

Kernel Regression Imputation in Manifolds Via Bi-Linear Modeling: The Dynamic-MRI Case

Konstantinos Slavakis ¹, Senior Member, IEEE, Gaurav N. Shetty, Loris Cannelli, Member, IEEE, Gesualdo Scutari ², Fellow, IEEE, Ukash Nakarmi, Member, IEEE, and Leslie Ying ³, Senior Member, IEEE

Abstract—This paper introduces a non-parametric approximation framework for imputation-by-regression on data with missing entries. The framework, coined kernel regression imputation in manifolds (KRIM), is built on the hypothesis that features, generated by the measured data, lie close to an unknown-to-the-user smooth manifold. A reproducing kernel Hilbert space (RKHS) forms the feature space where the smooth manifold is embedded in. Aiming at concise representations, KRIM identifies a small number of “landmark points” to define approximating “linear patches” that mimic tangent spaces to smooth manifolds. This geometric information is infused into the design through a novel bi-linear model which can be easily extended to accommodate multi-kernel contributions in the non-parametric approximations. To effect imputation-by-regression, a bi-linear inverse problem is solved by an iterative algorithm with guaranteed convergence to a stationary point of a non-convex loss function. To showcase KRIM’s modularity, the application of KRIM to dynamic magnetic resonance imaging (dMRI) is detailed, where reconstruction of images from severely under-sampled dMRI data is desired. Extensive numerical tests on synthetic and real dMRI data demonstrate the superior performance of KRIM over state-of-the-art approaches under several metrics and with a small computational footprint.

Index Terms—Regression, imputation, kernel, manifold, MRI.

I. INTRODUCTION

A. Motivation and Problem Statement

Dynamic magnetic resonance imaging (dMRI) is a non-invasive MRI modality aiming at observing possibly moving

Manuscript received June 20, 2021; revised September 30, 2021 and December 17, 2021; accepted December 29, 2021. Date of publication February 1, 2022; date of current version February 19, 2022. This work was supported by NSF (CCF/CIF) under Grant 1718796. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Se Young Chun. (Corresponding author: Konstantinos Slavakis.)

Konstantinos Slavakis is with the Department of Information and Communications Engineering, Tokyo Institute of Technology, Tokyo 226-8502, Japan (e-mail: slavakis.k.aa@m.titech.ac.jp).

Gaurav N. Shetty is with Geisinger Health Systems (Steele Institute), Danville, PA 17822 USA (e-mail: gshetty@geisinger.edu).

Loris Cannelli is with the Dalle Molle Institute for Artificial Intelligence, IDSIA/SUPSI, CH-6962 Lugano, Switzerland (e-mail: loris.cannelli@idsia.ch).

Gesualdo Scutari is with the Department of Industrial Engineering, Purdue University, West Lafayette, IN 47907-2023 USA (e-mail: gscutari@purdue.edu).

Ukash Nakarmi is with the Department of Computer Science and Computer Engineering, University of Arkansas, Fayetteville, AR 72701 USA (e-mail: unakarmi@uark.edu).

Leslie Ying is with the Departments of Electrical Engineering and Biomedical Engineering, University at Buffalo, SUNY, Buffalo, NY 14260 USA (e-mail: leiyng@buffalo.edu).

This article has supplementary downloadable material available at <https://doi.org/10.1109/TCI.2022.3148062>, provided by the authors.

Digital Object Identifier 10.1109/TCI.2022.3148062

body organs over time via sophisticated signal acquisition, with rich potential for cardiac, brain and neurological diagnosis [1]. Limitations in MRI scanners often compromise signal sampling at rates lower than the required ones, causing “holes/gaps” or “missing entries” in the collected data, which give rise in turn to aliasing and motion blurring in the observed image series [1]. Such after-effects may hamper clinical diagnosis.

Missing entries occur commonly in a wide range of applications owing to a number of reasons like damaged sensors, constraints on the signal-acquisition mechanism, or data-privacy issues. Missing entries translate to uncertainties that affect negatively the end result: distortion in seismic imaging [2], sub-optimal data modeling in recommender systems [3], as well as large numbers of false positives/negatives in detection methods for single-cell genomics [4].

Several strategies have been proposed to address the missing-entries problem. This study focuses on *imputation-by-regression*, where, typically, i) a regression model that fits the measured values is identified with the regressors being the measured data themselves, and ii) missing entries are imputed by the identified regression model and the measured data [5]. This paper presents a non-parametric approximation framework, coined *kernel regression imputation in manifolds (KRIM)*, to address imputation-by-regression via exploring/exploiting latent geometric information hidden beneath the measured data. Although KRIM can be tailored to fit into any application domain where data/feature approximation is needed, dMRI is used here to demonstrate the way data are collected, clarify concepts, and validate the effectiveness of the approach via extensive numerical tests.

B. Prior Art

This section offers a brief overview of the prior art on imputation-by-regression, especially in the context of dMRI. To keep the discussion short, details on similarities/differences between KRIM and state-of-the-art approaches are deferred to Section V, after KRIM and extensive numerical tests are presented in full detail.

The standard approach to imputation-by-regression is compressed sensing (CS) where signals (image frames or patches) are described by sparse representations of dictionary “atoms” (vectors) that reside in appropriately defined transform domains (feature spaces) [6]–[9]. Low-rank models [10]–[12] and dictionary learning [13], [14] have been also applied to MRI

as data-recovery tools. Unlike the previous methods, where no manifold structures are used, KRIM makes *explicit* use of (non-linear) manifold assumptions and geometry.

Recent efforts on imputation-by-regression revolve around deep-learning (DeepL) approaches [15]–[21] which rely on lengthy processes to learn from training data prior to reconstructing test images. Notwithstanding, concerns were raised in [22] via tests which demonstrated that mainstream DeepL approaches may yield undesired perturbations and structural changes (*e.g.*, small tumors) in reconstructed medical images.

Manifold learning builds on the hypothesis that measured data lie close to a low-dimensional manifold embedded in a high-dimensional space [23]–[29]. Typically, graph Laplacian matrices, which capture relations within data-point clouds, form quadratic regularizers of an inverse problem to recover data; *e.g.*, [27]. Following less conventional paths, [29] relies on affine relations between low-dimensional renditions of the acquired data to identify a temporal basis, and draws arguments from [10] to proceed with the computation also of a spatial basis to describe image frames.

Kernels have been also employed in MRI recovery tasks, *e.g.*, [29]–[32], following their success in the realms of non-parametric approximation and machine learning [33]–[36]. Loosely speaking, kernel methods map non-linearly the observed data from the input space to a high-dimensional feature one where linear operations are employed. A challenging task for kernel methods is to tune the parameters of a kernel function to “optimally” fit the measured data via cross-validation. Another obstacle, especially in regression tasks, is the need for explicit and usually costly pre-imaging steps from the feature space back to the original input one, *e.g.*, [30]. Lastly, kernel-based MRI schemes often leave the complex-valued nature of the (*k*-space) data underused, *e.g.*, [30].

C. Main Contributions and Components of KRIM

KRIM offers non-parametric approximation [35] in imputation-by-regression via the hypothesis that features, generated by the measured data, lie close to an unknown-to-the-user smooth manifold which is embedded in a reproducing kernel Hilbert space (RKHS) [33], [34]; *cf.* Appendix. This hypothesis serves also as a considerable generalization of the assumption which propels learning in RKHSs [36], where features are predominantly assumed to lie close to a linear (affine) subspace, that is, a prototypical smooth manifold.

KRIM needs no training data to operate, unlike mainstream DeepL approaches [15]–[21]. Moreover, KRIM follows less typical paths than methods which use graph Laplacian matrices to unveil data structures, *e.g.*, [27], [31]. Aiming at computationally efficient solutions, KRIM identifies first a potentially small number of vectors, coined “landmark points,” to describe concisely the data-cloud in its original input space (*cf.* Section III-A), and exploits the landmark points to extract geometric information from parsimonious affine combinations (“linear patches”) that mimic tangent spaces [37] to smooth manifolds in the RKHS feature space; see Section III-B.

The learned geometric information is incorporated into the design via a novel bi-linear regression model which allows the data to determine by themselves their most appropriate neighbors in the manifold to perform imputation-by-regression, and renders any pre-imaging steps from the feature space back to the original one unnecessary, unlike [30]. The bi-linear model allows also for multi-kernel contributions in data/feature approximations, surmounting the task of pin-pointing the “optimal” parameters of a single kernel through cross-validation, *e.g.*, [38]; *cf.* Section III-C. Moreover, KRIM is equipped with a dimensionality-reduction module to effect low-rank properties and computationally efficient solutions (*cf.* Section III-D), and is able also to employ complex RKHSs [39], [40]; *cf.* Appendix.

To carry out imputation-by-regression, an inverse problem is formed and solved by an algorithm with guaranteed convergence to a stationary point of a non-convex loss function; *cf.* Section III-E. To showcase KRIM’s modularity and flexibility in fitting into specific application domains, this paper details the application of KRIM to dMRI, where reconstruction of high-resolution images from severely under-sampled dMRI data is desired; *cf.* Section III-F.

This work extends [41] where preliminary results on the single-kernel case (KRIM[S] in the subsequent discussion) were reported. Numerical tests on synthetic and real dMRI data, many more than those in [41] even for the single-kernel KRIM[S] case, demonstrate the superior performance of KRIM over state-of-the-art approaches under several metrics and with a small computational footprint; *cf.* Section IV. Useful introductory information on the mathematical concepts which are used often in this manuscript, such as the proximal mapping and complex RKHSs, can be found in the Appendix. Due to space limitations, material that does not fit in the main document is provided in the supplementary file. To avoid confusion, the tag “Supp-” refers to material of the supplementary file.

II. DATA COLLECTION AND REPRESENTATION

Measured data \mathcal{Y} are considered to be complex valued. In dMRI, \mathcal{Y} takes the form of a tensor defined on the $(N_f \times N_p \times N_{fr})$ -sized “(k,t)-space”; *cf.* Fig. 1(a). Each “slice” \mathcal{Y}_t of tensor \mathcal{Y} , with $t \in \{1, \dots, N_{fr}\}$, corresponds to the $(N_f \times N_p)$ -sized “k-space” measurements collected at the time instance t . Positive integers N_f and N_p represent the numbers of frequency- and phase-encoding lines, respectively [1], while $N_k := N_f N_p$ stands for the number of entries required for a fully-scanned k-space frame. “Full-resolution” MR images are obtained from the “fully sampled” k-space data \mathcal{Y}_t by $\mathcal{X}_t := \mathcal{F}^{-1}(\mathcal{Y}_t)$ (*cf.* Fig. 1(b)), where $\mathcal{F}^{-1}(\cdot)$ is the 2D inverse DFT [1]. In other words, k-space can be viewed as the 2D “frequency domain” of MRI. To facilitate data processing via linear-algebra operations, $\text{vec}(\mathcal{Y}_t)$ stacks the columns of \mathcal{Y}_t one below the other to create a single $N_k \times 1$ vector $\mathbf{y}_t := \text{vec}(\mathcal{Y}_t)$. Define now $\mathbf{Y} := [\mathbf{y}_1, \dots, \mathbf{y}_{N_{fr}}] \in \mathbb{C}^{N_k \times N_{fr}}$.

To cover as many cases of practical interest as possible, it is assumed that data \mathbf{Y} can be represented by \mathbf{X} in a possibly different domain than that of \mathbf{Y} . In dMRI, for example, \mathbf{X} can be defined as follows: if $\mathcal{F}(\cdot)$

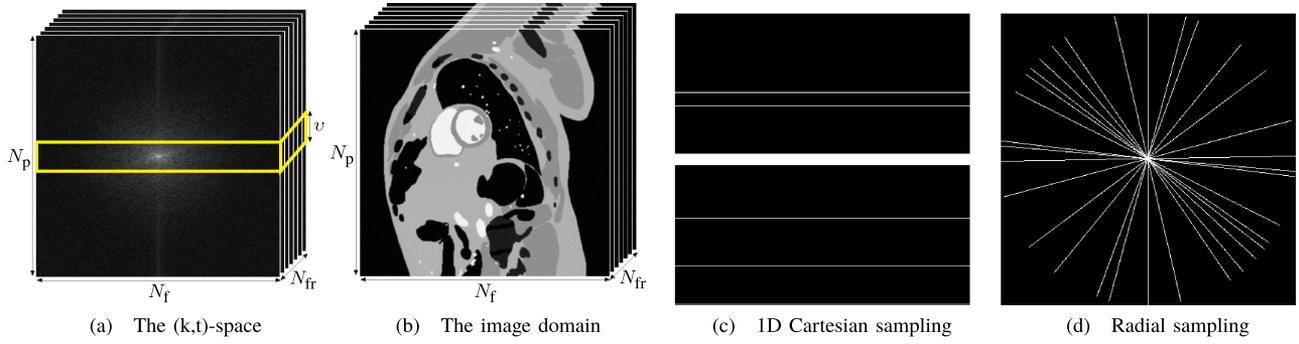


Fig. 1. (a) Complex-valued data \mathcal{Y} are collected over the $(N_f \times N_p \times N_{fr})$ -sized (k, t) -space in dMRI. The marked $N_f \times v \times N_{fr}$ box corresponds to the location of the faithful data, often called “navigator/pilot” data in dMRI. (b) The $N_f \times N_p \times N_{fr}$ image-domain data \mathcal{X} . (c) 1D Cartesian and (d) radial sampling trajectories in k-space.

denotes the 2D DFT even when applied to vectors, *i.e.*, $\mathcal{F}[\text{vec}(\mathcal{X}_t)] := \text{vec}[\mathcal{F}(\mathcal{X}_t)] = \mathbf{y}_t$, then $\mathbf{X} := \mathcal{F}^{-1}(\mathbf{Y}) := [\mathcal{F}^{-1}(\mathbf{y}_1), \mathcal{F}^{-1}(\mathbf{y}_2), \dots, \mathcal{F}^{-1}(\mathbf{y}_{N_{fr}})] \in \mathbb{C}^{N_k \times N_{fr}}$. In other words, \mathbf{X} can be viewed as the image-domain representation of \mathbf{Y} , bearing rich structure: \mathbf{X} comprises usually a moving foreground and a static background, while exhibiting low-rank properties owing to the high correlations among its column vectors.

Motivated by \mathbf{X} , which stands for the representation of \mathbf{Y} in a *transform domain* (*e.g.*, the image domain in dMRI), the following assumption postulates the availability of a linear transform mapping \mathcal{T} which maps \mathbf{X} to \mathbf{Y} .

1) *Modeling Assumption 1*: Available is a linear mapping $\mathcal{T} : \mathbb{C}^{\rho_1 \times \rho_2} \rightarrow \mathbb{C}^{N_k \times N_{fr}}$, for some positive integers ρ_1 and ρ_2 , such that (s.t.) there exists $\mathbf{X} \in \mathbb{C}^{\rho_1 \times \rho_2}$ with $\mathcal{T}(\mathbf{X}) = \mathbf{Y}$. In other words, it is assumed that the pre-image $\mathcal{T}^{-1}(\mathbf{Y}) := \{\mathbf{X} \in \mathbb{C}^{\rho_1 \times \rho_2} \mid \mathcal{T}(\mathbf{X}) = \mathbf{Y}\}$ of \mathbf{Y} under \mathcal{T} is non-empty. To facilitate computations, it is assumed that the adjoint mapping $\mathcal{T}^*(\cdot)$ is also available to the user (*cf.* Appendix).

Loosely speaking, $\mathcal{T}(\cdot)$ serves as a way to connect the domain where data \mathbf{Y} are measured and the transform domain where the measured data manifest rich properties and structure. In dMRI, for example, the 2D DFT $\mathcal{F}(\cdot)$ becomes $\mathcal{T}(\cdot)$. Moreover, for the DFT implementation in Julia [42], $\mathcal{F}^*(\cdot) = N_k \mathcal{F}^{-1}(\cdot)$, where $\mathcal{F}^{-1}(\cdot)$ is the 2D inverse DFT. Modeling Assumption 1 is general enough to accommodate also cases where a domain $\mathbb{C}^{\rho_1 \times \rho_2}$ is unavailable: set $\mathcal{T} := \text{Id} =: \mathcal{T}^*$, where Id stands for the identity mapping. Mapping \mathcal{T} will be used in (11).

Due to physical limitations in MRI scanners, data acquisition is a slow process with scanners being usually unable to keep up with the dynamic nature of the scanning area (patient/organ motion) [1]. Moreover, to reduce as much as possible the discomfort of patients during scanning procedures, it is a common practice to sample partially the $(N_f \times N_p)$ -sized k-space and to collect only a small number of data per frame, often much smaller than N_k which is required for a full scan. To express this partial observation in mathematical terms, define the *sampling mapping* $\mathcal{S}(\cdot) : \mathbb{C}^{N_k \times N_{fr}} \rightarrow \mathbb{C}^{N_k \times N_{fr}} : \mathbf{Y} \mapsto \mathcal{S}(\mathbf{Y})$, which operates entry-wise and nullifies the entry of \mathbf{Y} in the case that entry is missing, while leaves the entry intact in the case

where that entry is measured. Consequently, under severe under-sampling of the k-space, images $\mathcal{F}^{-1}[\mathcal{S}(\mathbf{Y})]$ will suffer from temporal bleeding and motion blurring, adding to the deformations caused by the “thermal” noise which is omnipresent in any data-collection procedure [1]. The missing-entries problem permeates any data-driven application, *e.g.*, unreported observations in recommender systems [3] and dropout values in gene expressions [4].

Under such stringent sampling conditions, several sampling modalities have been proposed to observe k-space partially; *e.g.*, 1D Cartesian (*cf.* Fig. 1(c)) and radial (*cf.* Fig. 1(d)) [1], [12]. A general trend is to heavily acquire data from the central (*i.e.*, “low frequency”) k-space region, which carries contrast information, and acquire data sparsely from the peripheral (*i.e.*, “high frequency”) region which reflects high-resolution image details. Cartesian sampling acquires data along horizontal/vertical lines, concentrating in the low-frequency area of k-space and appearing scarcely in the high-frequency one (*cf.* Fig. 1(c)). Radial sampling acquires data along radial spokes, resulting in dense measurements in low-frequency and sparse measurements in the high-frequency k-space area (*cf.* Fig. 1(d)).

In all imputation frameworks, the designer chooses a *faithful* subset of the measured/sampled data $\mathcal{S}(\mathbf{Y})$ to extract geometric patterns and features. Often, all of $\mathcal{S}(\mathbf{Y})$ are considered to be faithful. In the present context, the heavily sampled central region of k-space constitutes the faithful data, coined “navigator data” [27], [29], [30], [43]. More specifically, in 1D Cartesian sampling, few $v (\ll N_p)$ lines among the measured data $\mathcal{S}(\mathbf{Y})$ which cross the central region of each frame (yellow-colored box in Fig. 1(a)) are vectorized to form the $v \times 1$ vector $\check{\mathbf{y}}_t^f$, where $v := v N_f$, superscript “f” stresses the fact that the vector includes faithful data, and symbol $\check{(\cdot)}$ is used to highlight the fact that the dimension of these vectors is in general smaller than N_k . These vectors are finally stacked into columns of a matrix to form $\check{\mathbf{Y}}_f := [\check{\mathbf{y}}_1^f, \dots, \check{\mathbf{y}}_{N_{fr}}^f] \in \mathbb{C}^{v \times N_{fr}}$. Similarly, in radial sampling, a few v radial spokes among the measured data $\mathcal{S}(\mathbf{Y})$ can be carefully selected to form the faithful data. Faithful data are also met in other applications domains; in single-cell genomics, for example, they may take the form of gene-expression data extracted from the bulk analysis of similar cell types [4].

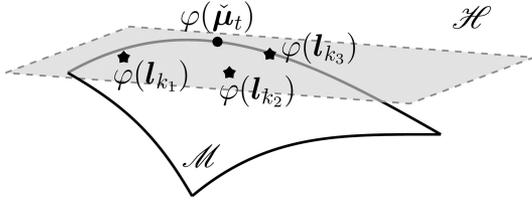


Fig. 2. Points $\{\varphi(\mathbf{l}_{k_j})\}_{j=1}^3$ lie into or close to the unknown-to-the-user manifold $\mathcal{M} \subset \mathcal{H}$ and are affinely combined to approximate $\varphi(\check{\boldsymbol{\mu}}_t)$. All affine combinations of $\{\varphi(\mathbf{l}_{k_j})\}_{j=1}^3$ define the approximating “linear patch” (gray-coloured plane), which mimics a tangent space to \mathcal{M} .

III. THE KRIM FRAMEWORK

A. Selecting Landmark/Representative Points

To promote parsimonious descriptions of the point-cloud $\{\check{\mathbf{y}}_t^f\}_{t=1}^{N_{\text{fr}}}$ as in [43], a subset $\{\mathbf{l}_k\}_{k=1}^{N_l}$, coined *landmark/representative points*, is selected from $\{\check{\mathbf{y}}_t^f\}_{t=1}^{N_{\text{fr}}}$, with N_l being smaller than N_{fr} . Randomly selecting the landmark points from $\{\check{\mathbf{y}}_t^f\}_{t=1}^{N_{\text{fr}}}$ is not the most appropriate strategy to capture the geometry of the faithful data. Although several strategies may be implemented to identify $\{\mathbf{l}_k\}_{k=1}^{N_l}$, the greedy “maxmin” methodology [44] is adopted here. In short, maxmin is a recursive scheme, taking a number of N_l steps to terminate, where at the k th step of the algorithm, a landmark point is selected from $\{\check{\mathbf{y}}_t^f\}_{t=1}^{N_{\text{fr}}}$ that maximizes, over all *un-selected* $\{\check{\mathbf{y}}_t^f\}_{t=1}^{N_{\text{fr}}}$, the minimum (Euclidean) distance to the landmark points which have been already selected up to the $(k-1)$ th step of the algorithm. In general, the larger N_{fr} is, the larger N_l should be to describe the point-cloud $\{\check{\mathbf{y}}_t^f\}_{t=1}^{N_{\text{fr}}}$, justifying thus the non-parametric character of the proposed framework. Maxmin algorithm scores a computational complexity of order $\mathcal{O}(N_l N_{\text{fr}})$, which is naturally heavier than the case of selecting $\{\mathbf{l}_k\}_{k=1}^{N_l}$ randomly from $\{\check{\mathbf{y}}_t^f\}_{t=1}^{N_{\text{fr}}}$. After $\{\mathbf{l}_k\}_{k=1}^{N_l}$ are identified, the $\nu \times N_l$ matrix $\mathbf{L} := [\mathbf{l}_1, \mathbf{l}_2, \dots, \mathbf{l}_{N_l}]$ is defined.

B. Approximating “Linear Patches” in RKHS

Motivated by the concept of tangent spaces to smooth manifolds, this section introduces “linear patches” (see Fig. 2) to approximate the features generated by the data. A user-defined complex RKHS \mathcal{H} , equipped with a reproducing kernel $\kappa(\cdot, \cdot)$ and a feature mapping $\varphi(\cdot)$, plays the role of the feature space in this study; see the Appendix for definitions. According to (18), let also $\Phi(\mathbf{L}) := [\varphi(\mathbf{l}_1), \dots, \varphi(\mathbf{l}_{N_l})]$.

1) *Modeling Assumption 2:* For every $t \in \{1, \dots, N_{\text{fr}}\}$ and every $i \in \{1, \dots, N_k\}$, the (i, t) th entry y_{it} of \mathbf{Y} is approximated as

$$y_{it} \approx f_i(\check{\boldsymbol{\mu}}_t), \quad (1)$$

where $f_i(\cdot) : \mathbb{C}^\nu \rightarrow \mathbb{C}$ is a non-linear function that belongs to the linear span $\text{span}\{\varphi(\mathbf{l}_k)\}_{k=1}^{N_l}$ in the RKHS \mathcal{H} , and $\check{\boldsymbol{\mu}}_t$ is a complex-valued $\nu \times 1$ vector s.t. $\varphi(\check{\boldsymbol{\mu}}_t) \in \text{span}\{\varphi(\mathbf{l}_k)\}_{k=1}^{N_l}$.

Approximation (1) offers smoothing/denoising at entries (i, t) where y_{it} is measured and imputation at entries (i, t) where y_{it} is missing. Hypothesis $\varphi(\check{\boldsymbol{\mu}}_t) \in \text{span}\{\varphi(\mathbf{l}_k)\}_{k=1}^{N_l}$ in \mathcal{H} comes

from the need to approximate y_{it} in (1) by a non-linear function of the measured faithful data. Using the landmark points instead of *all* the faithful data is motivated by the desire to reduce the computational burden of the proposed framework, especially whenever $N_{\text{fr}} \gg N_l$. Furthermore, hypothesis $f_i(\cdot) \in \text{span}\{\varphi(\mathbf{l}_k)\}_{k=1}^{N_l}$ is motivated by the celebrated representer theorem [34] which guarantees that the solution $f_{it}(\cdot)$:

$$f_{it} := \arg \min_{f \in \mathcal{H}} |y_{it} - f(\check{\boldsymbol{\mu}}_t)|^2 + \Omega(\|f\|_{\mathcal{H}}),$$

indeed belongs to $\text{span}\{\varphi(\mathbf{l}_k)\}_{k=1}^{N_l}$, under the assumption that $\varphi(\check{\boldsymbol{\mu}}_t) \in \text{span}\{\varphi(\mathbf{l}_k)\}_{k=1}^{N_l}$ and where $\Omega(\cdot) : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ is a strictly monotonically increasing function.

By Modeling Assumption 2, there exists a vector $\mathbf{a}_i := [a_{i1}, \dots, a_{iN_l}]^\top \in \mathbb{C}^{N_l}$ (\top denotes vector/matrix transposition) s.t. $f_i = \sum_{k=1}^{N_l} a_{ik} \varphi(\mathbf{l}_k) = \Phi(\mathbf{L}) \mathbf{a}_i$. The kernel trick and the Appendix suggest that $y_{it} \approx f_i(\check{\boldsymbol{\mu}}_t) = \langle f_i | \varphi(\check{\boldsymbol{\mu}}_t) \rangle_{\mathcal{H}} = \langle \Phi(\mathbf{L}) \mathbf{a}_i | \varphi(\check{\boldsymbol{\mu}}_t) \rangle_{\mathcal{H}} = \mathbf{a}_i^H \Phi(\mathbf{L})^H \varphi(\check{\boldsymbol{\mu}}_t)$, where \mathbf{H} represents the Hermitian operation onto vectors/matrices. In other words, if $\mathbf{A} := [\mathbf{a}_1, \dots, \mathbf{a}_{N_k}]^H \in \mathbb{C}^{N_k \times N_l}$, then (1) can be recast as: $\forall t$,

$$\mathbf{y}_t \approx \mathbf{A} \Phi(\mathbf{L})^H \varphi(\check{\boldsymbol{\mu}}_t). \quad (2)$$

The following modeling assumption postulates more structure within the data, moving beyond the typical linear-subspace constraints of Modeling Assumption 2.

2) *Modeling Assumption 3:* Points $\{\varphi(\mathbf{l}_k)\}_{k=1}^{N_l}$ and $\{\varphi(\check{\boldsymbol{\mu}}_t)\}_{t=1}^{N_{\text{fr}}}$ lie into or close to a smooth manifold \mathcal{M} embedded in the RKHS \mathcal{H} (cf. Fig. 2).

Since landmark points $\{\mathbf{l}_k\}_{k=1}^{N_l}$ are meant to capture the geometry of the point-cloud $\{\check{\mathbf{y}}_t^f\}_{t=1}^{N_{\text{fr}}}$, it is conceivable that a group of “closely related” points, for example, $\{\varphi(\mathbf{l}_{k_j})\}_{j=1}^3$ in Fig. 2, may collaborate to approximate point $\varphi(\check{\boldsymbol{\mu}}_t)$. Here, “closely related” means proximity within \mathcal{M} , which may capture similarities in the structures and attributes of the data in their original input space, e.g., spatio-temporal relations in the dMRI’s k-space. Putting together Modeling Assumption 3 and the concept of tangent spaces to a smooth manifold [37], $\varphi(\check{\boldsymbol{\mu}}_t)$ is considered to be *affinely* approximated by neighboring landmark points. For example, $\varphi(\check{\boldsymbol{\mu}}_t)$ is approximated in Fig. 2 by a “linear patch” (gray-colored plane) which gathers all affine combinations of $\{\varphi(\mathbf{l}_{k_j})\}_{j=1}^3$. This notion can be formalized by the existence of an $N_l \times 1$ vector \mathbf{b}_t s.t. $\varphi(\check{\boldsymbol{\mu}}_t) \approx \Phi(\mathbf{L}) \mathbf{b}_t$, under the constraint $\mathbf{1}_{N_l}^H \mathbf{b}_t = 1$, which enforces affine combinations, and with $\mathbf{1}_{N_l}$ being the $N_l \times 1$ all-one vector. Moreover, driven by the desire for parsimonious data descriptions—few points $\{\varphi(\mathbf{l}_{k_j})\}_{j=1}^3$ approximate $\varphi(\check{\boldsymbol{\mu}}_t)$ in Fig. 2—and the robustness that such descriptions entail, \mathbf{b}_t is considered to be sparse. Gather $\{\mathbf{b}_t\}_{t=1}^{N_{\text{fr}}}$ in the sparse matrix $\mathbf{B} := [\mathbf{b}_1, \dots, \mathbf{b}_{N_{\text{fr}}}]$ and define the kernel matrix

$$\mathbf{K}_L := \Phi(\mathbf{L})^H \Phi(\mathbf{L}). \quad (3)$$

The previous discussion and (2) yield

$$\mathbf{Y} = \mathbf{A} \mathbf{K}_L \mathbf{B} + \mathbf{E}_1, \quad (4)$$

where the complex-valued matrix \mathbf{E}_1 accumulates all mis-modeling and data-approximation errors.

C. Multiple Kernels

Given the rich variety of reproducing kernels in the literature (see the Appendix and [36]), selecting an appropriate kernel and fine-tuning its parameters through cross-validation to fit the available data are inconvenient tasks. A popular strategy to address those issues is to employ a dictionary of user-defined reproducing kernel functions $\{\kappa_m(\cdot, \cdot)\}_{m=1}^M$ [38], with their associated RKHSs $\{\mathcal{H}_m\}_{m=1}^M$ and feature mappings $\{\varphi_m(\cdot)\}_{m=1}^M$, and form the weighted kernel matrix $\sum_{m=1}^M \omega_m \mathbf{K}_L^{(m)}$, where $\omega_m \in \mathbb{R}_{>0}$ and $\mathbf{K}_L^{(m)}$ is the kernel matrix constructed according to (3) via $\kappa_m(\cdot, \cdot)$. The advantage behind this scheme is that instead of pin-pointing a single kernel and its parameters, a dictionary of kernel matrices $\{\mathbf{K}_L^{(m)}\}_{m=1}^M$ is chosen and $\{\omega_m\}_{m=1}^M$ are fine-tuned by the data themselves through some learning algorithm; *cf.* [38]. Rather than plugging the previous weighted kernel matrix into (4), an approach with more degrees of freedom in the employed parameters is introduced via the following assumption.

1) Modeling Assumption 4: There exist complex-valued $N_k \times N_l$ matrices $\{\mathbf{A}_m\}_{m=1}^M$ and $N_l \times N_{fr}$ sparse matrices $\{\mathbf{B}_m\}_{m=1}^M$, with $\mathbf{1}_{N_l}^H \mathbf{B}_m = \mathbf{1}_{N_{fr}}^H$, $\forall m$, s.t.

$$\mathbf{Y} = \sum_{m=1}^M \mathbf{A}_m \mathbf{K}_L^{(m)} \mathbf{B}_m + \mathbf{E}_2, \quad (5)$$

where the complex-valued matrix \mathbf{E}_2 accumulates all mis-modeling and data-approximation errors.

Matrices $\{\mathbf{A}_m, \mathbf{B}_m\}_{m=1}^M$ need to be identified under some ‘‘optimality’’ criterion to fit the contributions of the multiple kernels to the data, offering more degrees of freedom than typical multi-kernel models where scalars, like the $\{\omega_m\}_{m=1}^M$ of the previous discussion, need to be identified. Unlike [30], (5) employs kernel functions $\{\varphi_m(\cdot)\}_{m=1}^M$ with no need for pre-imaging from the feature spaces $\{\mathcal{H}_m\}_{m=1}^M$ back to \mathbb{C}^ν .

D. Dimensionality Reduction

Owing to the non-parametric character of KRIM, the number N_l of landmark points $\{\mathbf{l}_k\}_{k=1}^{N_l}$ and thus the size $N_l \times N_l$ of the kernel matrix $\mathbf{K}_L^{(m)}$ in (5) can become computationally unbearable if the number N_{fr} of the faithful data, from which $\{\mathbf{l}_k\}_{k=1}^{N_l}$ are extracted, gets large; *cf.* Section III-A. To meet the limitations of computational and time resources, and to impose a low-rank structure on (5), which appears to have a beneficial effect on dMRI [10], [11], a dimensionality-reduction module is introduced here to map each $\mathbf{K}_L^{(m)}$ to a $d \times N_l$ complex-valued matrix $\check{\mathbf{K}}_L^{(m)}$, with $d \ll N_l$. To this end, the two-step dimensionality-reduction scheme of [45], motivated originally by the popular local linear embedding (LLE) [46], is extended into the present RKHS case:

- 1) a matrix \mathbf{W}_m is identified first to gather all the coefficients of the affine dependencies within vectors $\{\varphi_m(\mathbf{l}_k)\}_{k=1}^{N_l}$, with $\varphi_m(\cdot)$ denoting the feature mapping from \mathbb{C}^ν to the user-defined RKHS \mathcal{H}_m ; and
- 2) the low-dimensional $\check{\mathbf{K}}_L^{(m)}$, which preserves the identified affine dependencies \mathbf{W}_m , is computed.

More specifically, along the lines of the discussion that follows Modeling Assumption 3, a sparse matrix $\mathbf{W}_m \in \mathbb{C}^{N_l \times N_l}$ is assumed s.t. $\Phi_m(\mathbf{L}) \approx \Phi_m(\mathbf{L}) \mathbf{W}_m$, under the constraints $\mathbf{1}_{N_l}^H \mathbf{W}_m = \mathbf{1}_{N_l}^H$ and $\text{diag}(\mathbf{W}_m) = \mathbf{0}$, where $\Phi_m(\mathbf{L})$ is defined as in (18) via the feature mapping $\varphi_m(\cdot)$. The constraint $\mathbf{1}_{N_l}^H \mathbf{W}_m = \mathbf{1}_{N_l}^H$ attributes to the desire for affine combinations, while $\text{diag}(\mathbf{W}_m) = \mathbf{0}$ excludes the trivial solution of the identity matrix \mathbf{I}_{N_l} for \mathbf{W}_m . To surmount computations on the possibly infinite dimensional $\Phi_m(\mathbf{L})$, the kernel trick turns out to be handy: multiplying the previous relation from the left by $\Phi_m(\mathbf{L})^H$ yields $\Phi_m(\mathbf{L})^H \Phi_m(\mathbf{L}) \approx \Phi_m(\mathbf{L})^H \Phi_m(\mathbf{L}) \mathbf{W}_m \Leftrightarrow \mathbf{K}_L^{(m)} \approx \mathbf{K}_L^{(m)} \mathbf{W}_m$. More precisely, for a user-defined $\lambda_W \in \mathbb{R}_{>0}$, \mathbf{W}_m is identified by

$$\mathbf{W}_m := \arg \min_{\mathbf{W} \in \mathbb{C}^{N_l \times N_l}} \|\mathbf{K}_L^{(m)} - \mathbf{K}_L^{(m)} \mathbf{W}\|_F^2 + \lambda_W \|\mathbf{W}\|_1$$

s.t. $\mathbf{1}_{N_l}^H \mathbf{W} = \mathbf{1}_{N_l}^H$, $\text{diag}(\mathbf{W}) = \mathbf{0}$, (6)

where $\|\cdot\|_F$ stands for the Frobenius norm of a matrix, while the ℓ_1 -norm of the complex-valued matrix $\mathbf{W} =: [w_{ij}]$ is defined as $\|\mathbf{W}\|_1 := \sum_{ij} [\Re(w_{ij})^2 + \Im(w_{ij})^2]^{1/2}$ [47, Lem. V.1], with $\Re(\cdot)$ and $\Im(\cdot)$ denoting the real and imaginary parts of a complex number. Task (6) is an affinely constrained and composite convex minimization task, and hence, [48] is used to solve it by virtue of the flexibility of [48] in dealing with affine constraints.

After \mathbf{W}_m is identified by (6), $\check{\mathbf{K}}_L^{(m)}$ is computed by

$$\check{\mathbf{K}}_L^{(m)} := \arg \min_{\check{\mathbf{K}} \in \mathbb{C}^{d \times N_l}} \|\check{\mathbf{K}} - \check{\mathbf{K}} \mathbf{W}_m\|_F^2 \quad \text{s.t. } \check{\mathbf{K}} \check{\mathbf{K}}^H = \mathbf{I}_d, \quad (7)$$

where the constraint is introduced to exclude the trivial solution $\mathbf{0}_{d \times N_l}$. The solution to (7) is the orthogonal $d \times N_l$ matrix $[\mathbf{v}_1, \dots, \mathbf{v}_d]^H$, where $\{\mathbf{v}_j\}_{j=1}^d$ are eigen-vectors corresponding to the d smallest eigenvalues of the positive-semidefinite matrix $(\mathbf{I}_{N_l} - \mathbf{W}_m)(\mathbf{I}_{N_l} - \mathbf{W}_m)^H$.

In the context of data-recovery problems, there is a need to unfold the compressed matrix $\check{\mathbf{K}}_L^{(m)}$ back to $\mathbf{K}_L^{(m)}$; *e.g.*, frameworks based on principal component analysis (PCA) [10]. This need invites the following assumption. The subsequent matrix \mathbf{G}_m acts as a ‘‘decompression’’ operator which maps $\check{\mathbf{K}}_L^{(m)} \in \mathbb{R}^{d \times N_l}$ back to $\mathbf{K}_L^{(m)} \in \mathbb{R}^{N_l \times N_l}$.

1) Modeling Assumption 5: There exist $N_l \times d$ complex-valued matrices $\{\mathbf{G}_m\}_{m=1}^M$ s.t. $\forall m \in \{1, \dots, M\}$,

$$\mathbf{K}_L^{(m)} = \mathbf{G}_m \check{\mathbf{K}}_L^{(m)} + \mathbf{E}_3^{(m)},$$

where the $N_l \times N_l$ matrices $\{\mathbf{E}_3^{(m)}\}_{m=1}^M$ gather mis-modeling and approximation errors.

Via Modeling Assumptions 4 and 5, as well as $\mathbf{U}_m := \mathbf{A}_m \mathbf{G}_m$, $\forall m \in \{1, \dots, M\}$, it can be verified that

$$\mathbf{Y} = \sum_{m=1}^M \mathbf{U}_m \check{\mathbf{K}}_L^{(m)} \mathbf{B}_m + \mathbf{E}, \quad (8)$$

where the complex-valued $N_k \times N_{fr}$ matrix \mathbf{E} accumulates all approximation and mis-modeling errors. Model (8) is abbreviated to KRIM[M] whenever $M > 1$ (multi-kernel case), while KRIM[S] is used in the case $M = 1$. KRIM[S], introduced in [41], requires effort via cross validation to fine-tune the parameters of its single kernel function to fit the data.

E. The Inverse Problem

To simplify (8), define $\tilde{\mathbf{U}} := [\mathbf{U}_1, \mathbf{U}_2, \dots, \mathbf{U}_M] \in \mathbb{C}^{N_k \times dM}$, $\tilde{\mathbf{B}} = [\mathbf{B}_1^H, \mathbf{B}_2^H, \dots, \mathbf{B}_M^H]^H \in \mathbb{C}^{MN_i \times N_{fr}}$, and

$$\tilde{\mathbf{K}}_L := \begin{bmatrix} \tilde{\mathbf{K}}_L^{(1)} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{K}}_L^{(2)} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \tilde{\mathbf{K}}_L^{(M)} \end{bmatrix} \in \mathbb{C}^{dM \times MN_i}. \quad (9)$$

Then, (8) takes the bi-linear form

$$\mathbf{Y} = \tilde{\mathbf{U}}\tilde{\mathbf{K}}_L\tilde{\mathbf{B}} + \mathbf{E}, \quad (10)$$

where ‘‘bi-linearity’’ refers to the behavior that if $\tilde{\mathbf{U}}$ (or $\tilde{\mathbf{B}}$) is fixed to a specific value, then (10) expresses \mathbf{Y} as a linear (better affine) model of the variable $\tilde{\mathbf{B}}$ (or $\tilde{\mathbf{U}}$).

A way to fit model (10) to \mathbf{Y} is to incorporate the popular quadratic ‘‘error’’ term $\|\mathbf{Y} - \tilde{\mathbf{U}}\tilde{\mathbf{K}}_L\tilde{\mathbf{B}}\|_F^2$ into the loss function of an inverse problem and to identify the $(\tilde{\mathbf{U}}, \tilde{\mathbf{B}})$ which minimize that loss function. Nevertheless, Modeling Assumption 1 can help recast the previous quadratic error term via \mathbf{Y} 's alternative representation \mathbf{X} . To this end, the following inverse problem is proposed to identify $(\tilde{\mathbf{U}}, \tilde{\mathbf{B}}, \mathbf{X})$ for some user-defined parameters $\lambda_1, C_U \in \mathbb{R}_{>0}$:

$$\min_{(\tilde{\mathbf{U}}, \tilde{\mathbf{B}}, \mathbf{X})} \underbrace{\frac{1}{2} \|\mathcal{F}(\mathbf{X}) - \tilde{\mathbf{U}}\tilde{\mathbf{K}}_L\tilde{\mathbf{B}}\|_F^2}_{\text{T1}} + \underbrace{\lambda_1 \|\tilde{\mathbf{B}}\|_1 + h(\mathbf{X})}_{\text{T2}} \quad (11a)$$

$$\text{s.to } \|\tilde{\mathbf{U}}\mathbf{e}_i\| \leq C_U, \forall i \in \{1, \dots, dM\}, \quad (11b)$$

$$\mathbf{1}_{N_i}^H \mathbf{B}_m = \mathbf{1}_{N_{fr}}^H, \forall m \in \{1, \dots, M\}, \quad (11c)$$

$$\mathcal{S}(\mathbf{Y}) = \mathcal{S}\mathcal{F}(\mathbf{X}),$$

$$\tilde{\mathbf{U}} \in \mathbb{C}^{N_k \times dM}, \tilde{\mathbf{B}} \in \mathbb{C}^{MN_i \times N_{fr}}, \mathbf{X} \in \mathbb{C}^{N_k \times N_{fr}}, \quad (11d)$$

where \mathbf{e}_i denotes the i th column of the identity matrix \mathbf{I}_{dM} . The recovery task comprises the following terms and constraints:

- 1) the data-fit term T1;
- 2) the sparsity-imposing term T2 along the lines of Modeling Assumption 4;
- 3) (11b) to prevent unbounded solutions $\tilde{\mathbf{U}}$ due to the bi-linear structure of the model;
- 4) (11c) to introduce affine constraints according to Modeling Assumption 4;
- 5) (11d) which builds on Modeling Assumption 1 to quantify the desired consistency between the transform-domain representation \mathbf{X} and the observed/sampled entries of \mathbf{Y} ; and
- 6) $h(\cdot)$ is a regularizing function that imposes structure in the variable \mathbf{X} [see the term T3 in (13a)].

The data identify via $(\tilde{\mathbf{U}}, \tilde{\mathbf{B}})$ the most appropriate neighbors in the manifold together with the most suitable collaborating weights to contribute into their own representations. In short, spatio-temporal patterns are exploited by the data themselves to impute their own missing entries.

Any solution $(\hat{\tilde{\mathbf{U}}}, \hat{\tilde{\mathbf{B}}})$ to (11) smooths/denoises the measured values and imputes the missing ones in \mathbf{Y} via the bi-linear model $\mathbf{Y} \approx \hat{\mathbf{Y}} := \hat{\tilde{\mathbf{U}}}\hat{\tilde{\mathbf{K}}}_L\hat{\tilde{\mathbf{B}}}$. The availability of the adjoint mapping \mathcal{F}^* , presumed in Modeling Assumption 1, facilitates computations in algorithms that generate iterates to estimate a solution $\hat{\mathbf{X}}$ to (11). This can be verified by the following discussion where an iterative algorithm that computes a solution to (11) in the context of dMRI is provided.

F. The dMRI Case Study

In the dMRI case, the general inverse problem (11) can be specified as follows. With $\mathcal{F}(\cdot)$ defined in Section II, let the $N_k \times dM$ complex-valued matrix $\tilde{\mathbf{D}} := \mathcal{F}^{-1}(\tilde{\mathbf{U}})$. By setting $\mathcal{F} := \mathcal{F}$ in the Modeling Assumption 1 and by virtue of the properties of \mathcal{F} , it can be verified that the term T1 in (11) becomes (see Sec. Supp-I for a proof)

$$\|\mathcal{F}(\mathbf{X}) - \tilde{\mathbf{U}}\tilde{\mathbf{K}}_L\tilde{\mathbf{B}}\|_F^2 = N_k \|\mathbf{X} - \tilde{\mathbf{D}}\tilde{\mathbf{K}}_L\tilde{\mathbf{B}}\|_F^2. \quad (12)$$

Moreover, scanners in dMRI usually observe regions within \mathbf{X} that demonstrate a periodic movement, *e.g.*, cardiac pumping, while being surrounded by a static background. To exploit this prior knowledge for better data reconstruction, with the $1 \times N_{fr}$ row vector \mathbf{x}_i^T denoting the time profile of the i th pixel ($i \in \{1, \dots, N_k\}$) in $\mathbf{X} =: [\mathbf{x}_1, \dots, \mathbf{x}_{N_k}]^T$, and with $\mathcal{F}_t(\cdot)$ standing for the 1D DFT along the time axis, it is conceivable that $\mathcal{F}_t(\mathbf{x}_i)$ becomes (almost) a sparse vector whenever the time profile \mathbf{x}_i records a periodic movement. This is also the case whenever the time profile of the i th pixel observes also a static background. In other words, $\mathcal{F}_t(\mathbf{X}) := [\mathcal{F}_t(\mathbf{x}_1), \dots, \mathcal{F}_t(\mathbf{x}_{N_k})]^T$ can be approximated by a sparse matrix \mathbf{Z} . By incorporating (12) and the previous arguments into (11) the following dMRI inverse problem is obtained: for the user-defined parameters $\lambda_1, \lambda_2, \lambda_3, C_D \in \mathbb{R}_{>0}$, solve

$$\min_{(\tilde{\mathbf{D}}, \tilde{\mathbf{B}}, \mathbf{X}, \mathbf{Z})} \frac{1}{2} \|\mathbf{X} - \tilde{\mathbf{D}}\tilde{\mathbf{K}}_L\tilde{\mathbf{B}}\|_F^2 + \lambda_1 \|\tilde{\mathbf{B}}\|_1 + \underbrace{\frac{\lambda_2}{2} \|\mathbf{Z} - \mathcal{F}_t(\mathbf{X})\|_F^2 + \lambda_3 \|\mathbf{Z}\|_1}_{\text{T3}} \quad (13a)$$

$$\text{s.to } \|\tilde{\mathbf{D}}\mathbf{e}_i\| \leq C_D, \forall i \in \{1, \dots, dM\}$$

$$\mathbf{1}_{N_i}^H \mathbf{B}_m = \mathbf{1}_{N_{fr}}^H, \forall m \in \{1, \dots, M\}$$

$$\mathcal{S}(\mathbf{Y}) = \mathcal{S}\mathcal{F}(\mathbf{X})$$

$$\tilde{\mathbf{D}} \in \mathbb{C}^{N_k \times dM}, \tilde{\mathbf{B}} \in \mathbb{C}^{MN_i \times N_{fr}}, \mathbf{X}, \mathbf{Z} \in \mathbb{C}^{N_k \times N_{fr}}, \quad (13b)$$

where term T3 in (13a) specifies the regularizing loss $h(\cdot)$ in (11a). Since the goal of T3 is to promote the sparsity of $\mathcal{F}_t(\mathbf{X})$, an ℓ_1 -norm loss of $\mathcal{F}_t(\mathbf{X})$ could have been used in (13a) instead of T3. However, the auxiliary variable \mathbf{Z} is introduced in (13a) to facilitate processing by decoupling \mathcal{F}_t and the ℓ_1 -norm loss.

Other than the introduction of kernel functions, (13) is also structurally different than the inverse problem in the predecessor [43], since the computationally intensive 2D DFT $\mathcal{F}(\cdot)$ is not employed at all at the loss function (13a), unlike [43], but it is only applied to the constraint (13b). Such a strategy yields

largely reduced computational times with regards to [43] as the extensive numerical tests demonstrate in Section IV.

G. Solving the dMRI Inverse Problem

This section discusses the details of Algorithm 1 which solves (13). Even though Algorithm 1 addresses the specific dMRI recovery task, its backbone iterations can be applied, with appropriate tweaks, also to other application domains.

The successive-convex-approximation framework of [49] is applied to solve (13) via steps 4–10 in Algorithm 1. Step 7 in Algorithm 1 focuses on solving convex minimization sub-tasks to update estimates $\hat{\mathbf{D}}_n, \hat{\mathbf{B}}_n, \hat{\mathbf{X}}_n, \hat{\mathbf{Z}}_n$. With user-defined parameters $\tau_D, \tau_B \in \mathbb{R}_{>0}$,

$$\hat{\mathbf{D}}_{n+1/2} \in \arg \min_{\mathbf{D}} \frac{1}{2} \|\hat{\mathbf{X}}_n - \tilde{\mathbf{D}} \tilde{\mathbf{K}}_L \hat{\mathbf{B}}_n\|_F^2 + \frac{\tau_D}{2} \|\tilde{\mathbf{D}} - \hat{\mathbf{D}}_n\|_F^2$$

s.to $\|\tilde{\mathbf{D}} \mathbf{e}_i\| \leq C_D, \forall i \in \{1, \dots, dM\}$, (14a)

$$\hat{\mathbf{B}}_{n+1/2} \in \arg \min_{\mathbf{B}} \frac{1}{2} \|\hat{\mathbf{X}}_n - \hat{\mathbf{D}}_n \tilde{\mathbf{K}}_L \hat{\mathbf{B}}\|_F^2 + \lambda_1 \|\hat{\mathbf{B}}\|_1$$

$$+ \frac{\tau_B}{2} \|\hat{\mathbf{B}} - \hat{\mathbf{B}}_n\|_F^2$$

s.to $\mathbf{1}_{N_i}^H \mathbf{B}_m = \mathbf{1}_{N_{fr}}^H, \forall m \in \{1, \dots, M\}$, (14b)

$$\hat{\mathbf{X}}_{n+1/2} \in \arg \min_{\mathbf{X}} \frac{1}{2} \|\mathbf{X} - \hat{\mathbf{D}}_n \tilde{\mathbf{K}}_L \hat{\mathbf{B}}_n\|_F^2 + \frac{\lambda_2}{2} \|\hat{\mathbf{Z}}_n - \mathcal{F}_t(\mathbf{X})\|_F^2$$

s.to $\mathcal{S}(\mathbf{Y}) = \mathcal{S}\mathcal{F}(\mathbf{X})$, (14c)

$$\hat{\mathbf{Z}}_{n+1/2} \in \arg \min_{\mathbf{Z}} \frac{1}{2} \|\mathbf{Z} - \mathcal{F}_t(\hat{\mathbf{X}}_n)\|_F^2 + \frac{\lambda_3}{\lambda_2} \|\mathbf{Z}\|_1$$

$$= \text{Soft}[\mathcal{F}_t(\hat{\mathbf{X}}_n), \lambda_3/\lambda_2],$$
 (14d)

where the extra regularizing quadratic terms in (14a) and (14b) are introduced to render the loss functions strongly convex and still allow Algorithm 1 to converge to a stationary point of the loss in (13a); see [49]. The unique solution $\hat{\mathbf{X}}_{n+1/2}$ to (14c) is

$$\hat{\mathbf{X}}_{n+1/4} := \frac{1}{1+\lambda_2 N_{fr}} \left(\hat{\mathbf{D}}_n \tilde{\mathbf{K}}_L \hat{\mathbf{B}}_n + \lambda_2 N_{fr} \mathcal{F}^{-1}(\hat{\mathbf{Z}}_n) \right), \quad (15a)$$

$$\hat{\mathbf{X}}_{n+1/2} := \mathcal{F}^{-1} \mathcal{S}(\mathbf{Y}) + \mathcal{F}^{-1} \mathcal{S}^c \mathcal{F}(\hat{\mathbf{X}}_{n+1/4}), \quad (15b)$$

where $\mathcal{S}^c(\cdot)$ is the complementary operator of $\mathcal{S}(\cdot)$, *i.e.*, $\mathcal{S}^c(\cdot)$ nullifies the entries of the matrix that are preserved by $\mathcal{S}(\cdot)$, leaving the remaining entries intact. A detailed derivation of (15) is provided in the supplementary file.

Updating $(\hat{\mathbf{D}}_n, \hat{\mathbf{B}}_n, \hat{\mathbf{X}}_n, \hat{\mathbf{Z}}_n)$ can be carried out in parallel, promoting thus efficient implementations. Since (14d) and (15) offer closed-form solutions, the bulk of computations per iteration n of Algorithm 1 comes from the solution of the convex sub-tasks (14a) and (14b); see Alg. 1. In practice, (14a) and (14b) are solved inexactly by employing only a finite number of steps of Alg. Supp-1. Convergence of Algorithm 1 to a stationary point of the non-convex loss in (13), which satisfies at the same time the constraints in (13), is guaranteed even if (14a) and (14b) are solved inexactly by Alg. Supp-1 via points in an ϵ_n -ball neighborhood of their exact solutions with diminishing radius ϵ_n as n increases [49].

Algorithm 1: Solving KRIM's Inverse Problem for dMRI.

Input: Available are data $\mathcal{S}(\mathbf{Y})$ which include the faithful $\check{\mathbf{Y}}_f$ ones. Choose parameters $\lambda_1, \lambda_2, \lambda_3, C_D, \tau_D, \tau_B \in \mathbb{R}_{>0}$, as well as $\zeta \in (0, 1)$ and $\gamma_0 \in (0, 1]$. Choose N_l landmark points, the dimensionality parameter d , as well as M kernels and their parameters to form the kernel matrices.

Output: Extract the limit points $\hat{\mathbf{D}}_*$ and $\hat{\mathbf{B}}_*$ of sequences $(\hat{\mathbf{D}}_n)_n$ and $(\hat{\mathbf{B}}_n)_n$, respectively, and recover the dMRI images/frames by $\hat{\mathbf{D}}_* \tilde{\mathbf{K}}_L \hat{\mathbf{B}}_*$.

- 1: Identify landmark points \mathbf{L} from $\check{\mathbf{Y}}_f$ (*cf.* Section III-A).
 - 2: Compute $\tilde{\mathbf{K}}_L$ via (9).
 - 3: Set the iteration index $n = 0$, and arbitrarily fix the starting points $(\hat{\mathbf{D}}_0, \hat{\mathbf{B}}_0, \hat{\mathbf{X}}_0, \hat{\mathbf{Z}}_0)$.
 - 4: **while** $n \geq 0$ **do**
 - 5: Available are $(\hat{\mathbf{D}}_n, \hat{\mathbf{B}}_n, \hat{\mathbf{X}}_n, \hat{\mathbf{Z}}_n)$ and γ_n .
 - 6: Set $\gamma_{n+1} := \gamma_n(1 - \zeta\gamma_n)$.
 - 7: Obtain $\hat{\mathbf{D}}_{n+1/2}$ and $\hat{\mathbf{B}}_{n+1/2}$ by (14a) and (14b), respectively (*cf.* Alg. Supp-1), $\hat{\mathbf{X}}_{n+1/2}$ by (14c) and (15), and $\hat{\mathbf{Z}}_{n+1/2}$ by (14d).
 - 8: Update $(\hat{\mathbf{D}}_{n+1}, \hat{\mathbf{B}}_{n+1}, \hat{\mathbf{X}}_{n+1}, \hat{\mathbf{Z}}_{n+1})$
 $:= \gamma_{n+1}(\hat{\mathbf{D}}_{n+1/2}, \hat{\mathbf{B}}_{n+1/2}, \hat{\mathbf{X}}_{n+1/2}, \hat{\mathbf{Z}}_{n+1/2})$
 $+ (1 - \gamma_{n+1})(\hat{\mathbf{D}}_n, \hat{\mathbf{B}}_n, \hat{\mathbf{X}}_n, \hat{\mathbf{Z}}_n)$.
 - 9: Set $n \leftarrow n + 1$ and go to step 4.
 - 10: **end while**
-

Alg. Supp-1 operates under a constant step size and needs only the computations of first-order information (gradients) of the loss function and proximal mappings, like soft-thresholding and projections; see the supplementary file for details. Although a thorough convergence-rate analysis of [48] is still pending, Alg. Supp-1 is expected to score in general a (sub-)linear rate of convergence to an ϵ -ball neighborhood of an exact solution, similarly to other iterative methods which are based on first-order information of a convex loss function. In the context of Algorithm 1, the sub-routine Alg. Supp-1 requires the computation of the Lipschitz coefficients (Supp-6) and (Supp-13) per iteration index n by singular-value-decomposition operations. Ways to simplify those computations are currently under investigation.

IV. NUMERICAL TESTS

A. Datasets and Competing Algorithms

The proposed framework (KRIM[M] and KRIM[S]) is validated on three datasets, used also in [10], [28]–[30], [43]: i) Magnetic resonance extended cardiac-torso (MRXCAT) cine phantom (Section IV-E) [50]; ii) in-vivo cardiac phantom generated from a real MR-scanned image (Section IV-F) [10]; and iii) free-breathing and real-time measurements of an MR-scanned cardiac region (Section IV-G) [51]. KRIM[M] and

KRIM[S] are compared against the popular FRIST [9], PS-Sparse [10], XD-GRASP [12], STORM [27], as well as the kernel-based KLR [30] and the predecessor of this work BiLMDM [43]. Detailed numerical tests and comparisons against the popular dictionary-learning method [14] can be found in [43], where BiLMDM is shown to outperform the algorithmic scheme of [14] in several scenarios and on datasets which are identical with the ones in this study. Representative videos of the reconstructed image time series are also provided as supplemental material to the manuscript [52].

B. Performance Metrics

The quality of reconstructions is quantified via the following performance metrics.

- i) The normalized root mean square error (NRMSE):

$$\text{NRMSE} := \frac{\|\mathbf{X} - \hat{\mathbf{X}}\|_F}{\|\mathbf{X}\|_F}, \quad (16)$$

where \mathbf{X} denotes the high-fidelity original image data, generated from a fully measured (k,t)-space (cf. Section II), and $\hat{\mathbf{X}}$ represents the estimate of \mathbf{X} . Low NRMSE values are desired.

- ii) The high frequency error norm (HFEN):

$$\text{HFEN} := \frac{\|\text{LoG}(\mathbf{X}) - \text{LoG}(\hat{\mathbf{X}})\|_2}{\|\text{LoG}(\mathbf{X})\|_2}, \quad (17)$$

where LoG is a rotationally symmetric Laplacian-of-Gaussian filter. HFEN assesses the effectiveness of an algorithm along the edges of an image in terms of sharpness. As in [13], a kernel filter of size 15×15 with standard deviation of 1.5 pixels is employed. Low HFEN scores are desired.

- iii) The structural similarity index measure (SSIM) [53] investigates local patterns in pixel intensities after normalizing for luminance and contrast, with values within $[0, 1]$. SSIM values close to 1 are desired.
- iv) The sharpness measures M1 (intensity-variance based) and M2 (energy of the image gradient) [54, (43) and (46)]. These gradient-based measures focus on the high-frequency (edge characteristics) variations in the images and quantify peaks. Large M1 and M2 values are desired.

Among the previous metrics, NRMSE, HFEN, and SSIM require the knowledge of the ground-truth image, unlike M1 and M2. All metric values which are reported in this manuscript are the sample means of the values obtained from 25 independent trials/experiments.

C. Experimental Setup

KRIM (Algorithm 1) was implemented in Julia (ver. 1.0.8) [42], while BiLMDM [41], [43] was implemented in MATLAB. The Julia [42] code of KRIM is available online [55]. MATLAB implementations for STORM [56], XD-GRASP [57], and FRIST [58] are publicly available, whereas implementations of PS-Sparse and KLR were made available by the Authors of [10] and [30], respectively. Experiments were carried out on a 12-core Intel(R) 2.40 GHz Linux based-system with 48 GB

RAM. All hyperparameters of all frameworks were fine-tuned to produce reconstructions with the lowest NRMSE score.

Among the three datasets of Section IV-A, ground-truth data are available only for the first two datasets. These datasets are retrospectively under-sampled in the (k,t) domain under different acceleration/sampling rates, defined as $N_k N_{fr} / (\# \text{ of acquired samples})$. Both the 1D Cartesian (Fig. 1(c)) and the radial (Fig. 1(d)) under-sampling schemes were employed. Apart from the navigator data, which follow a specific structure according to Section II, the rest of the measured data are sampled randomly and independently among frames provided they obey the adopted sampling modality, *i.e.*, 1D Cartesian or radial. Given the stringent space limitations of this journal, only results on the 1D Cartesian scheme are demonstrated for the first dataset, while only results on radial under-sampling for the second one. The third dataset of Section IV-A is prospectively under-sampled by default, since it comprises readings from an MR scanner.

D. Parameter Tuning

Tasks (13) and (14) contain parameters which relate to different aspects of the implementation; for example, data-modeling parameters ($\lambda_1, \lambda_2, \lambda_3, C_D, N_l, d$) are tuned for an optimal data fit, while optimization parameters ($\gamma_0, \zeta, \alpha, \tau_D, \tau_B$) govern the convergence of the sequence of estimates. Those parameters need tuning for Algs. 1 and Supp-1 to reach “desired performance.” To offer a concrete discussion about “desired performance,” parameters are carefully tuned for the first two datasets of Section IV-A where ground-truth dMRI data are available. “Optimal” parameter values are defined as those values that yield low NRMSE while, at the same time, the achieved low NRMSE value shows small fluctuations as parameters change.

Trial and error revealed the following: parameter λ_1 penalizes the sparsity of $\tilde{\mathbf{D}}$ to control the size of the neighborhood that participates in the approximation of a dMRI frame. Higher the value λ_1 is, lower the size of the neighborhood becomes, allowing only frames with very similar characteristics and structure to participate in approximations. It is worth noting that this neighborhood is made of landmark points and, hence, parameter N_l (total number of landmark points) plays an important role in defining the size of neighborhoods. Parameter N_l has to be made large enough so that there is enough representation from a group of frames with similar phase and characteristics, but not excessively large to keep computation times manageable. For the datasets used, $N_l \in N_{fr} \cdot [20\%, 35\%]$ turned out to be adequate. An increase of N_l beyond those values didn’t show any significant improvement in performance.

With regards to $\tilde{\mathbf{D}}$, the bounding constraint $C_D = 1$ usually leads to “optimal” estimates for $\tilde{\mathbf{D}}$ in the case where k-space is normalized in a way that pixel intensity values lie within $[0, 1]$. The d value imposes a low-rank structure to the model, and it was observed to be dependent on the dynamic nature of the image series. For the parameters (λ_2, λ_3), which exploit periodicity along the time series, higher the ratio λ_3/λ_2 is, the more temporal bleeding is observed. Hence, a value low enough to avoid temporal bleeding but high enough to exploit

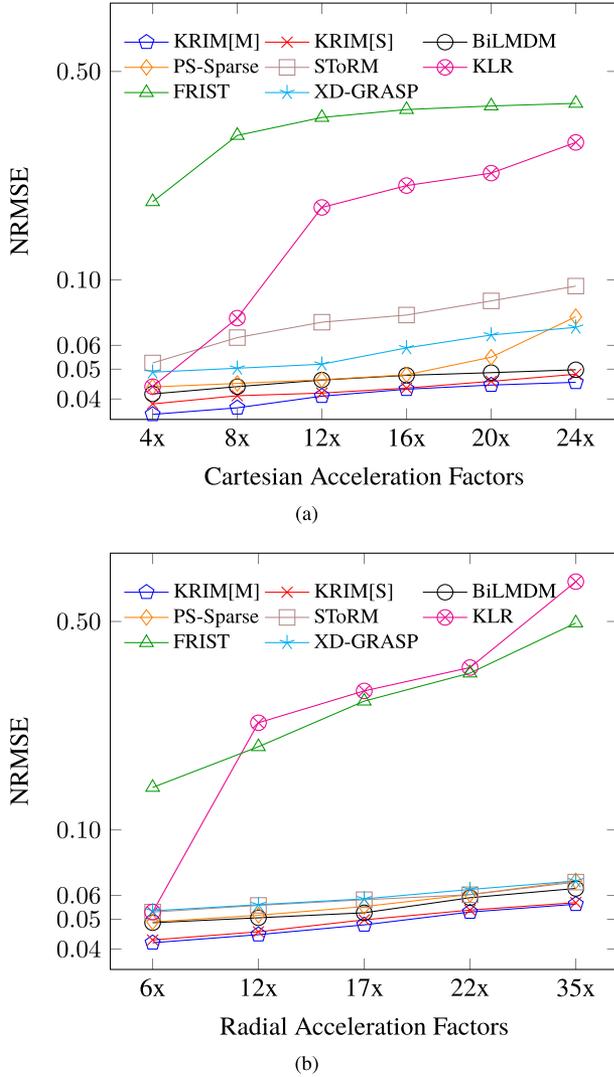


Fig. 3. NRMSE values at different acceleration rates for the reconstructions of the synthetically generated data (a) MRXCAT phantom and (b) cardiac cine. The curves for BiLMDM, KRIM[S] and KRIM[M] are generated using the mean over 25 independent trials to investigate the dependence of the solution on the initialization point of the non-convex inverse problem. The NRMSE values are plotted on a log-axis to provide enhanced visibility of the cluster of curves within the $[0.035, 0.065]$ NRMSE range.

the periodicity along the temporal axis is desired. Focusing on the parameters which affect the optimization procedures, the step sizes $\gamma_0 = 0.9$ and $\zeta = 0.001$ for Algorithm 1 and $\alpha = 0.5$ for Alg. Supp-1 ensure desirable results. Similarly, τ_D, τ_B values within $[10^{-7}, 10^{-5}]$ showed desirable results across different datasets.

Tests on the sensitivity of KRIM[M]’s NRMSE values vs. perturbations of the values of parameters λ_1 and d around their finely tuned $\hat{\lambda}_{1,*}$ and \hat{d}_* can be found in Fig. 5 in the context of Section IV-F. It is worth noticing that values of d larger than \hat{d}_* do not inflict any severe degradation on NRMSE, as opposed to scenarios where $d < \hat{d}_*$. Moreover, the larger d becomes, the smaller the deteriorating effect of the perturbations of λ_1 around $\hat{\lambda}_{1,*}$ on NRMSE. However, excessively large values of d should

TABLE I
QUANTITATIVE PERFORMANCE FOR THE MRXCAT PHANTOM
(ACCELERATION RATE: 8X)

	NRMSE	SSIM	HFEN	M1	M2	Time
KRIM[M]	0.0372	0.9052	0.1058	37.95	1.5×10^6	23min
KRIM[S]	0.0407	0.8869	0.1198	37.65	1.4×10^6	14min
KRIM[M] $\setminus_{(c_*, r_*)}$	0.0378	0.9052	0.1062	38.54	1.5×10^6	23min
BiLMDM	0.0438	0.8697	0.1314	28.69	1.1×10^6	5.5hrs
KLR	0.0744	0.6704	0.1580	34.98	1.4×10^6	6.5hrs
FRIST	0.3055	0.4572	0.7006	13.50	7.7×10^5	3.5hrs
SToRM	0.0644	0.7743	0.2478	31.06	1.5×10^6	43min
PS-Sparse	0.0449	0.8864	0.1338	30.66	1.2×10^6	10min
XD-GRASP	0.0505	0.7959	0.1659	35.43	1.4×10^5	1.5 hrs

TABLE II
QUANTITATIVE PERFORMANCE ANALYSIS FOR IN-VIVO CARDIAC
PHANTOM (ACCELERATION RATE: 12X)

	NRMSE	SSIM	HFEN	M1	M2	Time
KRIM[M]	0.0445	0.9586	0.0537	15.15	4.5×10^5	13 min
KRIM[S]	0.0455	0.9569	0.0633	14.77	4.4×10^5	7 min
KRIM[M] $\setminus_{(c_*, r_*)}$	0.0439	0.961	0.0543	15.04	4.5×10^5	13 min
BiLMDM	0.0507	0.9494	0.0903	14.87	4.4×10^5	4 hrs
KLR	0.2296	0.7059	0.3364	4.94	2.7×10^5	5 hrs
FRIST	0.1903	0.5048	0.5045	9.59	2.8×10^5	2 hrs
SToRM	0.0559	0.9394	0.1105	14.46	4.5×10^5	32 min
PS-Sparse	0.0517	0.9487	0.1061	15.13	4.5×10^5	4 min
XD-GRASP	0.0561	0.9288	0.1595	14.30	4.5×10^5	1 hr

be avoided to keep the computational footprint of the proposed scheme small.

A polynomial kernel (*cf.* Appendix), with $c_* := 3$ and $r_* := 0.5 \times$ (mean of all entries in \mathbf{L}), offered the “best” NRMSE results for KRIM[S] after careful but time consuming cross validation. For KRIM[M], a dictionary of $M = 7$ kernels is created which contains a group of Gaussian and polynomial kernels with varying σ and (c, r) , respectively. More specifically, four polynomial kernels with $c \in \{1, 2, c_*, 4\}$ and $r = r_*$, and three Gaussian ones with $\sigma \in \{0.2, 0.4, 0.8\}$.

Notice that KRIM[S]’s “optimal” polynomial kernel with (c_*, r_*) is included in KRIM[M]’s kernel dictionary. The reason for adopting this strategy is to build on the effort spent on KRIM[S] to identify the kernel parameters (c_*, r_*) , and to push for better performance than the case of the hand-tuned kernel in KRIM[S] by allowing the data themselves to fine-tune kernel contributions in the data-approximation formula of (8). Indeed, such a performance boost was observed throughout the numerical tests. Nevertheless, to provide a complete set of tests, KRIM[M] was also employed with a kernel dictionary that does not contain KRIM[S]’s “optimal” polynomial kernel with parameters (c_*, r_*) . In other words, KRIM[M] $\setminus_{(c_*, r_*)}$ will serve henceforth as the tag for KRIM[M], employed with a kernel dictionary of six kernels: three polynomial ones with $c \in \{1, 2, 4\}$ and $r = r_*$, and three Gaussian ones with $\sigma \in \{0.2, 0.4, 0.8\}$. The results on KRIM[M] $\setminus_{(c_*, r_*)}$ in Tables I and II, especially with regards to KRIM[M], attest to the robustness of KRIM and its ability to offer state-of-the-art performance without consuming time on pin-pointing kernel-function parameters via cross validation. To save space, average computation times and image reconstructions are not demonstrated for KRIM[M] $\setminus_{(c_*, r_*)}$ since they are almost identical to those of KRIM[M].

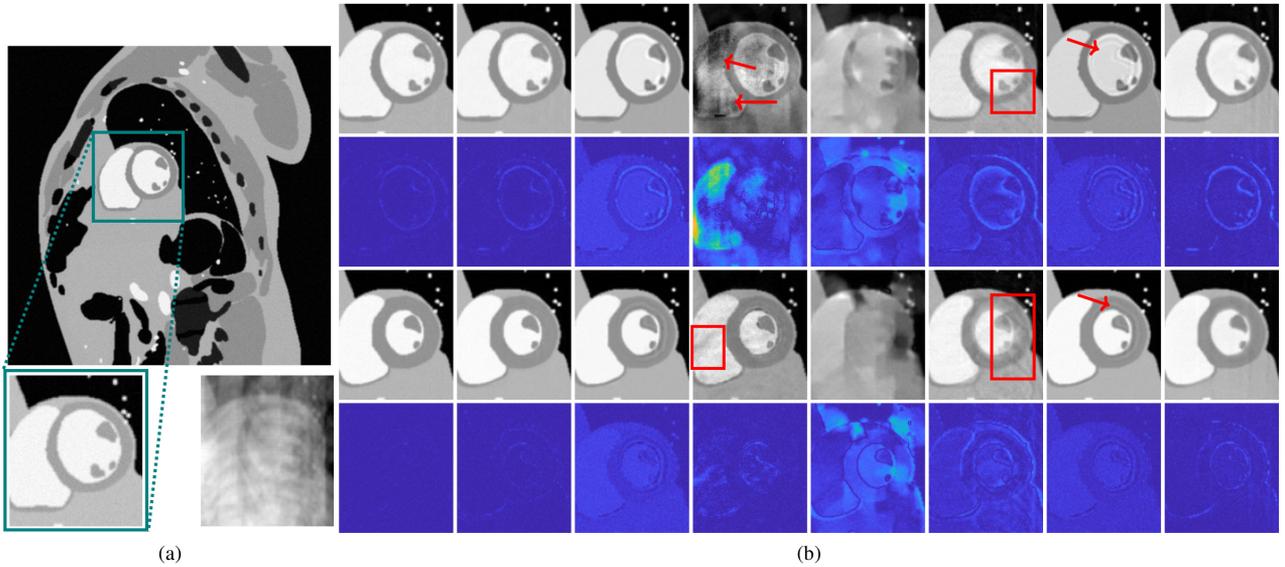


Fig. 4. (a) Ground-truth MRXCAT phantom. The ROI is extracted via the zero-filled DFT reconstruction from scanner data emulating an acceleration rate of $8\times$. Comparative results between the proposed KRIM[M], KRIM[S], BiLMDM, KLR, FRIST, STORM, PS-Sparse, XD-GRASP (left to right) are shown in (b). 1^{st} row: Diastole phase reconstructions (frame 1 of the image series); 2^{nd} row: Error maps for the diastole phase; 3^{rd} row: Systole phase reconstructions (frame 12 of the image series); 4^{th} row: Error maps for systole phase. Red markings point out distortions, aliasing, or undesired artifacts in the reconstructions. Reconstruction results along the time axis are also provided in Fig. 6.

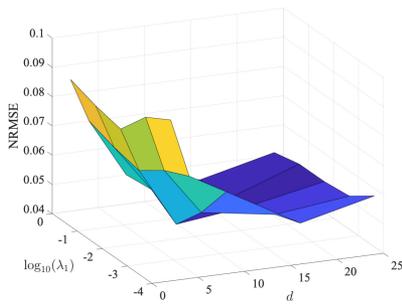


Fig. 5. Test results on the sensitivity of KRIM[M]'s NRMSE values vs. perturbations of the values of parameters λ_1 and d around their finely tuned ones in the context of Section IV-F for the acceleration rate of $12\times$.

KRIM with a set of parameters which are tuned on some training data is expected to behave similarly on data sets which “look alike,” or, are generated by similar probability distributions with the training data. Hence, to foresee the performance of KRIM on test data where ground-truth data are unavailable, it is necessary to tune (“train”) KRIM’s parameters on data that are similar to the test ones.

E. MRXCAT Cardiac Phantom

The MRXCAT phantom is a synthetically generated cardiac cine dMRI dataset using the extended cardiac torso (XCAT) framework [50]. The MRXCAT phantom emulates a breath-hold cardiac cine data of spatial size $(N_f, N_p) = (408, 408)$, with $N_{fr} = 360$ frames, spatial resolution $= 1.56 \times 1.56 \text{ mm}^2$, and field-of-vision (FOV) $= 400 \times 400 \text{ mm}^2$. The generated phantom data contains 15 cardiac cycles and 24 cardiac phases. It is characterized by its limited inter-frame variations and periodic

nature along the temporal direction. The KRIM parameters $d := 4$ and $N_l := 100$ are considered.

It can be seen from Fig. 3(a) that KRIM outperforms consistently the competing state-of-the-art schemes across different acceleration rates. Table I details the averages (over 25 runs) of all performance metrics for the acceleration rate of $8\times$. Unlike KLR, STORM and FRIST, KRIM reconstructions do not deteriorate significantly in NRMSE values as acceleration rates become larger. Notice also the large savings in computation time achieved by KRIM with regards to its predecessor BiLMDM [43].

To reinforce the results of Fig. 3(a) and Table I, 25 independent experiments were run and statistical-significance tests were carried out on the NRMSE values of KRIM[M] and KRIM[S] with respect to the rest of the methods. An one-sided Welch’s t-test was performed under the hypothesis $\text{NRMSE}(\text{Method}) \leq \text{NRMSE}(\text{KRIM[M]})$, where “Method” can be any of BiLMDM, KLR, FRIST, STORM, PS-Sparse, and XD-GRASP, including even KRIM[S]. The test produced p-values ≈ 0 (order of machine precision), establishing thus very strong evidence to reject the previous hypothesis [59, Chap. 10]. Similarly, the same statistical-significance test yielded p-values ≈ 0 under the hypothesis $\text{NRMSE}(\text{Method}) \leq \text{NRMSE}(\text{KRIM[S]})$, where $\text{Method} \in \{\text{BiLMDM, KLR, FRIST, STORM, PS-Sparse, XD-GRASP}\}$.

Spatial and temporal reconstructions, generated from (k,t) -space measurements available at the acceleration rate of $8\times$, are exhibited for comparison in Figs. 4 and 6, respectively, alongside with the ground-truth dMRI data for reference. Examination of these images sheds light onto the fact that KRIM produces images which are almost distortion free (unlike FRIST), artifact free (unlike PS-Sparse), alias free (unlike KLR and STORM) and sharper (compared to XD-GRASP). These deformities are

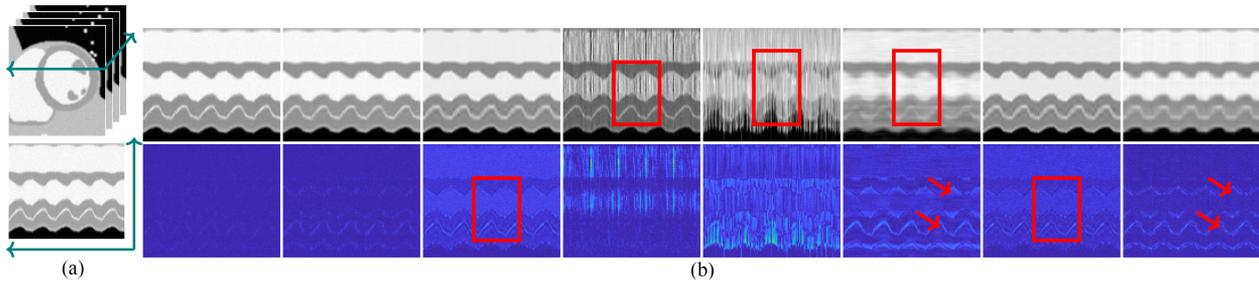


Fig. 6. Reconstruction results along the time axis for the MRXCAT cardiac phantom measured at an acceleration rate of 8x; *cf.* Section IV-E. (a) Ground-truth image stack and temporal cross-section. The location of the cross-section is marked on the image with a red axis. (b) Comparative results of the temporal cross-section via methods KRIM[M], KRIM[S], BiLMDM, KLR, FRIST, SToRM, PS-Sparse, XD-GRASP (left to right) are exhibited in the first row, followed by the error maps in the second row. Red markings point out the distortions, temporal bleeding, or artifacts in the reconstructions.

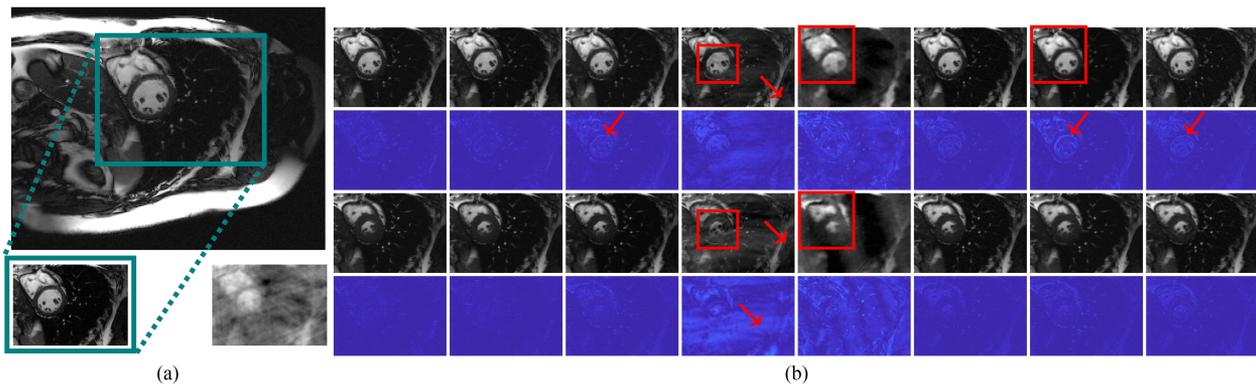


Fig. 7. Reconstruction results across the spatial domain for the in-vivo human cardiac phantom at an acceleration rate of 12x; *cf.* Section IV-F. (a) Ground-truth data. The ROI is extracted via the zero-filled DFT from scanner data. (b) Results for KRIM[M], KRIM[S], BiLMDM, KLR, FRIST, SToRM, PS-Sparse, XD-GRASP (left to right). 1st row: Diastole phase reconstructions (frame 1 of the image series); 2nd row: Error maps for the diastole phase; 3rd row: Systole phase reconstructions (frame 12 of the image series); 4th row: Error maps for systole phase. Red markings point out distortions, aliasing, or undesired artifacts in the reconstructions.

highlighted in Fig. 4. Speaking in terms of contrast, KRIM reconstructions provide the best results, closely resembling the ground-truth dMRI data. Images reconstructed by KRIM also look sharper compared to the other schemes.

E. In-Vivo Human Cardiac Phantom

Real human and cardiac cine MR data [10], with size $(N_f, N_p) = (256, 200)$, $FOV = 273 \times 300 \text{ mm}^2$ and spatial resolution $1.36 \times 1.36 \text{ mm}^2$ are considered. The data were acquired during a single breath hold. Multiple time-wraps (introducing temporal variations) and quasi-periodic spatial deformation [60] (to model respiration) were then used to generate a temporal sequence with $N_{fr} = 256$ frames. KRIM parameters $(d, N_t) = (16, 80)$.

Fig. 3(b) shows that KRIM produces images with the least NRMSE among all competing methods for all acceleration rates within the range 6x–35x. Table II summarizes the averages (over 25 runs) of all performance metrics for the acceleration rate of 12x. Similar statistical-significance tests to those of Section IV-E were carried out, producing p-values ≈ 0 and reaching thus to similar conclusions with regards to the NRMSE performance of KRIM[M] and KRIM[S].

Quantitative results are supported by Figs. 7 and 8. Those images were generated from (k, t) data which were

retrospectively under-sampled, emulating a radial sampling trajectory at an acceleration rate of 12x. As exhibited by HFEN, M1 and M2 measures, the cardiac images generated via KRIM are sharper and exhibit no blurring compared to PS-Sparse, SToRM and XD-GRASP (as pointed by the arrows). There are no patchy distortions or evidence of aliasing in the images recovered by KRIM, unlike FRIST and KLR. The temporal cross sections of the reconstructions provide better insights and reveal temporal blurring (marked with arrows) in PS-Sparse, presence of distortions and artifacts in KLR and FRIST. On the other hand, KRIM doesn't show any signs of distortions, artifacts or temporal blurring. Average computation times are also reported in Table II for 25 runs of the reconstruction schemes.

G. Acquired dMRI Cardiac Scan

Measurements were acquired via a FLASH sequence [51] from a volunteer breathing normally under $TR/TE = 5.8/4\text{ms}$, $FOV = 284 \times 350 \text{ mm}^2$ and spatial resolution $= 1.8 \times 1.8 \text{ mm}^2$. A 12-channel scanner was used to continuously measure 4500 phase-encoding lines under 1D Cartesian sampling. The phase-encoding lines are further grouped into groups of 15 (containing 5 navigator lines) to obtain $N_{fr} = 300$ frames and a data matrix of $192 \times 156 \times 300$ per channel. This corresponds to an acceleration/sampling rate of 10x. All methods

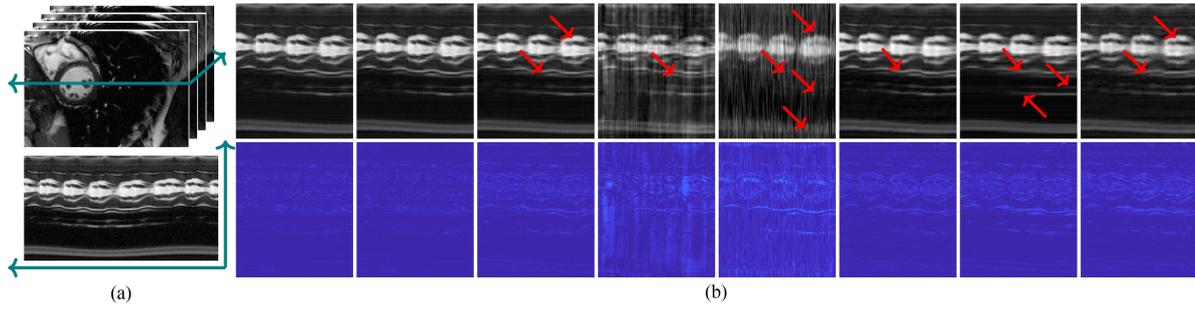


Fig. 8. Reconstruction results along the time domain for the in-vivo human cardiac phantom at an acceleration rate of 12x; *cf.* Section IV-F. (a) Ground-truth image stack and temporal cross-section. The location of the cross-section is marked on the image with a red axis. (b) Reconstruction results on the temporal cross-section via KRIM[M], KRIM[S], BiLMDM, KLR, FRIST, STORM, PS-Sparse, XD-GRASP (left to right) are exhibited in the first row, followed by the error maps in the second row. Red markings point out distortions, temporal bleeding, or artifacts in the reconstructions.

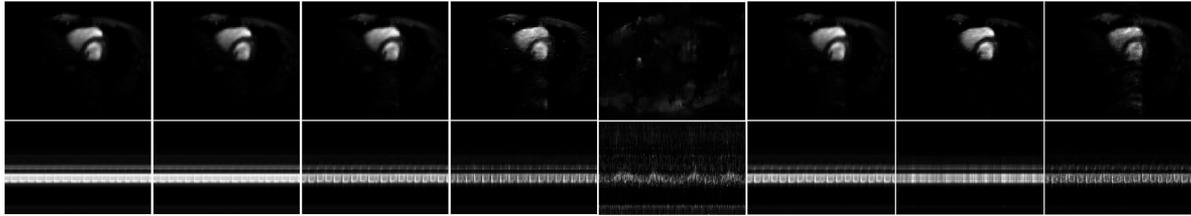


Fig. 9. Reconstructions from measurements acquired from a volunteer are exhibited for schemes KRIM[M], KRIM[S], BiLMDM, KLR, FRIST, STORM, PS-Sparse, XD-GRASP (left to right). The first row shows the spatial results for frame 1, while the second row shows the temporal cross section per scheme.

TABLE III
QUANTITATIVE PERFORMANCE ANALYSIS FOR THE ACQUIRED
DMRI CARDIAC SCAN (ACCELERATION RATE: 10X)

	M1	M2	Time
KRIM[M]	6.144	1.2×10^5	27 min
KRIM[S]	6.002	1.2×10^5	25 min
BiLMDM	5.443	1.1×10^5	5.25 hrs
KLR	4.06	1.2×10^5	14 hrs
FRIST	1.583	7×10^4	17 hrs
STORM	5.991	1.1×10^5	35 min
PS-Sparse	5.016	1.2×10^5	16 min
XD-GRASP	3.761	1.2×10^5	1.5 hrs

were fine-tuned to reach “optimal” performance per channel. Consequently, the finely tuned hyperparameters which yield “optimal” performance differ from channel to channel for all employed methods. Combining all channel data to achieve desired performance needs different data modeling and inverse-problem formulation than the ones used in this work, and for this reason, a more thorough treatment of multi-channel dMRI data is deferred to a future work.

Due to the unavailability of the ground-truth images, comparisons are qualitative in nature. Results for a single channel are exhibited in Fig. 9. Reconstructions via FRIST show severe distortions and suffer from aliasing (evident in the temporal cross section). Although it is hard to spot differences visually, the M1/M2 metrics (Table III) demonstrate the improvement in sharpness offered by KRIM over all other schemes. Average computation times of 25 trials on the multi-coil image series are provided in Table III.

V. KRIM AND PRIOR ART IN DMRI: A DETAILED DISCUSSION

This section accentuates the differences of KRIM with XD-GRASP [12], PS-Sparse [10], FRIST [9], STORM [27], [31],

[32], KLR [30], the manifold-alignment (MA) method [26], as well as BiLMDM [43].

The CS-based XD-GRASP [12] employs a golden-angle sampling scheme and applies a pre-processing sorting step to identify multiple motion-states based on respiratory and/or cardiac motion signals derived directly from the data. KRIM does not require any such pre-processing, and identifies the intrinsic correlations within different states of the same type of motion (*e.g.*, different cardiac phases) automatically via landmark points and bi-linear modeling. In addition, after each type of motion signal is separated and sorted, XD-GRASP still needs to choose a sparsifying transform in the temporal direction, while KRIM achieves that inherently via bi-linear modeling. Even though XD-GRASP produces good MRI reconstructions, KRIM produces sharper images as showed in Figs. 4, 7 and Tables I, II.

Matrix factorization schemes, similar to (12), have been also used in PS-Sparse [10] and KLR [30]. Nevertheless, KRIM differentiates itself from the two previous methods in several aspects. The PCA-based PS-Sparse [10], for example, searches for local affine/linear patches to manifolds, as in Fig. 2, and determines its temporal basis using the SVD of the faithful or navigator/pilot data, something that can be indeed effective if data show affine/linear structures within the raw-data input space, which is not often true. In contrast, KRIM searches for affine/linear patterns in an RKHS-feature space by the same token that celebrated kernel-based methods use to apply standard linear-estimation arguments in the RKHS-feature space, *e.g.*, [36]. Section IV provides empirical evidence that KRIM offers better data/feature approximation than PS-Sparse with no signs of temporal bleeding in images. Moreover, there is no need for the matrix factors $\tilde{\mathbf{D}}$ and $\tilde{\mathbf{B}}$ in (12) to be orthogonal, unlike PS-Sparse.

The transform-learning based FRIST scheme performs image reconstruction on a frame-by-frame basis, ignoring the temporal information of the dMRI data onto which KRIM capitalizes. In addition to the lack of temporal information, FRIST relies on dividing images in patches prior to reconstruction. This does help with exploiting the structural similarities in the image, but as the acceleration rate increases, the aliasing effect severely affects the image-domain data resulting in patchy distortion and blurring. This is verified in Figs. 3 and 4 by the deterioration of NRMSE with increasing acceleration rates.

On the other hand, the kernel (K)PCA-driven KLR [30] applies PCA in RKHS-feature spaces to capture potential non-linearities in respiratory and cardiac motions. Notwithstanding, KLR requires a computationally expensive pre-imaging step to map points from the RKHS-feature space back to the original input one. KRIM’s bi-linear formulation in (12) captures the latent point-cloud geometry in the RKHS-feature space without any need for pre-imaging steps. KRIM employs also complex-valued kernels to take into consideration the information carried in both real and imaginary parts of the complex-valued (k, t) -space measurements \mathbf{Y} [40], as opposed to KLR which uses only real-valued kernel functions. Furthermore, the single-kernel approaches KLR [30] and [41], as well as KRIM[S], require “optimal” tuning of the parameters of the kernel function, which can be a daunting task with the availability of the wide range of kernels in the literature [36]. KRIM[M] escapes this trap by the use of a kernel dictionary and lets the data decide on the “optimal” kernel combinations through bi-linear modeling in (8). This doesn’t only eliminate the need for pin-pointing the optimal kernel but shows better results than the ones by KLR and KRIM[S] as summarized in Tables I and II. The difference in the results between KRIM and KLR is also evident by Fig. 3 where KLR reconstructions show sharp increase in NRMSE values due to aliasing as acceleration rates increase.

SToRM [27], [31], [32] enforces “smoothness” to the underlying low-dimensional manifold by penalizing the loss of the recovery problem with several norms of a graph-Laplacian matrix, amounting to a “global” smoothness constraint over the data point-cloud. In contrast, to prevent losing local structural information and to offer as much flexibility as possible in data/feature approximations, KRIM enforces smoothness locally over the manifold via point-cloud neighborhoods and affine approximations within those neighborhoods to mimic the concept of tangent spaces. The “union” of those affine combinations is modeled by the sparse matrix $\tilde{\mathbf{B}}$ in (12). KRIM performs better than SToRM along edges (as verified by the HFEN values in Tab. I and II), and produces sharper images with almost no aliasing effect and temporal bleeding. The manifold-alignment (MA) method [26] follows a two-step approach, where i) an LLE-inspired [46] module is employed to “align” k -space data into a low-dimensional manifold via user-defined weighting functions, and ii) a module found also in XD-GRASP [12] is used to impute the high-dimensional data by allowing contributions from neighboring points in the manifold which was learned during the first step of the MA approach. User-defined weighting functions are also introduced to determine the contributions of the neighboring manifold points in data imputations. In

contrast, KRIM employs no user-defined weighting functions to determine contributions of neighboring manifold points, but allows those points to determine by themselves the appropriate contributions, both in the module of Section III-D and in the bi-linear model (4).

KRIM exhibited improvement in performance over its predecessor BiLMDM [43]. This is mainly a result of reaping the benefits of non-parametric approximations in the potentially infinite dimensional and functional RKHS feature spaces over the Euclidean-space setting of BiLMDM, by the same token that kernel-based methods offer more approximation capabilities than classical learning schemes in Euclidean spaces [36]. Further, the inverse problems of KRIM and BiLMDM are different; the requirement for data consistency has been moved from the loss function (BiLMDM) to a constraint in KRIM [see (11d) and (13b)]. As such, any DFT operations are employed only in the sub-task (14c), and thus less often than BiLMDM, putting KRIM way ahead of BiLMDM in terms of computation times, and paving the way for more efficient implementations in future studies. Numerical evidence of the improvements were summarized in Tables I, II and III.

Recently, DeepL methods have been developed for accelerated cardiac cine imaging [15]–[21]. These methods require a *large* number of training samples acquired *beforehand* to train the deep neural network, while KRIM’s training data are a *small* number of navigator data that are acquired *during* the accelerated scan. Consequently, KRIM’s “training” is performed on a much less complicated model than standard DeepL methods, avoiding typical issues such as overfitting. Because of these distinctions, a fair comparison of KRIM vs. DeepL methods is challenging and thus not performed here.

VI. CONCLUSION AND FUTURE DIRECTIONS

This manuscript laid down the foundations for a novel imputation-by-regression scheme based on classic kernel-approximation arguments for features of the measured data which lie into or close to an unknown-to-the-user smooth manifold embedded in potentially infinite dimensional RKHSs. A novel bi-linear data/feature-approximation model employs multiple kernel functions, bears the advantage of not requiring tuning for an optimal kernel, avoids computationally expensive pre-imaging steps, accommodates complex-valued data, and demonstrates a low-rank structure to promote computationally efficient solutions. The proposed framework, coined kernel regression imputation in manifolds (KRIM), facilitates data reconstruction from partial observations with missing entries, without the need for a fully observed training dataset. This work provided also evidence of KRIM’s rich potential via a dMRI setting, where high-resolution medical images need to be recovered from severely under-sampled data. The manuscript has also put forth elaborate quantitative and qualitative results which highlight the superior performance of KRIM over state-of-the-art dMRI schemes under several performance metrics and with a small computational footprint.

The more data are gathered, *e.g.*, excessively large N_k and N_{fr} , the heavier the computational burden of Algorithm 1

becomes. Future research paths include new online-learning ways to surmount such obstacles, and methods to simplify or overcome the cumbersome computations of the Lipschitz coefficients (Supp-6) and (Supp-13) which are currently based on singular-value-decomposition operations. The assessment of clinical data necessitate expertise from medical professionals, *e.g.*, [61], and is thus deferred to a future work in a more clinically oriented publication venue. Finally, the application of KRIM to other domains, such as recommender systems [3] and single-cell genomics [4], is currently under investigation.

APPENDIX

Consider the space $\mathbb{C}^{m \times n}$ of all $m \times n$ complex-valued matrices, equipped with the inner product $\langle \mathbf{A} | \mathbf{B} \rangle := \text{trace}(\mathbf{A}^H \mathbf{B})$, $\forall \mathbf{A}, \mathbf{B} \in \mathbb{C}^{m \times n}$, where \mathbf{H} represents the Hermitian operation onto vectors/matrices. The induced norm of $\mathbb{C}^{m \times n}$ by the previous inner product coincides with the Frobenius one: $\|\cdot\|_F = \langle \cdot | \cdot \rangle^{1/2}$. It is worth noting that $\langle \mathbf{B} | \mathbf{A} \rangle = \overline{\langle \mathbf{A} | \mathbf{B} \rangle}$ with the overline symbol representing entry-wise complex conjugation of a scalar/vector/matrix. Moreover, the spectral norm $\|\mathbf{A}\|_2$ of matrix $\mathbf{A} \in \mathbb{C}^{m \times n}$ is defined as $\lambda_{\max}^{1/2}(\mathbf{A}^H \mathbf{A})$, where $\lambda_{\max}(\cdot)$ stands for the maximum eigenvalue of a symmetric matrix. In the case of $m = n$, then $\|\mathbf{A}\|_2 = \lambda_{\max}(\mathbf{A})$.

Given the positive integers ρ_1, ρ_2 and the linear mapping $\mathcal{T} : \mathbb{C}^{\rho_1 \times \rho_2} \rightarrow \mathbb{C}^{N_k \times N_{fr}}$, the adjoint of \mathcal{T} is the linear mapping $\mathcal{T}^* : \mathbb{C}^{N_k \times N_{fr}} \rightarrow \mathbb{C}^{\rho_1 \times \rho_2}$ defined as $\langle \mathbf{A} | \mathcal{T}(\mathbf{B}) \rangle = \langle \mathcal{T}^*(\mathbf{A}) | \mathbf{B} \rangle$, $\forall \mathbf{A} \in \mathbb{C}^{N_k \times N_{fr}}, \forall \mathbf{B} \in \mathbb{C}^{\rho_1 \times \rho_2}$. For example, with regards to the sampling mapping $\mathcal{S}(\cdot)$ in Section II, its adjoint $\mathcal{S}^*(\cdot) = \mathcal{S}(\cdot)$, *i.e.*, $\mathcal{S}(\cdot)$ is self-adjoint, and $\mathcal{S}^2(\cdot) := \mathcal{S}[\mathcal{S}(\cdot)] = \mathcal{S}(\cdot)$. Moreover, for the DFT implementation in Julia [42], $\mathcal{F}^* = N_k \mathcal{F}^{-1}$ and $\mathcal{F}_t^* = N_{fr} \mathcal{F}_t^{-1}$. If matrix $\mathbf{A} \in \mathbb{C}^{m \times n}$ is viewed as a linear mapping $\mathbf{A} : \mathbb{C}^n \rightarrow \mathbb{C}^m$ on vectors via matrix-vector multiplication, then $\mathbf{A}^* = \mathbf{A}^H$. A generally non-linear mapping $T(\cdot)$ is called Lipschitz continuous with coefficient $L \in \mathbb{R}_{>0}$, if $\|T(\mathbf{A}) - T(\mathbf{B})\|_F \leq L \|\mathbf{A} - \mathbf{B}\|_F$, $\forall \mathbf{A}, \mathbf{B}$.

A complex reproducing kernel Hilbert space (RKHS) \mathcal{H} is a Hilbert space consisting of functions $f : \mathbb{C}^\nu \rightarrow \mathbb{C} : \mathbf{l} \mapsto f(\mathbf{l})$ such that (s.t.) for any \mathbf{l} the mapping $f \mapsto f(\mathbf{l})$ is continuous over \mathcal{H} [33], [39], [40]. For a Hilbert space \mathcal{H} , the bivariate functional $\kappa : \mathbb{C}^\nu \times \mathbb{C}^\nu \rightarrow \mathbb{C} : (\mathbf{l}, \mathbf{l}') \mapsto \kappa(\mathbf{l}, \mathbf{l}')$ is called *reproducing* if i) $\kappa(\mathbf{l}, \cdot) \in \mathcal{H}$ for any \mathbf{l} ; and ii) $\langle f | \kappa(\mathbf{l}, \cdot) \rangle_{\mathcal{H}} = f(\mathbf{l})$ for any \mathbf{l} and any $f \in \mathcal{H}$, where $\langle \cdot | \cdot \rangle_{\mathcal{H}}$ stands for the inner product in \mathcal{H} . It turns out that every reproducing kernel κ turns its associated \mathcal{H} into an RKHS, and that every RKHS \mathcal{H} possesses a unique reproducing kernel κ [39]. The *feature mapping* $\varphi(\cdot)$ associated with the RKHS \mathcal{H} is defined as $\varphi(\cdot) : \mathbb{C}^\nu \rightarrow \mathcal{H} : \mathbf{l} \mapsto \varphi(\mathbf{l}) := \kappa(\mathbf{l}, \cdot)$.

Notice that in a complex RKHS the inner product is Hermitian: $\langle f | f' \rangle_{\mathcal{H}} = \overline{\langle f' | f \rangle_{\mathcal{H}}}$. For this reason and to revert to familiar notation, let $f^H f' := \langle f | f' \rangle_{\mathcal{H}}$, where superscript \mathbf{H} denotes the usual Hermitian operation on vectors/matrices. The reproducing property of kernel κ yields the celebrated “kernel trick”: $\kappa(\mathbf{l}, \mathbf{l}') = \varphi(\mathbf{l})^H \varphi(\mathbf{l}')$, for any \mathbf{l} and $\mathbf{l}' \in \mathbb{C}^\nu$. It can be thus verified that $\kappa(\mathbf{l}, \mathbf{l}') = \kappa(\mathbf{l}', \mathbf{l})$ [39]. Popular kernels for complex RKHSs are: i) The linear kernel $\kappa_L(\mathbf{l}, \mathbf{l}') := \mathbf{l}^H \mathbf{l}'$; ii) the Gaussian kernel $\kappa_G; \gamma(\mathbf{l}, \mathbf{l}') := \exp[-\gamma(\mathbf{l} - \mathbf{l}')^T (\mathbf{l} - \mathbf{l}')]^2$ [39], [40], where $\gamma \in \mathbb{R}_{>0}$ and superscript \mathbf{T} stands for vector/matrix

transposition; and iii) the polynomial kernel $\kappa_{P; (c, r)}(\mathbf{l}, \mathbf{l}') := (\mathbf{l}^H \mathbf{l}' + c)^r$, where $r \in \mathbb{Z}_{>0}$, $c \in \mathbb{R}_{>0}$.

For vectors $\{\mathbf{l}_k\}_{k=1}^{N_l} \subset \mathbb{C}^\nu$ and $\mathbf{L} := [\mathbf{l}_1, \dots, \mathbf{l}_{N_l}]$, let

$$\Phi(\mathbf{L}) := [\varphi(\mathbf{l}_1), \dots, \varphi(\mathbf{l}_{N_l})]. \quad (18)$$

The associated $N_l \times N_l$ kernel matrix \mathbf{K}_L is defined as the matrix whose (k, k') th entry is equal to $\kappa(\mathbf{l}_k, \mathbf{l}_{k'})$, or $\mathbf{K}_L := \Phi(\mathbf{L})^H \Phi(\mathbf{L})$ for simplicity.

For a convex function $g(\cdot) : \mathbb{C}^{m \times n} \rightarrow \mathbb{R} \cup \{+\infty\}$ and $\lambda \in \mathbb{R}_{>0}$, the proximal mapping $\text{Prox}_{\lambda g}(\cdot) : \mathbb{C}^{m \times n} \rightarrow \mathbb{C}^{m \times n}$ is defined as $\text{Prox}_{\lambda g}(\mathbf{A}) := \arg \min_{\mathbf{B} \in \mathbb{C}^{m \times n}} \lambda g(\mathbf{B}) + (1/2) \|\mathbf{A} - \mathbf{B}\|_F^2$ [62]. For example, in the case where g becomes the indicator function ι_C with respect to a closed convex set $\mathcal{C} \subset \mathbb{C}^{m \times n}$, *i.e.*, $\iota_C(\mathbf{A}) := 0$, if $\mathbf{A} \in \mathcal{C}$, while $\iota_C(\mathbf{A}) := +\infty$, if $\mathbf{A} \notin \mathcal{C}$, then $\text{Prox}_{\lambda \iota_C}$ becomes the metric projection mapping onto \mathcal{C} : $\text{Prox}_{\lambda \iota_C}(\mathbf{A}) = \arg \min_{\mathbf{B} \in \mathcal{C}} \|\mathbf{A} - \mathbf{B}\|_F$. Moreover, in the case where g is the ℓ_1 -norm $\|\cdot\|_1$, then the (i, j) th entry of $\text{Prox}_{\lambda \|\cdot\|_1}(\mathbf{A})$ is obtained by the soft-thresholding rule [62]:

$$[\text{Prox}_{\lambda \|\cdot\|_1}(\mathbf{A})]_{ij} = [\mathbf{A}]_{ij} \left(1 - \frac{\lambda}{\max\{\lambda, |[\mathbf{A}]_{ij}|\}} \right). \quad (19)$$

Notation $\text{Soft}(\mathbf{A}, \lambda)$ is often used instead of $\text{Prox}_{\lambda \|\cdot\|_1}(\mathbf{A})$.

ACKNOWLEDGMENT

The authors would like to thank the anonymous Reviewers for their constructive comments, and the Institute of Biomedical Engineering, ETH Zurich as well as the Cardiovascular Magnetic Resonance Society for making the data used in Section IV.

REFERENCES

- [1] Z.-P. Liang and P. C. Lauterbur, *Principles of Magnetic Resonance Imaging: A Signal Processing Perspective*. Piscataway, NJ, USA: IEEE Press, 2000.
- [2] Y. Yang, J. Ma, and S. Osher, “Seismic data reconstruction via matrix completion,” *Inverse Problems Imag.*, vol. 7, no. 4, 2013, Art. no. 1379.
- [3] G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions,” *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, Jun. 2005.
- [4] D. Lähnemann *et al.*, “Eleven grand challenges in single-cell data science,” *Genome Biol.*, vol. 21, no. 1, pp. 1–35, 2020.
- [5] N. J. Horton and K. P. Kleinman, “Much ado about nothing: A comparison of missing data methods and software to fit incomplete data regression models,” *Amer. Statistician*, vol. 61, no. 1, pp. 79–90, 2007.
- [6] M. Lustig, D. Donoho, and J. M. Pauly, “Sparse MRI: The application of compressed sensing for rapid MR imaging,” *Magn. Reson. Med.*, vol. 58, no. 6, pp. 1182–1195, 2007.
- [7] H. Jung, J. C. Ye, and E. Y. Kim, “Improved k-t BLAST and k-t SENSE using FOCUS,” *Phys. Med. Biol.*, vol. 52, no. 11, pp. 3201–3226, 2007.
- [8] Y. Wang and L. Ying, “Compressed sensing dynamic cardiac cine MRI using learned spatiotemporal dictionary,” *IEEE Trans. Biomed. Eng.*, vol. 61, no. 4, pp. 1109–1120, Apr. 2014.
- [9] B. Wen, S. Ravishanker, and Y. Bresler, “FRIST-Flipping and rotation invariant sparsifying transform learning and applications,” *Inverse Problems*, vol. 33, no. 7, 2017, Art. no. 074007.
- [10] 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.
- [11] R. Otazo, E. Candès, 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, Mar. 2015.
- [12] 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.

- [13] S. Ravishanker and Y. Bresler, "MR image reconstruction from highly undersampled k-space data by dictionary learning," *IEEE Trans. Med. Imag.*, vol. 30, no. 5, pp. 1028–1041, May 2011.
- [14] S. Ravishanker, 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.
- [15] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: Going beyond Euclidean data," *IEEE Signal Process. Mag.*, vol. 34, no. 4, pp. 18–42, Jul. 2017.
- [16] J. Schlemper, J. Caballero, J. V. Hajnal, A. N. Price, and D. Rueckert, "A deep cascade of convolutional neural networks for dynamic MR image reconstruction," *IEEE Trans. Med. Imag.*, vol. 37, no. 2, pp. 491–503, Feb. 2018.
- [17] C. Qin, J. Schlemper, J. Caballero, A. N. Price, J. V. Hajnal, and D. Rueckert, "Convolutional recurrent neural networks for dynamic MR image reconstruction," *IEEE Trans. Med. Imag.*, vol. 38, no. 1, pp. 280–290, Jan. 2019.
- [18] M. Chen, H. Jiang, W. Liao, and T. Zhao, "Nonparametric regression on low-dimensional manifolds using deep ReLU networks: Function approximation and statistical recovery," 2019, *arXiv:1908.01842*.
- [19] S. Biswas, H. K. Aggarwal, and M. Jacob, "Dynamic MRI using model-based deep learning and SToRM priors: MoDL-SToRM," *Magn. Reson. Med.*, vol. 82, no. 1, pp. 485–494, Jul. 2019.
- [20] D. Liang, J. Cheng, Z. Ke, and L. Ying, "Deep magnetic resonance image reconstruction: Inverse problems meet neural networks," *IEEE Signal Process. Mag.*, vol. 37, no. 1, pp. 141–151, Jan. 2020.
- [21] C. M. Sandino, J. Y. Cheng, F. Chen, M. Mardani, J. M. Pauly, and S. S. Vasanawala, "Compressed sensing: From research to clinical practice with deep neural networks: Shortening scan times for magnetic resonance imaging," *IEEE Signal Process. Mag.*, vol. 37, no. 1, pp. 117–127, Jan. 2020.
- [22] V. Antun, F. Renna, C. Poon, B. Adcock, and A. C. Hansen, "On instabilities of deep learning in image reconstruction and the potential costs of AI," *Proc. Nat. Acad. Sci.*, vol. 117, no. 48, pp. 30088–30095, 2020.
- [23] J. Nilsson, F. Sha, and M. I. Jordan, "Regression on manifolds using kernel dimension reduction," in *Proc. Int. Conf. Mach. Learn.*, 2007, pp. 697–704.
- [24] A. V. Little, M. Maggioni, and L. Rosasco, "Multiscale geometric methods for data sets I: Multiscale SVD, noise and curvature," *Appl. Comput. Harmon. Anal.*, vol. 43, no. 3, pp. 504–567, 2017.
- [25] M. Usman, D. Atkinson, C. Kolbitsch, T. Schaeffter, and C. Prieto, "Manifold learning based ECG-free free-breathing cardiac CINE MRI," *J. Magn. Reson. Imag.*, vol. 41, no. 6, pp. 1521–1527, 2015.
- [26] X. Chen *et al.*, "High-resolution self-gated dynamic abdominal MRI using manifold alignment," *IEEE Trans. Med. Imag.*, vol. 36, no. 4, pp. 960–971, Apr. 2017.
- [27] S. Poddar and M. Jacob, "Dynamic MRI using smoothness regularization on manifolds (SToRM)," *IEEE Trans. Med. Imag.*, vol. 35, no. 4, pp. 1106–1115, Apr. 2016.
- [28] U. Nakarmi, K. Slavakis, J. Lyu, and L. Ying, "M-MRI: A manifold-based framework to highly accelerated dynamic magnetic resonance imaging," in *Proc. Int. Symp. Biomed. Imag.*, 2017, pp. 19–22.
- [29] U. Nakarmi, K. Slavakis, and L. Ying, "MLS: Joint manifold-learning and sparsity-aware framework for highly accelerated dynamic magnetic resonance imaging," in *Proc. Int. Symp. Biomed. Imag.*, 2018, pp. 1213–1216.
- [30] U. Nakarmi, Y. Wang, J. Lyu, D. Liang, and L. Ying, "A kernel-based low-rank (KLR) model for low-dimensional manifold recovery in highly accelerated dynamic MRI," *IEEE Trans. Med. Imag.*, vol. 36, no. 11, pp. 2297–2307, Nov. 2017.
- [31] S. Poddar, Y. Q. Mohsin, D. Ansah, B. Thattaliyath, R. Ashwath, and M. Jacob, "Manifold recovery using kernel low-rank regularization: Application to dynamic imaging," *IEEE Trans. Comput. Imag.*, vol. 5, no. 3, pp. 478–491, Sep. 2019.
- [32] O. Arif, H. Afzal, H. Abbas, M. F. Amjad, J. Wan, and R. Nawaz, "Accelerated dynamic MRI using kernel-based low rank constraint," *J. Med. Syst.*, vol. 43, no. 8, pp. 1–11, 2019.
- [33] N. Aronszajn, "Theory of reproducing kernels," *Trans. Amer. Math. Soc.*, vol. 68, no. 3, pp. 337–404, 1950.
- [34] G. S. Kimeldorf and G. Wahba, "A correspondence between Bayesian estimation on stochastic processes and smoothing by splines," *Ann. Math. Statist.*, vol. 41, no. 2, pp. 495–502, 1970.
- [35] A. B. Tsybakov, *Introduction to Nonparametric Estimation*. New York, NY, USA: Springer, 2008.
- [36] B. Schölkopf and A. J. Smola, *Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2001.
- [37] L. W. Tu, *An Introduction to Manifolds*. New York, NY, USA: Springer, 2008.
- [38] S. S. Bucak, R. Jin, and A. K. Jain, "Multiple kernel learning for visual object recognition: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1354–1369, Jul. 2014.
- [39] I. Steinwart, D. Hush, and C. Scovel, "An explicit description of the reproducing kernel Hilbert spaces of Gaussian RBF kernels," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4635–4643, Oct. 2006.
- [40] P. Bouboulis and S. Theodoridis, "Extension of Wirtinger's calculus to reproducing kernel Hilbert spaces and the complex kernel LMS," *IEEE Trans. Signal Process.*, vol. 59, no. 3, pp. 964–978, Mar. 2011.
- [41] G. N. Shetty, K. Slavakis, U. Nakarmi, G. Scutari, and L. Ying, "Kernel Bilinear modeling for reconstructing data on manifolds: The dynamic-MRI case," in *Proc. Eur. Signal Process. Conf.*, 2021, pp. 1482–1486.
- [42] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A fresh approach to numerical computing," *SIAM Rev.*, vol. 59, no. 1, pp. 65–98, 2017.
- [43] G. N. Shetty, K. Slavakis, A. Bose, U. Nakarmi, G. Scutari, and L. Ying, "Bi-linear modeling of data manifolds for dynamic-MRI recovery," *IEEE Trans. Med. Imag.*, vol. 39, no. 3, pp. 688–702, Mar. 2020.
- [44] V. De Silva and J. B. Tenenbaum, "Sparse multidimensional scaling using landmark points," Stanford Univ., Tech. Rep., vol. 120, 2004.
- [45] K. Slavakis, G. B. Giannakis, and G. Leus, "Robust sparse embedding and reconstruction via dictionary learning," in *Proc. Conf. Inf. Sci. Syst.*, Baltimore, USA, 2013, pp. 1–6.
- [46] L. K. Saul and S. T. Roweis, "Think globally, fit locally: Unsupervised learning of low dimensional manifolds," *J. Mach. Learn. Res.*, vol. 4, pp. 119–155, 2003.
- [47] A. Maleki, L. Anitori, Z. Yang, and R. G. Baraniuk, "Asymptotic analysis of complex LASSO via complex approximate message passing (CAMP)," *IEEE Trans. Inf. Theory*, vol. 59, no. 7, pp. 4290–4308, Jul. 2013.
- [48] K. Slavakis and I. Yamada, "Fejér-monotone hybrid steepest descent method for affinely constrained and composite convex minimization tasks," *Optimization*, vol. 67, no. 11, pp. 1963–2001, 2018.
- [49] F. Facchinei, G. Scutari, and S. Sagratella, "Parallel selective algorithms for nonconvex Big Data optimization," *IEEE Trans. Signal Process.*, vol. 63, no. 7, pp. 1874–1889, Apr. 2015.
- [50] L. Wissmann, C. Santelli, W. P. Segars, and S. Kozierke, "MRXCAT: Realistic numerical phantoms for cardiovascular magnetic resonance," *J. Cardiovasc. Magn. Reson.*, vol. 16, no. 1, pp. 1–11, 2014.
- [51] A. Haase, J. Frahm, D. Matthaei, W. Hancicke, and K.-D. Merboldt, "FLASH imaging. Rapid NMR imaging using low flip-angle pulses," *J. Magn. Reson. (1969)*, vol. 67, no. 2, pp. 258–266, 1986.
- [52] K. Slavakis and G. Shetty, "Kernel regression imputation in manifolds: Videos of dMRI data," 2021. [Online]. Available: <https://dx.doi.org/10.21227/tary-nv39>
- [53] W. Zhou, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Imag. Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
- [54] M. Subbarao, T.-S. Choi, and A. Nikzad, "Focusing techniques," *Opt. Eng.*, vol. 32, no. 11, pp. 2824–2837, 1993.
- [55] G. N. Shetty, "Kernel regression imputation in manifolds: Julia code." [Online]. Available: <https://github.com/KostasSlavakis/Kernel-Regression-Imputation-in-Manifolds-KRIM>
- [56] S. Poddar and M. Jacob, "SToRM MATLAB package." Accessed: 2018. [Online]. Available: <https://github.com/sunrita-poddar/12SToRM>
- [57] L. Feng, L. Axel, H. Chandarana, K. T. Block, D. K. Sodickson, and R. Otazo, "XD-GRASP MATLAB package." Accessed: 2019. [Online]. Available: <https://cai2r.net/resources/software/xd-grasp-matlab-code>
- [58] B. Wen, S. Ravishanker, and Y. Bresler, "FRIST MATLAB package." Accessed: 2018. [Online]. Available: https://github.com/wenbihan/FRIST_ivp2017
- [59] L. A. Wasserman, *All of Statistics: A Concise Course in Statistical Inference*. New York, NY, USA: Springer, 2004.
- [60] Y.-C. Tsai, H.-D. Lin, Y.-C. Hu, C.-L. Yu, and K.-P. Lin, "Thin-plate spline technique for medical image deformation," *J. Med. Biol. Eng.*, vol. 20, no. 4, pp. 203–210, 2000.
- [61] ACR-NASCI-SPR, "practice parameter for the performance and interpretation of cardiac magnetic resonance imaging (MRI)," *Amer. College Radiol.*, 2021. [Online]. Available: <https://www.acr.org/-/media/ACR/Files/Practice-Parameters/Quant-CardiacCT-MR.pdf>
- [62] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. New York, NY, USA: Springer, 2011.