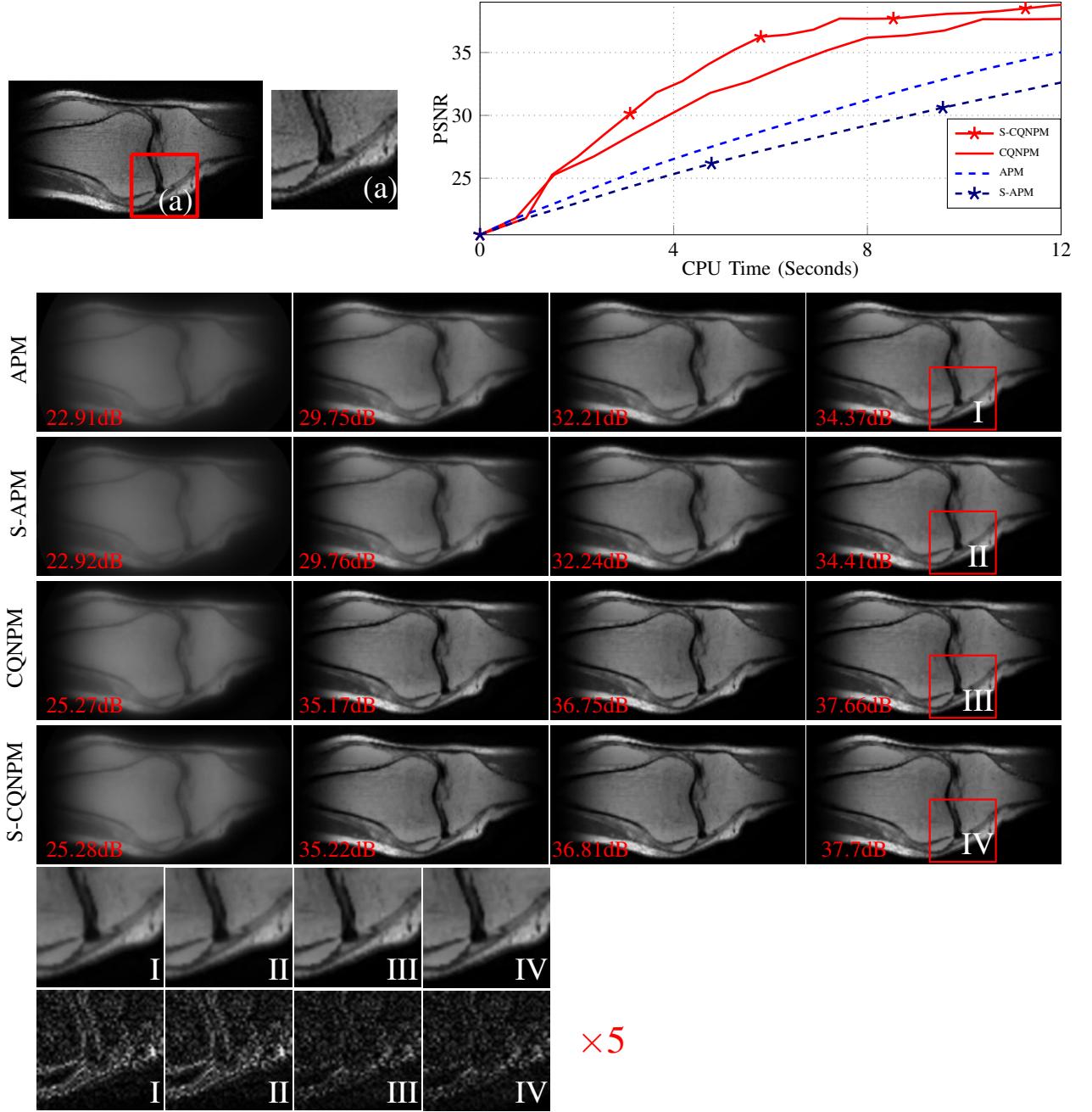


Fig. S.14. Performance of different algorithms with  $h(\mathbf{x}) = \alpha\|\mathbf{T}\mathbf{x}\|_1 + (1 - \alpha)\text{TV}(\mathbf{x})$  for the spiral acquisition and  $\sim 30\text{dB}$  data input SNR. The parameters  $\lambda = 2 \times 10^{-4}$  and  $\alpha = 10^{-4}$ . First row: the ground truth image and PSNR values versus CPU time; second to fifth row: the reconstructed knee images at 3, 10, 13, and 16th iteration with Figure S.5 setting. The sixth and seventh rows represent the zoomed-in regions and the corresponding error mapps ( $\times 5$ ) of the 16th iteration reconstructed images with APM  $\rightarrow$  S-APM  $\rightarrow$  CQNPM  $\rightarrow$  S-CQNPM.

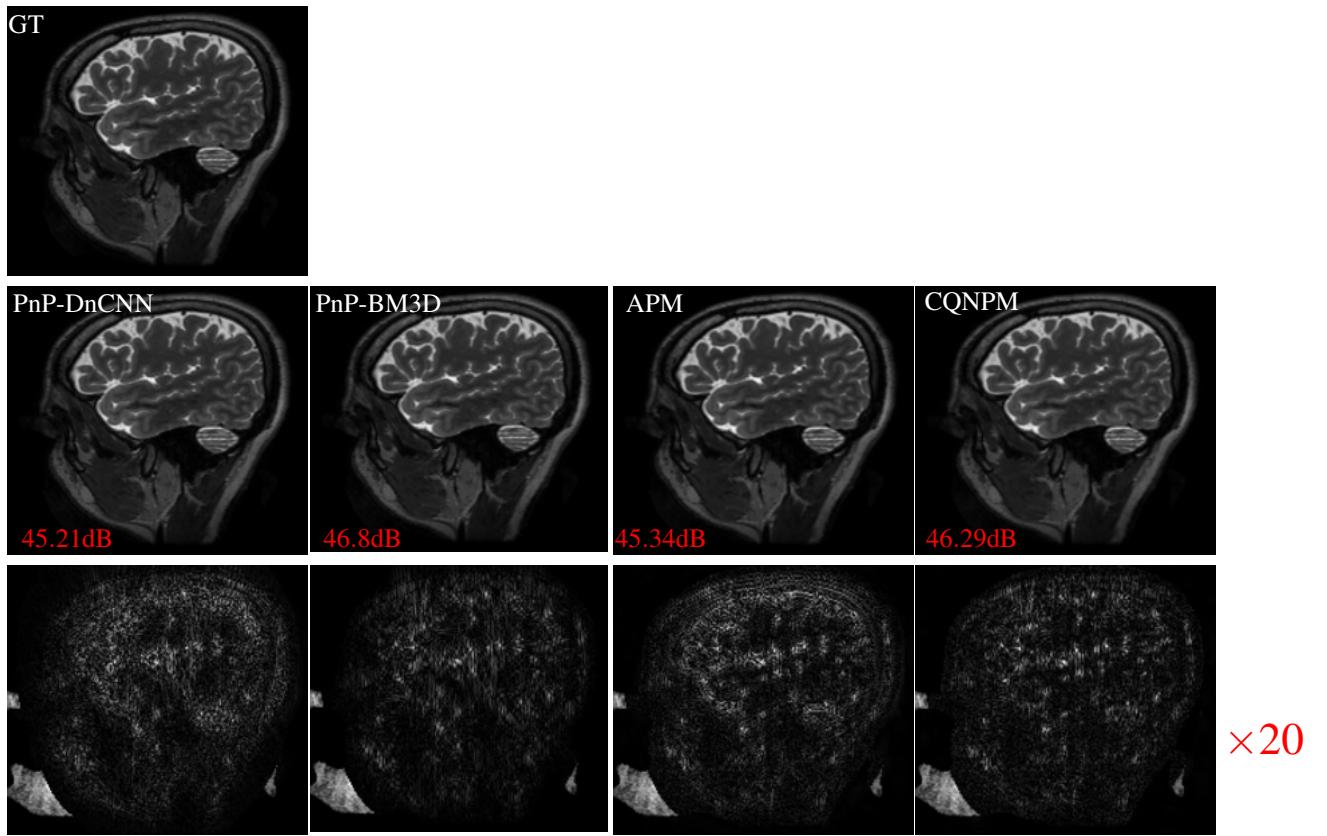


Fig. S.15. The reconstruction of PnP with DnCNN and BM3D denoisers, APM, and CQNPM with the spiral acquisition and  $\sim 30$ dB data input SNR. APM and CQNPM used the wavelet and total variation regularizers.