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

Konstantinos Slavakis,^{*} Gaurav N. Shetty,[†] Loris Cannelli,[‡] Gesualdo Scutari,[§] Ukash Nakarmi,[¶] and Leslie Ying^{||}

Abstract—This paper introduces a non-parametric kernel-based modeling framework for imputation by regression on data that are assumed to lie close to an unknown-to-the-user smooth manifold in a Euclidean space. The proposed framework, coined kernel regression imputation in manifolds (KRIM), needs no training data to operate. Aiming at computationally efficient solutions, KRIM utilizes a small number of “landmark” data-points to extract geometric information from the measured data via parsimonious affine combinations (“linear patches”), which mimic the concept of tangent spaces to smooth manifolds and take place in functional approximation spaces, namely reproducing kernel Hilbert spaces (RKHSs). Multiple complex RKHSs are combined in a data-driven way to surmount the obstacle of pinpointing the “optimal” parameters of a single kernel through cross-validation. The extracted geometric information is incorporated into the design via a novel bi-linear data-approximation model, and the imputation-by-regression task takes the form of an inverse problem which is solved by an iterative algorithm with guaranteed convergence to a stationary point of the non-convex loss function. To showcase the modular character and wide applicability of KRIM, this paper highlights the application of KRIM to dynamic magnetic resonance imaging (dMRI), where reconstruction of high-resolution 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, non-parametric, kernel, imputation, manifold, MRI.

I. INTRODUCTION

A. Motivation and Problem Statement

Dynamic magnetic resonance imaging (dMRI) is a non-invasive MRI modality that aims at observing possibly moving body organs over time via sophisticated signal acquisition, and shows rich potential as a medium for cardiac, brain and neurological diagnosis [1, 2]. Physical limitations in MRI scanners often compromise signal sampling at rates lower than

the required ones, generating “holes/gaps” or “missing data” in measurements, causing distortion, aliasing, and motion blurring in the observed image series [1]. Such after-effects may hamper the diagnosis ability of a physician.

Missing values/data commonly occur in a wide range of applications owing to a number of reasons like damaged sensors, physical constraints on the signal-acquisition mechanism, or simply because some of the information is not available. Missing observations translate to uncertainties that negatively affect the end result: distortion in seismic imaging [3], sub-optimal data modeling in recommender systems [4], as well as large numbers of false positives/negatives in detection methods for single-cell genomics [5].

Several strategies have been proposed to address the missing-data problem. This study focuses on *imputation by regression*, where, typically, (i) a regression model that fits the measured data values is identified with the regressors being the measured data themselves, and (ii) unknown data values are imputed by the identified regression model and the measured data [6]. Along these lines, this paper puts forth an efficient non-parametric framework, coined *kernel regression imputation in manifolds (KRIM)*, to address imputation-by-regression tasks via exploration and exploitation of latent geometric information which is hidden beneath the measured data. Although KRIM can be tailored to fit into any application domain where data approximation is needed, the dMRI setting is used here to demonstrate the way data are collected and side-information is employed, solidify arguments, clarify concepts, and validate the effectiveness of the approach via extensive numerical tests.

B. Prior Art

The standard approach to imputation by regression is compressed sensing (CS) where signals (image frames or patches) are sparsely represented by dictionary “atoms” (vectors) that reside in appropriately defined transform domains (feature spaces) [7–11]. CS-MRI approaches witnessed also the development of low-rank models to exploit inherent correlations for MRI recovery even at the image-patch analysis level [12, 13].

Recent efforts on imputation by regression revolve around deep-learning (DeepL) approaches [14–18] that rely on lengthy processes to learn models from training data prior to reconstructing MR test images. Notwithstanding, concerns were brought to light in [19] with regards to DeepL approaches for medical imaging through tests which demonstrated undesired perturbations and structural changes (e.g., small tumors) in reconstructed images.

This work was supported in part by the NSF (CCF/CIF) grant 1718796.

^{*}K. Slavakis is with the Department of Information and Communications Engineering of Tokyo Institute of Technology, Tokyo, Japan (e-mail: slavakis.k.aa@m.titech.ac.jp).

[†]G. N. Shetty is with the Medical School of University of Michigan, Ann Arbor MI, USA (e-mail: gshetty@med.umich.edu).

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

[§]G. Scutari is with the Department of Industrial Engineering of Purdue University, West Lafayette IN, USA (e-mail: gscutari@purdue.edu).

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

^{||}L. Ying is with the Departments of Electrical Engineering and Biomedical Engineering of University at Buffalo, SUNY, Buffalo NY, USA (e-mail: leiying@buffalo.edu).

Manifold learning, built on the hypothesis that measured data lie onto or close to a low-dimensional manifold embedded in a high-dimensional space [20–22], has found applications also in dMRI [23–26]. According to the standard manifold-learning route, graph Laplacian matrices, which capture relations within data-point clouds, are employed in quadratic regularizers of an inverse problem to recover dMR images in [24]. Following a less conventional path, [26] relies on affine relations between low-dimensional renditions of the acquired data to identify a temporal basis, and follows the lines of [12] to proceed with the computation also of a spatial basis to describe dMR image frames.

Kernels have been also employed in MRI data-recovery tasks, *e.g.*, [26–29], following their success in the realms of non-parametric approximation [30–33] and machine learning [34]. Loosely speaking, the premise of kernel methods is to non-linearly map the observed data from the input space to a high-dimensional feature one where linear operations are employed. For instance, [27] approaches manifold learning via the application of the celebrated principal component analysis (PCA) into a kernel-induced feature space, while [28] proposes a kernel graph-Laplacian-matrix extension of [24]. A challenging task for kernel-based schemes is to tune kernel parameters to “optimally” fit the measured data via cross-validation. Another obstacle for kernel approaches, especially in regression tasks, is the need for explicit pre-imaging steps from the feature space back to the original input one. Pre-imaging procedures are often computationally costly and rely on the existence of an invertible function that may not even exist for a specific choice of a kernel function, *e.g.*, [27]. Lastly, kernel-based MRI schemes often leave the complex-valued nature of the measured dMRI data underused, *e.g.*, [27].

C. Main Contributions

The proposed KRIM framework offers a novel non-parametric approach to kernel-based imputation by regression via the hypothesis that the measured data lie into or close to an unknown-to-the-user smooth manifold in a high-dimensional Euclidean space. KRIM requires no lengthy learning procedures on training data to identify latent geometric structures, perform regression on the measured data and impute their missing values, unlike mainstream DeepL approaches [14–18].

KRIM follows less typical manifold-learning paths than those which capitalize on graph Laplacian matrices to unveil data structures, *e.g.*, [24, 28]. More specifically and 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.* Sec. III-A. Moving ahead, KRIM exploits the landmark points to extract geometric information from the data via parsimonious affine combinations (“linear patches”) that mimic the concept of tangent spaces to smooth manifolds [35], and take place in functional approximation spaces, namely reproducing kernel Hilbert spaces (RKHSs) [30–33]; *cf.* Sec. III-B. The learned geometric information is incorporated into the design via a novel bi-linear regression approach, rooted in classic RKHS-based approximation principles, able to describe the

measured data and impute their missing values, and promote computationally efficient solutions by a dimensionality-reduction module [36]; *cf.* Secs. III-C and III-D. It is worth noticing here that the introduction of the bi-linear model enables KRIM to avoid pre-imaging steps which map points of an RKHS space back to points of the input one, unlike [27]. Moreover, to accommodate complex-valued data, the employed approximation space comprises multiple complex RKHSs [37, 38], where multiplicity in kernels is introduced to surmount the usually cumbersome task of pin-pointing the “optimal” parameters of a single kernel through cross-validation, *e.g.*, [39]; *cf.* Sec. III-C.

The imputation-by-regression task takes the form of an inverse problem which is solved by an algorithmic procedure with guaranteed convergence to a stationary point of the non-convex loss function; *cf.* Sec. III-E. To showcase the modular character of KRIM, as well as its 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.* Sec. III-F. Extensive numerical tests on synthetic and real dMRI data demonstrate the superior performance of KRIM, under several metrics and with a small computational footprint, over state-of-the-art approaches; *cf.* Sec. IV.

KRIM differentiates itself from its predecessor [40] not only in the usage of RKHSs, but also in the structure of the inverse dMRI problem, where the computationally intensive 2D discrete Fourier transform (DFT) is employed in the learning task only as a constraint and is not involved at all at the loss function, unlike [40]. This approach yields large savings in computation times when compared with [40], as extensive numerical tests show in Sec. IV. A more detailed discussion on similarities and differences between KRIM and state-of-the-art approaches can be found in Sec. V.

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 Appendix A. Preliminary results of this study were documented in [41].

II. DATA COLLECTION

Measured data are denoted by $\mathcal{Y}$ and are considered to be complex valued. In dMRI, $\mathcal{Y}$ takes the form of a tensor defined on the $(N_p \times N_f \times N_{fr})$ -sized “(k,t)-space”; *cf.* Fig. 1a. Each “slice” $\mathcal{Y}_t$ of tensor $\mathcal{Y}$, with $t \in \{1, \dots, N_{fr}\}$, corresponds to the $(N_p \times N_f)$ -sized “k-space” measurements collected at the time instance t . Positive integers N_p and N_f represent the numbers of phase- and frequency-encoding lines, respectively [1], while $N_k := N_p N_f$ stands for the number of measurement values required for a fully-scanned k-space frame. “Full-resolution” MR images can be obtained $\forall t$ by $\mathcal{X}_t := \mathcal{F}^{-1}(\mathcal{Y}_t)$ (*cf.* Fig. 1b), 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

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

$\mathbf{Y} := [\mathbf{y}_1, \dots, \mathbf{y}_{N_{fr}}] \in \mathbb{C}^{N_k \times N_{fr}}$. Abusing notation to avoid clutter in symbols, $\mathcal{F}(\cdot)$ still represents the 2D DFT even when applied to vectors, *i.e.*, $\mathcal{F}[\text{vec}(\mathcal{X}_t)] := \text{vec}[\mathcal{F}(\mathcal{X}_t)] = \mathbf{y}_t$, so that $\mathbf{X} := \mathcal{F}^{-1}(\mathbf{Y}) := [\mathcal{F}^{-1}(\mathbf{y}_1), \mathcal{F}^{-1}(\mathbf{y}_2), \dots, \mathcal{F}^{-1}(\mathbf{y}_{N_{fr}})]$.

Due to physical limitations in MR scanners, data acquisition in dMRI is a slow process with scanners being usually unable to keep up with the dynamic nature of the scanning area (patient/organ motion) [2]. Moreover, to reduce as much as possible the discomfort of patients during scanning procedures, it is a common practice to partially sample the $(N_p \times N_f)$ -sized k -space and to collect only a small number of data per frame, often much smaller than the $N_k = N_p N_f$ one which is required for a full scan. To express this partial observation in mathematical terms, define the 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 the 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. The missing-data problem permeates any data-analytic application, *e.g.*, unreported observations in recommender systems [4] and dropout values in gene expressions [5].

Given the previous stringent sampling conditions, the MR-research community has proposed several strategies to sample the partially observed k -space; *e.g.*, 1D Cartesian (*cf.* Fig. 1c) and radial trajectories (*cf.* Fig. 1d) [1, 42]. A general trend is to heavily acquire data from the central (*i.e.*, “low frequency”) k -space region, which carries contrast information, while sparsely acquire data 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 scarcely appearing in the high-frequency one (*cf.* Fig. 1c). 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. 1d).

All imputation frameworks, including the proposed one, rely on a *faithful* subset of the measured data to impute the missing observations of the partially observed $\mathcal{S}(\mathbf{Y})$. In

MRI, for example, the heavily sampled central region of the k -space, usually called “navigator data” [24, 26, 27, 40], constitutes the faithful data. More specifically, in 1D Cartesian sampling, few v ($\ll N_f$) lines, which are marked by the yellow-colored box in Fig. 1a and cross the central region of each frame, are vectorized to form the $v \times 1$ vector $\tilde{\mathbf{y}}_t^f$, where $v := v N_p$, with superscript “f” stressing the fact that the vector includes faithful data and symbol $(\cdot)$ 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 $\tilde{\mathbf{Y}}_f := [\tilde{\mathbf{y}}_1^f, \dots, \tilde{\mathbf{y}}_{N_{fr}}^f] \in \mathbb{C}^{v \times N_{fr}}$. Similarly, in radial sampling, a few v radial spokes can be carefully selected by the user to form the faithful data. The availability of faithful data can be also met in applications other than MRI; in single-cell genomics, for example, they may take the form of gene-expression data extracted from the bulk analysis of similar cell types [5].

III. THE KRIM FRAMEWORK

A. Selecting Landmark/Representative Points

To promote parsimonious descriptions of the point-cloud $\{\tilde{\mathbf{y}}_t^f\}_{t=1}^{N_{fr}}$ as in [40], a subset $\{\mathbf{l}_k\}_{k=1}^{N_l}$, coined *landmark/representative points*, is selected from $\{\tilde{\mathbf{y}}_t^f\}_{t=1}^{N_{fr}}$, with N_l being smaller than N_{fr} . Randomly selecting the landmark points from $\{\tilde{\mathbf{y}}_t^f\}_{t=1}^{N_{fr}}$ 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 [43] is adopted here. In short, maxmin is a recursive scheme, taking a number of N_l steps to terminate, where at the n th step of the algorithm, a landmark point is selected from $\{\tilde{\mathbf{y}}_t^f\}_{t=1}^{N_{fr}}$ that maximizes, over all *un-selected* $\{\tilde{\mathbf{y}}_t^f\}_{t=1}^{N_{fr}}$, the minimum (Euclidean) distance to the landmark points which have been already selected up to the $(n-1)$ th step of the algorithm. In general, the larger N_{fr} is, the larger N_l should be to describe the point-cloud $\{\tilde{\mathbf{y}}_t^f\}_{t=1}^{N_{fr}}$, justifying thus the non-parametric label of the proposed framework. Maxmin algorithm scores a computational complexity of order $\mathcal{O}(N_l N_{fr})$, which is naturally heavier than the case of selecting $\{\mathbf{l}_k\}_{k=1}^{N_l}$ randomly from $\{\tilde{\mathbf{y}}_t^f\}_{t=1}^{N_{fr}}$. After $\{\mathbf{l}_k\}_{k=1}^{N_l}$ are identified,

the complex-valued $\nu \times N_l$ matrix $\mathbf{L} := [\mathbf{l}_1, \mathbf{l}_2, \dots, \mathbf{l}_{N_l}]$ is defined.

B. Data Approximation in RKHS via Landmark Points

The subsequent discussion assumes a user-defined complex RKHS $\mathcal{H}$ equipped with a reproducing kernel $\kappa(\cdot, \cdot)$ and a feature mapping $\varphi(\cdot)$; cf. Appendix A. According to (17), let also $\Phi(\mathbf{L}) := [\varphi(\mathbf{l}_1), \dots, \varphi(\mathbf{l}_{N_l})]$.

Modeling Assumption 1. 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 follows:

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

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 $\tilde{\mu}_t$ is a complex-valued $\nu \times 1$ vector s.t. $\varphi(\tilde{\mu}_t) \in \text{span}\{\varphi(\mathbf{l}_k)\}_{k=1}^{N_l}$.

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 $\varphi(\tilde{\mu}_t) \in \text{span}\{\varphi(\mathbf{l}_k)\}_{k=1}^{N_l}$ in the feature space $\mathcal{H}$ 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, since employing the richer $\text{span}\{\varphi(\tilde{\mathbf{y}}_t^f)\}_{t=1}^{N_{\text{fr}}}$ would inflict a heavy load on computations, 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 [31] which guarantees that the solution $f_{it}(\cdot)$ to the following variational problem:

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

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

By Modeling Assumption 1, there exists a vector $\mathbf{a}_i := [a_{i1}, \dots, a_{iN_l}]^T \in \mathbb{C}^{N_l}$ ($\top$ 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 Appendix A suggest that $y_{it} \approx f_i(\tilde{\mu}_t) = \langle f_i | \varphi(\tilde{\mu}_t) \rangle_{\mathcal{H}} = \langle \Phi(\mathbf{L})\mathbf{a}_i | \varphi(\tilde{\mu}_t) \rangle_{\mathcal{H}} = \mathbf{a}_i^H \Phi(\mathbf{L})^H \varphi(\tilde{\mu}_t)$, where 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 expressed also as follows: $\forall t$,

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

The following modeling assumption postulates more structure within the data, moving beyond the typical linear-subspace constraints such as Modeling Assumption 1.

Modeling Assumption 2. Points $\{\varphi(\mathbf{l}_k)\}_{k=1}^{N_l}$, as well as $\{\varphi(\tilde{\mu}_t)\}_{t=1}^{N_{\text{fr}}}$, lie onto or close to a smooth manifold $\mathcal{M}$ embedded in the RKHS $\mathcal{H}$ (cf. Fig. 2).

Since landmark points $\{\mathbf{l}_k\}_{k=1}^{N_l}$ have been identified to capture the geometry of the point-cloud $\{\tilde{\mathbf{y}}_t^f\}_{t=1}^{N_{\text{fr}}}$, 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 $\varphi(\tilde{\mu}_t)$. Here, “closely related” means proximity

Fig. 2. Points $\{\varphi(\mathbf{l}_{k_j})\}_{j=1}^3$ lie onto or close to the unknown-to-the-user manifold $\mathcal{M} \subset \mathcal{H}$ and are affinely combined to approximate $\varphi(\tilde{\mu}_t)$. All affine combinations of $\{\varphi(\mathbf{l}_{k_j})\}_{j=1}^3$ are represented by the gray-coloured plane which mimics the concept of a tangent space to $\mathcal{M}$.

within $\mathcal{M}$, which may translate into similarities in data-structures and attributes in the original input space (k-space in dMRI). Putting together the modeling assumption 2 and the concept of tangent spaces to a smooth manifold [35], $\varphi(\tilde{\mu}_t)$ may be *affinely* approximated by neighboring landmark points. For example, $\varphi(\tilde{\mu}_t)$ is approximated in Fig. 2 by the gray-colored plane which gathers all affine combinations of $\{\varphi(\mathbf{l}_{k_1}), \varphi(\mathbf{l}_{k_2}), \varphi(\mathbf{l}_{k_3})\}$. This notion can be formalized by the existence of an $N_l \times 1$ vector $\mathbf{b}_t$ s.t. $\varphi(\tilde{\mu}_t) \approx \Phi(\mathbf{L})\mathbf{b}_t$, 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 $\{\varphi(\mathbf{l}_{k_1}), \varphi(\mathbf{l}_{k_2}), \varphi(\mathbf{l}_{k_3})\}$ to describe $\varphi(\tilde{\mu}_t)$ in Fig. 2—and the robustness that such descriptions entail, $\mathbf{b}_t$ is considered to be sparse. Gathering all vectors $\{\mathbf{b}_t\}_{t=1}^{N_{\text{fr}}}$ in the sparse matrix $\mathbf{B} := [\mathbf{b}_1, \dots, \mathbf{b}_{N_{\text{fr}}}]$, and by defining the kernel matrix $\mathbf{K}_L := \Phi(\mathbf{L})^H \Phi(\mathbf{L})$, the previous discussion and (2) yield the following data-approximation model:

$$\mathbf{Y} = \mathbf{A} \mathbf{K}_L \mathbf{B} + \mathbf{E}_1, \quad (3)$$

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

C. Multiple Kernels

Given the rich variety of reproducing kernels in the literature, e.g., Appendix A and [34], it becomes a tough task to select a specific kernel and fine-tune its parameters through cross-validation to fit the available data. A popular strategy to address this issue is to employ a dictionary of user-defined reproducing kernel functions $\{\kappa_m(\cdot, \cdot)\}_{m=1}^M$, 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 each $\omega_m \in \mathbb{R}_{>0}$ and $\mathbf{K}_L^{(m)}$ is the kernel matrix constructed according to the preceding discussion via $\kappa_m(\cdot, \cdot)$; cf. [39]. 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 from the data themselves through some learning algorithm. Rather than simply plugging the previous weighted kernel matrix into (3), an approach with more degrees of freedom in the employed parameters is followed here via the following assumption.

Modeling Assumption 3. There exist complex-valued $N_k \times N_l$ matrices $\{\mathbf{A}_m\}_{m=1}^M$ and $N_l \times N_{\text{fr}}$ 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.

$$\mathbf{Y} = \sum_{m=1}^M \mathbf{A}_m \mathbf{K}_L^{(m)} \mathbf{B}_m + \mathbf{E}_2, \quad (4)$$

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

It is worth noticing here that the proposed modeling framework employs kernel functions for data approximations without any need for explicit pre-imaging from the feature spaces $\{\mathcal{H}_m\}_{m=1}^M$ back to the input $\mathbb{C}^\nu$ one, as in [27].

D. Dimensionality Reduction

Owing to the non-parametric character of the proposed scheme, 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 (4) 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. Sec. III-A. To meet the limitations of computational and time resources, and to impose a low-rank structure on (4), which appears to have a beneficial effect on dMRI [12, 13], a dimensionality-reduction module is introduced here to map each $\mathbf{K}_L^{(m)}$ to a $d \times N_l$ complex-valued matrix $\tilde{\mathbf{K}}_L^{(m)}$, with $d \ll N_l$. To this end, the two-step dimensionality-reduction scheme of [36] is extended into the present RKHS case: (i) 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 $\mathbb{C}^\nu$ to the user-defined RKHS $\mathcal{H}_m$; and (ii) the low-dimensional $\tilde{\mathbf{K}}_L^{(m)}$, which preserves the identified affine dependencies $\mathbf{W}_m$, is computed.

More specifically, along the lines of the discussion that follows modeling assumption 2, 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 (17) 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 $\mathbf{W}_m$. Further, to surmount computations on the possibly infinite dimensional $\Phi_m(\mathbf{L})$, the kernel trick turns out to be handy also here: Multiplying the previous relation from the left by $\Phi_m(\mathbf{L})^H$ yields $\Phi_m(\mathbf{L})^H \Phi_m(\mathbf{L}) \approx \Phi_m(\mathbf{L})^H \Phi_m(\mathbf{L})\mathbf{W}_m \Leftrightarrow \mathbf{K}_L^{(m)} \approx \mathbf{K}_L^{(m)} \mathbf{W}_m$. In a more formal way, for a user-defined $\lambda_W \in \mathbb{R}_{>0}$, $\mathbf{W}_m$ is identified by

$$\begin{aligned} \mathbf{W}_m := \arg \min_{\mathbf{W} \in \mathbb{C}^{N_l \times N_l}} & \|\mathbf{K}_L^{(m)} - \mathbf{K}_L^{(m)} \mathbf{W}\|_F^2 + \lambda_W \|\mathbf{W}\|_1 \\ \text{s.t. } & \mathbf{1}_{N_l}^H \mathbf{W} = \mathbf{1}_{N_l}^H, \quad \text{diag}(\mathbf{W}) = \mathbf{0}, \end{aligned} \quad (5)$$

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 $\|\mathbf{W}\|_1 := \sum_{ij} [\Re(w_{ij})^2 + \Im(w_{ij})^2]^{1/2}$ [44, Lem. V.1], with $\Re(\cdot)$ and $\Im(\cdot)$ denoting the real and imaginary parts of a complex number. Task (5) is an affinely constrained and composite convex minimization task, and, hence, [45] is used to solve it by virtue of the fast convergence and flexibility of [45] in dealing with affine constraints.

After $\mathbf{W}_m$ is identified by (5), $\tilde{\mathbf{K}}_L^{(m)}$ is computed by

$$\tilde{\mathbf{K}}_L^{(m)} := \arg \min_{\tilde{\mathbf{K}} \in \mathbb{C}^{d \times N_l}} \|\tilde{\mathbf{K}} - \tilde{\mathbf{K}} \mathbf{W}_m\|_F^2 \quad \text{s.t. } \tilde{\mathbf{K}} \tilde{\mathbf{K}}^H = \mathbf{I}_d, \quad (6)$$

where the constraint is employed for $\tilde{\mathbf{K}}_L^{(m)}$ to avoid the trivial solution $\mathbf{0}_{d \times N_l}$. It turns out that the solution to (6) 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 $\tilde{\mathbf{K}}_L^{(m)}$ back to $\mathbf{K}_L^{(m)}$; e.g., PCA-driven frameworks. This need calls for the following assumption. The subsequent matrix $\mathbf{G}_m$ acts as a “decompression” operator which maps the $d \times N_l$ matrix $\tilde{\mathbf{K}}_L^{(m)}$ back to the $N_l \times N_l$ matrix $\mathbf{K}_L^{(m)}$.

Modeling Assumption 4. There exist $N_l \times d$ complex-valued matrices $\{\mathbf{G}_m\}_{m=1}^M$ s.t. $\forall m \in \{1, \dots, M\}$,

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

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

Via Modeling Assumptions 3 and 4, as well as $\mathbf{U}_m := \mathbf{A}_m \mathbf{G}_m$, $\forall m \in \{1, \dots, M\}$, it can be verified that

$$\mathbf{Y} = \sum_{m=1}^M \mathbf{U}_m \tilde{\mathbf{K}}_L^{(m)} \mathbf{B}_m + \mathbf{E}, \quad (7)$$

where the complex-valued $N_k \times N_{fr}$ matrix $\mathbf{E}$ accumulates all approximation and mis-modeling errors. The KRIM data-approximation model in (7) is abbreviated to KRIM[M] whenever $M > 1$ (multi-kernel case), while tag KRIM[S] is used in the single-kernel case $M = 1$. It is worth noticing here that KRIM[S], introduced in [41], requires effort to fine-tune the parameters of the single kernel function to “optimal” values via cross validation.

E. The Inverse Problem

To simplify (7), define the following complex-valued matrices: the $N_k \times dM$ matrix $\tilde{\mathbf{U}} := [\mathbf{U}_1, \mathbf{U}_2, \dots, \mathbf{U}_M]$, the $MN_l \times N_{fr}$ matrix $\tilde{\mathbf{B}} = [\mathbf{B}_1^H, \mathbf{B}_2^H, \dots, \mathbf{B}_M^H]^H$, and the $dM \times MN_l$ matrix

$$\tilde{\mathbf{K}}_L := \begin{bmatrix} \tilde{\mathbf{K}}_L^{(1)} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{K}}_L^{(2)} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \tilde{\mathbf{K}}_L^{(M)} \end{bmatrix}. \quad (8)$$

Then, (7) takes the bi-linear form of

$$\mathbf{Y} = \tilde{\mathbf{U}} \tilde{\mathbf{K}}_L \tilde{\mathbf{B}} + \mathbf{E}, \quad (9)$$

where “bi-linearity” refers to the behavior that if $\tilde{\mathbf{U}}$ (or $\tilde{\mathbf{B}}$) is fixed to a specific value, then (9) expresses $\mathbf{Y}$ as a linear (better affine) model of the variable $\tilde{\mathbf{B}}$ (or $\tilde{\mathbf{U}}$).

Having established (9), the following inverse problem is proposed to identify $(\tilde{\mathbf{U}}, \tilde{\mathbf{B}})$ for some user-defined parameters $\lambda_1, C_U \in \mathbb{R}_{>0}$:

$$\begin{aligned} \min_{(\tilde{\mathbf{U}}, \tilde{\mathbf{B}})} & \overbrace{\frac{1}{2} \|\mathcal{S}(\mathbf{Y}) - \mathcal{S}(\tilde{\mathbf{U}} \tilde{\mathbf{K}}_L \tilde{\mathbf{B}})\|_F^2}^{\text{T1}} + \overbrace{\lambda_1 \|\tilde{\mathbf{B}}\|_1}^{\text{T2}} \\ \text{s.t. } & \|\tilde{\mathbf{U}} \mathbf{e}_i\| \leq C_U, \forall i \in \{1, \dots, dM\} \end{aligned} \quad (10a)$$

$$\mathbf{1}_{N_l}^H \mathbf{B}_m = \mathbf{1}_{N_{fr}}^H, \forall m \in \{1, \dots, M\} \quad (10b)$$

$$\tilde{\mathbf{U}} \in \mathbb{C}^{N_k \times dM}, \tilde{\mathbf{B}} \in \mathbb{C}^{MN_l \times N_{fr}},$$

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: **(i)** the data-fit term T1; **(ii)** the sparsity-imposing term T2 along the lines of Modeling Assumption 3; **(iii)** constraint (10a) to prevent unbounded solutions in $\tilde{\mathbf{U}}$ due to the bilinear structure of the model; and **(iv)** constraint (10b) to introduce affine constraints according to Modeling Assumption 3. A solution $(\hat{\tilde{\mathbf{U}}}, \hat{\tilde{\mathbf{B}}})$ to (10) smoothes/denoises the measured values and imputes the missing ones in $\mathbf{Y} \approx \hat{\mathbf{Y}} := \hat{\tilde{\mathbf{U}}} \tilde{\mathbf{K}}_{\mathbf{L}} \hat{\tilde{\mathbf{B}}}$. An iterative algorithm that computes a solution $(\hat{\tilde{\mathbf{U}}}, \hat{\tilde{\mathbf{B}}})$ in the context of dMRI is provided in the following subsection.

F. The dMRI Case Study

With $\mathcal{F}(\cdot)$ defined in Sec. II, let the $N_k \times dM$ complex-valued matrix $\tilde{\mathbf{D}} := \mathcal{F}^{-1}(\tilde{\mathbf{U}})$. By virtue of the linearity of $\mathcal{F}(\cdot)$, the general KRIM model in (9) takes the following form in the present dMRI context:

$$\mathbf{Y} = \mathcal{F}(\tilde{\mathbf{D}}) \tilde{\mathbf{K}}_{\mathbf{L}} \tilde{\mathbf{B}} + \mathbf{E} = \mathcal{F}(\tilde{\mathbf{D}} \tilde{\mathbf{K}}_{\mathbf{L}} \tilde{\mathbf{B}}) + \mathbf{E},$$

which leads in turn to

$$\mathbf{X} = \mathcal{F}^{-1}(\mathbf{Y}) = \tilde{\mathbf{D}} \tilde{\mathbf{K}}_{\mathbf{L}} \tilde{\mathbf{B}} + \mathcal{F}^{-1}(\mathbf{E}). \quad (11)$$

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 $1 \times N_{\text{fr}}$ row vector $\mathbf{x}_i^{\text{T}}$ denoting the time profile of the i th pixel in $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_{N_k}]^{\text{T}}$, $i \in \{1, \dots, N_k\}$, 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 $\mathbf{x}_i^{\text{T}}$ of the i th pixel 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, $\mathcal{F}_t(\mathbf{X}) := [\mathcal{F}_t(\mathbf{x}_1), \dots, \mathcal{F}_t(\mathbf{x}_{N_k})]^{\text{T}}$ can be approximated by a sparse matrix $\mathbf{Z}$. Infusing (11) and the previous arguments into the general formulation (10), the following dMRI inverse problem is obtained: for user-defined parameters $\lambda_1, \lambda_2, \lambda_3, C_D \in \mathbb{R}_{>0}$, solve

$$\min_{(\tilde{\mathbf{D}}, \tilde{\mathbf{B}}, \mathbf{X}, \mathbf{Z})} \underbrace{\frac{1}{2} \|\mathbf{X} - \tilde{\mathbf{D}} \tilde{\mathbf{K}}_{\mathbf{L}} \tilde{\mathbf{B}}\|_{\text{F}}^2 + \lambda_1 \|\tilde{\mathbf{B}}\|_1 + \frac{\lambda_2}{2} \|\mathbf{Z} - \mathcal{F}_t(\mathbf{X})\|_{\text{F}}^2 + \lambda_3 \|\mathbf{Z}\|_1}_{\text{T3}} \quad (12a)$$

$$\begin{aligned} \text{s.to } & \|\tilde{\mathbf{D}} \mathbf{e}_i\| \leq C_D, \forall i \in \{1, \dots, dM\} \\ & \mathbf{1}_{N_t}^{\text{H}} \mathbf{B}_m = \mathbf{1}_{N_{\text{fr}}}^{\text{H}}, \forall m \in \{1, \dots, M\} \\ & \mathcal{S}(\mathbf{Y}) = \mathcal{S} \mathcal{F}(\mathbf{X}) \\ & \tilde{\mathbf{D}} \in \mathbb{C}^{N_k \times dM}, \tilde{\mathbf{B}} \in \mathbb{C}^{M N_t \times N_{\text{fr}}} \\ & \mathbf{X}, \mathbf{Z} \in \mathbb{C}^{N_k \times N_{\text{fr}}}, \end{aligned} \quad (12b)$$

where term T3 in the loss function (12a) is employed to promote the potential sparsity of $\mathcal{F}_t(\mathbf{X})$, and (12b) is introduced for $\mathcal{F}(\mathbf{X})$ to be consistent with the measured data in $\mathbf{Y}$.

Other than the introduction of kernel functions, (12) is also structurally different than the inverse problem in [40], since the computationally intensive 2D DFT $\mathcal{F}(\cdot)$ is not employed at all at the loss function (12a), unlike [40], but it is only applied to

the constraint (12b). Such a strategy yields largely reduced computational times with regards to [40] as the extensive numerical tests demonstrate in Sec. IV.

G. Solving the dMRI inverse problem

This section discusses the details of Alg. 1 which solves (12). Even though Alg. 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 [46] is applied to solve (12) via steps 4–10 in Alg. 1. Convergence to a stationary solution of (12) is guaranteed [46]. Step 7 in Alg. 1 focuses on solving convex minimization sub-tasks to update estimates $\hat{\tilde{\mathbf{D}}}_n, \hat{\tilde{\mathbf{B}}}_n, \hat{\mathbf{X}}_n, \hat{\mathbf{Z}}_n$. With user-defined parameters $\tau_D, \tau_B \in \mathbb{R}_{>0}$,

$$\begin{aligned} \hat{\tilde{\mathbf{D}}}_{n+1/2} \in \arg \min_{\tilde{\mathbf{D}}} & \frac{1}{2} \|\hat{\mathbf{X}}_n - \tilde{\mathbf{D}} \tilde{\mathbf{K}}_{\mathbf{L}} \hat{\tilde{\mathbf{B}}}_n\|_{\text{F}}^2 + \frac{\tau_D}{2} \|\tilde{\mathbf{D}} - \hat{\tilde{\mathbf{D}}}_n\|_{\text{F}}^2 \\ \text{s.to } & \|\tilde{\mathbf{D}} \mathbf{e}_i\| \leq C_D, \forall i \in \{1, \dots, dM\}, \end{aligned} \quad (13a)$$

$$\begin{aligned} \hat{\tilde{\mathbf{B}}}_{n+1/2} \in \arg \min_{\tilde{\mathbf{B}}} & \frac{1}{2} \|\hat{\mathbf{X}}_n - \hat{\tilde{\mathbf{D}}}_n \tilde{\mathbf{K}}_{\mathbf{L}} \tilde{\mathbf{B}}\|_{\text{F}}^2 + \lambda_1 \|\tilde{\mathbf{B}}\|_1 \\ & + \frac{\tau_B}{2} \|\tilde{\mathbf{B}} - \hat{\tilde{\mathbf{B}}}_n\|_{\text{F}}^2 \\ \text{s.to } & \mathbf{1}_{N_t}^{\text{H}} \mathbf{B}_m = \mathbf{1}_{N_{\text{fr}}}^{\text{H}}, \forall m \in \{1, \dots, M\}, \end{aligned} \quad (13b)$$

$$\begin{aligned} \hat{\mathbf{X}}_{n+1/2} \in \arg \min_{\mathbf{X}} & \frac{1}{2} \|\mathbf{X} - \hat{\tilde{\mathbf{D}}}_n \tilde{\mathbf{K}}_{\mathbf{L}} \hat{\tilde{\mathbf{B}}}_n\|_{\text{F}}^2 + \frac{\lambda_2}{2} \|\hat{\mathbf{Z}}_n - \mathcal{F}_t(\mathbf{X})\|_{\text{F}}^2 \\ \text{s.to } & \mathcal{S}(\mathbf{Y}) = \mathcal{S} \mathcal{F}(\mathbf{X}), \end{aligned} \quad (13c)$$

$$\begin{aligned} \hat{\mathbf{Z}}_{n+1/2} \in \arg \min_{\mathbf{Z}} & \frac{1}{2} \|\mathbf{Z} - \mathcal{F}_t(\hat{\mathbf{X}}_n)\|_{\text{F}}^2 + \frac{\lambda_3}{\lambda_2} \|\mathbf{Z}\|_1 \\ = & \text{Soft}[\mathcal{F}_t(\hat{\mathbf{X}}_n), \lambda_3/\lambda_2]. \end{aligned} \quad (13d)$$

Tasks (13a) and (13b) are solved by [45] as described in Alg. 2. Notice that for Alg. 2 to operate, only first-order information (gradients) and proximal mappings, like soft-thresholding and projections, are required; *cf.* Appendix B. Moreover, the unique solution $\hat{\mathbf{X}}_{n+1/2}$ to (13c) can be obtained as follows:

$$\hat{\mathbf{X}}_{n+1/4} := \frac{1}{1 + \lambda_2 N_{\text{fr}}} \left(\hat{\tilde{\mathbf{D}}}_n \tilde{\mathbf{K}}_{\mathbf{L}} \hat{\tilde{\mathbf{B}}}_n + \lambda_2 N_{\text{fr}} \mathcal{F}^{-1}(\hat{\mathbf{Z}}_n) \right), \quad (14a)$$

$$\hat{\mathbf{X}}_{n+1/2} := \mathcal{F}^{-1} \mathcal{S}(\mathbf{Y}) + \mathcal{F}^{-1} \mathcal{S}^c \mathcal{F}(\hat{\mathbf{X}}_{n+1/4}), \quad (14b)$$

where $\mathcal{S}^c(\cdot)$ operates on a matrix in a complementary way to the sampling operator $\mathcal{S}(\cdot)$, *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 (14) is provided in Appendix B-C. The updates of $(\hat{\tilde{\mathbf{D}}}_n, \hat{\tilde{\mathbf{B}}}_n, \hat{\mathbf{X}}_n, \hat{\mathbf{Z}}_n)$ can be carried out independently of each other and computed in parallel, offering room for efficient implementations with short computation times. Furthermore, tasks (13) can be also solved inexactly at every iteration, with increasing precision as described in [46], offering space for further reduction in computation times.

IV. NUMERICAL TESTS

A. Datasets and Competing Algorithms

The efficacy of the proposed framework (KRIM[M] and KRIM[S]) is validated on three datasets, widely used in [12,

Algorithm 1 KRIM Reconstruction for dMRI data

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 also M kernels and their parameters to form the kernel dictionary.

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 image by $\hat{\mathbf{D}}_* \tilde{\mathbf{K}}_{\mathbf{L}} \hat{\mathbf{B}}_*$.

- 1: Identify landmark points $\mathbf{L}$ from $\check{\mathbf{Y}}_f$ (cf. Sec. III-A).
 - 2: Compute $\tilde{\mathbf{K}}_{\mathbf{L}}$ via (8).
 - 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 (13a) and (13b), respectively (cf. Alg. 2), $\hat{\mathbf{X}}_{n+1/2}$ by (13c) and (14), and $\hat{\mathbf{Z}}_{n+1/2}$ by (13d).
 - 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**
-

25–27, 40]: (i) Magnetic resonance extended cardiac-torso (MRXCAT) cine phantom (Sec. IV-E) [47]; (ii) in-vivo cardiac phantom generated from real MR-scanned image (Sec. IV-F) [12]; and (iii) free-breathing and real-time measurements of an MR-scanned cardiac region (Sec. IV-G). KRIM[M] and KRIM[S] are compared with the state-of-the-art FRIST [11], PS-Sparse [12], STORM [24], KLR [27], and BiLMDM [40].

B. Performance Metrics

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

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

$$\text{NRMSE} := \frac{\|\mathbf{X} - \hat{\mathbf{X}}\|_F}{\|\mathbf{X}\|_F}, \quad (15)$$

where $\mathbf{X}$ denotes the high-fidelity original image data, generated from a fully measured (k,t)-space (cf. Sec. II), and $\hat{\mathbf{X}}$ represents an estimate of $\mathbf{X}$ computed via the competing reconstruction schemes using the undersampled (k,t)-space. NRMSE highlights the pixel-by-pixel accuracy of a reconstruction scheme. Low NRMSE values are desired.

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

$$\text{HFEN} := \frac{\|\text{LoG}(\mathbf{X}) - \text{LoG}(\hat{\mathbf{X}})\|_2}{\|\text{LoG}(\mathbf{X})\|_2}, \quad (16)$$

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. Deviation from the sample mean was recorded to be within $[0.0001, 0.00045]$ across all acceleration rates. 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.

where LoG is a rotationally symmetric Laplacian of Gaussian filter. HFEN explores the effectiveness of a reconstruction algorithm along the edges of an image. As in [48], a kernel filter of size 15×15 with standard deviation of 1.5 pixels is employed. A low HFEN score is an indicator of how close the reconstructed image is to the ground-truth one in terms of sharpness.

- (iii) The sharpness measures M1 (intensity-variance based) and M2 (energy of the image gradient) [49, (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.
- (iv) The structural similarity index measure (SSIM) [50] investigates local patterns in pixel intensities after normalizing for luminance and contrast, with values within $[0, 1]$. SSIM values close to 1 are desired since they indicate the

similarity between the reconstructed data and the ground-truth image.

Among the previous metrics, NRMSE, HFEN, and SSIM require the knowledge of the ground-truth image, unlike the M1 and M2 measures.

C. Experimental Setup

KRIM[M] and KRIM[S] (Alg. 1) were implemented in Julia (ver. 1.0.8) [51], while BiLMDM [40, 41] was implemented in MATLAB [52]. MATLAB implementations for STORM [53] and FRIST [54] are publicly available, whereas implementations of PS-Sparse and KLR were made available by the Authors of [12] and [27], respectively. Experiments were carried out on a 12-core Intel(R) 2.40GHz Linux based-system with 48GB RAM. The hyperparameters for the frameworks, with which the proposed schemes are compared, were tuned to produce reconstructions with the lowest NRMSE score. KRIM's Julia code will be made available online via GitHub as soon as the manuscript is accepted for publication.

Among the three datasets of Sec. 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. 1c) and the radial (Fig. 1d) under-sampling schemes were employed. Given the stringent space limitations of this journal, only the results on the 1D Cartesian scheme are demonstrated for the first dataset, while only the results on radial under-sampling for the second one. Nevertheless, similar behavior of results was also verified for the other under-sampling scenarios. The third dataset of Sec. IV-A is prospectively under-sampled by default, since it comprises readings from an MR scanner.

D. Parameter Tuning

Tasks (12) and (13) 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 2 to reach “desired performance.” To offer a concrete discussion about “desired performance,” parameters are carefully tuned for the first two datasets of Sec. IV-A where ground-truth dMRI data are available. The following discussion identifies “optimal ranges” of parameter values based on the behavior of the NRMSE metric as a function of the parameters: “optimal” parameter values are defined as those values that yield low NRMSE while, at the same time, that achieved low NMRSE value shows small fluctuations as parameters change.

Trial and error revealed the following: parameter λ_1 penalizes the sparsity of $\tilde{\mathbf{B}}$ to control the size of the neighborhood that participates in the approximation of a time-series 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 data 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, it is observed that $N_l \in N_{fr} \cdot [15\%, 25\%]$ is adequate. An increase of N_l beyond those values didn't show any significant improvement in performance.

With regards to the variable $\tilde{\mathbf{D}}$, the bounding constraint $C_D = 1$ usually leads to “optimal” estimates for $\tilde{\mathbf{D}}$ in the case where the k-space is normalized in a way that the pixel intensity values lies 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) exploiting the 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 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 Alg. 1 and $\alpha = 0.5$ for Alg. 2 ensure desirable results. Similarly, τ_D, τ_B values within the range $[10^{-7}, 10^{-5}]$ of orders have shown desirable results across different datasets.

A polynomial kernel (cf. Appendix A), with $c_* := 3$ and $r_* := 0.5 \times (\text{mean of all entries in } \mathbf{L})$, 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 data-recovery 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 (7). Indeed, such a performance boost was observed throughout all of the numerical tests. Nevertheless, to provide a complete set of numerical 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, $\text{KRIM[M]}_{\setminus(c_*, r_*)}$ 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 $\text{KRIM[M]}_{\setminus(c_*, r_*)}$ in Tables I and III, 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 $\text{KRIM[M]}_{\setminus(c_*, r_*)}$ 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 8x. Comparative results between the proposed KRIM[M], KRIM[S], BiLMDM, KLR, FRIST, STORM, PS-Sparse (left to right) are shown in (b). 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. Reconstruction results along the time axis are also provided in Fig. 5.

E. MRXCAT Cardiac Phantom

The MRXCAT phantom is a synthetically generated cardiac cine dMRI dataset using the extended cardiac torso (XCAT) framework [47]. The MRXCAT phantom emulates a breath-hold cardiac cine data of spatial size $(N_p, N_f) = (408, 408)$ with $N_{fr} = 360$ frames. The data have spatial resolution $1.56 \times 1.56 \text{ mm}^2$ for a 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 motion along the temporal direction. The KRIM parameters $d := 4$ and $N_l := 100$ are considered.

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

	NRMSE	SSIM	HFEN	M1	M2
KRIM[M]	0.0372	0.9052	0.1058	37.95	1.5×10^6
KRIM[S]	0.0407	0.8869	0.1198	37.65	1.4×10^6
KRIM[M]_(c,r,s)	0.0378	0.9052	0.1062	38.54	1.5×10^6
BiLMDM	0.0438	0.8697	0.1314	28.69	1.1×10^6
KLR	0.0744	0.6704	0.1580	34.98	1.4×10^6
FRIST	0.3055	0.4572	0.7006	13.50	7.7×10^5
SToRM	0.0644	0.7743	0.2478	31.06	1.5×10^6
PS-Sparse	0.0449	0.8864	0.1338	30.66	1.2×10^6

It can be seen from Fig. 3a that KRIM outperforms consistently the competing state-of-the-art schemes across different acceleration rates. Table I details all performance metrics for the acceleration rate of 8x. Unlike KLR, STORM and FRIST, KRIM reconstructions do not deteriorate significantly in NRMSE values as acceleration rates become larger.

Spatial and temporal reconstructions, generated from (k,t)-space measurements available at the acceleration rate of 8x, are exhibited for comparison in Figs. 4 and 5, 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) and alias free (unlike KLR and STORM). These deformities are highlighted in Fig. 4. It is also worth noticing that 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. Average computation times are reported in Table II. Notice the large savings in computation time achieved by KRIM with regards to its predecessor BiLMDM [40].

TABLE II
AVERAGE COMPUTATION TIMES (25 RUNS) FOR THE MRXCAT PHANTOM

	Computation time
KRIM[M]	23min
KRIM[S]	14min
BiLMDM	5.5hrs
KLR	6.5hrs
FRIST	3.5hrs
SToRM	43min
PS-Sparse	10min

F. In-Vivo Human Cardiac Phantom

Real, human, cardiac cine MR data [12], with size $(N_p, N_f) = (200, 256)$, encompassing a FOV = $273 \times 300 \text{ mm}^2$

and with a 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 [55] (to model respiration) were then used to generate a temporal sequence with $N_{\text{fr}} = 256$ frames. For KRIM, parameters d and N_l are tuned to 16 and 80, respectively.

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

	NRMSE	SSIM	HFEN	M1	M2
KRIM[M]	0.0445	0.9586	0.0537	15.15	4.5×10^5
KRIM[S]	0.0455	0.9569	0.0633	14.77	4.4×10^5
KRIM[M]_{\setminus(c_s, r_s)}	0.0439	0.961	0.0543	15.04	4.5×10^5
BiLMDM	0.0507	0.9494	0.0903	14.87	4.4×10^5
KLR	0.2296	0.7059	0.3364	4.94	2.7×10^5
FRIST	0.1903	0.5048	0.5045	9.59	2.8×10^5
SToRM	0.0559	0.9394	0.1105	14.46	4.5×10^5
PS-Sparse	0.0517	0.9487	0.1061	15.13	4.5×10^5

Fig. 3b shows that KRIM produces images with the least NRMSE among all competing methods for all acceleration rates within the range 6x–35x. Table III summarizes all performance metrics for the acceleration rate of 12x. Similarly to the case of the MRXCAT phantom, the highest values for the M1 and M2 metrics are recorded for KRIM[M] (also KRIM[M]_{\setminus(c_s, r_s)}). The lowest HFEN score was exhibited by KRIM[M], indicating efficacy in reconstruction along the edges. The SSIM values for all KRIM versions are the highest ones approaching 1.

Quantitative results are supported by Figs. 6 and 7. 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 and STORM (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. To conclude, KRIM appears to score the best performance in dMRI reconstruction among all competing methods. Average computation times are reported in Table IV.

TABLE IV
AVERAGE COMPUTATION TIMES (25 RUNS) FOR IN-VIVO CARDIAC PHANTOM

	Computation time
KRIM[M]	13min
KRIM[S]	7min
BiLMDM	4hrs
KLR	5hrs
FRIST	2hrs
SToRM	32min
PS-Sparse	4min

G. Acquired dMRI Cardiac Scan

Measurements were acquired via a FLASH sequence [56] from a volunteer breathing normally under the following acquisition parameters: TR/TE = 5.8/4ms, 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-encoded lines under 1D Cartesian sampling. The phase-encoded lines are further grouped into groups of 15 (containing 5 navigator lines) to obtain $N_{\text{fr}} = 300$ frames and a data matrix of $156 \times 192 \times 300$ per channel. This corresponds to an acceleration/sampling rate of 10x. The reconstructed multi-channel data were combined via the sum-of-squares strategy for all reconstruction frameworks.

Due to the unavailability of the ground-truth images, comparisons are qualitative in nature, as exhibited in Fig. 8. Reconstructions via FRIST show severe distortions (highlighted by box) and suffer from aliasing (evident in the temporal cross section), while those via STORM show signs of blurring along the edges (in the area marked by the box). For the PS-Sparse reconstructions, some image structures seemed to be missing (marked by arrows) which are clearly present in images reconstructed by other schemes. Moreover, significant motion blurring occurs as exhibited in the temporal cross-section. On the other hand, images generated via KRIM look sharper with little or no signs of distortion and aliasing, unlike KRIM's competitors. Even though no significant difference could be observed when compared to BiLMDM, the M1 and M2 measures (Table V) quantify the improvement in sharpness using KRIM over BiLMDM. The average computation times for the reconstruction of the multi-coil image series are provided in Table VI.

TABLE V
QUANTITATIVE PERFORMANCE ANALYSIS FOR THE ACQUIRED dMRI CARDIAC SCAN (ACCELERATION RATE: 10x)

	M1	M2
KRIM[M]	7.15	1.5×10^5
KRIM[S]	6.77	1.4×10^5
BiLMDM	5.87	1.4×10^5
KLR	1.98	8×10^4
FRIST	1.59	7×10^4
SToRM	4.46	1.4×10^5
PS-Sparse	5.33	1.4×10^5

TABLE VI
AVERAGE COMPUTATION TIMES (25 RUNS) FOR THE ACQUIRED dMRI CARDIAC SCAN

	Computation time
KRIM[M]	27min
KRIM[S]	25min
BiLMDM	5.25hrs
KLR	14hrs
FRIST	17hrs
SToRM	35min
PS-Sparse	16min

V. DISCUSSION

This section accentuates the differences of KRIM with the state-of-the-art dMRI schemes PS-Sparse [12], STORM [24] and its variants [28, 29], KLR [27], as well as BiLMDM [40].

Matrix factorization schemes, similar to (11), have been also used in PS-Sparse [12] and KLR [27]. Nevertheless, KRIM differentiates itself from the two previous methods in several aspects. The PCA-based PS-Sparse [12], 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.*, [34]. Sec. IV provides empirical evidence that KRIM offers better data approximation than PS-Sparse in terms of performance metrics 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 (11) to be orthogonal, unlike the PCA-driven PS-Sparse.

On the other hand, the kernel (K)PCA-driven KLR [27] 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 (11) 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 $\mathbf{Y}$ [38], as opposed to KLR which uses only real-valued kernel functions. Furthermore, the single-kernel approaches KLR [27] 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 [34]. 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 (7). 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 III. The difference in the results between KRIM and KLR is also evident by the observation in Fig. 3 where KLR reconstructions show sharp increase in NRMSE values due to aliasing as acceleration rates increase.

SToRM [24, 28, 29] 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 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 (11). KRIM performs better than SToRM along edges (as verified by the HFEN values in Tab. I and III), and produces sharper images with almost no aliasing effect and temporal bleeding.

The transform-learning based FRIST scheme performs im-

age 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. 4 and 3 by the deterioration of NRMSE with increasing acceleration rates.

KRIM has also exhibited improvement in performance over its predecessor BiLMDM [40]. This comes as result of reaping the benefits of similarities and bi-linear modeling via operating in the high dimensional RKHS-feature spaces. Numerical evidence of the improvements are summarized in Tables I, III, and VI. It is also worth noticing that the formulation and algorithmic design of KRIM puts itself way ahead of BiLMDM in terms of computation times, paving the way for more efficient implementations in future studies.

VI. CONCLUSIONS

This manuscript laid down the foundations for a novel imputation-by-regression scheme based on classic kernel-approximation arguments for data which lie into or close an unknown-to-the-user manifold embedded in a high-dimensional Euclidean space. To this end, a bi-linear data-approximation model was introduced which 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 partially observed data with missing measurements, without the need for a fully observed training data-set. 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 schemes under several performance metrics and with a small computational footprint.

ACKNOWLEDGMENT

The authors thank the Institute of Biomedical Eng., ETH Zurich, and the Cardiovascular Magnetic Resonance Society for making the data used in Sec. IV.

APPENDIX A MATHEMATICAL BACKGROUND

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 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 m_1, m_2, n_1, n_2 and the linear mapping $\mathcal{L} : \mathbb{C}^{m_1 \times n_1} \rightarrow \mathbb{C}^{m_2 \times n_2}$, the adjoint of $\mathcal{L}$ is the linear mapping $\mathcal{L}^* : \mathbb{C}^{m_2 \times n_2} \rightarrow \mathbb{C}^{m_1 \times n_1}$ defined as $\langle \mathbf{A} | \mathcal{L}(\mathbf{B}) \rangle = \langle \mathcal{L}^*(\mathbf{A}) | \mathbf{B} \rangle$, $\forall \mathbf{A} \in \mathbb{C}^{m_2 \times n_2}, \forall \mathbf{B} \in \mathbb{C}^{m_1 \times n_1}$. For example, with regards to the sampling mapping $\mathcal{S}(\cdot)$ in Sec. 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 [51], $\mathcal{F}^* = N_k \mathcal{F}^{-1}$ 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 $\mathbf{A}^* = \mathbf{A}^H$. Mapping $\mathcal{L}(\cdot)$ is called Lipschitz continuous, with coefficient $L \in \mathbb{R}_{>0}$, if $\|\mathcal{L}(\mathbf{A}) - \mathcal{L}(\mathbf{B})\|_F \leq L \|\mathbf{A} - \mathbf{B}\|_F$, $\forall \mathbf{A}, \mathbf{B}$.

A complex reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ is a Hilbert space consisting of functions $f : \mathbb{C}^\nu \rightarrow \mathbb{C} : \xi \mapsto f(\xi)$ such that (s.t.) for any ξ the mapping $f \mapsto f(\xi)$ is continuous over $\mathcal{H}$ [30, 37, 38]. For a Hilbert space $\mathcal{H}$, the bivariate functional $\kappa : \mathbb{C}^\nu \times \mathbb{C}^\nu \rightarrow \mathbb{C} : (\xi, \xi') \mapsto \kappa(\xi, \xi')$ is called *reproducing* if (i) $\kappa(\xi, \cdot) \in \mathcal{H}$ for any ξ ; and (ii) $\langle f | \kappa(\xi, \cdot) \rangle_{\mathcal{H}} = f(\xi)$ for any ξ 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 [37]. The *feature mapping* $\varphi(\cdot)$ associated with the RKHS $\mathcal{H}$ is defined as $\varphi(\cdot) : \mathbb{C}^\nu \rightarrow \mathcal{H} : \xi \mapsto \varphi(\xi) := \kappa(\xi, \cdot)$.

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 H denotes the usual Hermitian operation on vectors/matrices. The reproducing property of kernel κ yields the celebrated “kernel trick”: $\kappa(\xi, \xi') = \varphi(\xi)^H \varphi(\xi')$, for any ξ and $\xi' \in \mathbb{C}^\nu$. It can be thus verified that $\kappa(\xi, \xi') = \kappa(\xi', \xi)$ [37].

Popular kernels for complex RKHSs are: (i) The linear kernel $\kappa_L(\xi, \xi') := \xi^H \xi'$; (ii) the Gaussian kernel $\kappa_{G;\gamma}(\xi, \xi') := \exp[-\gamma(\xi - \xi')^H(\xi - \xi')]$ [37, 38], where $\gamma \in \mathbb{R}_{>0}$ and superscript $\dagger$ stands for vector/matrix transposition; and (iii) the polynomial kernel $\kappa_{P;(c,r)}(\xi, \xi') := (\xi^H \xi' + c)^r$, where $r \in \mathbb{Z}_{>0}$, $c \in \mathbb{R}_{>0}$.

For vectors $\{\xi_j\}_{j=1}^J \subset \mathbb{C}^\nu$ and $\Xi := [\xi_1, \dots, \xi_J]$, let the “matrix”:

$$\Phi(\Xi) := [\varphi(\xi_1), \dots, \varphi(\xi_J)]. \quad (17)$$

Then, the $J \times J$ kernel matrix $\mathbf{K}_\Xi$ associated with $\{\xi_j\}_{j=1}^J$ is defined as the matrix whose (j, j') th entry is equal to $\kappa(\xi_j, \xi_{j'})$, or $\mathbf{K}_\Xi := \Phi(\Xi)^H \Phi(\Xi)$ 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 $\text{Prox}_{\lambda g}(\mathbf{A}) := \arg \min_{\mathbf{B} \in \mathbb{C}^{m \times n}} \lambda g(\mathbf{B}) + (1/2) \|\mathbf{A} - \mathbf{B}\|_F^2$ [57]. For example, in the case where g becomes the indicator function ι_C with respect to a closed convex set $C \subset \mathbb{C}^{m \times n}$, i.e., $\iota_C(\mathbf{A}) := 0$, if $\mathbf{A} \in C$, while $\iota_C(\mathbf{A}) := +\infty$, if $\mathbf{A} \notin C$, then $\text{Prox}_{\lambda \iota_C}$ becomes the metric projection mapping onto C : $\text{Prox}_{\lambda \iota_C}(\mathbf{A}) = \arg \min_{\mathbf{B} \in C} \|\mathbf{A} - \mathbf{B}\|_F$. 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 [57]:

$$[\text{Prox}_{\lambda \|\cdot\|_1}(\mathbf{A})]_{ij} = [\mathbf{A}]_{ij} \left(1 - \frac{\lambda}{\max\{\lambda, |[\mathbf{A}]_{ij}|\}} \right). \quad (18)$$

Notation $\text{Soft}(\mathbf{A}, \lambda)$ is often used instead of $\text{Prox}_{\lambda \|\cdot\|_1}(\mathbf{A})$ [44, Lemma V.I].

APPENDIX B SOLVING (13)

This section details the solutions to sub-tasks (13a), (13b) and (13c). The solutions to (13a) and (13b) are computed iteratively via the general framework of [45] which takes the specific form Alg. 2 to address the sub-tasks at hand. Alg. 2 is presented in a form to fit both (13a) and (13b). Moreover, all derivations in this section, as well as Sec. B-B, are performed on the basis of Wirtinger’s calculus [38] which fits well complex-valued vectors/matrices [38].

Algorithm 2 Computing $\hat{\tilde{\mathbf{D}}}_n$ of (13a) and $\hat{\tilde{\mathbf{B}}}_n$ of (13b)

Input: $\mathbf{H}_0$ is either $\hat{\tilde{\mathbf{D}}}_n$ or $\hat{\tilde{\mathbf{B}}}_n$ in step 7 of Alg. 1. Choose parameters $K_0 \in \mathbb{Z}_{>0}$ and $\alpha \in [0.5, 1)$.

Output: $\hat{\tilde{\mathbf{D}}}_n$ and $\hat{\tilde{\mathbf{B}}}_n$ are set equal to the point $\mathbf{H}_{K_0}$ of the sequence $(\mathbf{H}_k)_{k \in \mathbb{Z}_{>0}}$.

- 1: Compute the Lipschitz coefficient L via (24) or (31), and choose $\lambda \in (0, 2(1 - \alpha)/L)$.
 - 2: Let $T(\cdot)$ be either (26) or (33) and define $T_\alpha := \alpha T + (1 - \alpha) \text{Id}$, where Id denotes the identity operator.
 - 3: Set $\mathbf{H}_{1/2} := T_\alpha(\mathbf{H}_0) - \lambda \nabla g_1(\mathbf{H}_0)$, where the gradient $\nabla g_1(\mathbf{H}_0)$ takes the form of either (22) or (30).
 - 4: Set $\mathbf{H}_1 := \text{Prox}_{\lambda g_2}(\mathbf{H}_{1/2})$, where the proximal operator takes the form of either (25) or (32).
 - 5: **while** $0 \leq k \leq K_0 - 2$ **do**
 - 6: $\mathbf{H}_{k+3/2} := \mathbf{H}_{k+1/2} + T(\mathbf{H}_{k+1}) - \lambda \nabla g_1(\mathbf{H}_{k+1}) - T_\alpha(\mathbf{H}_k) + \lambda \nabla g_1(\mathbf{H}_k)$.
 - 7: $\mathbf{H}_{k+2} := \text{Prox}_{\lambda g_2}(\mathbf{H}_{k+3/2})$.
 - 8: Set k equal to $k + 1$ and go to step 5.
 - 9: **end while**
-

A. Solving for $\hat{\tilde{\mathbf{D}}}_n$ in (13a)

Upon defining the convex set $\mathcal{C}_b := \{\tilde{\mathbf{D}} | \|\tilde{\mathbf{D}} \mathbf{e}_i\| \leq C_D, \forall i \in \{1, \dots, d\}\}$, (13a) can be expressed as

$$\hat{\tilde{\mathbf{D}}}_n \in \arg \min_{\tilde{\mathbf{D}}} g_1(\tilde{\mathbf{D}}) + g_2(\tilde{\mathbf{D}}), \quad (19)$$

where

$$g_1(\tilde{\mathbf{D}}) := \frac{1}{2} \|\hat{\mathbf{X}}_n - \tilde{\mathbf{D}} \tilde{\mathbf{K}}_L \hat{\tilde{\mathbf{B}}}_n\|_F^2 + \frac{\tau_D}{2} \|\tilde{\mathbf{D}} - \hat{\tilde{\mathbf{D}}}_n\|_F^2, \quad (20)$$

$$g_2(\tilde{\mathbf{D}}) := \iota_{\mathcal{C}_b}(\tilde{\mathbf{D}}), \quad (21)$$

where $\iota_{\mathcal{C}_b}$ stands for the indicator function w.r.t. $\mathcal{C}_b$; cf. Appendix A. It can be verified that $\forall \tilde{\mathbf{D}}$,

$$\begin{aligned} \nabla g_1(\tilde{\mathbf{D}}) &= \tilde{\mathbf{D}} \tilde{\mathbf{K}}_L \hat{\tilde{\mathbf{B}}}_n \hat{\tilde{\mathbf{B}}}_n^H \tilde{\mathbf{K}}_L^H - \hat{\mathbf{X}}_n \hat{\tilde{\mathbf{B}}}_n^H \tilde{\mathbf{K}}_L^H \\ &\quad + \tau_D (\tilde{\mathbf{D}} - \hat{\tilde{\mathbf{D}}}_n). \end{aligned} \quad (22)$$

It can be also verified that $\forall \tilde{\mathbf{D}}_1, \tilde{\mathbf{D}}_2$,

$$\begin{aligned} & \|\nabla g_1(\tilde{\mathbf{D}}_1) - \nabla g_1(\tilde{\mathbf{D}}_2)\|_F \\ & \leq \|\tilde{\mathbf{K}}_L \hat{\tilde{\mathbf{B}}}_n \hat{\tilde{\mathbf{B}}}_n^H \tilde{\mathbf{K}}_L^H\|_2 \cdot \|\tilde{\mathbf{D}}_1 - \tilde{\mathbf{D}}_2\|_F + \tau_D, \end{aligned} \quad (23)$$

which yields the following Lipschitz coefficient,

$$L := \lambda_{\max}(\tilde{\mathbf{K}}_L \hat{\tilde{\mathbf{B}}}_n \hat{\tilde{\mathbf{B}}}_n^H \tilde{\mathbf{K}}_L^H) + \tau_D. \quad (24)$$

Moreover, $\text{Prox}_{\lambda g_2}(\cdot)$ turns out to be the (metric) projection mapping onto $\mathcal{C}_b$: the i th column of $\text{Prox}_{\lambda g_2}(\tilde{\mathbf{D}})$ is provided by

$$\text{Prox}_{\lambda g_2}(\tilde{\mathbf{D}})\mathbf{e}_i = \frac{C_D}{\max\{C_D, \|\tilde{\mathbf{D}}\mathbf{e}_i\|\}} \tilde{\mathbf{D}}\mathbf{e}_i. \quad (25)$$

Lastly, since (13a) does not include any explicit affine constraint, the affine mapping $T(\cdot)$ in Alg. 2 [45] can be taken as the identity one $\text{Id}(\cdot)$:

$$T(\tilde{\mathbf{D}}) = \tilde{\mathbf{D}}, \forall \tilde{\mathbf{D}}. \quad (26)$$

B. Solving for $\hat{\tilde{\mathbf{B}}}_n$ in (13b)

Recollecting the discussion from Sec. III, define the affine constraint $\mathcal{C}_{\text{aff}} := \{\tilde{\mathbf{B}} := [\mathbf{B}_1^H, \mathbf{B}_2^H, \dots, \mathbf{B}_M^H]^H \mid \mathbf{1}_{N_i}^H \mathbf{B}_m = \mathbf{1}_{N_{\text{fr}}}^H, \forall m \in \{1, \dots, M\}\}$. Then, (13b) can be recast as:

$$\hat{\tilde{\mathbf{B}}}_n \in \arg \min_{\tilde{\mathbf{B}} \in \mathcal{C}_{\text{aff}}} g_1(\tilde{\mathbf{B}}) + g_2(\tilde{\mathbf{B}}), \quad (27)$$

where

$$g_1(\tilde{\mathbf{B}}) := \frac{1}{2} \|\hat{\tilde{\mathbf{X}}}_n - \hat{\tilde{\mathbf{D}}}_n \tilde{\mathbf{K}}_L \tilde{\mathbf{B}}\|_F^2 + \frac{\tau_B}{2} \|\tilde{\mathbf{B}} - \hat{\tilde{\mathbf{B}}}_n\|_F^2, \quad (28)$$

$$g_2(\tilde{\mathbf{B}}) := \|\tilde{\mathbf{B}}\|_1. \quad (29)$$

It can be verified that $\forall \tilde{\mathbf{B}}$,

$$\begin{aligned} \nabla g_1(\tilde{\mathbf{B}}) &= \tilde{\mathbf{K}}_L^H \hat{\tilde{\mathbf{D}}}_n^H \hat{\tilde{\mathbf{D}}}_n \tilde{\mathbf{K}}_L \tilde{\mathbf{B}} - \tilde{\mathbf{K}}_L^H \hat{\tilde{\mathbf{D}}}_n^H \hat{\tilde{\mathbf{X}}}_n \\ &+ \tau_B (\tilde{\mathbf{B}} - \hat{\tilde{\mathbf{B}}}_n). \end{aligned} \quad (30)$$

Following the steps that led to (24), the Lipschitz coefficient for $\nabla g_1(\cdot)$ is computed here as:

$$L = \lambda_{\max}(\tilde{\mathbf{K}}_L^H \hat{\tilde{\mathbf{D}}}_n^H \hat{\tilde{\mathbf{D}}}_n \tilde{\mathbf{K}}_L) + \tau_B. \quad (31)$$

Using (18), $\text{Prox}_{\lambda g_2}(\cdot)$ becomes:

$$\text{Prox}_{\lambda g_2}(\tilde{\mathbf{B}}) = \text{Soft}(\tilde{\mathbf{B}}, \lambda \lambda_1). \quad (32)$$

Finally, owing to the existence of the affine constraint $\mathcal{C}_{\text{aff}}$, the $T(\cdot)$ operator in Alg. 2 [45] can be defined as the projection mapping onto $\mathcal{C}_{\text{aff}}$: $\tilde{\mathbf{B}} := [\mathbf{B}_1^H, \mathbf{B}_2^H, \dots, \mathbf{B}_M^H]^H \mapsto T(\tilde{\mathbf{B}}) := [\mathbf{B}'_1^H, \mathbf{B}'_2^H, \dots, \mathbf{B}'_M^H]^H$, where $\forall i \in \{1, \dots, M\}$,

$$\mathbf{B}'_i := \mathbf{B}_i - \frac{1}{N_i} \mathbf{1}_{N_i} (\mathbf{1}_{N_i}^H \mathbf{B}_i - \mathbf{1}_{N_{\text{fr}}}^H). \quad (33)$$

C. Solving for $\hat{\tilde{\mathbf{X}}}_n$ in (13c)

Upon defining the affine constraint $\mathcal{C}_{\mathcal{F}} := \{\mathbf{X} \mid \mathcal{S}(\mathbf{Y}) = \mathcal{S}\mathcal{F}(\mathbf{X})\}$, problem (13c) can be recast as:

$$\hat{\tilde{\mathbf{X}}}_n \in \arg \min_{\mathbf{X} \in \mathcal{C}_{\mathcal{F}}} g(\mathbf{X}), \quad (34)$$

where

$$g(\mathbf{X}) := \frac{1}{2} \|\mathbf{X} - \hat{\tilde{\mathbf{D}}}_n \tilde{\mathbf{K}}_L \hat{\tilde{\mathbf{B}}}_n\|_F^2 + \frac{\lambda_2}{2} \|\hat{\tilde{\mathbf{Z}}}_n - \mathcal{F}_t(\mathbf{X})\|_F^2. \quad (35)$$

It can be verified that

$$\begin{aligned} \nabla g(\mathbf{X}) &= -\hat{\tilde{\mathbf{D}}}_n \tilde{\mathbf{K}}_L \hat{\tilde{\mathbf{B}}}_n - \lambda_2 N_{\text{fr}} \mathcal{F}_t^{-1}(\hat{\tilde{\mathbf{Z}}}_n) \\ &+ (1 + \lambda_2 N_{\text{fr}}) \mathbf{X}. \end{aligned} \quad (36)$$

The minimizer $\mathbf{X}_*$ of the convex function (35) is obtained by $\nabla g(\mathbf{X}_*) = \mathbf{0}$:

$$\mathbf{X}_* = \frac{1}{1 + \lambda_2 N_{\text{fr}}} [\hat{\tilde{\mathbf{D}}}_n \tilde{\mathbf{K}}_L \hat{\tilde{\mathbf{B}}}_n + \lambda_2 N_{\text{fr}} \mathcal{F}_t^{-1}(\hat{\tilde{\mathbf{Z}}}_n)]. \quad (37)$$

According to [57, Prop 26.13], the solution $\hat{\tilde{\mathbf{X}}}_n$ to (34) is provided by $\hat{\tilde{\mathbf{X}}} = P_{\mathcal{C}_{\mathcal{F}}}(\mathbf{X}_*)$, where $P_{\mathcal{C}_{\mathcal{F}}}(\cdot)$ is the projection mapping onto $\mathcal{C}_{\mathcal{F}}$. Following [57, Prop 3.17], $P_{\mathcal{C}_{\mathcal{F}}}(\mathbf{X}_*) = \mathbf{X}_{\sharp} + P_{\mathcal{C}}(\mathbf{X}_* - \mathbf{X}_{\sharp})$, where $\mathbf{X}_{\sharp}$ is any point in $\mathcal{C}_{\mathcal{F}}$, and $P_{\mathcal{C}}$ stands for the projection operator onto the linear subspace $\mathcal{C} := \{\mathbf{X} \mid \mathbf{0} = \mathcal{S}\mathcal{F}(\mathbf{X})\}$. For example, $\mathbf{X}_{\sharp} := \mathcal{F}^{-1} \mathcal{S}(\mathbf{Y}) \in \mathcal{C}_{\mathcal{F}}$ (notice that $\mathcal{S}^2 = \mathcal{S}$). By virtue of the definition of the projection operator, the fact that $\mathcal{F}^* = N_k \mathcal{F}^{-1}$ (cf. Appendix A) and $\mathcal{S}^c \mathcal{S}(\cdot) = \mathbf{0}(\cdot)$, $P_{\mathcal{C}}(\mathbf{W})$ becomes:

$$\begin{aligned} & P_{\mathcal{C}}(\mathbf{X}_* - \mathbf{X}_{\sharp}) \\ &= \arg \min_{\mathbf{X} \mid \mathbf{0} = \mathcal{S}\mathcal{F}(\mathbf{X})} \|(\mathbf{X}_* - \mathbf{X}_{\sharp}) - \mathbf{X}\|_F^2 \\ &= \arg \min_{\mathbf{X} \mid \mathbf{0} = \mathcal{S}\mathcal{F}(\mathbf{X})} \|\mathcal{F}(\mathbf{X}_* - \mathbf{X}_{\sharp}) - \mathcal{F}(\mathbf{X})\|_F^2 \\ &= \mathcal{F}^{-1} \left(\arg \min_{\mathbf{V} \mid \mathbf{0} = \mathcal{S}(\mathbf{V})} \|\mathcal{F}(\mathbf{X}_* - \mathbf{X}_{\sharp}) - \mathbf{V}\|_F^2 \right) \\ &= \mathcal{F}^{-1} \mathcal{S}^c \mathcal{F}(\mathbf{X}_* - \mathbf{X}_{\sharp}) \\ &= \mathcal{F}^{-1} \mathcal{S}^c \mathcal{F}(\mathbf{X}_*) - \mathcal{F}^{-1} \mathcal{S}^c \mathcal{F} \mathcal{F}^{-1} \mathcal{S}(\mathbf{Y}) \\ &= \mathcal{F}^{-1} \mathcal{S}^c \mathcal{F}(\mathbf{X}_*), \end{aligned} \quad (38)$$

which leads in turn to (14).

REFERENCES

- [1] Z.-P. Liang and P. C. Lauterbur, *Principles of Magnetic Resonance Imaging: A Signal Processing Perspective*. IEEE Press, 2000.
- [2] —, “An efficient method for dynamic magnetic resonance imaging,” *IEEE Trans. Medical Imag.*, vol. 13, no. 4, pp. 677–686, 1994.
- [3] Y. Yang, J. Ma, and S. Osher, “Seismic data reconstruction via matrix completion,” *Inverse Problems & Imaging*, vol. 7, no. 4, p. 1379, 2013.
- [4] G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005.
- [5] D. Lähnemann *et al.*, “Eleven grand challenges in single-cell data science,” *Genome Biology*, vol. 21, no. 1, pp. 1–35, 2020.
- [6] 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,” *The American Statistician*, vol. 61, no. 1, pp. 79–90, 2007.
- [7] M. Lustig, D. Donoho, and J. M. Pauly, “Sparse MRI: The application of compressed sensing for rapid MR imaging,” *Magnetic Resonance in Medicine*, vol. 58, no. 6, pp. 1182–1195, 2007.
- [8] H. Jung, J. C. Ye, and E. Y. Kim, “Improved k-t BLAST and k-t SENSE using FOCUSS,” *Physics in Medicine & Biology*, vol. 52, no. 11, pp. 3201–3226, 2007.
- [9] U. Gampfer, P. Boesiger, and S. Kozerke, “Compressed sensing in dynamic MRI,” *Magnetic Resonance in Medicine*, vol. 59, no. 2, pp. 365–373, Feb. 2008.
- [10] 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, 2014.
- [11] B. Wen, S. Ravishanker, and Y. Bresler, “FRIST—Flipping and rotation invariant sparsifying transform learning and applications,” *Inverse Problems*, vol. 33, no. 7, p. 074007, 2017.

- [12] 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. Medical Imag.*, vol. 31, no. 9, pp. 1809–1820, 2012.
- [13] 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," *Magnetic Resonance in Medicine*, vol. 73, no. 3, pp. 1125–1136, Mar. 2015.
- [14] 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, 2017.
- [15] B. Zhu, J. Z. Liu, S. F. Cauley, B. R. Rosen, and M. S. Rosen, "Image reconstruction by domain-transform manifold learning," *Nature*, vol. 555, no. 7697, pp. 487–492, 2018.
- [16] M. Chen, H. Jiang, W. Liao, and T. Zhao, "Nonparametric regression on low-dimensional manifolds using deep ReLU networks: Function approximation and statistical recovery," *arXiv:1908.01842*, 2019.
- [17] 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, 2020.
- [18] C. M. Sandino *et al.*, "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, 2020.
- [19] 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," *Proceedings of the National Academy of Sciences*, vol. 117, no. 48, pp. 30 088–30 095, 2020.
- [20] J. Nilsson, F. Sha, and M. I. Jordan, "Regression on manifolds using kernel dimension reduction," in *Proc. International Conference on Machine Learning*, 2007, pp. 697–704.
- [21] M. Á. Carreira-Perpiñán and Z. Lu, "Manifold learning and missing data recovery through unsupervised regression," in *Proc. International Conference on Data Mining*, IEEE, 2011, pp. 1014–1019.
- [22] A. V. Little, M. Maggioni, and L. Rosasco, "Multiscale geometric methods for data sets I: Multiscale SVD, noise and curvature," *Applied and Computational Harmonic Analysis*, vol. 43, no. 3, pp. 504–567, 2017.
- [23] M. Usman, D. Atkinson, C. Kolbitsch, T. Schaeffter, and C. Prieto, "Manifold learning based ECG-free free-breathing cardiac CINE MRI," *J. Magnetic Resonance Imag.*, vol. 41, no. 6, pp. 1521–1527, 2015.
- [24] S. Poddar and M. Jacob, "Dynamic MRI using smoothness regularization on manifolds (SToRM)," *IEEE Trans. Medical Imag.*, vol. 35, no. 4, pp. 1106–1115, 2016.
- [25] U. Nakarmi, K. Slavakis, J. Lyu, and L. Ying, "M-MRI: A manifold-based framework to highly accelerated dynamic magnetic resonance imaging," in *Proc. International Symposium on Biomedical Imaging (ISBI)*, 2017, pp. 19–22.
- [26] U. Nakarmi, K. Slavakis, and L. Ying, "MLS: Joint manifold-learning and sparsity-aware framework for highly accelerated dynamic magnetic resonance imaging," in *Proc. International Symposium on Biomedical Imaging (ISBI)*, 2018, pp. 1213–1216.
- [27] 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. Medical Imag.*, vol. 36, no. 11, pp. 2297–2307, 2017.
- [28] S. Poddar *et al.*, "Manifold recovery using kernel low-rank regularization: Application to dynamic imaging," *IEEE Trans. Computational Imaging*, vol. 5, no. 3, pp. 478–491, 2019.
- [29] O. Arif *et al.*, "Accelerated dynamic MRI using kernel-based low rank constraint," *Journal of Medical Systems*, vol. 43, no. 8, p. 271, 2019.
- [30] N. Aronszajn, "Theory of reproducing kernels," *Trans. American Math. Society*, vol. 68, no. 3, pp. 337–404, 1950.
- [31] G. S. Kimeldorf and G. Wahba, "A correspondence between Bayesian estimation on stochastic processes and smoothing by splines," *The Annals of Mathematical Statistics*, vol. 41, no. 2, pp. 495–502, 1970.
- [32] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992.
- [33] A. B. Tsybakov, *Introduction to Nonparametric Estimation*. New York: Springer, 2008.
- [34] B. Schölkopf and A. J. Smola, *Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT press, 2001.
- [35] L. W. Tu, *An Introduction to Manifolds*. New York: Springer, 2008.
- [36] K. Slavakis, G. B. Giannakis, and G. Leus, "Robust sparse embedding and reconstruction via dictionary learning," in *Proc. Conference on Information Sciences and Systems (CISS)*, Baltimore: USA, Mar. 2013.
- [37] I. Steinwart, D. Hush, and C. Scovel, "An explicit description of the reproducing kernel Hilbert spaces of Gaussian RBF kernels," *IEEE Trans. Information Theory*, vol. 52, no. 10, pp. 4635–4643, 2006.
- [38] 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, 2011.
- [39] S. S. Bucak, R. Jin, and A. K. Jain, "Multiple kernel learning for visual object recognition: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1354–1369, 2013.
- [40] G. N. Shetty *et al.*, "Bi-linear modeling of data manifolds for dynamic-MRI recovery," *IEEE Trans. Medical Imag.*, vol. 39, no. 3, pp. 688–702, 2020.
- [41] G. N. Shetty, K. Slavakis, U. Nakarmi, G. Scutari, and L. Ying, "Kernel bi-linear modeling for reconstructing data on manifolds: The dynamic-MRI case," in *Proc. European Signal Processing Conference (EUSIPCO)*, EURASIP, 2021, pp. 1482–1486.
- [42] L. Feng *et al.*, "Golden-angle radial sparse parallel MRI: Combination of compressed sensing, parallel imaging, and golden-angle radial sampling for fast and flexible dynamic volumetric MRI," *Magnetic Resonance in Medicine*, vol. 72, no. 3, pp. 707–717, 2014.
- [43] V. De Silva and J. B. Tenenbaum, "Sparse multidimensional scaling using landmark points," Stanford University, Tech. Rep., 2004.
- [44] A. Maleki, L. Anitori, Z. Yang, and R. G. Baraniuk, "Asymptotic analysis of complex LASSO via complex approximate message passing (CAMP)," *IEEE Trans. Information Theory*, vol. 59, no. 7, pp. 4290–4308, 2013.
- [45] 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.
- [46] 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, 2015.
- [47] L. Wissmann, C. Santelli, W. P. Segars, and S. Kozierke, "MRXCAT: Realistic numerical phantoms for cardiovascular magnetic resonance," *J. Cardiovascular Magnetic Resonance*, vol. 16, no. 1, p. 63, 2014.
- [48] S. Ravishanker and Y. Bresler, "MR image reconstruction from highly undersampled k-space data by dictionary learning," *IEEE Trans. Medical Imag.*, vol. 30, no. 5, pp. 1028–1041, 2011.
- [49] M. Subbarao, T.-S. Choi, and A. Nikzad, "Focusing techniques," *Optical Engineering*, vol. 32, no. 11, pp. 2824–2837, 1993.
- [50] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, *et al.*, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Imaging Process.*, vol. 13, no. 4, pp. 600–612, 2004.
- [51] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A fresh approach to numerical computing," *SIAM Review*, vol. 59, no. 1, pp. 65–98, 2017.
- [52] *MATLAB (R2018b)*, Natick, Massachusetts: The Mathworks, Inc., 2018. [Online]. Available: https://www.mathworks.com/?s_tid=gn_logo/.
- [53] S. Poddar and M. Jacob, *SToRM MATLAB Package*, accessed in 2018. [Online]. Available: <https://github.com/sunrita-poddar/l2SToRM>.
- [54] B. Wen, S. Ravishanker, and Y. Bresler, *FRIST MATLAB Package*, accessed in 2018. [Online]. Available: https://github.com/wenbihan/FRIST_ivp2017.
- [55] 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. Medical and Biological Eng.*, vol. 20, no. 4, pp. 203–210, 2000.
- [56] A. Haase, J. Frahm, D. Matthaei, W. Hancin, and K.-D. Merboldt, "FLASH imaging. Rapid NMR imaging using low flip-angle pulses," *Journal of Magnetic Resonance (1969)*, vol. 67, no. 2, pp. 258–266, 1986.
- [57] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer, 2011.

Fig. 5. Reconstruction results along the time axis for the MRXCAT Cardiac Phantom measured at an acceleration rate of 8x; *cf.* Sec. 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 box. (b) Comparative results of the temporal cross-section via methods KRIM[M], KRIM[S], BiLMDM, KLR, FRIST, STORM, PS-Sparse (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. 6. Reconstruction results across the spatial domain for the in-vivo human cardiac Phantom at an acceleration rate of 8x; *cf.* Sec. 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 (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 the systole phase. Red markings point out distortions, aliasing, or undesired artifacts in the reconstructions.

Fig. 7. Reconstruction results along the time domain for the in-vivo human cardiac phantom at an acceleration rate of 8x: cf. Sec. 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 (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. 8. Reconstructions from measurements acquired from a volunteer are exhibited for schemes KRIM[M], KRIM[S], BiLMDM, KLR, FRIST, STORM, PS-Sparse (left to right). The first row shows the spatial results for frame 1, while the second row shows the temporal cross section per scheme.