Fast Low Rank Column-Wise Compressive Sensing for Accelerated Dynamic MRI

Silpa Babu, Sajan Goud Lingala, and Namrata Vaswani, *Fellow, IEEE*

***Abstract*—This work develops a novel set of algorithms, alternating Gradient Descent (GD) and minimization for MRI (altGDmin-MRI1 and altGDmin-MRI2), for accelerated dynamic MRI by assuming an approximate low-rank (LR) model on the matrix formed by the vectorized images of the sequence. The LR model itself is well-known in the MRI literature; our contribution is the novel GD-based algorithms which are much faster, memory-efficient, and 'general' compared with existing work; and careful use of a 3-level hierarchical LR model. By 'general,' we mean that, with a single choice of parameters, our method provides accurate reconstructions for multiple accelerated dynamic MRI applications, multiple sampling rates and sampling schemes. We show that our methods outperform many of the popular existing approaches while also being faster than all of them, on average. This claim is based on comparisons on 8 different retrospectively undersampled multi-coil dynamic MRI applications, sampled using either 1D Cartesian or 2D pseudo-radial undersampling, at multiple sampling rates. Evaluations on some prospectively undersampled datasets are also provided. Our second contribution is a mini-batch subspace tracking extension that can process new measurements and return reconstructions within a short delay after they arrive. The recovery algorithm itself is also faster than its batch counterpart.**

***Index Terms*—Compressed sensing, low-rank, MRI.**

I. Introduction

DYNAMIC Magnetic Resonance Imaging (MRI) is a powerful imaging modality to non-invasively capture time evolving phenomena in the human body, such as the beating heart, motion of vocal tract during speaking, or dynamics of contrast uptake in brain. A long standing challenge in MRI is its slow imaging speed which restricts its full potential in the achievable spatial or temporal resolution. From a signal processing standpoint, in MRI, one measures the 2D discrete Fourier transform (FT) of a slice of the organ being imaged, one FT coefficient (or one line of coefficients) at a time. This makes the imaging slow. Accelerated/undersampled/compressive MRI is one of the key practical applications where Compressive Sensing (CS) ideas have been extensively used for speeding up the scan. This includes both work that uses traditional (sparse) CS [2], [3] for single image MRI, as well as later work that relies on the low-rank (LR) assumption, e.g., [4], [5], [6], [7].

A. Our Contributions

This work develops a fast, memory-efficient, and *'general'* algorithm, called altGDmin-MRI, for accelerated dynamic MRI by assuming an approximate LR model on the matrix formed by the vectorized images of the sequence. In analogy with traditional (sparse) CS, we refer to the problem of reconstruction with this modeling assumption as approximate LR column-wise CS (LRcCS). We should mention here that LRcCS based models have been extensively used in past work in MRI [4], [5], [6], [7], [8], [9]. Our contribution is a novel set of algorithms that assume a 3-level hierarchical LR model; and extensive experiments to demonstrate that these are both fast and "general" (with a single set of parameters, these provide good enough reconstructions for many different MRI applications, sampling schemes and rates). Our methods are modifications of a fast GD-based algorithm that was developed and theoretically analyzed in our recent work [10].

Our second contribution is mini-batch and online "subspace tracking" extensions of altGDmin-MRI that can process new measurements and return reconstructions after a much shorter data acquisition delay than the full batch solution. The online extension needs to be initialized with a mini-batch of measurements, but, after that, it returns the reconstruction as soon as a new frame of measurements arrives. The reconstruction algorithms are also faster and more memory-efficient than their batch counterpart, but with a gradual degradation in quality as batch size is reduced.

Reconstruction algorithm speed is an important concern in applications needing low latency such as real-time interactive MRI, interventional MRI, or biofeedback imaging. Moreover, immediate reconstructions can also allow for on the fly identification of certain artifacts, which can be immediately corrected (e.g., adjusting center frequency to minimize off-resonance artifacts, re-scanning if subject experiences sudden motion such as cough, etc). Finally, even in offline settings, if a reconstruction can be obtained without making the patient wait too long, it would considerably improve clinical workflow and overall throughput.

Manuscript received 6 August 2022; revised 5 December 2022, 30 January 2023, and 9 March 2023; accepted 14 March 2023. Date of publication 17 April 2023; date of current version 24 April 2023. This work was supported by NSF under Grant CIF-2115200. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Mariya Doneva. An early version of this work with the same title was presented at ICASSP 2022 [DOI: 10.1109/ICASSP43922.2022.9747549]. *(Corresponding author: Namrata Vaswani.)*

Silpa Babu and Namrata Vaswani are with the Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011 USA (e-mail: sbabu@iastate.edu; namrata@iastate.edu).

Sajan Goud Lingala is with the Department of Biomedical Engineering, University of Iowa, Iowa city, IA 52242 USA (e-mail: sajangoud-lingala@uiowa.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/TCI.2023.3263810, provided by the authors.

Digital Object Identifier 10.1109/TCI.2023.3263810

TABLE I
WE REPORT **ERROR (RECONSTRUCTION TIME IN SECONDS)**. HERE ERROR IS THE MONTE CARLO AVERAGE OF $\|\boldsymbol{X}-\boldsymbol{X}^*\|_F^2/\|\boldsymbol{X}^*\|_F^2$, OVER 50 REALIZATIONS. COMPARISONS ON A 30 X 30 X 90 IMAGE PIECE OF THE PINCAT SEQUENCE ($n=900, q=50$) USING $m=n/10$ RANDOM GAUSSIAN (GAUSSIAN) OR RANDOM FOURIER (FOUR.) MEASUREMENTS.

	altGDmin	altGDmin-mean	AltMin	MixedNorm
Gauss.	0.010 (0.15)	0.009 (0.18)	0.113 (13)	0.029 (42)
Four.	0.331 (0.50)	0.002 (0.55)	0.025 (86)	1.0 (145)

Fast reconstructions therefore help reduce the overhead cost associated with re-scheduling patient scans.

B. Existing Work

Provable LRcCS Solutions With Only Simulation Experiments. There are three existing provable solutions to the LRcCS problem. The first is an Alternating Minimization (AltMin) solution that is designed to solve a generalization of LRcCS [11], [12], and hence also solves LRcCS. The second studies a convex relaxation called mixed norm minimization (MixedNorm) [13]. The third is the altGDmin solution [10] that we modify in the current work. The convex solution is extremely slow, both theoretically and experimentally; and it has a worse sample complexity in regimes of practical interest; see Table I and see [10]. The AltMin solution is faster than the convex one, but still significantly slower than AltGDmin [10]. All the proven theoretical guarantees are for random Gaussian measurements (each entry of each $\boldsymbol{A}_k$ is an independent identically distributed standard Gaussian) but, as in case of (sparse) CS [2], [14], [15], we expect the qualitative implications to remain true also for MRI which involves use of undersampled Fourier measurements.

MRI Literature: LR and Sparsity Based Approaches. Since the work on CS in the early 2000s there has been extensive work on exploiting sparsity of the image or of the sequence in different dictionaries and bases in order to enable accelerated MRI, e.g., see [2], [16] and follow-up work. For settings where joint reconstruction of a set of similar images is needed, LR is a more flexible model since it does not require knowledge of the sparsifying basis or dictionary. MR images change slowly over time and hence are well-modeled as being approximately LR. Prior LR model based solutions from the MRI literature include [5], [6], [7], [8], [9], [17] can be classified into two broad categories: (a) methods which enforce the LR constraint explicitly, e.g., via explicit estimation of the temporal subspace from low spatial, but high temporal resolution, training data [4], [8], [9] and follow-up works in which improved self navigated Partially Separable Function (PSF) models were proposed [18], [19], and (b) methods that enforce the LR constraint in an implicit manner, e.g., via the nuclear or Schatten-p norm regularization with $p<1$ as in k-t-SLR [5] and follow-up work [6], [7]. Some of these, such as k-t-SLR [5] and PSF-sparse [9], assume both sparsity and LR models on the sequence.

A related line of work models the matrix formed by the MRI sequence as being LR plus sparse (L+S). These methods decompose the dynamic time series as a sum of a LR component modeling smoothly varying time series (e.g. object background, and/or smooth contrast changes as in perfusion MRI), and a sparse component which models the other changes in the image; see [20] (L+S-Otazo), [21] (L+S-Lin), and follow-up works, e.g. [22], [23]. Furthermore, motion often breaks down the assumption of low rank in dynamic MRI. There has been extensive work on motion estimation and compensation before imposing the structural assumptions [16], [24], [25], [26].

An important challenge with k-t-SLR, L+S-Otazo, L+S-Lin, and most of the above works, is the need for carefully tuning the parameters (regularization parameters and other hyper-parameters associated with the iterative optimization algorithm) for different dynamic MRI applications. Most published work provides results and code/parameters that work well for only the chosen application (e.g., different set of parameters are provided for cardiac perfusion, and cardiac cine MRI in the open source codes of k-t SLR and L+S-Otazo). A second limitation of the iterative optimization algorithms developed in the above works is that they are slow and memory-inefficient (process the entire matrix $\boldsymbol{X}$ at each iteration). Both these limitations are exaggerated for the motion compensation methods: these have even more parameters and are even slower.

MRI Literature: Deep Learning (DL) Methods. There has been much recent work on the use of various DL techniques in the MRI literature. The most common ones are supervised DL reconstruction schemes, e.g., [27], [28], [29], [30], [31], [32], [33], [34], [35]. These need a large numbers of fully sampled training data points. While such data can be acquired in static imaging applications (e.g., by extending scan times from cooperative volunteers, or compliant patients), it is not straightforward to acquire sufficient number of fully sampled image sequences for dynamic imaging applications, and definitely not for high time resolution applications, which warrant the need for highly under-sampled acquisitions in the first place. For this reason, a majority of supervised DL models have been used to perform reconstruction frame by frame in dynamic MRI [27], [28], [29], [30], [31], [32], [33]. This approach does not fully exploit redundancies along the temporal dimension and hence often provides worse reconstructions than sparsity or LR based methods. Moreover, DL model learning/training can be very computationally, and hence energy-wise, expensive since each parameter requires retraining the network. Finally, the parameters are learned for one specific MRI application, and the same network does not give good results for another application. Good quality training sequences can be acquired in situations where the motion may be freezed, e.g., in breath held segmented cardiac cine MRI by breathholding, and ECG gating [32], [33], [34], [35]. These remove the first limitation above, but not the other two. Also, their memory requirement is the biggest limitation.

In recent literature, unsupervised DL based approaches have been proposed and evaluated, these can exploit spatio-temporal redundancies without the need for fully sampled training datasets [36], [37]. However, since these approaches do not use a pre-trained network, but instead train the network on the test/query data, they are orders of magnitude slower compared

with both query processing time of supervised DL methods or ours or any of the LR or sparsity based methods.

Other Related Work on Online or Mini-Batch Algorithms. Other somewhat related work from the compressive sensing MRI literature that also develops online or minibatch algorithms includes [38], [39], and follow-up methods.

C. Paper Organization

We provide the notation and the 3-level approximate-LRcCS problem formulation in Section II. The algorithms are developed in Section III. Mini-batch and online subspace tracking approaches are described in Section IV. Detailed experimental evaluations and comparisons are provided in Section V. Our experimental conclusions and various other issues are discussed in Section VI. We conclude in Section VII.

II. Notation and Problem Formulation

A. Notation and Problem Setting

We use $[q] := \{1, 2, \ldots, q\}$. Everywhere, $\|.\|_F$ denotes the Frobenius norm, $\|.\|$ without a subscript denotes the (induced) l_2 norm, $^\top$ denotes (conjugate) transpose, and $\boldsymbol{M}^\dagger := (\boldsymbol{M}^\top \boldsymbol{M})^{-1}\boldsymbol{M}^\top$. For a vector $\boldsymbol{w}$, $|\boldsymbol{w}|$ computes the magnitude of each on entry of $\boldsymbol{w}$. For a scalar γ, $\mathbb{1}(\boldsymbol{w} \leq \gamma)$ returns a vector of 1s and 0s of the same size as $\boldsymbol{w}$ with 1s where $\boldsymbol{w}(k) \leq \gamma$ and zero everywhere else. Here $\boldsymbol{w}(k)$ is the k-th entry of $\boldsymbol{w}$. We use $\circ$ to denote the Hadamard product (.* operation in MATLAB). Thus $\boldsymbol{z} := \boldsymbol{w} \circ \mathbb{1}(|\boldsymbol{w}| \leq \gamma)$ zeroes out entries of $\boldsymbol{w}$ with magnitude larger than γ. For two $n \times r$ matrices $\boldsymbol{U}_1, \boldsymbol{U}_2$ with orthonormal columns, we use $\mathrm{SD}(\boldsymbol{U}_1, \boldsymbol{U}_2) := \|(\boldsymbol{I} - \boldsymbol{U}_1\boldsymbol{U}_1{}^\top)\boldsymbol{U}_2\|_F$ as the Subspace Distance (SD) between the subspaces spanned by their columns.

Let n be the number of pixels in each (unknown) image of the sequence and let q be the total number of images in the sequence. We denote the vectorized (unknown) image at frame k by $\boldsymbol{z}_k^*$; this is an n-length vector. We denote the matrix formed by all the q images in the sequence by $\boldsymbol{Z}^*$. Thus $\boldsymbol{Z}^* := [\boldsymbol{z}_1^*, \boldsymbol{z}_2^*, \ldots, \boldsymbol{z}_k^*, \ldots, \boldsymbol{z}_q^*]$ is an $n \times q$ matrix. The acquired undersampled MRI data/measurements (after some pre-processing) are linear functions of each image. For simplicity of explaining the algorithms (and for comparing the with older theoretical work in this area), we model this linear transformation using a matrix $\boldsymbol{A}_k$ of size $m_k \times n$. Thus, our goal is to recover the image sequence matrix $\boldsymbol{Z}^*$ from

$$\boldsymbol{y}_k := \boldsymbol{A}_k \boldsymbol{z}_k^*, \; k \in [q] \tag{1}$$

when $m_k \ll n$, by making structural assumptions on the matrix $\boldsymbol{Z}^*$. For single-coil dynamic MRI,

$$\boldsymbol{A}_k = \boldsymbol{H}_k \boldsymbol{F}$$

where $\boldsymbol{F}$ is an $n \times n$ matrix that models computing the 2D discrete Fourier Transform (DFT) of the vectorized image as a matrix-vector product. The matrix $\boldsymbol{H}_k$ is a matrix of size $m_k \times n$ with entries being either one or zero. It contains exactly one 1 in each row (corresponding to the observed DFT frequency location converted to 1D coordinates). The mask matrix $\boldsymbol{H}_k$ is decided by the sampling trajectory (specified in Section V). In case of multi-coil dynamic MRI with mc coils, there are mc receive channels with each measuring a subset of Fourier coefficients of a differently weighted version of the cross-section to be imaged.

In matrix-vector notation, this can be modeled as follows. Let $\boldsymbol{y}_{k,j}$ denote the measurements at the j-th coil. Then,

$$\boldsymbol{y}_k = \begin{bmatrix} \boldsymbol{y}_{k,1} \\ \boldsymbol{y}_{k,2} \\ \vdots \\ \boldsymbol{y}_{k,mc} \end{bmatrix} = \underbrace{\begin{bmatrix} \boldsymbol{H}_k\boldsymbol{F}\boldsymbol{D}_1 \\ \boldsymbol{H}_k\boldsymbol{F}\boldsymbol{D}_2 \\ \vdots \\ \boldsymbol{H}_k\boldsymbol{F}\boldsymbol{D}_{mc} \end{bmatrix}}_{\boldsymbol{A}_k} \boldsymbol{z}_k^*$$

where $\boldsymbol{D}_j = diag(\boldsymbol{d}_j, j = 1, 2, \ldots, n)$ are $n \times n$ diagonal matrices with diagonal entries (entries of the vector $\boldsymbol{d}_j$) being the coil sensitivities of the j-th coil. We should point out that $\boldsymbol{D}_j\boldsymbol{x} = \boldsymbol{d}_j \circ \boldsymbol{x}$, thus $\boldsymbol{D}_j\boldsymbol{x}$ is equivalent to weighting the l-th pixel $\boldsymbol{x}_l$ by $(\boldsymbol{d}_j)_l$. Each $\boldsymbol{y}_{k,j}$ is of length m_k, thus $\boldsymbol{y}_k$ is of length $m_k \cdot mc$.

Let $m = \max_k(m_k)$. We define the $m \times n$ matrix $\boldsymbol{Y} = [(\boldsymbol{y}_1)_{\mathrm{long}}, (\boldsymbol{y}_2)_{\mathrm{long}}, \ldots, (\boldsymbol{y}_q)_{\mathrm{long}}]$ with $(\boldsymbol{y}_k)_{\mathrm{long}}$ being the vector $\boldsymbol{y}_k$ followed by $(m - m_k)$ zeros. Similarly let $(\boldsymbol{A}_k)_{\mathrm{long}}$ be an $m \times n$ matrix with $(m - m_k)$ rows of zeros at the end. Then, the above model can also be expressed as

$$\boldsymbol{Y} = \mathcal{A}(\boldsymbol{X}^*) := [(\boldsymbol{A}_1)_{\mathrm{long}}(\boldsymbol{x}_1^*), (\boldsymbol{A}_2)_{\mathrm{long}}(\boldsymbol{x}_2^*), \ldots, (\boldsymbol{A}_q)_{\mathrm{long}}(\boldsymbol{x}_q^*)] \tag{2}$$

Similarly $\mathcal{A}^\top(\boldsymbol{Y})$ returns the $n \times q$ matrix $\boldsymbol{X} = \mathcal{A}^\top(\boldsymbol{Y}) := [(\boldsymbol{A}_1)_{\mathrm{long}}^\top(\boldsymbol{y}_1)_{\mathrm{long}}, (\boldsymbol{A}_2)_{\mathrm{long}}^\top(\boldsymbol{y}_2)_{\mathrm{long}}, \ldots, (\boldsymbol{A}_q)_{\mathrm{long}}^\top(\boldsymbol{y}_q)_{\mathrm{long}}]$.

The above matrix vector model remains valid when the observed samples are available on Cartesian grid; either they are acquired on a Cartesian grid or are mapped onto a Cartesian grid. In actual algorithm implementation, all of the above is implemented efficiently using the 2D fast Fourier transform (fft2) function along with appropriate undersampling or use of Hadamard product. When directly using true radial samples, the main ideas above and in our algorithms given below are still exactly the same, except that fft2 gets replaced by non-uniform FT (NUFT). We use the method of [40] for a fast NUFT.

Writing the model as above makes the ideas easier to follow for readers who are not MRI experts.

B. Hierarchical LR Model on Image Sequence Matrix, Z*

Most MRI sequences have a certain baseline component that is roughly constant across the entire sequence. Denote this baseline component or "mean" image by $\bar{\boldsymbol{z}}^*$. It can be verified experimentally that this mean image has significantly larger energy compared to the residual image obtained after subtracting it out. Secondly, even after mean subtraction, MR image sequences are only approximately LR, i.e., the residual image obtained after subtracting the mean and the LR components is still not zero, but has a small magnitude compared to the LR component. It is therefore easier to estimate it once the projections of the estimates of the first two components have been subtracted out. Similarly the LR component is easier to estimate once the

projections of the mean estimate have been subtracted out. Thus the following 3-level model is the most appropriate for dynamic MRI: the k-th vectorized MR image, $\boldsymbol{z}_k^*$, satisfies

$$\boldsymbol{z}_k^* = \bar{\boldsymbol{z}}^* + \boldsymbol{x}_k^* + \boldsymbol{e}_k^*, \ \text{ for all } \ k \in [q],$$

with the assumption that $\|\boldsymbol{e}_k^*\| \ll \|\boldsymbol{x}_k^*\| \ll \|\bar{\boldsymbol{z}}^*\|$, and the $\boldsymbol{x}_k^*$'s form a rank r matrix $\boldsymbol{X}^* := [\boldsymbol{x}_1^*, \dots, \boldsymbol{x}_k^*, \dots, \boldsymbol{x}_q^*]$ with $r \ll \min(n, q)$. Here $\boldsymbol{e}_k^*$ is the unstructured residual signal component, which we refer to as the modeling error. We will consider two models on $\boldsymbol{e}_k^*$. The first does not assume any structure on $\boldsymbol{e}_k^*$s except assuming that their magnitude is small. The second assumes that $\boldsymbol{e}_k^*$s are small magnitude, and sparse in the temporal Fourier domain (rows of the matrix $\boldsymbol{E}^*$ are Fourier sparse).

The first model can be interpreted as a 3-level LR model: the first level is a special case of the LR model with rank 1: the "mean image" matrix, $\bar{\boldsymbol{z}}^* \boldsymbol{1}^\top$ (where $\boldsymbol{1}$ is a vector of q 1's) has $rank = 1$; the second level is matrix $\boldsymbol{X}^*$ which has $rank = r$; and the third level is the $rank = \min(n, q)$ matrix $\boldsymbol{E}^*$. Our assumption implies $\|\bar{\boldsymbol{z}}^* \boldsymbol{1}^\top\|_F \gg \|\boldsymbol{X}^*\|_F \gg \|\boldsymbol{E}^*\|_F$.

III. AltGDmin-MRI Algorithms

A. AltGDmin-MRI Overall Idea

We develop a 3-level hierarchical algorithm that first recovers $\bar{\boldsymbol{z}}^*$, then the $\boldsymbol{x}_k^*$s, and then $\boldsymbol{e}_k^*$s. Under the modeling assumption that $\|\bar{\boldsymbol{z}}^*\| \gg \|\boldsymbol{x}_k^*\| \gg \|\boldsymbol{e}_k^*\|$, the recovery of $\bar{\boldsymbol{z}}^*$ becomes the following least squares (LS) problem:

$$\min_{\tilde{\boldsymbol{z}}} \sum_{k=1}^{q} \|\boldsymbol{y}_k - \boldsymbol{A}_k \tilde{\boldsymbol{z}}\|^2.$$

Denote its solution by $\bar{\boldsymbol{z}}$. Next, we estimate the rank-r matrix $\boldsymbol{X}^*$ (and the rank r itself) from the measurement residuals,

$$\tilde{\boldsymbol{y}}_k := \boldsymbol{y}_k - \boldsymbol{A}_k \bar{\boldsymbol{z}}, \ k \in [q]$$

by using an automated version of altGDmin for LRcCS [10] applied to $\tilde{\boldsymbol{y}}_k$s. This is described below in Section III-B. Denote its output by $\boldsymbol{X}$. The last step, which we refer to as Modeling Error Correction (MEC), involves estimating the modeling error $\boldsymbol{e}_k^*$ from the new measurement residuals

$$\tilde{\tilde{\boldsymbol{y}}}_k := \boldsymbol{y}_k - \boldsymbol{A}_k \bar{\boldsymbol{z}} - \boldsymbol{A}_k \boldsymbol{x}_k, \ k \in [q]$$

Depending on which of the two models is assumed on $\boldsymbol{e}_k^*$, the steps to estimate it are different. We describe them in Section III-C. Denote the output of either step by $\boldsymbol{E}$.

The final output is $\boldsymbol{Z} := [\boldsymbol{z}_1, \boldsymbol{z}_2, \dots, \boldsymbol{z}_q]$ with $\boldsymbol{z}_k = \bar{\boldsymbol{z}} + \boldsymbol{x}_k + \boldsymbol{e}_k$. We summarize these steps in Algorithm 1.

B. Automated AltGDmin

Recall that $\tilde{\boldsymbol{y}}_k$'s are the measurement residuals after subtracting the projections of the estimated mean. Our next goal is to estimate a rank r matrix $\boldsymbol{X}$ that minimizes

$$\tilde{f}(\boldsymbol{X}) := \sum_{k=1}^{q} \|\tilde{\boldsymbol{y}}_k - \boldsymbol{A}_k \boldsymbol{x}_k\|^2$$

Motivation for a Novel GD-Based Algorithm. We would like a GD based solution since those are known to be much faster than both AltMin and convex relaxation methods [10], [42]. As explained in detail in [10], neither of the two commonly used GD approaches from LR recovery literature, and LR matrix completion (LRMC) in particular, – projected GD on $\boldsymbol{X}$ (projGD-$\boldsymbol{X}$) [42] or alternating GD with a norm balancing term (altGDnormbal) [43], [44] – provably works for the LRcCS

Algorithm 1: *altGDmin-MRI. CGLS is the Code From [41].*

1) Solve $\min_{\bar{\boldsymbol{z}}} \sum_{k=1}^{q} \|\boldsymbol{y}_k - \boldsymbol{A}_k \bar{\boldsymbol{z}}\|^2$ using CGLS with tolerance 10^{-3} and maximum number of iterations 10. Denote the solution by $\bar{\boldsymbol{z}}$.
2) a) For each $k \in q$, compute $\tilde{\boldsymbol{y}}_k := \boldsymbol{y}_k - \boldsymbol{A}_k \bar{\boldsymbol{z}}$.
   b) Run Algorithm 2 (auto-altGDmin) with $\tilde{\boldsymbol{y}}_k, \boldsymbol{A}_k$ as its inputs. Its output is $\boldsymbol{X}$.
3) a) For each $k \in q$, compute $\tilde{\tilde{\boldsymbol{y}}}_k := \boldsymbol{y}_k - \boldsymbol{A}_k \bar{\boldsymbol{z}} - \boldsymbol{A}_k \boldsymbol{x}_k$.
   b) Run step 1 (altGDmin-MRI1) OR step 2 (altGDmin-MRI2) of Algorithm 3 The output of either is the matrix $\boldsymbol{E}$.

Output $\boldsymbol{Z} := [\boldsymbol{z}_1, \boldsymbol{z}_2, \dots, \boldsymbol{z}_q]$ with $\boldsymbol{z}_k = \bar{\boldsymbol{z}} + \boldsymbol{x}_k + \boldsymbol{e}_k$.

Algorithm 2: *auto-altGDmin: altGDmin With Automated Parameter Setting. Let* $\boldsymbol{M}^\dagger := (\boldsymbol{M}^\top \boldsymbol{M})^{-1} \boldsymbol{M}^\top$.

Input: $\tilde{\boldsymbol{y}}_k, \boldsymbol{A}_k, k \in [q]$.
Initialization:
Compute $\gamma = 36 \sum_{ki} |\tilde{\boldsymbol{y}}_{ki}|^2 / mq$, $\tilde{\boldsymbol{y}}_{k,tnc} = \tilde{\boldsymbol{y}}_k \circ \mathbb{1}\{|\tilde{\boldsymbol{y}}_k| \le \sqrt{\gamma}\}$, $\bar{m} = \sum_{k=1}^{q} m_k / q$, and compute

$$\boldsymbol{X}_0 := \Big[\frac{1}{\sqrt{m_1 \bar{m}}} (\boldsymbol{A}_1^\top \tilde{\boldsymbol{y}}_{1,tnc}), \dots, \frac{1}{\sqrt{m_k \bar{m}}} (\boldsymbol{A}_k^\top \tilde{\boldsymbol{y}}_{k,tnc}), \dots, \frac{1}{\sqrt{m_q \bar{m}}} (\boldsymbol{A}_q^\top \tilde{\boldsymbol{y}}_{q,tnc}) \Big]$$

Let $\sigma_j = \sigma_j(\boldsymbol{X}_0)$. Set $\hat{r}$ as the smallest integer for which

$$\sum_{j=1}^{r} \sigma_j^2 \ge (b/100) \cdot \sum_{j=1}^{\min(n, q, mc \min_k m_k)/10} \sigma_j^2, \ b = 85.$$

Set $\boldsymbol{U}_0 \leftarrow$ top $\hat{r}$ left singular vectors of $\boldsymbol{X}_0$
GDmin iterations: Set $T_{\max} = 70$
for $t = 1$ **to** $T_{\max}$ **do**
  Let $\boldsymbol{U} \leftarrow \boldsymbol{U}_{t-1}$.
  Update $\boldsymbol{B}$: For all $k \in [q]$, $\boldsymbol{b}_k \leftarrow (\boldsymbol{A}_k \boldsymbol{U})^\dagger \tilde{\boldsymbol{y}}_k$.
  Update $\boldsymbol{X}$: For all $k \in [q]$, $\boldsymbol{x}_k \leftarrow \boldsymbol{U} \boldsymbol{b}_k$
  Gradient compute:

$$\nabla_U f(\boldsymbol{U}, \boldsymbol{B}) \leftarrow \sum_{k=1}^{q} \boldsymbol{A}_k^\top (\boldsymbol{A}_k \boldsymbol{U} \boldsymbol{b}_k - \tilde{\boldsymbol{y}}_k) \boldsymbol{b}_k^\top$$

  If $t = 1$, set $\eta = 0.14 / \|\nabla_U f(\boldsymbol{U}, \boldsymbol{B})\|$.
  GD step for $\boldsymbol{U}$: $\boldsymbol{U}^+ \leftarrow \boldsymbol{U} - \eta \nabla_U f(\boldsymbol{U}, \boldsymbol{B})$
  Projection for $\boldsymbol{U}$: $\boldsymbol{U}^+ \overset{\text{QR}}{=} \boldsymbol{U}^+ \boldsymbol{R}^+$. Set $\boldsymbol{U}_t \leftarrow \boldsymbol{U}^+$.
  EXIT loop if $\text{SD}(\boldsymbol{U}, \boldsymbol{U}^+) / \sqrt{\hat{r}} < \epsilon_{exit} = 0.01$
end for
Output: $\boldsymbol{X} := [\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_q]$.

measurement model. The reason is that, in both cases, the estimates of the columns $\boldsymbol{x}_k^*$ are too coupled. Moreover, even for LRMC for which these approaches do work, projGD-$\boldsymbol{X}$ is memory-intensive: it requires memory of order nq; while altGDnormbal is slow: it needs a GD step size that is $1/r$ or smaller [43], [44], making it r-times slower than GD with a constant step size. The following modification, that we dub *altGDmin*, is as fast per-iteration as projGD-$\boldsymbol{X}$, as memory-efficient as altGDnormbal, and yet its estimate are only mildly coupled (they are uncoupled given an estimate of the column span of $\boldsymbol{X}^*$). Because of this, AltGDmin is amenable to analysis that helps show that, in the random Gaussian measurement setting, the algorithm converges fast – the required number of iterations to achieve ϵ accuracy grows as $\log(1/\epsilon)$ – while needing a small number of samples [10].

AltGDmin Algorithm. Rewrite the unknown matrix $\boldsymbol{X}$ as $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{B}$, where $\boldsymbol{U}$ is $n \times r$ and $\boldsymbol{B}$ is $r \times q$, and consider

$$f(\boldsymbol{U}, \boldsymbol{B}) := \tilde{f}(\boldsymbol{U}\boldsymbol{B}) = \sum_{k=1}^{q} \|\tilde{\boldsymbol{y}}_k - \boldsymbol{A}_k \boldsymbol{U} \boldsymbol{b}_k\|^2.$$

AlGDmin involves iterating over the following two steps after starting with a carefully designed initialization for $\boldsymbol{U}$.

1) For each new estimate of $\boldsymbol{U}$, we solve for $\boldsymbol{B}$ by minimizing over it while keeping $\boldsymbol{U}$ fixed at its current value. Because our measurements are column-wise decoupled ($\tilde{\boldsymbol{y}}_k$ does not depend on any other image except the k-th one), the minimization step gets decoupled for the different columns of $\boldsymbol{B}$, i.e.,

$$\min_{\boldsymbol{B}} f(\boldsymbol{U}, \boldsymbol{B}) = \sum_{k=1}^{q} \min_{\boldsymbol{b}_k} \|\tilde{\boldsymbol{y}}_k - \boldsymbol{A}_k \boldsymbol{U} \boldsymbol{b}_k\|^2.$$

   This problem is now a very quick column-wise least squares (LS) problem with $\boldsymbol{b}_k$ being an r-length vector and $\boldsymbol{A}_k\boldsymbol{U}$ being an $m_k \times r$ matrix. Thus, the complexity is only $m_k r^2 \cdot q$ (for solving q individual LS problems) plus the cost of computing $\boldsymbol{A}_k\boldsymbol{U}$ for all $k \in [q]$. The total complexity for this step is thus similar to that of one GD step for $\boldsymbol{U}$.
2) We use projected GD for updating $\boldsymbol{U}$: one GD step w.r.t. $\boldsymbol{U}$ followed by projecting onto the space of orthonormal matrices (by using QR decomposition).

Finally, since $f(\boldsymbol{U}, \boldsymbol{B})$ is not convex in the unknowns $\{\boldsymbol{U}, \boldsymbol{B}\}$, the above algorithm needs a careful initialization of one of them. Using the standard approach from LR recovery literature, we can initialize $\boldsymbol{U}$ by computing the top r left singular vectors of the matrix

$$\boldsymbol{X}_0 = \left[\frac{1}{m}\boldsymbol{A}_1^\top \tilde{\boldsymbol{y}}_1, \ldots, \frac{1}{m}\boldsymbol{A}_k^\top \tilde{\boldsymbol{y}}_k, \ldots, \frac{1}{m}\boldsymbol{A}_q^\top \tilde{\boldsymbol{y}}_q\right]$$

When the measurement matrices $\boldsymbol{A}_k$ are random Gaussian, a truncation/thresholding step, that zeros out entries of $\tilde{\boldsymbol{y}}_k$ with magnitude much larger than the root mean squared value $\sqrt{\sum_{ki} \tilde{\boldsymbol{y}}_{ki}^2/mq}$, is required on each $\tilde{\boldsymbol{y}}_k$ before computing the above matrix. This step helps to filter out the very large measurements (those whose $\tilde{\boldsymbol{y}}_{ki}^2$ is much larger than the expected value) and helps ensure that the new matrix has entries which are sub-Gaussian[1]. In the MRI setting, since the matrices $\boldsymbol{A}_k$ are subsampled Fourier, if the measurements are indeed noise-free as assumed in (1) and the matrix satisfies the incoherence assumption given earlier, then the above is not needed. The reason is, in this case, $\|\boldsymbol{a}_{ki}\|^2 = 1$ [2] and so, $\tilde{\boldsymbol{y}}_{ki}^2 \le \max_k \|\boldsymbol{z}_k^* - \bar{\boldsymbol{z}}\|^2$ for all i, k. However, in practice, there have be either measurement or image outliers: in some acquisitions, there may be occasional large noise or, certain images $\boldsymbol{z}_k^*$ may be outliers, e.g., this would happen if the subject took a deep breath during acquisition. The truncation step helps filter out such measurements. If there are no outliers or large noise, then it does nothing and hence it does not worsen performance either. Another minor change is needed: since $m_k \neq m_1$ (time varying number of measurements), in order to prove a guarantee similar to our result from [10], one needs to replace the $1/m$ factor by $1/\sqrt{m_k \bar{m}}$ where $\bar{m} = \sum_{k=1}^{q} m_k/q$. We specify $\boldsymbol{X}_0$ with these modifications in line 5 of Algorithm 2.

AltGDmin Parameter Setting. The parameters for AltGDmin are the rank r, the GD step size η, and the maximum number of iterations T along with a stopping criterion to exit the loop sooner if the estimates do not change much.

For approximately LR matrices, there is no one correct choice of r. We use the following constraints to find a good approach. We need our choice of rank, $\hat{r}$, to be sufficiently small compared to $\min(n, q)$ for the algorithm to take advantage of the LR assumption. Moreover, for the LS step for updating $\boldsymbol{b}_k$'s to work well (for its error to be small), we also need it to be small compared with $mc \min_k m_k$. Based on just these constraints, one can set $\hat{r} = \min(n, q, mc \min_k m_k)/10$. Or, one can compute the "b% energy threshold" of the first $\min(n, q, mc \min_k m_k)/10$ singular values, i.e., compute $\hat{r}$ as the smallest value of r for which

$$\sum_{j=1}^{\hat{r}} \sigma_j(\boldsymbol{X}_0)^2 \ge (b/100) \cdot \sum_{j=1}^{\min(n,q,mc\min_k m_k)/10} \sigma_j(\boldsymbol{X}_0)^2.$$

for a $b \le 100$. Here $\sigma_j(\boldsymbol{X}_0)$ is its j-th singular value. We use this latter approach with $b = 85$. We have experimented with other values as well in the 80-95% range, and the algorithm is not very sensitive to this choice.

We set the GD step size $\eta = 0.14/\|\nabla_U f(\boldsymbol{U}_0, \boldsymbol{B}_0)\|$ where $\boldsymbol{U}_0, \boldsymbol{B}_0$ are the initial estimates. Assuming that the gradient norm decreases over iterations, this implies that $\eta \cdot \|\nabla_U f(\boldsymbol{U}_t, \boldsymbol{B}_t)\| \le 0.14 < 1$ always. Since $\|\boldsymbol{U}_t\| = 1$ (due to the QR decomposition step), this ensures that a GD step is never too big. To decide T (maximum number of iterations), we stop the GD loop when $\text{SD}(\boldsymbol{U}_{t-1}, \boldsymbol{U}_t) < \epsilon_{exit}\sqrt{r}$ while setting $T_{\max} = 70$ so that no more than 70 iterations are run. We set $\epsilon_{exit} = 0.01$.

We summarize the complete algorithm, with the above parameter settings, in Algorithm 2.

As suggested in [10], one can also set η as $\eta = c/(m\|\boldsymbol{U}_0\boldsymbol{B}_0\|^2)$ with a $c < 1$. This is a conservative approach

[1] In our proofs, this allows us to use the sub-Gaussian Hoeffding inequality [45] to get our desired bound the initialization error.

[2] or some constant that is the same for all i, k and depends on how the Fourier matrix is normalized

that is needed for proving guarantees which are only sufficient conditions and will lead to slower convergence.

AltGDmin Guarantee. Theorem 2.1 of [10] proved the following for AltGDmin with parameters as specified there.

Theorem 3.1: Suppose that $\boldsymbol{Z}^* = \boldsymbol{X}^*$, i.e., it is a matrix with rank r, and $\tilde{\boldsymbol{y}}_k = \boldsymbol{y}_k$. Suppose also that each $\boldsymbol{A}_k$ is $m \times n$ and i.i.d. random Gaussian. Assume that $\max_k \|\boldsymbol{x}_k^*\|^2 \leq \mu^2 \|\boldsymbol{X}^*\|^2/q$ for a constant μ (incoherence parameter) that is only a little larger than one. Let $\boldsymbol{x}_k$ denote the AltGDmin estimates after $T = C\kappa^2 \log(1/\epsilon)$ iterations. If $mq \geq C\kappa^4\mu^2(n+q)r^2 \log(1/\epsilon)$, if the algorithm parameters are set as described in [10, Theorem 2.1], then, with probability at least $1 - n^{-10}$, $||\boldsymbol{x}_k^* - \boldsymbol{x}_k|| \leq \epsilon ||\boldsymbol{x}_k^*||$ for all $k = 1, 2, \ldots, q$.

In the above result κ is the ratio of the first to the r-th singular value of $\boldsymbol{X}^*$. Treating κ, μ as numerical constants, this is equivalent to requiring that $\max_k \|\boldsymbol{x}_k^*\|^2 \leq C\|\boldsymbol{X}^*\|_F^2/q$ for a constant C. In practice, this means that, the different (vectorized) images $\boldsymbol{x}_k^*$ in the sequence have similar enough energy so that the maximum energy of any one of them is not much larger than its average, $\|\boldsymbol{X}^*\|_F^2/q$. This fact is very valid for MRI datasets.

C. Model Error Correction (MEC): Two Models and Algorithms

MEC Using Model 1 (No Structure on $\boldsymbol{E}^*$): AltGDmin-MRI1. Recall that this model assumes no structure on $\boldsymbol{e}_k^*$ except that it has small magnitude. We thus recover each $\boldsymbol{e}_k$ individually by solving

$$\min_{\boldsymbol{e}} \|\tilde{\boldsymbol{y}}_k - \boldsymbol{A}_k \boldsymbol{e}\|^2$$

for each k, while imposing the assumption that $\|\boldsymbol{e}\|^2$ is small. An indirect way to enforce this, while also getting a fast algorithm, is to start with a zero initialization and run only a few iterations of GD to solve the above minimization problem.

Parameter Setting for MEC-1. We use the Stanford Conjugate Gradient LS (CGLS) code [41] for solving the above minimization. We used this code with tolerance of 10^{-36} and maximum number of iterations 3.

This is summarized in Algorithm 3.

MEC Using Model 2 (Temporal Fourier Sparse $\boldsymbol{E}^*$): AltGDmin-MRI-2. Our second model assumes Fourier sparsity of the modeling error along the time axis. To be precise, we are assuming that

$$\boldsymbol{S}^* := \mathcal{F}_{row}(\boldsymbol{E}^*)$$

is a row sparse matrix (matrix whose rows are sparse vectors). Here, the operator $\mathcal{F}_{row}$ computes the 1D DFT of each row of its argument. We thus have the following model on the images' matrix $\boldsymbol{Z}^*$:

$$\boldsymbol{Z}^* = (\bar{\boldsymbol{z}}^* \boldsymbol{1}^\top) + \boldsymbol{X}^* + \boldsymbol{E}^*, \;\; \boldsymbol{E}^* := \mathcal{F}_{row}^{-1}(\boldsymbol{S}^*)$$

with $\|\boldsymbol{E}^*\|_F \ll \|\boldsymbol{X}^*\|_F \ll \sqrt{q}\|\bar{\boldsymbol{z}}^*\|$ and $\boldsymbol{S}^*$ being row sparse.

To estimate $\boldsymbol{E}^*$ under this model, we use the Iterative Soft Thresholding Algorithm (ISTA) for sparse recovery [46] which was also used in [20]. For recovering an unknown sparse $\boldsymbol{s}$ from $\boldsymbol{y} := \boldsymbol{A}\boldsymbol{s}$, this starts with a zero initialization, $\boldsymbol{s} = 0$, and runs the iterations: $\boldsymbol{s} \leftarrow \text{SThr}_\omega(\boldsymbol{s} + \boldsymbol{A}^\top(y - \boldsymbol{A}\boldsymbol{s}))$. Here $\text{SThr}_\omega(\boldsymbol{s})$ is the Soft-Thresholding operator; it zeroes out entries of $\boldsymbol{s}$ that are smaller than ω while shrinking the larger magnitude entries towards zero by ω, i.e. $[\text{SThr}_\omega(\boldsymbol{s})]_i = sign(\boldsymbol{s}_i)(|\boldsymbol{s}_i| - \omega)$ if $|\boldsymbol{s}_i| > \omega$ and $[\text{SThr}_\omega(\boldsymbol{s})]_i = 0$ otherwise.

For our model, this translates to the following iteration. Compute the residual $\tilde{\tilde{\boldsymbol{Y}}} := \boldsymbol{Y} - \mathcal{A}(\bar{\boldsymbol{z}}\boldsymbol{1}^\top) - \mathcal{A}(\boldsymbol{X})$. Update $\boldsymbol{E}$ by running the following iteration starting with $\boldsymbol{E} = 0$:

$$\boldsymbol{E} \leftarrow \mathcal{F}_{row}^{-1}(\text{SThr}_\omega(\mathcal{F}_{row}(\boldsymbol{E} + \mathcal{A}^\top(\tilde{\tilde{\boldsymbol{Y}}} - \mathcal{A}(\boldsymbol{E})))))$$

This is summarized in Algorithm 3.

The temporal Fourier sparsity model has been used for imposing the L+S assumption for dynamic MRI in [20], [21], and follow-up works. We should clarify that, in this work, we are not imposing the L+S model, instead we are assuming a 3-level hierarchical model, with sparsity being used to model the residual in the third level. The assumption $\|\boldsymbol{E}^*\|_F \ll \|\boldsymbol{X}^*\|_F$ makes our model different from the regular L+S model which assumes $\boldsymbol{Z}^* = \boldsymbol{X}^* + \boldsymbol{E}^*$ with no assumption on one of them being smaller in magnitude than the other. From our experiments in Section V, in an average sense, our algorithm, altGDmin-MRI2 that uses our model, gives better reconstructions (both in terms of error and visually). However, this may be either because the assumed models are different or because the reconstruction algorithms are very different too: we use a GD-based algorithm, while [20], [21], use different algorithms to solve the convex relaxation of the L+S model. This issue will be explored in more detail in future work where we plan to also develop and evaluate a GD-based algorithm under the L+S assumption.

Parameter Setting for MEC-2. We used soft thresholding with threshold as given in Algorithm 3.

D. Implementation

We write things as above only for ease of explanation. In our implementation, we never use matrix-vector multiplication for computing $\boldsymbol{A}_k\boldsymbol{x}$ or $\boldsymbol{A}_k^\top\boldsymbol{y}$, since that is much more memory intensive and much slower than using Fast FT (FFT). Also, we use various MATLAB features and linear algebra tricks to remove "for" loops wherever possible. Code is provided at the link given in Section V.

IV. AltGDmin Based Subspace Tracking

The algorithms discussed so far are batch methods, i.e., they require waiting for all the measurements to be taken. This means that they cannot be used in applications that require near real-time reconstructions (explained in Section I-A).

A. Mini-Batch Subspace Tracking

Consider the pseudo-real-time setting in which the algorithm processes each new mini-batch of the data (here a set of α consecutive $\boldsymbol{y}_k$s) as soon as it arrives. Thus, instead of waiting for all q measurements $\boldsymbol{y}_k$ to be obtained, it only waits for a new set of $\alpha < q$ measurements before processing them. For algorithms that use the LR assumption on the data, such an algorithm can be referred to as a *Subspace Tracking* solution since it is implicitly assuming that consecutive mini-batches of

Algorithm 3: *altGDmin-MRI1 or altGDmin-MRI2.*

- **Unstructured MEC (altGDmin-MRI1)**
  Run the following
  – For each $k \in [q]$, run 3 iterations of CGLS to solve $\min_{\boldsymbol{e}} \|\tilde{\tilde{\boldsymbol{y}}}_k - \boldsymbol{A}_k \boldsymbol{e}\|^2$. Denote the output by $\boldsymbol{e}_k$.

or

- **Sparse MEC (altGDmin-MRI2)**
  Run the following ISTA algorithm
  – $\boldsymbol{E}_0 = \boldsymbol{0}$, $\tau = 0$. Repeat the following steps
    * $\boldsymbol{M}_\tau \leftarrow \mathcal{F}_{row}(\boldsymbol{E}_\tau + \mathcal{A}^\top(\tilde{\tilde{\boldsymbol{Y}}} - \mathcal{A}(\boldsymbol{E}_\tau)))$
    * If $\tau = 0$, set $\omega = 0.001 \cdot \|\boldsymbol{M}_0\|_{\max}$ where $\|.\|_{\max}$ is the maximum magnitude entry of the matrix.
    * $\tau \leftarrow \tau + 1$
    * $\boldsymbol{E}_\tau \leftarrow \mathcal{F}_{row}^{-1}(\text{SThr}_\omega(\boldsymbol{M}_\tau))$
    Until $\tau = 10$ or $\frac{\|\boldsymbol{M}_\tau - \boldsymbol{M}_{\tau-1}\|_F}{\|\boldsymbol{M}_{\tau-1}\|_F} < 0.0025$

Output $\boldsymbol{E}$.

Algorithm 4: *altGDminMRI-ST1 and altGDminMRI-ST2: Mini-batch Subspace Tracking with MEC model 1 (ST1) or model 2 (ST2).*

1) Let $j = 1$. Run Algorithm 1 on first mini-batch of α_1 $\boldsymbol{y}_k$s. Thus $q \equiv \alpha_1$ and we let $T_{\max,1} = 70$.
Denote its final subspace estimate by $\boldsymbol{U}^{(1)}$.
2) For each $j > 1$ do
  a) Run Algorithm 1 with the following two changes: (i) replace the Initialization step of altGDmin (Algorithm 2) by $\boldsymbol{U}_0 \leftarrow \boldsymbol{U}^{(j-1)}$; and (ii) set $T_{\max,j} = 5$.
  Denote its final subspace estimate by $\boldsymbol{U}^{(j)}$.
End For

data lie close to the same or slightly different r-dimensional subspaces. One can utilize this "slow subspace change" assumption in the following fashion. For the first mini-batch, use the altgdmin-MRI algorithm with q replaced by α. For later mini-batches, use altgdmin-MRI with two changes. (1) Use the final estimated $\boldsymbol{U}$ from the previous mini-batch, denoted $\boldsymbol{U}^{(j-1)}$, as the initialization for the current one. This means that we replace lines 2-6 of Algorithm 2 by $\boldsymbol{U}_0 \leftarrow \boldsymbol{U}^{(j-1)}$. (2) Second, reduce the maximum number of iterations for the j-th minibatch, denoted $T_{\max,j}$, to a much lower value for $j > 1$ than for $j = 1$.

The complete algorithm is summarized in Algorithm 4. We use $T_{\max,1} = 70$ and $T_{\max,j} = 5$ for $j \geq 2$. Depending on the MEC step being used, we refer to the resulting algorithms as altGDminMRI-ST1 and altGDminMRI-ST2.

B. Online Subspace Tracking

In certain other applications, after an initial short delay, a true real-time (fully online) algorithm is needed. This means that, each time a new $\boldsymbol{y}_k$ is obtained, it should return a new estimate $\boldsymbol{x}_k$. To obtain such an algorithm we eliminate the mean computation step and the $\boldsymbol{U}$ update steps in online mode. Both these are computed only for the first mini-batch and used at all later times. Suppose the first mini-batch consists of α_1 frames. For all times $k > \alpha_1$, we use the estimated mean $\bar{\boldsymbol{z}}$ and the estimated $\boldsymbol{U}$ from the first mini-batch. For $k > \alpha_1$, for each new $\boldsymbol{y}_k$, we only update $\boldsymbol{b}_k$ and $\boldsymbol{e}_k$ and only using MEC model 1 (unstructured $\boldsymbol{e}_k^*$). We summarize this algorithm, altGDminMRI-onlineST, in Algorithm 5.

Algorithm 5: *altGDminMRI-onlineST: Online Subspace Tracking.*

1) For the first mini-batch of α frames, run Algorithm 1. Denote the computed mean image by $\bar{\boldsymbol{z}}^{(1)}$ and the final subspace estimate by $\boldsymbol{U}^{(1)}$.
2) For $k > \alpha + 1$ do:
  a) Compute $\tilde{\boldsymbol{y}}_k = \boldsymbol{y}_k - \boldsymbol{A}_k \bar{\boldsymbol{z}}^{(1)}$
  b) Let $\boldsymbol{U} = \boldsymbol{U}^{(1)}$.
  c) Compute $\boldsymbol{b}_k \leftarrow (\boldsymbol{A}_k \boldsymbol{U})^\dagger \tilde{\boldsymbol{y}}_k$.
  d) Compute $\tilde{\tilde{\boldsymbol{y}}}_k = \tilde{\boldsymbol{y}}_k - \boldsymbol{A}_k \boldsymbol{U} \boldsymbol{b}_k$
  e) Compute $\boldsymbol{e}_k$ by running 3 iterations of CGLS to solve $\min_{\boldsymbol{e}} \|\tilde{\tilde{\boldsymbol{y}}}_k - \boldsymbol{A}_k \boldsymbol{e}\|^2$. Denote the output by $\boldsymbol{e}_k$.
  f) Output $\boldsymbol{x}_k = \bar{\boldsymbol{z}}_{(1)} + \boldsymbol{U} \boldsymbol{b}_k + \boldsymbol{e}_k$
End For

V. Experiments

The code for all our experiments is posted at https://github.com/Silpa1/comparison_of_algorithms. We show results on both retrospective and prospective datasets. The retrospective undersampling is either 2D golden-angle based pseudo-radial[3] or 1D Cartesian undersampling using the sampling scheme of [20]. Radial sampling provides a way to non-uniformly undersample the 2D Fourier plane in such a way that more samples are acquired in the center of the 2D Fourier plan (low frequency regions in both dimensions). Directly using radially sampled data requires use of the computationally expensive non-uniform FT (NUFT) which makes the algorithm very slow. Previous research has shown negligible loss in image quality if the polar coordinates of radially undersampled data are regridded onto the Cartesian grid, followed by use of fast FT (FFT) algorithms in the reconstruction algorithms [48], [49], [50]. In fact, for CS-based methods, there is sometimes an improvement in reconstruction quality as well when using this type of regridded data (pseudo-radial data) and we observe this in some of our experiments too.

All experiments were conducted in MATLAB on the same PC. AltGDmin-MRI1 and AltGDmin-MRI2 were compared with (1) the three provable techniques – mixed norm min (MixedNorm) [13], AltMin (changed for the current linear setting) [12] and basic AltGDmin [10]; with 4 state-of-the-art methods from the LR-based MRI literature – k-t-SLR (uses an L&S model) [5], L+S-Otazo [20], L+S-Lin [21], and PSF-sparse [4], [9]. For all comparisons, we used author provided code: mixed norm min (MixedNorm): https://www.dropbox.com/sh/lywtzc0y9awpvgz/AABbjuiuLWPy_8y7C3GQKo8pa?dl=0,

[3]The angular increment between successive radial spokes is determined by the golden angle (111.25 degrees) [47]. The starting point is changed over time so that the sampling masks are different for different images in the sequence. The golden angle ensures maximum incoherent kspace coverage over time.

AltMin: https://github.com/praneethmurthy/, k-t-SLR: code was emailed by the author to us, L+S-Lin: https://github.com/JeffFessler/reproduce-l-s-dynamic-mri, L+S-Otazo: https://cai2r.net/resources/ls-reconstruction-matlab-code/, PSF-sparse: http://mri.beckman.illinois.edu/software.html.

In all our experiments, one set of parameters was used. For our two methods, parameters were set as given earlier in the stepwise algorithms. For L+S-Lin, we evaluated it with using author-provided parameters for cardiac data and for PINCAT. Overall the cardiac parameters gave reduced errors and hence we used these in all experiments. For L+S-Otazo and ktSLR also, author-provided cardiac parameters were used, since these gave the best results. In the codes of kt-SLR, L+S-Otazo and L+S-Lin, the input k-space data is needed in a different format (the sequence of frequency locations is different, due to different uses of the "fftshift" in MATLAB). To deal with this, we converted all algorithms and ours so that all took input k-space data in the same format as L+S-Lin. L+S-Lin code has parameter settings that require the input k-space data to be normalized in a certain way. To generate the best performance for it, we did this normalization where needed.

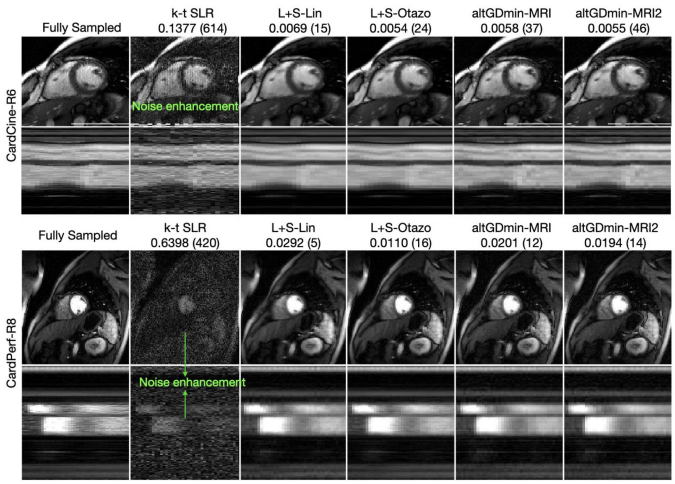


Fig. 1. Retrospective, Cartesian, CardPerf and CardCine: Comparisons of reconstruction algorithms on CardPerf-R8 and CardCine-R6 datasets. In row 1 and row 3, we show one original frame (14th frame) and its reconstructions. In row 2 and row 4, we show the corresponding time profile images. The chosen cut line is shown in Fig. 3. **Error (recon time)** of each algorithm is reported below the algorithm name. Observe that kt-SLR has considerable noise enhancement in both the datasets. AltGDmin-MRI2 (proposed) provides qualitatively similar results to L+S-Lin and L+S-Otazo.

A. Comparison With Provably Correct Algorithms

We compared altGDmin-basic and altGDmin-mean (altGDmin-MRI1 without the last MEC step) with MixedNorm [13] and the AltMin algorithm of [11], [12], modified for the linear LRcCS problem (replace the PR step for updating $\boldsymbol{b}_k$'s by a simple LS step). Since these algorithms were designed and evaluated only for random Gaussian measurements, their code cannot be easily modified to handle large-sized image sequences (requires use of the fft operator to replace actual matrix-vector multiplications) or complicated MRI sampling patterns. Hence, for this experiment, we use a 30 x 30 piece of the PINCAT image sequence with 50 frames ($n = 900, q = 50$ and simulate (i) random Gaussian $\boldsymbol{A}_k$'s and (ii) random Fourier $\boldsymbol{A}_k$'s (the sampling mask is obtained by selecting m 2D-DFT frequencies uniformly at random from all n possible ones). We report the results in Table I for $m = n/10$. As can be seen, altGDmin and altGDmin-mean are more than 90-times faster than both altMin and MixedNorm. Also, altGDmin-mean has the lowest error. In the Gaussian setting, both altGDmin and altGDmin-mean have similar and small errors. In the Fourier setting, since we are selecting random frequencies, if enough lower frequencies are not selected, the error is large. When the mean image is estimated and subtracted, the energy in the lower spatial frequencies is a lot lower. This is why use of mean subtraction (altGDmin-mean) significantly reduces the error in this case.

B. Comparisons on Retrospectively Undersampled Datasets

The error value that we report in this and later sections is normalized scale-invariant mean squared error (N-S-MSE) computed as follows $Error = (\sum_{k=1}^{q} \text{dist}^2(\boldsymbol{x}_k^*, \hat{\boldsymbol{x}}_k))/\|\boldsymbol{X}^*\|_F^2$ where $\text{dist}^2(\boldsymbol{x}^*, \hat{\boldsymbol{x}}) = \|\boldsymbol{x}^* - \hat{\boldsymbol{x}}\frac{\hat{\boldsymbol{x}}^\top \boldsymbol{x}^*}{\|\boldsymbol{x}\|^2}\|^2$ is the scale invariant distance between two vectorized images with "scale" being a complex number. The reconstructed images can be complex-valued. We also report the time taken to reconstruct the entire sequence. The reporting format is **Error (Reconstruction Time in seconds)**.

We used a total of 20 datasets: 2 datasets from [21] which were retrospectively undersampled using Cartesian variable density random undersampling at reduction factors (R); R=8, R=6 respectively – cardiac perfusion R8 (CardPerf-R8) and cardiac cine R6 (CardCine-R6); and 6 other applications that were retrospectively pseudo-radially undersampled with 4, 8 and 16 radial lines – brain-T2-T1-rho (Brain), free breathing ungated cardiac perfusion (UnCardPerf), a long but low-resolution speech sequence (Speech), two cardiac cine datasets from the OCMR database (CardOCMR16, CardOCMR19) [51], and PINCAT. PINCAT data was single-coil, while all others were multi-coil. Image sequence sizes: CardPerf ($n = 16384, q = 40$), CardCine ($n = 65536, q = 24$), Brain ($n = 16384, q = 24$), Speech ($n = 4624, q = 2048$), UnCardPerf ($n = 31104, q = 200$), CardOCMR16 ($n = 28800, q = 15$), CardOCMR19 ($n = 27648, q = 25$), PINCAT ($n = 16384, q = 50$).

Error and time comparisons are reported in Table II. In its last row, we display Average Error (Average Reconstruction Time) with the average taken over the 20 previous rows. Visual comparisons are shown in Figs. 1–3. Observe that our approaches have the best errors and are also the fastest. Both visually, and error-wise, AltGDmin-MRI2 has either the best or a close second-best reconstruction quality in all cases, while also being very fast. It is not always the fastest, but it is the fastest for long sequences and fast-enough for all. AltGDmin-MRI1 also has low errors (only slightly higher than MRI2), and is faster than MRI2. On the other hand, no other approach is consistently good across all 20 datasets. Lastly, for the most undersampled (4 radial lines) case, our methods have much lower errors than

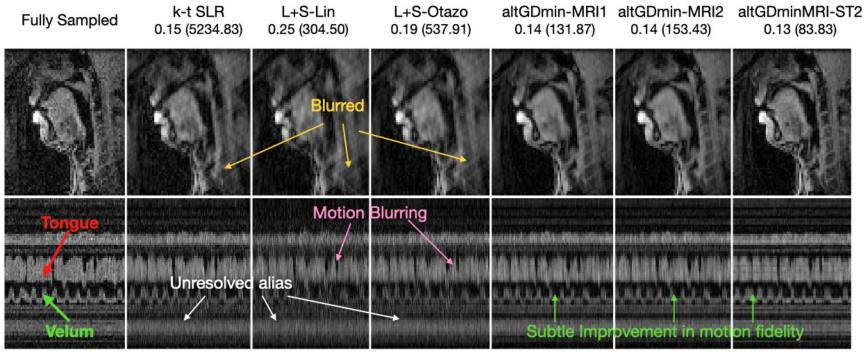
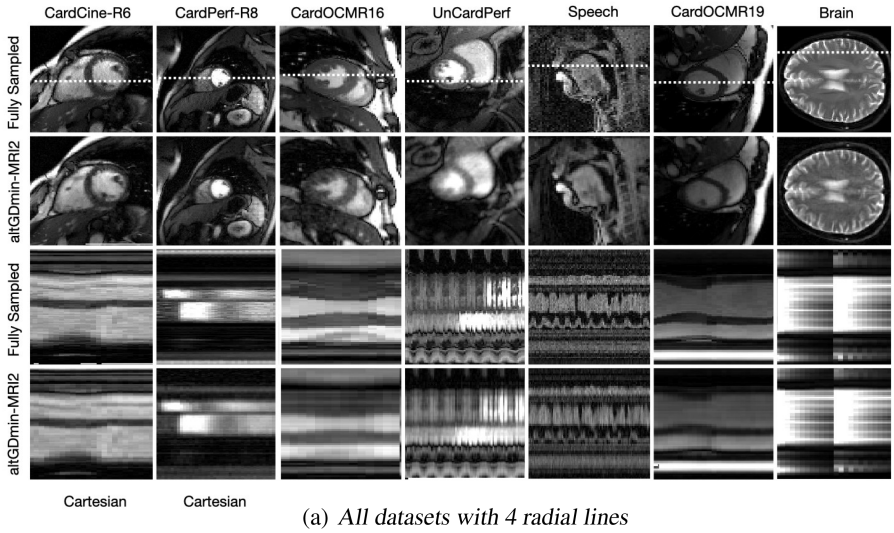
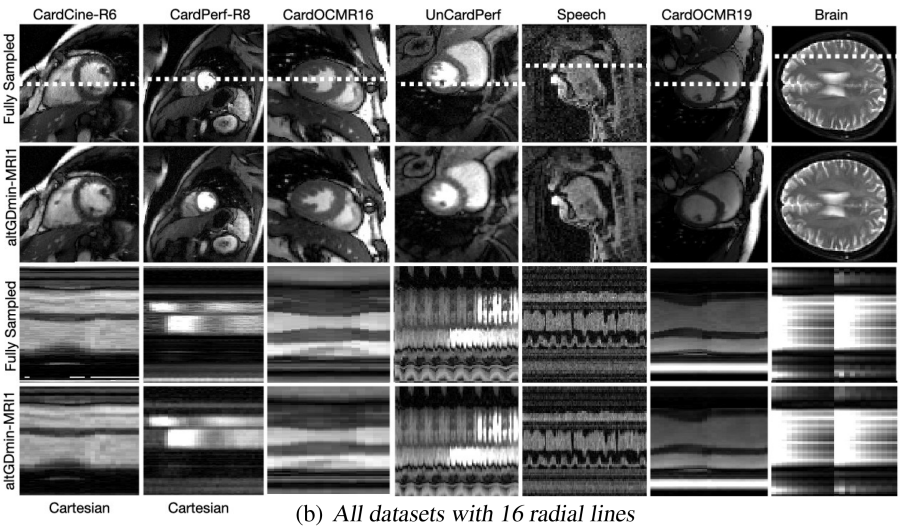


Fig. 2. Retrospective, Pseudo-radial (4 radial lines), Speech: In row 1, we show one original frame and its reconstructions. In row 2, we show the time profile image. This is a cut through the tongue and velum depicting the motion of these articulators. Only 100 out of 2048 image frames are shown for the sake of brevity. The chosen cut line is shown in Fig. 3. **Error (recon. time)** of each algorithm is reported below the algorithm name. Observe that k-t-SLR, L+S-Lin, and L+S-Otazo reconstructions have motion blurring and/or alias artifacts. In contrast, altGDmin-MRI2 reconstructions even with smaller batch sizes produce superior reconstructions. Last column shows a subspace tracking result.



(a) *All datasets with 4 radial lines*



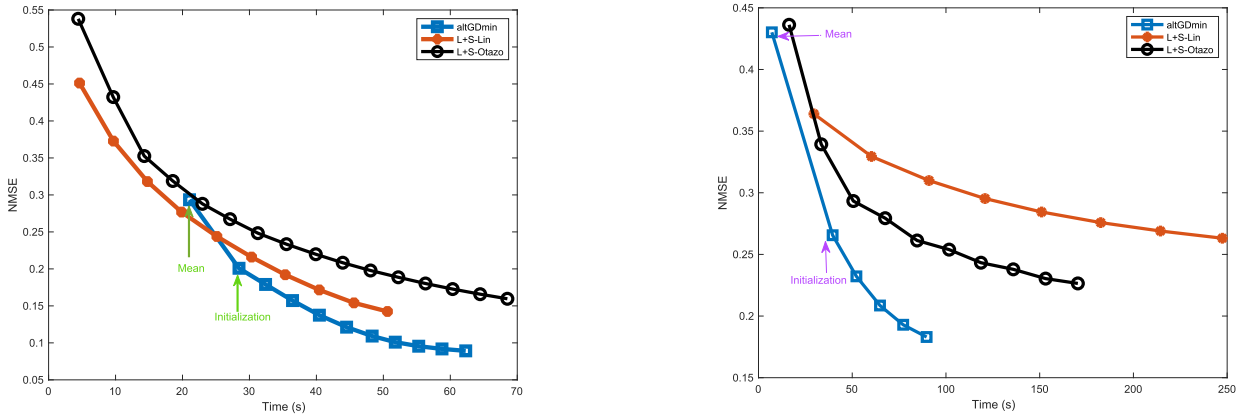
(b) *All datasets with 16 radial lines*

Fig. 3. Retrospective, Pseudo-radial, All datasets: This figure is organized differently than the previous ones. Row 1 shows one original (fully sampled) image for all datasets, row 2 shows reconstruction using only AltGDmin-MRI2. Row 3 and row 4 are the original and reconstructed time profile images. Observe that it gives good results in all applications without any parameter tuning. Comparing Fig(a) and Fig(b), we observe that with 16 radial lines the blurring effect in the recons reduced.

TABLE II

TABLE FORMAT IS **ERROR (RECON TIME IN SECONDS)**. THE LAST ROW SHOWS **AVERAGE-ERROR (AVERAGE-RECONSTRUCTION TIME IN SECONDS)** OVER ALL 20 ROWS OF RESULTS. FOR PSF-SPARSE, WE GENERATED DATA USING THE K-T SAMPLING SCHEME FROM THEIR PAPER [9] WHILE ENSURING SAME TOTAL NUMBER OF SAMPLES AS THE REST OF THE COMPARED METHODS.

Dataset	kt-SLR	L+S-Otazo	L+S-Lin	altGDmin-MRI1	altGDmin-MRI2	altGDminMRI-ST2, α=64,100	PSF-sparse
Cartesian							
CardPerf-R8	0.6398 (420.71)	0.0110 (16.93)	0.0292 (5.99)	0.0201 (12.01)	0.0194 (14.74)		
CardCine-R6	0.1377 (614.53)	0.0054 (24.57)	0.0069 (15.51)	0.0058 (37.55)	0.0055 (46.08)		
Pseudo-radial							
Brain(4)	0.0093 (127.92)	0.0167 (5.65)	0.0173 (2.53)	0.0125 (3.70)	0.0121 (4.86)		
Brain(8)	0.0034 (102.33)	0.0086 (4.15)	0.0095 (2.48)	0.0054 (3.87)	0.0051 (5.11)		
Brain(16)	0.0014 (75.57)	0.0049 (2.81)	0.0062 (2.44)	0.0027 (3.84)	0.0024 (4.98)		
Speech(4)	0.1543 (5234.83)	0.1991 (537.91)	0.2545 (304.50)	0.1416 (131.87)	0.1395 (153.43)	0.1174 (86.57)	
Speech(8)	0.0593 (5261.49)	0.1107 (491.11)	0.1284 (306.16)	0.0991 (152.35)	0.0952 (176.35)	0.0873 (87.64)	
Speech(16)	0.0203 (5288.61)	0.0550 (426.10)	0.0557 (304.45)	0.0580 (236.85)	0.0540 (261.39)	0.0542 (100.75)	
UnCardPerf(4)	0.0894 (4150.72)	0.0910 (189.88)	0.1424 (50.68)	0.0695 (70.97)	0.0684 (92.31)		
UnCardPerf(8)	0.0442 (3472.96)	0.0591 (120.55)	0.0632 (50.44)	0.0470 (67.79)	0.0451 (90.66)	0.0531 (81.48)	
UnCardPerf(16)	0.0206 (2873.30)	0.0370 (88.44)	0.0329 (50.47)	0.0298 (69.08)	0.0275 (90.16)	0.0328 (70.77)	
CardOCMR16(4)	0.0362 (227.92)	0.0293 (10.73)	0.0515 (2.75)	0.0092 (10.52)	0.0092 (15.42)		0.3803 (2.39)
CardOCMR16(8)	0.0045 (225.98)	0.0064 (8.37)	0.0101 (2.73)	0.0033 (7.26)	0.0033 (8.64)		0.1200 (7.91)
CardOCMR16(16)	0.0015 (162.12)	0.0035 (4.64)	0.0030 (2.73)	0.0015 (5.39)	0.0014 (6.69)		0.0020 (5.24)
CardOCMR19(4)	0.0216 (399.70)	0.0251 (18.70)	0.0698 (5.06)	0.0095 (11.58)	0.0094 (14.10)		
CardOCMR19(8)	0.0043 (409.01)	0.0092 (13.05)	0.0149 (5.06)	0.0051 (10.49)	0.0050 (12.91)		
CardOCMR19(16)	0.0020 (269.01)	0.0052 (7.20)	0.0044 (5.05)	0.0032 (9.47)	0.0030 (12.00)		
PINCAT(4)	0.0445 (34.85)	0.0381 (8.04)	0.1054 (2.26)	0.0278 (1.63)	0.0278 (1.77)		
PINCAT(8)	0.0216 (31.54)	0.0162 (3.91)	0.0208 (2.22)	0.0166 (1.36)	0.0166 (1.31)		
PINCAT(16)	0.0095 (23.31)	0.0065 (2.70)	0.0047 (2.25)	0.0097 (1.08)	0.0097 (1.13)		
av-Err (av-Time)	0.0663 (1470.3)	0.0369 (99.3)	0.0515 (56.3)	0.0289 (42.4)	0.0280 (50.7)		



(a) *Error-Time: UnCardPerf* (b) *Error-Time: Speech*

Fig. 4. We plot the Error after each iteration t (y-axis) and the time taken until iteration t (x-axis) for the UnCardPerf and Speech datasets for alGDminMRI (without the MEC steps), L+S-Lin, and L+S-Otazo. Observe that altGDmin-MRI converges fastest.

all the others. We also compare with the PSF-sparse algorithm of [9] for one dataset, UnCardPerf. This is an improved version of the original kt-PCA method of [4]. This approach does not work for any given sampling scheme. Hence, for it, we used the author provided "k-t sampling" code (Cartesian undersampling with certain k-space locations sampled at each time, and the rest of the locations being highly undersampled). We changed its sampling rate parameter to make its undersampling factor, $\sum_k m_k/(nq)$, similar to ours. From Table II, clearly, the error of PSF-sparse is much higher (38- and 12- times higher) in the 4 and 8 radial lines cases.

In this table, we have also added results for one mini-batch size for altGDminMRI-ST2. As can be seen, in the speech sequence case, the recovery error is in fact smaller. In the cardiac case, there is marginal increase in recovery error. In both cases, the time taken is lesser than that of altGDmin-MRI2. More detailed evaluation is described below in Section V-E.

C. Error Versus Iteration Time Plot

In Fig. 4, we plot the error after each iteration t (y-axis) and the time taken until iteration t (x-axis) for altGDmin-MRI without the MEC step (just altGDmin+mean), L+S-Lin and L+S-Otazo for the UnCardPerf and Speech datasets with 4 radial lines. Such a plot is more informative than error versus iteration since it allows one to both see the error decay with iterations and to also see the time taken for each iteration by each method. The first marker in the altGDmin-MRI plot is time taken after the

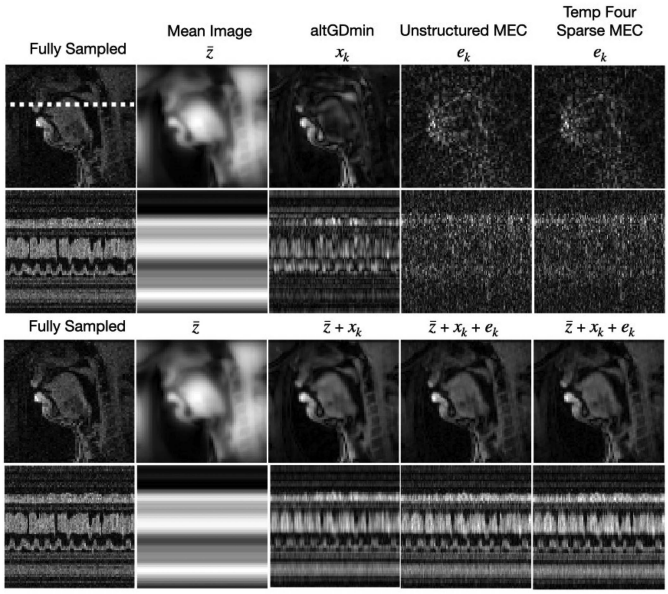


Fig. 5. Demonstrating the utility of each step of our algorithms. In row 1, we show one original frame (1470th frame) and the estimates of each step: the mean estimate $\bar{\boldsymbol{z}}$, the LR estimate $\boldsymbol{x}_k$ obtained using altGDmin, and the modeling error estimates $\boldsymbol{e}_k$ under our two models (unstructured and sparse in temporal Fourier domain). In row 2, we show the corresponding time profile images. In row 3, we show how each component improves image quality by showing $\bar{\boldsymbol{z}}$, $\bar{\boldsymbol{z}}+\boldsymbol{x}_k$, and $\bar{\boldsymbol{z}}+\boldsymbol{x}_k+\boldsymbol{e}_k$. In row 4, we show the corresponding time profile images. Observe that altGDmin contributes to the details of each image. The last MEC steps improves finer details. .

mean computation, the second marker also adds the time taken after the initialization step, the third also adds the time taken by first altGDmin iteration, and so on. All three algorithms have their own exit loop and maximum number of iterations and hence each plot ends at a different time. For both the datasets, observe that altGDmin converges much faster than the other two compared methods. kt-SLR is not compared because it is much slower. From both figures, we can also observe that use of our initialization step is very useful, it helps reduce the error significantly. With just the mean and initialization steps, the normalized error gets reduced to 0.2 and 0.27 respectively.

D. Effect of Each Step of altGDmin-MRI

In Fig. 5, for one dataset, we show the output of each step of our two algorithms. As can be seen each of the 3 steps improves the reconstruction quality.

E. Subspace Tracking Algorithms Evaluation

We evaluate these algorithms on the Speech and the UnCardPerf sequence pseudo-radially undersampled using 16, 8, and 4 radial lines. We use these two sequences since these have a larger value of q. This is needed to ensure that the mini-batch sizes are not too small or at least the first mini-batch size is large enough. We provide the results in Table III. We evaluate mini-batch ST (Algorithm 4), with $T_{\max,1} = 70$ and $T_{\max,j} = 5$ for $j > 1$ for decreasing values of α. Observe that for mini-batch sizes up to $\alpha \geq 64$, there is no appreciable increase in error. But the improvement in reconstruction time is very significant. It

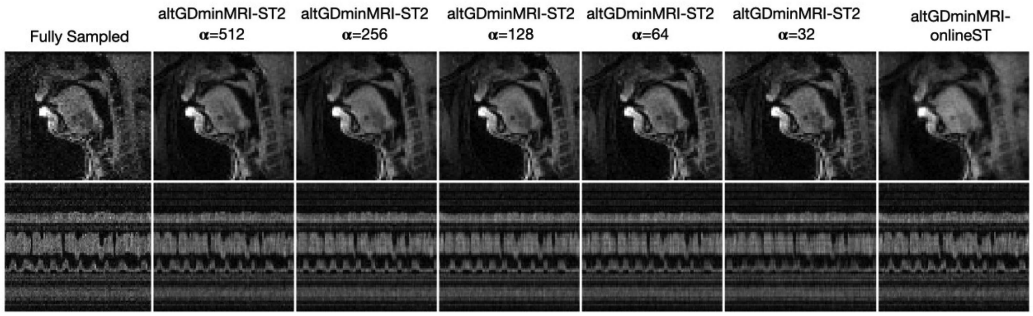
TABLE III
SUBSPACE TRACKING RESULTS: COMPARING ALTGDMINMRI-ST1, ALTGDMINMRI-ST2, AND ONLINEST ALGORITHMS FOR THE SPEECH AND THE UNCARDPERF SEQUENCES RETROSPECTIVELY UNDERSAMPLED USING 16, 8, 4 RADIAL LINES AND DIFFERENT CHOICES OF MINI-BATCH SIZE α.

α	altGDminMRI-ST1	altGDminMRI-ST2	altGDminMRI-onlineST
Cardiac 16 radial lines			
100	0.0355 (50.20)	0.0328 (70.77)	0.0782 (37.61)
50	0.0349 (45.29)	0.0323 (67.47)	0.0958 (27.35)
Cardiac 8 radial lines			
100	0.0556 (59.51)	0.0531 (81.48)	0.1038 (48.05)
50	0.0579 (52.17)	0.0555 (75.73)	0.1271 (36.61)
Cardiac 4 radial lines			
100	0.0705 (135.54)	0.0695 (164.97)	0.1273 (138.44)
50	0.0750 (87.31)	0.0737 (109.71)	0.1586 (70.05)
Speech 16 radial lines			
1024	0.0575 (192.93)	0.0534 (219.95)	0.0895 (136.76)
512	0.0578 (148.53)	0.0534 (175.08)	0.1139 (94.19)
256	0.0584 (109.91)	0.0538 (136.52)	0.1318 (63.22)
128	0.0584 (86.09)	0.0538 (113.76)	0.1468 (46.94)
64	0.0589 (74.05)	0.0542 (100.75)	0.1613 (40.43)
32	0.0604 (67.24)	0.0554 (93.57)	0.1771 (37.45)
Speech 8 radial lines			
1024	0.0888 (189.69)	0.0858 (212.63)	0.1260 (143.03)
512	0.0882 (124.93)	0.0850 (154.88)	0.1525 (89.61)
256	0.0885 (93.39)	0.0851 (119.25)	0.1735 (59.94)
128	0.0889 (74.42)	0.0853 (98.52)	0.1933 (45.93)
64	0.0911 (66.05)	0.0873 (87.64)	0.2115 (40.23)
32	0.1007 (61.85)	0.0960 (83.86)	0.2292 (36.87)
Speech 4 radial lines			
1024	0.1202 (576.38)	0.1191 (594.78)	0.1642 (537.66)
512	0.1257 (212.70)	0.1242 (241.69)	0.2023 (184.72)
256	0.1201 (93.79)	0.1183 (119.46)	0.2226 (63.93)
128	0.1177 (71.25)	0.1157 (96.15)	0.2490 (46.26)
64	0.1195 (65.27)	0.1174 (86.57)	0.2715 (41.74)
32	0.1321 (62.40)	0.1293 (83.83)	0.2977 (37.70)

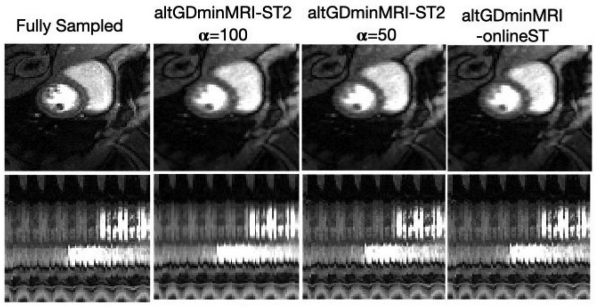
is much faster than any of the other algorithms compared in Table II. We also compare with full online ST (Algorithm 5). In this case, there is a significant increase in error as the value of initial mini-batch α decreases. But the speed is even better. We show visuals for different values of α in Fig. 6. As can be seen, the quality is as good as that of the full batch one. The reason is this approach is modeling a slowly changing subspace rather than a fixed one, and this can be a better assumption for speech sequences.

F. Experiments on 3 Prospectively Under-Sampled Radial Data

We compare altGDmin-MRI2 with L+S-Lin (the overall best algorithm in terms of performance and speed amongst all compared methods) and with the baseline reconstruction obtained using direct inverse Fourier Transform (FT), this uses zeros where data is not observed. Results are show in Figs. 7, 8(a), and 8(b). Our first dataset is a radially undersampled dynamic contrast enhanced (DCE) abdomen taken from [20], [21]. The k-space data dimensions in the dataset were: 384 read out points, 21 radial spokes (with golden angle based angular increments) per frame, 28 time frames, and 7 virtual coils after PCA based coil compression. Here 21 spokes per frame means the total $21 * 28 = 588$ spokes were arranged into 28 time frames with



(a) *Speech: 4 radial lines*



(b) *Ungated Cardiac Perfusion (UnCardPerf): 16 radial lines*

Fig. 6. Subspace Tracking results: Retrospective, Pseudo-radial, Speech(4 radial lines) and UnCardPerf (16 radial lines) reconstructed using altGDminMRI-ST2. In row 1, we show one original frame and its reconstructions. In row 2, we show the time profile image. As expected, smaller batch sizes produce faster reconstructions. With larger batch sizes, the temporal sharpness (or motion fidelity) improve but only subtly.

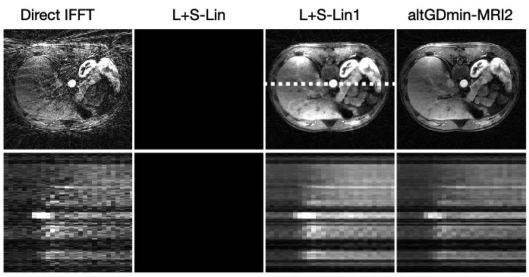


Fig. 7. Prospective, Radial, Abdomen: We compare our approach with direct iNUFFT (baseline) and with L+S-Lin. When running L+S-Lin with the cardiac perfusion parameters (the ones used in all earlier experiments), the algorithm completely fails, see column 2. Thus, we also implemented it using author-provided parameters for this dataset; we refer to this as L+S-Lin1 which is shown in column 3. This gives a good recovery similar to that of altGDmin-MRI2.

the first 21 spokes forming the first time frame, the next 21 forming the next time frame and so on.

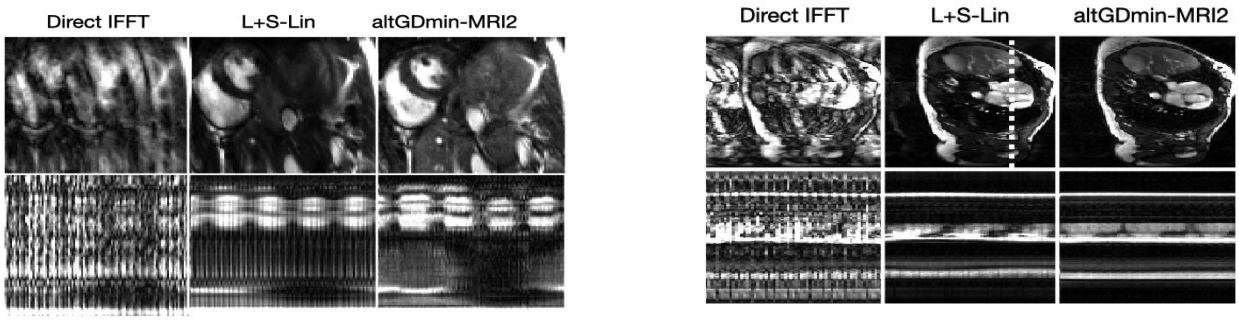
For this dataset, we modified our code to use the non-uniform Fast FT (NUFFT) code from [40] to replace FFT. From Fig. 7, notice that, in the altGDmin-MRI2 (atGDmin2) reconstruction, we can observe the contrast uptake dynamics through the liver. The other blood vessels are well resolved too. L+S-Lin (with the parameters used in all previous experiments) failed completely. It returns a black image. So we used the author provided parameters to see if that works, we label it L+S-Lin1. This provides qualitatively similar results to ours.

Our next dataset is a Cartesian undersampled breath held cardiac cine from the OCMR database with k-space dataset dimensions 384 read out points, 14 phase encode lines per time frame, 137 time frames and 18 coils. From Fig. 8(a), observe that L+S-Lin reconstruction has motion blurring and/or alias artifacts. In contrast, altGDmin-MRI2 result is good comparatively. Our last dataset is another Cartesian undersampled cardiac dataset from the OCMR database with k-space data dimensions 384 read out points, 16 phase encode lines per time frame, 65 time frames and 34 coils. In Fig. 8(b), we compare the reconstructions. Both L+S-Lin and altGDmin-MRI2 reconstructions are good.

VI. Discussion

We first explain why our algorithms are "general," memory-efficient, and fast. Next, we provide a summary of our experimental conclusions, a discussion of the most related works, and of our subspace tracking methods (ST). We end with describing the limitations of our work and ways to improve it.

AltGDmin-MRI Methods are "General". This is because these have only a few parameters and are not very sensitive to their choices. Moreover, our goal was to develop a *single algorithm, with one fixed set of parameters*, that provides a good enough performance across a wide range of applications, sampling schemes, and sampling rates, while also being very fast; and not necessarily the best one for each case. Hence we do not do any application-specific tuning. The reason our methods have only a few parameters is two-fold. First, the LR model does not require any parameter except the assumed rank (or parameters for the algorithm to estimate the rank). This is unlike



(a) OCMR dataset 1 (b) OCMR dataset 2

Fig. 8. Prospective, Cartesian, Cardiac (OCMR): In both figures, in row 1, we show one image for direct inverse FFT (IFFT) of undersampled kspace data (column 1), reconstructions using L+S-Lin (column 2) and our method, altGDmin-MRI2 (column 3). In row 2, we show the time profile images for the three reconstructions in the same order. Observe that for both OCMR cardiac datasets, altGDmin-MRI2 reconstructions are qualitatively better compared to L+S-Lin.

sparsity or structured sparsity models, which require either picking the most appropriate sparsifying basis or dictionary or learning one, and in either case there are many parameters that need to be carefully set to pick the best basis/dictionary. Second, AltGDmin is a simple GD based algorithm. Besides rank, its only other parameters are the GD step size η and the maximum number of iterations $T_{\max}$, along with a loop exit threshold ϵ_{exit}. The algorithm is not sensitive to the choice of $T_{\max}$ as long as it is large enough. AltGDmin-MRI1 is its modification that can be understood as assuming a 3-level hierarchical LR model that (i) first estimates the baseline/mean image across all frames (approximately the $r = 1$ case), and computes the measurement residual by removing this estimate; (ii) next it uses the residual as input to auto-altGDmin ($r = \hat{r}$ case, where $\hat{r}$ is the automatically estimated rank), and (iii) finally it estimates the residual error in the above mean + LR model column-wise (this is the $r = \min(n, q)$ case). In case of altGDmin-MRI2, this last residual error is assumed to be temporally Fourier sparse. The mean computation step (LS problem solved using the Stanford CGLS code) and the MEC steps also require only two parameters each, while being sensitive to only one of them: loop exit tolerance and maximum number of iterations. The ISTA algorithm used in case of altGDmin-MRI2 needs one more parameter - the threshold for soft thresholding.

Memory-Efficiency. The effect of memory complexity is not very evident in this paper since we only do single-slice imaging, but will be when working with dynamic multi-slice imaging. In that case, n would be the number of voxels in each volume (all slices at one time) with q still being the sequence length. AltGDmin uses the $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{B}$ factorization, with $\boldsymbol{U}$ and $\boldsymbol{B}$ being matrices with r columns and rows respectively. Here r is the assumed (low) rank. Storing and processing $\boldsymbol{U}, \boldsymbol{B}$ requires memory of size only $\max(n, q)r$ instead of nq. Our approach for estimating r caps its value at $r \leq \min(n, q, m)/10$. The initial mean computation step estimates an n-length vector $\bar{\boldsymbol{z}}$ using GD (CGLS code); it can be made memory-efficient by using a for-loop to compute the gradient sum at each iteration. For AltGDmin-MRI1, the last MEC step is done individually for each $\boldsymbol{e}_k^*$. Thus both these steps have memory complexity of order n only. The MEC step of AltGDmin-MRI2 can be made memory-efficient by processing each row separately using a for-loop over all n rows. This step thus has memory complexity of order q. Thus the overall memory complexity of both AltGDmin-MRI1 and AltGDmin-MRI2 is order $\max(n, q)r$ with $r \ll n, q$, while that of most other LR-based methods (except AltMin and PSF-sparse) is nq.

Time Complexity and Speed. The most computationally expensive part of both algorithms is altGDmin. The two expensive steps of this algorithm are (i) computing the gradient w.r.t. $\boldsymbol{U}$, and (ii) computing $\boldsymbol{A}_k\boldsymbol{U}$ for the LS step to estimate $\boldsymbol{b}_k$s. When implemented for the MRI setting using the 2D-FFT operators, $\boldsymbol{A}_k\boldsymbol{U}$ requires computing r 2D-FFTs for $n_1 \times n_2$ images with $n = n_1 \cdot n_2$. One 2D-FFT needs time of order $n_2 n_1 \log n_1 + n_1 n_2 \log n_2 \leq 2n \log n$. Thus this step needs time of order $nr \log n$. The gradient computation needs q 2D inverse FFTs, thus its cost is $nq \log n$. Since $r \leq q$, the overall cost is order $nq \log n$ per iteration. Without provable guarantees for the MRI setting, we cannot say anything theoretically about the number of iterations required. From our experiments (see Fig. 4 and Tables I, II), our algorithm error decays faster than that of all the other compared approaches.

Summary of Experiments. In the highly undersampled setting of only 4 radial lines (32 times acceleration), AltGDmin-MRI2 and AltGDmin-MRI1 have much lower errors and better visual recon quality than all compared methods, while also being the fastest. In all cases, on average, the AltGDmin-MRI2 errors and time taken are still the best (lowest). Our Subspace Tracking based mini-batch modifications are even faster, while providing almost comparable, or better, quality reconstructions for batch sizes $\alpha \geq 64$. For this reason, these can only be used on longer sequences. Notice that the last model error correct (MEC) step of altGDmin-MRI1 uses a maximum of only 3 iterations, while that of altGDmin-MRI2 uses a maximum of 10 iterations. As noted by an anonymous reviewer, this may be one reason for the latter having slightly better performance. We tried using 10 maximum iterations also for MEC of altGDmin-MRI1 (not shown); with this, its computed error does reduce further to almost the same level as MRI2. However, for a some cases, the visual quality of MRI2 still is better.

Discussion of Compared Methods and DL-Based Methods. MixedNorm, AltMin, and kt-SLR are much slower compared with AltGDmin-MRI. For many of the pseudo-radial datasets,

kt-SLR has the lowest errors and visual performance, with that of AltGDmin-MRI2 being either as good or only slightly worse. But kt-SLR has very large errors for the Cartesian undersampled ones and consequently its average error is large. A possible reason for this is that its parameters have been tuned for the pseudo-radial sampling. On the other hand, L+S-Otazo and L+S-Lin have low errors for the two Cartesian undersampled datasets (both of these were used in their papers); but have larger errors for most of the pseudo-radial 4 and 8 radial lines (highly undersampled) datasets. The likely reason again is similar.

We also compared with the PSF-sparse algorithm of [52] for one dataset; see Table II. This only works with the kt-sampling scheme developed by the authors, so we used this sampling for it while changing the code parameters to ensure comparable $mc\sum_k m_k/(nq)$ (comparable acceleration factor). PSF-sparse first estimates the row space of the unknown image sequence matrix using the data at the frequency locations fully sampled along time. This means it estimates the row span of $\boldsymbol{B}$ first. Next, it estimates the column space, span of $\boldsymbol{U}$, by minimizing the error w.r.t. all observed data while also imposing a (temporal Fourier) sparsity constraint. The first step needs sufficient number of low frequency samples for accurate recovery. This is why, when $\sum_k m_k$ is small, either the first step recon is bad or there are almost no samples left to get a good estimate at all frequencies. In contrast, our algorithms simultaneously estimate $\boldsymbol{U}, \boldsymbol{B}$ using multiple alternating iterations initialized using a carefully designed spectral initialization for $\boldsymbol{U}$.

Supervised DL methods need a lot of training data. Hence for most MRI applications (except breath held cardiac or ECG gating), these are designed for image-based reconstruction and cannot model the spatiotemporal correlations across the sequence. Consequently, their performance on dynamic MRI is often worse than that of LR or LR+S based methods which do not need any training data. Moreover, the energy cost for training the DLs for entire image sequences can be prohibitive.

On the other hand, the new unsupervised DL methods can model the spatiotemporal correlations without training data, but these are slower on query data by orders of magnitude. The reason is these do not use a pre-trained network but instead train the deep network on the query data. As an example, to reconstruct a typical speech sequence with $n = 100^2 = 10000$ pixels and $q = 500$ frames, this class of approaches needs 45mins to an hour on a GPU. Our algorithm only needs 1-2 minutes for a similar dataset. Also, these need careful hyperparameter tuning (cannot be used for different MRI applications without tuning) while our methods do not.

Clarification That No Binning is Needed in This Work. None of our proposed approaches (not even the subspace tracking ones) need raw data sorted into various predefined temporal phases (such as defining cardiac phases) or "bins". As an example, for the cardiac cine sequence, we did not need knowledge of which frames are pre-contrast and which are post-contrast.

Practical Utility of Mini-Batch and Online Subspace Tracking (ST). Besides operating in mini-batch mode, the algorithm speed of mini-batch ST is also much faster, and, from our experiments, the increase in recovery error is insignificant for mini-batch size $\alpha \geq 64$. In the speech sequence, the error actually decreases when using ST. The reason is this is a very long sequence and the subspace likely changes over time. The ST method tracks this change. With online ST, the reconstruction quality does suffer. But its speed is very fast, and the algorithm works in true real time mode after the first mini-batch of α_1 frames is processed. Hence, in practice, a combination of the two approaches would be the most useful: in online mode, obtain real-time reconstructions which are very fast but often not very accurate; and follow it up with mini-batch updates (after the mini-batch has arrived) that are much more accurate.

Because we do not need any binning, and because the ST approaches provide a reconstruction as soon as a small mini-batch of time frames are acquired, these would be suitable for a variety of real-time ungated type of applications, where the raw data is continuously being acquired without an a-priori definition of temporal phases. Example applications include free breathing ungated cardiac cine, real-time dynamic MRI of vocal tract shaping during speech production, free breathing dynamic contrast enhanced MRI (also see example in Fig. 8). There is a need to have a fast on-the-fly reconstructions to inspect quality of the dynamic reconstructions. For example, in dynamic speech MRI, low latency reconstructions are useful to adjust for localization planes, adjust center frequency to minimize off-resonance artifacts, and to visualize articulatory movements in biofeedback type experiments. Similarly, in real-time ungated free breathing cardiac MRI experiments, a fast reconstruction without a prior definition of cardiac phases allows one to visualize arrhythmic events on the fly.

Limitations and How to Tune Parameters to Tailor to Applications. As can be seen from our results, the reconstruction performance is not the best for all applications. In a few cases, there is some visible blurring. The reason is our goal was to show what our algorithm can achieve using a single set of parameters. The blurring can be reduced by increasing the rank $\hat{r}$ that is used in AltGDmin while still ensuring it is sufficiently smaller than $\min(mc\min_k m_k, n, q)$; and/or increasing the number of MEC step iterations. Making the loop exit criterion more robust can also help, e.g., instead of exiting after only two consecutive estimates of $\boldsymbol{U}$ are close in subspace distance, one could exit if this happens consecutively for a few iterations. The algorithm speed can be improved further by using a variable step size η_t for the GD step: use larger values in the initial iterations and reduce it over time. Lastly, the initial mean computation step of altGDmin-MRI1 and altGDmin-MRI2 can be made more robust by using truncation similar to that used in the initialization of altGDmin.

For both mini-batch and online ST methods, we need to initialize using a mini-batch. From our experiments, the first mini-batch needs to be at least 32-64 frames long. It is possible to replace even the first mini-batch step by a fully-online one if we replace the GD for updating $\boldsymbol{U}$ by stochastic GD that only uses the gradient w.r.t. the data term for the current $\boldsymbol{y}_k$. However, the tradeoff will be a worsened reconstruction quality. This approach will be explored in future work. Moreover, our ST methods are not very robust to outliers, e.g., due to a deep breath by the subject. One way to make them somewhat robust is to re-initialize every so often. The second solution is to develop an L+S model based algorithm.

VII. Conclusions and Future Work

We developed a set of fast, memory-efficient, and "general" algorithms for accelerated/undersampled approximately LR dynamic MRI. Here, "general" means that it works with the same set of parameters for multiple MRI applications, sampling schemes and rates. We also developed an altGDmin-MRI-based subspace tracking solution that operates in mini-batch mode and provides comparable reconstruction quality while being even faster. Unlike the supervised DL methods, our algorithms do not need any training data, and also do not need the hours or days of training time and computational power. Unlike the newer unsupervised DL based methods, our methods are orders of magnitude faster. In experimental comparisons with the best known existing LR, L+S or L&S based methods (kt-SLR, L+S-Otazo, L+S-Lin, PSF-sparse, AltMin, MixedNorm), on average, our methods have the best reconstruction quality, and are also the fastest. Future work will explore reconstruction of 3D+t data (multi-slice dynamic imaging) using the proposed approach and also try to develop tensor-based modification of our ideas. A second goal will be to design a fast and memory-efficient altGDmin-based algorithm for the L+S model. We will also try to design a truly online ST method that does not need mini-batch initialization.

References

[1] S. Babu, S. Nayer, S. G. Lingala, and N. Vaswani, "Fast low rank compressive sensing for accelerated dynamic MRI," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2022, pp. 1346–1350.

[2] M. Lustig, D. Donoho, and J. M. Pauly, "Sparse MRI: The application of compressed sensing for rapid mr imaging," *Magn. Reson. Med.*, vol. 58, 6, pp. 1182–1195, Dec. 2007.

[3] U. Gamper, P. Boesiger, and S. Kozerke, "Compressed sensing in dynamic MRI," *Magn. Reson. Med.*, vol. 59, no. 2, pp. 365–373, Jan. 2008.

[4] Z.-P. Liang, "Spatiotemporal imagingwith partially separable functions," in *Proc. IEEE 4th Int. Symp. Biomed. Imag.: From Nano Macro*, 2007, pp. 988–988.

[5] S. G. Lingala, Y. Hu, E. DiBella, and M. Jacob, "Accelerated dynamic MRI exploiting sparsity and low-rank structure: Kt SLR," *IEEE Trans. Med. Imag.*, vol. 30, no. 5, pp. 1042–1054, May 2011.

[6] M. Chiew, S. M. Smith, P. J. Koopmans, N. N. Graedel, T. Blumensath, and K. L. Miller, "k-t faster: Acceleration of functional MRI data acquisition using low rank constraints," *Magn. Reson. Med.*, vol. 74, no. 2, pp. 353–364, 2015.

[7] M. Chiew, N. N. Graedel, J. A. McNab, S. M. Smith, and K. L. Miller, "Accelerating functional MRI using fixed-rank approximations and radial-cartesian sampling," *Magn. Reson. Med.*, vol. 76, no. 6, pp. 1825–1836, 2016.

[8] H. Pedersen, S. Kozerke, S. Ringgaard, K. Nehrke, and W. Y. Kim, "k-t PCA: Temporally constrained K-T blast reconstruction using principal component analysis," *Magn. Reson. Med.*, vol. 62, no. 3, pp. 706–716, 2009.

[9] B. Zhao, J. P. Haldar, A. G. Christodoulou, and Z.-P. Liang, "Image reconstruction from highly undersampled-space data with joint partial separability and sparsity constraints," *IEEE Trans. Med. Imag.*, vol. 31, no. 9, pp. 1809–1820, Sep. 2012.

[10] S. Nayer and N. Vaswani, "Fast and sample-efficient federated low rank matrix recovery from column-wise linear and quadratic projections," *IEEE Trans. Inf. Theory*, vol. 69, no. 2, pp. 1177–1177, Feb. 2023.

[11] S. Nayer, P. Narayanamurthy, and N. Vaswani, "Phaseless PCA: Low-rank matrix recovery from column-wise phaseless measurements," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 4762–4762.

[12] S. Nayer and N. Vaswani, "Sample-efficient low rank phase retrieval," *IEEE Trans. Inf. Theory*, vol. 67, no. 12, pp. 8190–8190, Dec. 2021.

[13] R.S. Srinivasa, K. Lee, M. Junge, and J. Romberg, "Decentralized sketching of low rank matrices," in *Proc. Neural Inf. Process. Syst.*, 2019, pp. 10101–10101.

[14] E. J. Candes and T. Tao, "Near optimal signal recovery from random projections: Universal encoding strategies?," *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5406–5425, Dec. 2006.

[15] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[16] L. Feng, L. Axel, H. Chandarana, K. T. Block, D. K. Sodickson, and R. Otazo, "Xd-grasp: Golden-angle radial MRI with reconstruction of extra motion-state dimensions using compressed sensing," *Magn. Reson. Med.*, vol. 75, no. 2, pp. 775–788, 2016.

[17] A. S. Gupta and Z. Liang, "Dynamic imaging by temporal modeling with principal component analysis," in *Proc. 9th Annu. Meeting Int. Soc. Magn. Reson. Med.*2001, pp. 21–21.

[18] A. G. Christodoulou, T. K. Hitchens, Y. L. Wu, C. Ho, and Z.-P. Liang, "Improved subspace estimation for low-rank model-based accelerated cardiac imaging," *IEEE Trans. Biomed. Eng.*, vol. 61, no. 9, pp. 2451–2457, Sep. 2014.

[19] L. Feng, Q. Wen, C. Huang, A. Tong, F. Liu, and H. Chandarana, "Grasp-pro: Improving grasp DCE-MRI through self-calibrating subspace-modeling and contrast phase automation," *Magn. Reson. Med.*, vol. 83, no. 1, pp. 94–108, 2020.

[20] R. Otazo, E. Candes, and D. K. Sodickson, "Low-rank plus sparse matrix decomposition for accelerated dynamic MRI with separation of background and dynamic components," *Magn. Reson. Med.*, vol. 73, no. 3, pp. 1125–1136, 2015.

[21] C. Y. Lin and J. A. Fessler, "Efficient dynamic parallel MRI reconstruction for the low-rank plus sparse model," *IEEE Trans. Comput. Imag.*, vol. 5, no. 1, pp. 17–26, Mar. 2019.

[22] Y. Liu, T. Liu, J. Liu, and C. Zhu, "Smooth robust tensor principal component analysis for compressed sensing of dynamic MRI," *Pattern Recognit.*, vol. 102, 2020, Art. no. 107252.

[23] F. Ong and M. Lustig, "Beyond low rank+ sparse: Multiscale low rank matrix decomposition," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 4, pp. 672–687, Jun. 2016.

[24] S. G. Lingala, E. DiBella, and M. Jacob, "Deformation corrected compressed sensing (DC-CS): A novel framework for accelerated dynamic MRI," *IEEE Trans. Med. Imag.*, vol. 34, no. 1, pp. 72–85, Jan. 2015.

[25] X. Chen, M. Salerno, Y. Yang, and F. H. Epstein, "Motion-compensated compressed sensing for dynamic contrast-enhanced MRI using regional spatiotemporal sparsity and region tracking: Block low-rank sparsity with motion-guidance (BLOSM)," *Magn. Reson. Med.*, vol. 72, no. 4, pp. 1028–1028, 2014.

[26] A. Tolouee, J. Alirezaie, and P. Babyn, "Nonrigid motion compensation in compressed sensing reconstruction of cardiac cine MRI," *Magn. Reson. Imag.*, vol. 46, pp. 114–120, 2018.

[27] Z. Ke et al., "Learned low-rank priors in dynamic MR imaging," *IEEE Trans. Med. Imag.*, vol. 40, no. 12, pp. 3698–3710, Dec. 2021.

[28] A. H. Ahmed, H. Aggarwal, P. Nagpal, and M. Jacob, "Dynamic MRI using deep manifold self-learning," in *Proc. IEEE 17th Int. Symp. Biomed. Imag.*, 2020, pp. 1052–1055.

[29] H. K. Aggarwal, M. P. Mani, and M. Jacob, "MoDL: Model-based deep learning architecture for inverse problems," *IEEE Trans. Med. Imag.*, vol. 38, no. 2, pp. 394–405, Feb. 2019.

[30] M. Arvinte, S. Vishwanath, A. H. Tewfik, and J. I. Tamir, "Deep J-Sense: Accelerated MRI reconstruction via unrolled alternating optimization," in *Proc. Int. Conf. Med. Image Comput. Comput.- Assist. Interv.*, 2021, pp. 350–360.

[31] J. Herrmann et al., "Feasibility and implementation of a deep learning MR reconstruction for TSE sequences in musculoskeletal imaging," *Diagnostics*, vol. 11, no. 8, 2021, Art. no. 1484.

[32] C. M. Sandino, P. Lai, S. S. Vasanawala, and J. Y. Cheng, "Accelerating cardiac cine MRI using a deep learning-based espirit reconstruction," *Magn. Reson. Med.*, vol. 85, no. 1, pp. 152–167, 2021.

[33] E. J. Zucker, C. M. Sandino, A. Kino, P. Lai, and S. S. Vasanawala, "Free-breathing accelerated cardiac MRI using deep learning: Validation in children and young adults," *Radiology*, vol. 300, no. 3, pp. 539–548, 2021.

[34] V. Ghodrati et al., "Temporally aware volumetric generative adversarial network-based mr image reconstruction with simultaneous respiratory motion compensation: Initial feasibility in 3D dynamic cine cardiac MRI," *Magn. Reson. Med.*, vol. 86, no. 5, pp. 2666–2683, 2021.

[35] T. Küstneret al., "Cinenet: Deep learning-based 3D cardiac cine MRI reconstruction with multi-coil complex-valued 4D spatio-temporal convolutions," *Sci. Rep.*, vol. 10, no. 1, pp. 1–13, 2020.

[36] A. H. Ahmed, Q. Zou, P. Nagpal, and M. Jacob, "Dynamic imaging using deep bi-linear unsupervised representation (DEBLUR)," *IEEE Trans. Med. Imag.*, vol. 41, no. 10, pp. 2693–2693, Oct. 2022.

[37] E. K. Cole, F. Ong, S. S. Vasanawala, and J. M. Pauly, "Fast unsupervised MRI reconstruction without fully-sampled ground truth data using generative adversarial networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 3988–3997.
[38] S. Ravishankar, B. E. Moore, R. R. Nadakuditi, and J. A. Fessler, "Low-rank and adaptive sparse signal (LASSI) models for highly accelerated dynamic imaging," *IEEE Trans. Med. Imag.*, vol. 36, no. 5, pp. 1116–1128, May 2017.
[39] B. E. Moore, S. Ravishankar, R. R. Nadakuditi, and J. A. Fessler, "Online adaptive image reconstruction (OnAIR) using dictionary models," *IEEE Trans. Comput. Imag.*, vol. 6, pp. 153–166, 2020.
[40] J. A. Fessler and B. P. Sutton, "Nonuniform fast fourier transforms using min-max interpolation," *IEEE Trans. Signal Process.*, vol. 51, no. 2, pp. 560–574, Feb. 2003.
[41] "Stanford cgls code," [Online]. Available: https://web.stanford.edu/group/SOL/software/cgls/
[42] Y. Cherapanamjeri, K. Gupta, and P. Jain, "Nearly-optimal robust matrix completion," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 797–797.
[43] X. Yi, D. Park, Y. Chen, and C. Caramanis, "Fast algorithms for robust PCA via gradient descent," in *Proc. Neural Inf. Process. Syst.*, 2016, pp. 4152–4160.
[44] Q. Zheng and J. Lafferty, "Convergence analysis for rectangular matrix completion using burer-monteiro factorization and gradient descent," 2016, *arXiv:1605.07051*.
[45] R. Vershynin, *High-Dimensional Probability: An Introduction With Applications in Data Science*, vol. 47. Cambridge, U.K.: Cambridge Univ. Press, 2018.
[46] J.-L. Starck, D. L. Donoho, and E. J. Candès, "Astronomical image representation by the curvelet transform," *Astron. Astrophys.*, vol. 398, no. 2, pp. 785–800, 2003.
[47] S. Winkelmann, T. Schaeffter, T. Koehler, H. Eggers, and O. Doessel, "An optimal radial profile order based on the golden ratio for time-resolved MRI," *IEEE Trans. Med. Imag.*, vol. 26, no. 1, pp. 68–76, Jan. 2007.
[48] Y. Tian et al., "Evaluation of pre-reconstruction interpolation methods for iterative reconstruction of radial k-space data," *Med. Phys.*, vol. 44, no. 8, pp. 4025–4034, 2017.
[49] T. Benkert, Y. Tian, C. Huang, E. V. DiBella, H. Chandarana, and L. Feng, "Optimization and validation of accelerated golden-angle radial sparse MRI reconstruction with self-calibrating grappa operator gridding," *Magn. Reson. Med.*, vol. 80, no. 1, pp. 286–293, 2018.
[50] L. Feng, C. Huang, K. Shanbhogue, D. K. Sodickson, H. Chandarana, and R. Otazo, "Racer-grasp: Respiratory-weighted, aortic contrast enhancement-guided and coil-unstreaking golden-angle radial sparse MRI," *Magn. Reson. Med.*, vol. 80, no. 1, pp. 77–89, 2018.
[51] C. Chen et al., "OCMR (v1. 0)-open-access multi-coil k-space dataset for cardiovascular magnetic resonance imaging,"2020, *arXiv:2008.03410*.
[52] B. Zhao, J. P. Haldar, A. G. Christodoulou, and Z.-P. Liang, "Image reconstruction from highly undersampled (k, t)-space data with joint partial separability and sparsity constraints," *IEEE Trans. Med. Imag.*, vol. 31, no. 9, pp. 1809–1820, Sep. 2012.
[53] D. O. Walsh, A. F. Gmitro, and M. W. Marcellin, "Adaptive reconstruction of phased array MR imagery," *Magn. Reson. Med.*, vol. 43, no. 5, pp. 682–690, 2000.

Silpa Babu received the M.Tech degree in communication engineering from the Vellore Institute of Engineering,Vellore, India, in 2018. She is currently working toward the Ph.D. degree with the Department of Electrical and Computer Engineering, Iowa State University, Ames, IA, USA. Her research interests include machine learning and medical imaging generally. Specific interests include fast and efficient recovery of MRI images from under sampled kspace data.(Email: sbabu@iastate.edu)

Sajan Goud Lingala received the bachelor's degree in biomedical engineering from the Osmania University, Amberpet, India, from 2002 to 2006, the master's degree in biomedical engineering Indian Institute of Technology from 2006 to 2008, and Ph.D. degrees in biomedical engineering University of Iowa, Iowa City, IA, USA, from 2008 to 2014. He is currently an assistant professor of biomedical engineering and radiology with the University of Iowa. Between 2014 and 2017, he completed a Postdoctoral Fellowship with the University of Southern California, Los Angeles, CA, USA. His research interests include rapid MRI sequence design, dynamic MRI, and model-based reconstruction using adaptive priors which include learning based. (Email: sajangoud-lingala@uiowa.edu)

Namrata Vaswani (Fellow, IEEE) received the B.Tech degree electrical engineering from IIT-Delhi, Delhi, India, in 1999 and the Ph.D. degree in electrical engineering from the University of Maryland, College Park, MA, USA, in 2004. Since Fall 2005, she has been with the Iowa State University where she is currently the Anderlik Professor of Electrical and Computer Engineering. Her research interests include data science, with a particular focus on statistical machine learning and signal processing. She has served two terms as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING, as the Lead Guest-Editor of a 2018 Proceedings of the IEEE Special Issue (Rethinking PCA for modern datasets), and as the Area Editor of the IEEE Signal Processing Magazine during 2018–2020. She was the recipient of the Iowa State Early Career Engineering Faculty Research Award in 2014, Iowa State University Mid-Career Achievement in Research Award in 2019 and University of Maryland's ECE Distinguished Alumni Award in 2019. She was also recipient of the 2014 IEEE Signal Processing Society Best Paper Award for her 2010 IEEE Transactions on Signal Processing paper co-authored with her student Wei Lu on Modified-CS: Modifying compressive sensing for problems with partially known support. She is a Fellow of the IEEE Fellow (class of 2019). (Email: namrata@iastate.edu)