Bayesian Optimization of Sampling Densities in MRI

Alban Gossard, Frédéric de Gournay, Pierre Weiss

Bayesian Optimization of Sampling Densities in MRI

Alban Gossard[1,3] alban.paul.gossard@gmail.com
Frédéric de Gournay[1,2] degourna@insa-toulouse.fr
Pierre Weiss[1,4] pierre.weiss@cnrs.fr

[1] Institut de Mathématiques de Toulouse; UMR5219; Université de Toulouse; CNRS
[2] INSA, F-31077 Toulouse, France [3] UPS, F-31062 Toulouse Cedex 9, France
[4] CNRS, Université de Toulouse, France

September 14, 2022

Abstract

Data-driven optimization of sampling patterns in MRI has recently received a significant attention. Following recent observations on the combinatorial number of minimizers in off-the-grid optimization, we propose a framework to globally optimize the sampling densities using Bayesian optimization. Using a dimension reduction technique, we optimize the sampling trajectories more than 20 times faster than conventional off-the-grid methods, with a restricted number of training samples. This method – among other benefits – discards the need of automatic differentiation. Its performance is slightly worse than state-of-the-art learned trajectories since it reduces the space of admissible trajectories, but comes with significant computational advantages. Other contributions include: i) a careful evaluation of the distance in probability space to generate trajectories ii) a specific training procedure on families of operators for unrolled reconstruction networks and iii) a gradient projection based scheme for trajectory optimization.

1 Introduction

The quest for efficient acquisition and reconstruction mechanisms in Magnetic Resonance Imaging (MRI) has been ongoing since its invention in the 1970's. This led to a few major breakthrough, which comprise the design of efficient pulse sequences [9], the use of parallel imaging [50, 10], the theory and application of compressed sensing [42] and its recent improvements thanks to the progresses in learning and GPU computing [38]. While the first attempts to use neural networks in this field were primarily focused on the efficient design of reconstruction algorithms [32], some recent works began investigating the design of efficient sampling schemes or joint sampling/reconstruction schemes. The aim of this paper is to make progress in the numerical analysis of this nascent and challenging field.

1.1 Some sampling theory

In a simplified way, an MRI scanner measures values of the Fourier transform of the image to reconstruct at different locations $(\xi_m)_{1\leq m\leq M}$ in the so-called k-space. The locations (ξ_m) are obtained by sampling a continuous trajectory defined through a gradient sequence. The problem we tackle in this paper is: how to choose the points (ξ_m) or the underlying trajectories in an efficient or optimal way?

Shannon-Nyquist This question was first addressed using Shannon-Nyquist theorem, which certifies that sampling the k-space on a sufficiently fine Euclidean grid provides exact reconstructions using linear reconstructors. This motivated the design of many trajectories,

such as the ones in echo-planar imaging (EPI) [53]. Progresses on non uniform sampling theory [21] then provided guidelines to produce efficient sampling/reconstruction schemes for linear reconstructors. This theory is now mature for the reconstruction of bandlimited functions. In a nutshell, it advocates the use of a sampling set which covers the k-space sufficiently densely with well spread samples.

Compressed sensing theory Shannon-Nyquist theory requires sampling the k-space densely, resulting in long scanning times. It was observed in the 1980's that subsampling the high frequencies using variable density radial patterns did not compromise the image quality too much [5, 31]. The first theoretical elements justifying this evidence were provided by the theory of compressed sensing, when using nonlinear reconstructors. This seminal theory is based on concepts such as the restricted isometry property (RIP) or the incoherence between the measurements [14, 43]. However it soon became evident that these concepts were not suited to the practice of MRI and a refined theory based on local coherence appeared in [3, 11]. The main teaching is that a good sampling scheme for ℓ^1-based reconstruction methods must have a variable density that depends on the sparsity basis and on the sparsity pattern of the images. To the best of our knowledge, this theory is currently the one that provides the best explanation of the success of sub-sampling. In particular, analytical expressions of the optimal densities [1] can be derived and fit relatively well with the best empirical ones.

The main teachings To date, there is still a significant discrepancy between the theory and practice of sampling in MRI. A mix between theory and common sense however provides the following main insights. A good sampling scheme should [12]:

- have a variable density, decaying with the distance to the center of the k-space,
- have a sufficiently high density in the center to comply with the Shannon-Nyquist criterion, and sufficiently low to avoid dense clusters which would not bring additional information,
- have a locally uniform coverage of the k-space. In particular, nearby samples are detrimental to the reconstruction since they are highly correlated and increase the condition number of partial Fourier matrices.

These considerations are all satisfied when using Poisson disk sampling with an adequate density [56] for pointwise sampling. They also led to the development of the SPARKLING trajectories [18, 40], which incorporate additional trajectory constraints in the design.

What can still be optimized? Given the previous remarks, an important question remains open: how to choose the sampling density? An axiomatic approach leads to choosing radial densities with a plateau (constant value) at the center. The radial character ensures rotation invariance, which seems natural to image organs in arbitrary orientations. The plateau enforces Nyquist rate at the center. However, it may still be possible to improve the results for specific datasets.

1.2 Data-driven sampling schemes

The first attempts to learn a sampling density [37, 63] were based on the average energy of the k-space coefficients on a collection of reference images. While this principle is valid for linear reconstructions, it is not supported by a theoretical background when using nonlinear reconstructors. Motivated by the recent breakthroughs of learning and deep learning, many

authors recently proposed to learn either the reconstructor [30], the sampling pattern [8, 28, 64, 54], or both [33, 6, 59, 4, 58]. Data-driven optimization has emerged as a promising approach to tailor the sampling schemes with respect to the reconstructor and to the image structure. In [8, 28, 54, 51, 64], the authors look for an optimal subset of a fixed set of k-space positions. The initial algorithms are based on simple greedy approaches that generated a sampling pattern by iteratively selecting a discrete horizontal line that minimizes the residual error of the reconstructed image. This approach is limited to low dimensional sets of parallel lines. Some efforts have been spent on finding better and more scalable solutions to this hard combinatorial problem using stochastic greedy algorithms [51], ℓ^1-relaxation and bi-level programming [54] or bias-accelerated subset selection [64]. This method is reported to provide results over 3D images and seems to have solved some of the scalability issues.

To the best of our knowledge, the first work investigating the joint optimization of a sampling pattern and a reconstruction algorithm was proposed in [33]. In this work, the authors use a Monte Carlo Tree Search which allows them to optimize a policy that determines the positions to sample. This sampling relies on lines and the reconstruction process is an image to image domain with an inverse Fourier transform performed on the data before the denoising step. In the same spirit, [6] proposes to learn MRI trajectories by optimizing a binary mask over a Cartesian grid with some sparsity constraint. The reconstruction is decomposed into two steps: a regridding using an inverse Fourier transform and a U-NET for de-aliasing. Finally, a new class of reconstruction methods called *algorithm unrolling*, mimicking classical variational approaches have emerged. These approaches improve the interpretability of deep learning based methods. Optimizing the weights of a CNN that plays the role of a denoiser in a conjugate gradient descent has been investigated in [4]. The authors jointly optimize the sampling pattern and a denoising network based on an unrolled conjugate gradient scheme. The sampling scheme is expressed as the tensor product of 1D sampling patterns which significantly restricts the possible sampling schemes.

Overall, the previous works suffer from some limitations: the sampling points are required to live on a Cartesian grid, which may be non physical and lead to combinatorial problems; the methods cannot incorporate advanced constraints on the sampling trajectory and therefore focus on "rigid" constraints such as selecting a subset of horizontal lines.

To address these issues, some recent works propose to optimize points that can move freely in a continuous domain [59, 58]. This approach allows handling real kinematic constraints. In [59], the authors propose to reconstruct an image using a rough inversion of the partial Fourier transform, followed by a U-NET to eliminate the residual artifacts. They optimize jointly the weights of the U-NET together with the k-space positions using a stochastic gradient method. The physical kinematic constraints are handled using two different ingredients. First, the k-space points are regularly ordered by solving a traveling salesman problem, ensuring a low distance between consecutive points. Second, the constraints are promoted using a penalization function. This re-ordering step was then abandoned in [58], where the authors use a B-spline parameterization of the trajectories with a penalization over the constraints in the cost function. Instead of using a rough inversion with a U-NET, the authors opted for an unrolled ADMM reconstructor where the proximal operator is replaced by a DIDN CNN [60]. The k-space locations and the CNN weights are optimized jointly. In both works, long computation times and memory requirements are reported. We also observed significant convergence issues related to the existence of spurious minimizers [26].

1.3 Our contribution

The purpose of this work is to improve the process of optimizing sampling schemes from a methodological perspective. We propose a framework that optimizes the sampling density using Bayesian Optimization (BO). Our method has a few advantages compared to recent learning based approaches: i) it globalizes the convergence by reducing the dimensionality of the optimization problem, ii) it reduces the computing times drastically, iii) it requires only a small number of reference images and iv) it works off-the-grid and handles arbitrary physical constraints. The first three features are essential to make sampling scheme optimization tractable in a wide range a different MRI scanners. The last one allows more versatility in the sampling patterns that can take advantage of all the degrees of freedom offered by an MRI scanner.

2 The proposed approach

In this section, we describe the main ideas of this work after having introduced the notation.

2.1 Preliminaries

Images Let $\mathcal{X}$ denote the set of K training images $\mathcal{X} = \{x_1, \dots, x_K\}$. A D-dimensional image is a vector of $\mathbb{C}^N$, where $N = N_1 \dots N_D$ and $N_d \in 2\mathbb{N}$ denotes the number of pixels in the d-th direction. In this work, each index $n \in [\![1, N]\!]$, is associated to a position $p_n \in [\![-\frac{N_1}{2}, \frac{N_1}{2} - 1]\!] \times \ldots \times [\![-\frac{N_D}{2}, \frac{N_D}{2} - 1]\!]$ on Euclidean grid. It describes the location of the n-th pixel in the k-space. With a slight abuse of notation, we associate to each discrete image $x_k \in \mathbb{C}^N$, a function still denoted x_k, defined by

$$x_k = \left(\sum_{n=1}^{N} x_k[n] \delta_{p_n} \right) \star \psi,$$

where $\star$ denotes the convolution-product and where ψ is an interpolation function. For instance, we can set ψ as the indicator of a grid element to generate piece-wise constant images.

Image quality To measure the reconstruction quality, we consider an image quality metric $\eta : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}_+$. The experiments in this work are conducted using the squared ℓ^2 distance $\eta(\tilde{x}, x) = \frac{1}{2}\|\tilde{x} - x\|_2^2$. Any other metric could be used instead with the proposed approach.

The Non Uniform Fourier Transform Throughout the paper, we let $\xi = (\xi_1, \dots, \xi_M) \in (\mathbb{R}^D)^M$ denote a set of locations in the k-space (or Fourier domain). Let $A(\xi) \in \mathbb{C}^{M \times N}$ denote the forward non uniform Fourier transform defined for all $m \in [\![1, M]\!]$ and $x \in \mathbb{C}^N$ by

$$\begin{aligned} [A(\xi)(x)]_m &= \int_{t \in \mathbb{R}^D} \exp(-i\langle t, \xi_m \rangle) x(t)\, dt \\ &= \Psi(\xi_m) \cdot \sum_{n=1}^{N} x[n] \exp(-i\langle p_n, \xi_m \rangle), \end{aligned} \tag{1}$$

where Ψ is the Fourier transform of the interpolation function ψ.

Image reconstruction We let $R : \mathbb{C}^M \times (\mathbb{R}^D)^M \times \mathbb{R}^J \to \mathbb{C}^N$ denote an image reconstruction mapping. For a measurement vector $y \in \mathbb{C}^M$, a sampling scheme $\xi \in (\mathbb{R}^D)^M$, and a parameter $\lambda \in \mathbb{R}^J$, we let $\tilde{x} = R(\xi, y, \lambda)$ denote the reconstructed image. In this paper, we will consider two different reconstructors:

- A Total Variation (TV) reconstructor [42], which is a standard baseline:

$$R_1(\xi, y, \lambda) = \underset{x \in \mathbb{C}^N}{\arg\min} \frac{1}{2}\|A(\xi)x - y\|_2^2 + \lambda\|\nabla x\|_1, \tag{2}$$

  where $\lambda \geq 0$ is a regularization parameter. The approximate solution of this problem is obtained with an iterative algorithm run for a fixed number of iterations. We refer the reader to Appendix A.1 for the algorithmic details. This allows us to use the automatic differentiation of PyTorch as described in [48].

- An unrolled neural network $R_2(\xi, y, \lambda)$, where λ denotes the weights of the neural network. There is now a multitude of such reconstructors available in the literature [45]. They draw their inspiration from classical model-based reconstructors with handcrafted priors. The details are provided in Appendix A.2.

Constraints on the sampling scheme As mentioned in the introduction, the sampling positions $\xi = (\xi_1, \ldots, \xi_M)$ correspond to the discretization of a k-space trajectory subject to kinematic constraints. Throughout the paper, we let $\Xi \subset (\mathbb{R}^D)^M$ denote the constraint set for ξ. A sampling set consists of $N_s \in \mathbb{N}$ trajectories (shots) with P measurements per shot. We consider realistic kinematic constraints on these trajectories. Let α denote the maximal speed of a discrete trajectory and β denote its maximal acceleration (the slew rate). We let

$$Q_P^{\alpha,\beta} = \left\{ \xi \in ([-\pi, \pi]^D)^P, \|\dot{\xi}\|_\infty \leq \alpha, \|\ddot{\xi}\|_\infty \leq \beta, C\xi = b \right\}, \tag{3}$$

where

$$\|\dot{\xi}\|_\infty = \max_{1 \leq p \leq P-1} \|\xi_{p+1} - \xi_p\|_2$$
$$\|\ddot{\xi}\|_\infty = \max_{2 \leq p \leq P-1} \|\xi_{p+1} + \xi_{p-1} - 2\xi_p\|_2,$$

where b is a vector and C a matrix encoding some position constraints. For instance, we enforce the first point of each trajectory to start at the origin. Since the sampling schemes consists of N_s trajectories, the constraint set on the sampling is $\Xi = (Q_P^{\alpha,\beta})^{N_s}$. The total number of measurements M equal to $M = N_s \cdot P$. We refer the reader to [19] for more details on these constraints.

2.2 The challenges of sampling scheme optimization

In this paper, we consider the optimization of a sampling scheme for a fixed reconstruction mapping R. A good sampling scheme should reconstruct the images in the training set $\mathcal{X}$ efficiently in average. Hence, a natural optimization criterion is

$$\min_{\xi \in \Xi} \mathbb{E}\left(\frac{1}{K} \sum_{k=1}^{K} \eta\left(R(\xi, A(\xi)x_k + n, \lambda), x_k\right) \right). \tag{4}$$

The term $A(\xi)x_k$ corresponds to the measurements of the image x_k associated to the sampling scheme ξ. The expectation is taken with respect to the term $n \in \mathbb{C}^N$ which models

noise on the measurements. More elaborate forward models can be designed to account for sensibility matrices in multi-coil imaging or for trajectory errors. We will not consider these extensions in this paper. Their integration is straightforward – at least at an abstract level.

Even if problem (4) is simple to state (and very similar to [59, 58]), the practical optimization looks extremely challenging for the following reasons:

- The computation of the cost function is very costly.
- Computing the derivative of the cost function using backward differentiation requires differentiating a Non-uniform Fast Fourier Transform (NFFT). It also requires a consequent quantity of memory that limits the complexity of the reconstruction mapping.
- The energetic landscape of the functional is usually full of spurious minimizers [26].
- The minimization of an expectation calls for the use of stochastic gradient descent, but the additional presence of a constraint set Ξ reduces the number of solvers available.

Hence, the design of efficient computational solutions is a major issue. It will be the main focus of this paper. The following sections are dedicated to the simplification of (4) and to the design of a lightweight solver. We also propose a home-made solver that attacks (4) directly. Since similar ideas were proposed in [58], we describe the main ideas and differences in Appendix A.5.1 only.

2.3 Regularization and dimensionality reduction

The non-convexity of (4) is a major hurdle inducing spurious minimizers [26]. We discuss the existing solutions to mitigate this problem and give our solution of choice.

2.3.1 Existing strategies and their limitation

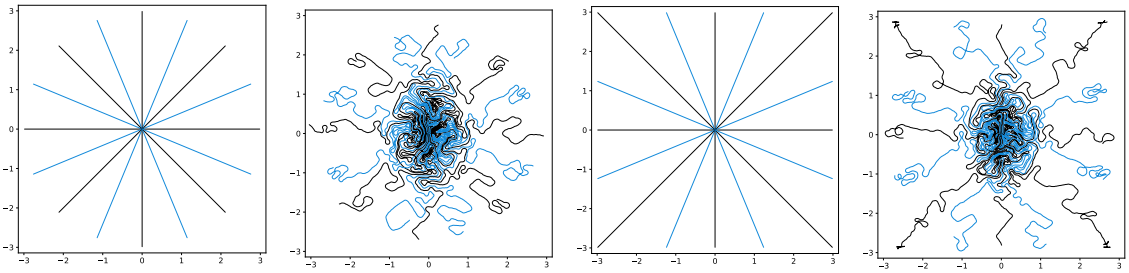
In [59, 58], the authors propose to avoid local minima by using a multi-scale optimization approach starting from a trajectory described through a small number of control points and progressively getting more complex through the iterations. The use of the stochastic Adam optimizer can also allow escaping from narrow basins of attraction. In addition, Adam optimizer can be seen as a preconditioning technique, which can accelerate the convergence, especially for the high frequencies [26]. This optimizer together with a multi-scale approach can yield sampling schemes with improved reconstruction quality at the cost of a long training process. However, despite heuristic approaches to globalize the convergence, we experienced significant difficulties in getting reproducible results.

To illustrate this fact, we conducted a simple experiment in Fig. 1. Starting from two similar initial sampling trajectories, we let a multi-scale solver run for 14 epochs and 85 hours on the fastMRI knee database. We then evaluate the average reconstruction PSNR on the validation set. As can be seen, the final point configuration and the average performance varies significantly.

2.3.2 Optimizing a sampling density

The key idea in this paper is to regularize the problem by optimizing a sampling density rather than the point positions directly. To formalize this principle we need to introduce two additional ingredients:

1. A probability density generator $\rho : \mathbb{R}^L \to \mathcal{P}$, where $\mathcal{P}$ is the set of probability distributions on $\mathbb{R}^D$. In this paper, ρ will be defined as a simple affine mapping, but we could also consider more advanced generators such as Generative Adversarial Networks.



(a) Radial init. (b) Optimized scheme 33.48 ± 3.31dB (c) Box init. (d) Optimized scheme 33.17 ± 3.23dB

Figure 1: The globalization issue: optimizing a scheme with an advanced multi-scale approach yields different average PSNR when starting from different point configurations. In this experiment, we used a total variation reconstruction algorithm and 10% undersampling.

2. A trajectory sampler $\mathcal{S}_M : \mathcal{P} \to (\mathbb{R}^D)^M$, which maps a density ρ to a point configuration $\mathcal{S}_M(\rho) \in (\mathbb{R}^D)^M$. Various possibilities could be considered such as Poisson point sampling, Poisson disk sampling. In this paper, we will use discrepancy based methods [12].

Instead of minimizing (4), we propose to work directly with the density. Letting $\xi : \mathbb{R}^L \to (\mathbb{R}^D)^M$ denote the mapping defined by

$$\xi(z) \stackrel{\text{def}}{=} \mathcal{S}_M(\rho(z)), \tag{5}$$

we propose to minimize:

$$F(z) \stackrel{\text{def}}{=} \min_{z \in \mathcal{C} \subset \mathbb{R}^L} \frac{1}{K} \sum_{k=1}^{K} \mathbb{E}\left(\eta\left[R(\xi(z), A(\xi(z))x_k + n, \lambda), x_k\right]\right), \tag{6}$$

where the expectation is taken with respect to the noise term n. A schematic illustration of this approach is proposed in Fig. 2.

2.3.3 The density generator

Various approaches could be used to define a density generator ρ. In this work, we simply define $\rho(z)$ as an affine mapping, i.e.

$$\rho(z) \stackrel{\text{def}}{=} \mu_0 + \sum_{l=1}^{L} z_l \mu_l, \tag{7}$$

where z belongs to a properly defined convex set $\mathcal{C}$. We describe hereafter how the eigenelements $(\mu_l)_l$ and the set $\mathcal{C}$ are constructed.

A candidate space of densities The general idea of our construction is to define a family of elementary densities and to enrich it by taking convex combinations of its elements.

Let $\theta \in [0, \pi[$ denote a rotation angle, σ_x, σ_y denote lengths, $r > 0$ denote a density at the center and $\gamma > 0$ a decay rate. For $(x, y) \in \mathbb{R}^2$, let $x_\theta = x\cos(\theta) + y\sin(\theta)$, $y_\theta = -\sin(\theta)x + \cos(\theta)y$. We define

$$\Psi(x, y; \sigma_x, \sigma_y, \theta, r, \gamma) = \frac{1}{c} \min\left(r, \frac{1}{\left(\left(\frac{x_\theta}{\sigma_x}\right)^2 + \left(\frac{y_\theta}{\sigma_y}\right)^2 + \epsilon\right)^\gamma}\right), \tag{8}$$

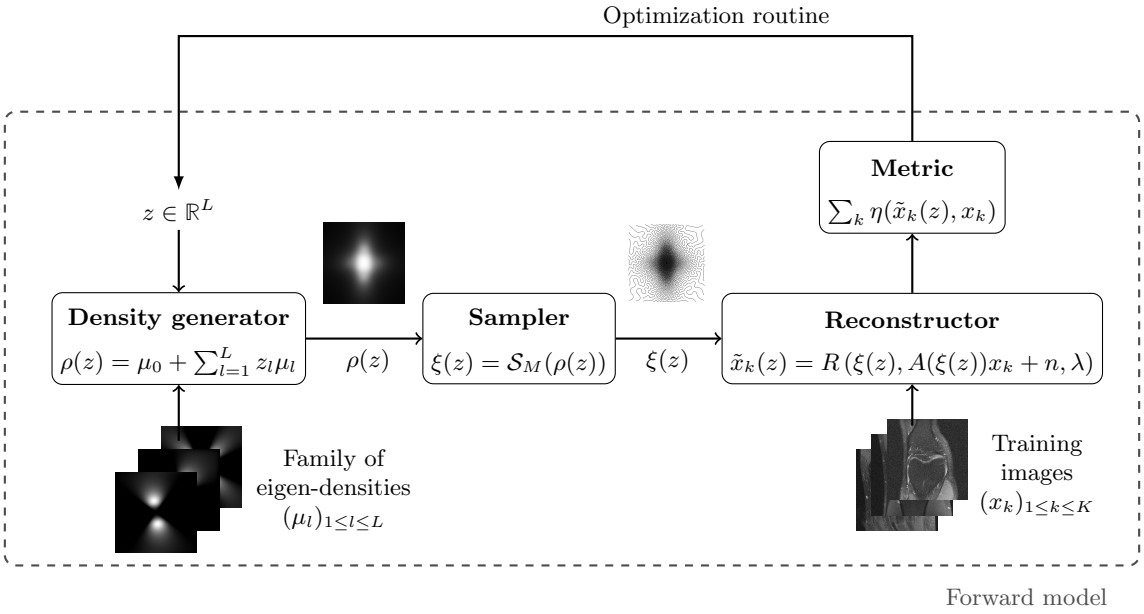


Figure 2: A schematic illustration of the proposed algorithm. We generate a sampling density $\rho(z)$ through an affine combination of eigen-elements (μ_l). The density is then used in a sampling pattern generator $\mathcal{S}_M$ which yields a sampling trajectory $\xi(z)$. A set of training images are then reconstructed using this scheme. This allows computation of the (batch) average error. A zero-th, or first order (automatic differentiation) optimization routine optimizes the sampling density iteratively.

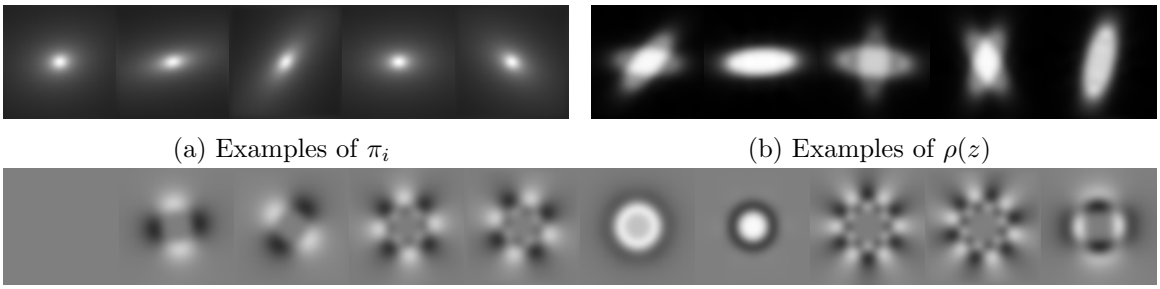
where c is a normalizing constant such that $\int_{\mathbb{R}^2} \Psi = 1$. We then smooth the function Ψ by convolving it with a Gaussian function G_κ of standard deviation $\kappa > 0$:

$$\pi = G_\kappa \star \Psi. \tag{9}$$

The elements in this family are good candidates for sampling densities: i) they are nearly constant and approximately equal to r at the center of the k-space, ii) they can be anisotropic to accommodate for specific image orientations and iii) they have various decay rates, allowing sampling the high frequencies more or less densely. Some examples of such density are displayed in Fig. 3a. However, the family of densities generated by this procedure is quite poor. For instance, it is impossible to sample densely both the x and y axes simultaneously. In order to enrich it, we propose to consider the set of convex combinations of these elementary densities. This allows us to construct more general multi-modal densities, see Fig. 3b for examples of such convex combinations.

Dimensionality reduction In order to construct the family $(\mu_0, \dots, \mu_L)$, we first draw a large family of $I \gg L$ densities $(\pi_i)_{1 \leq i \leq I}$. They are generated at random by uniform draws of the parameters $(\sigma_x, \sigma_y, \theta, t, \gamma)$ inside a box. We then perform a principal component analysis (PCA) on this family to generate some eigen-elements $(\nu_l)_{0 \leq l \leq L}$. We set $\mu_0 = \nu_0 / \langle \nu_0, \mathbb{1} \rangle$. Since probability densities must sum to 1, we orthogonalize the family (ν_l) with respect to the vector μ_0. Thereby, we obtain a second family $(\mu_l)_{0 \leq l \leq L}$ that satisfies $\langle \mu_0, \mathbb{1} \rangle = 1$ and $\langle \mu_l, \mathbb{1} \rangle = 0$ for all $1 \leq l \leq L$. This procedure discards one dimension. The resulting PCA basis is illustrated in Fig. 3c.

Let $\mathcal{E}$ denote the intersection of the span of $(\mu_l)_{l \leq L}$ with the probability densities and $\Pi_\mathcal{E}$ the orthogonal projection on $\mathcal{E}$. The space of densities is the convex hull of the family



(a) Examples of π_i (b) Examples of $\rho(z)$

(c) μ_l for $0 \leq l \leq 9$

Figure 3: Examples of densities using the proposed parameterization.

$(\pi_i)_i$ projected on $\mathcal{E}$:

$$\mathcal{C} \stackrel{\text{def}}{=} \text{Conv}\left(\Pi_{\mathcal{E}}\left(\pi_i\right), 1 \leq i \leq I\right). \tag{10}$$

As illustrated in Fig. 3b, this process overall provides a rather rich and natural family with a low dimensionality (here $L = 20$).

2.3.4 The sampler

The sampler $\mathcal{S}_M : \mathcal{P} \to (\mathbb{R}^D)^M$ is based on discrepancy minimization [52, 29, 18]. It is defined as an approximate solution of

$$\mathcal{S}_M(\rho) = \arg\min_{\xi \in \Xi} \text{dist}\left(\frac{1}{n}\sum_{m=1}^{M} \delta_{\xi[m]}, \rho\right), \tag{11}$$

where $\Xi \subset (\mathbb{R}^D)^M$ takes into account the trajectory constraints and dist is a discrepancy defined by

$$\text{dist}(\mu, \nu) = \sqrt{\langle h \star (\mu - \nu), (\mu - \nu)\rangle_{L^2(\mathbb{R}^D)}},$$

where h is a positive definite kernel (i.e. a function with a real positive Fourier transform). Other metrics on the set of probability distributions could be used such as the transportation distance [41]. The formulation (11) has already been proposed in [18] and it is at the core of the Sparkling scheme generation [40]. We will discuss the choice of the kernel h in the numerical experiments: it turns out to play a critical role.

In practice (11) is not solved exactly: an iterative solver [19] is ran for a fixed number of iterations. This allows use of automatic differentiation in order to compute the Jacobian of ξ w.r.t. z. Technical details about the implementation of this sampler are provided in Appendix A.5.

2.3.5 The pros and cons of this strategy

The optimization problem (6) presents significant advantages compared to the original one (4):

- The number of optimization variables is considerably reduced: instead of working with $D \cdot M$ variables, we now only work with $L \ll D \cdot M$ variables defining a continuous density. In this paper we set $L = 20$ which is considerably smaller in comparison to the $M = 25801$ 2D sampling points for the formulation of (4) with 25% undersampling on 320×320 images. This allows resorting to global optimization routines. Hereafter, we will describe a Bayesian optimization approach.

- The point configurations generated by this algorithm are always locally uniform since they correspond to the minimizers of a discrepancy. Clusters are therefore naturally discarded, which can be seen as a natural regularization scheme.
- As discussed in the numerical experiments, the regularization effect allows optimizing the sampling density with a small dataset with a similar performance. Optimizing the function with as little as 32 reference images yields a near optimal density. This aspect might be critical for small databases.

On the negative side, notice that we considerably constrained the family of achievable trajectories, thereby reducing the maximal achievable gain. We will show later that the trajectories obtained by minimizing (6) are indeed slightly less efficient than those obtained with (4). This price might be affordable if we compare it to the advantages of having a significantly faster and more robust solver requiring only a fraction of the data needed for solving (4).

2.4 The optimization routine

In this section, we describe an algorithmic approach to attack the problem (6).

2.4.1 The non informativeness of the gradient

A natural approach to solve (6) is to optimize the coefficients $z \in \mathbb{R}^L$ using a gradient based algorithm. Unfortunately, the reparameterization of the cost function with a density still makes the energy profile full of spurious minimizers. The presence of these oscillations would trap a gradient based algorithm in local minimizers. Fig. 4 illustrates this fact. In the "Shift" row, we display the energy profile when shifting the sampling pattern on the top-left by 4 pixels in the horizontal and vertical direction. In the "Density" row, we display the energy profile with a family of $L = 2$ eigen-elements (μ_1, μ_2). The 3×3 red dots on the energy profiles corresponds to the 3×3 sampling densities on the top-right.

Overall, this experiment shows that the gradient direction is not meaningful: it oscillates in an erratic way. This advocates for the use of 0^{th} order optimization methods. A significant advantage of this observation is that it allows discarding the memory and time issues related to automatic differentiation.

2.4.2 Bayesian optimization

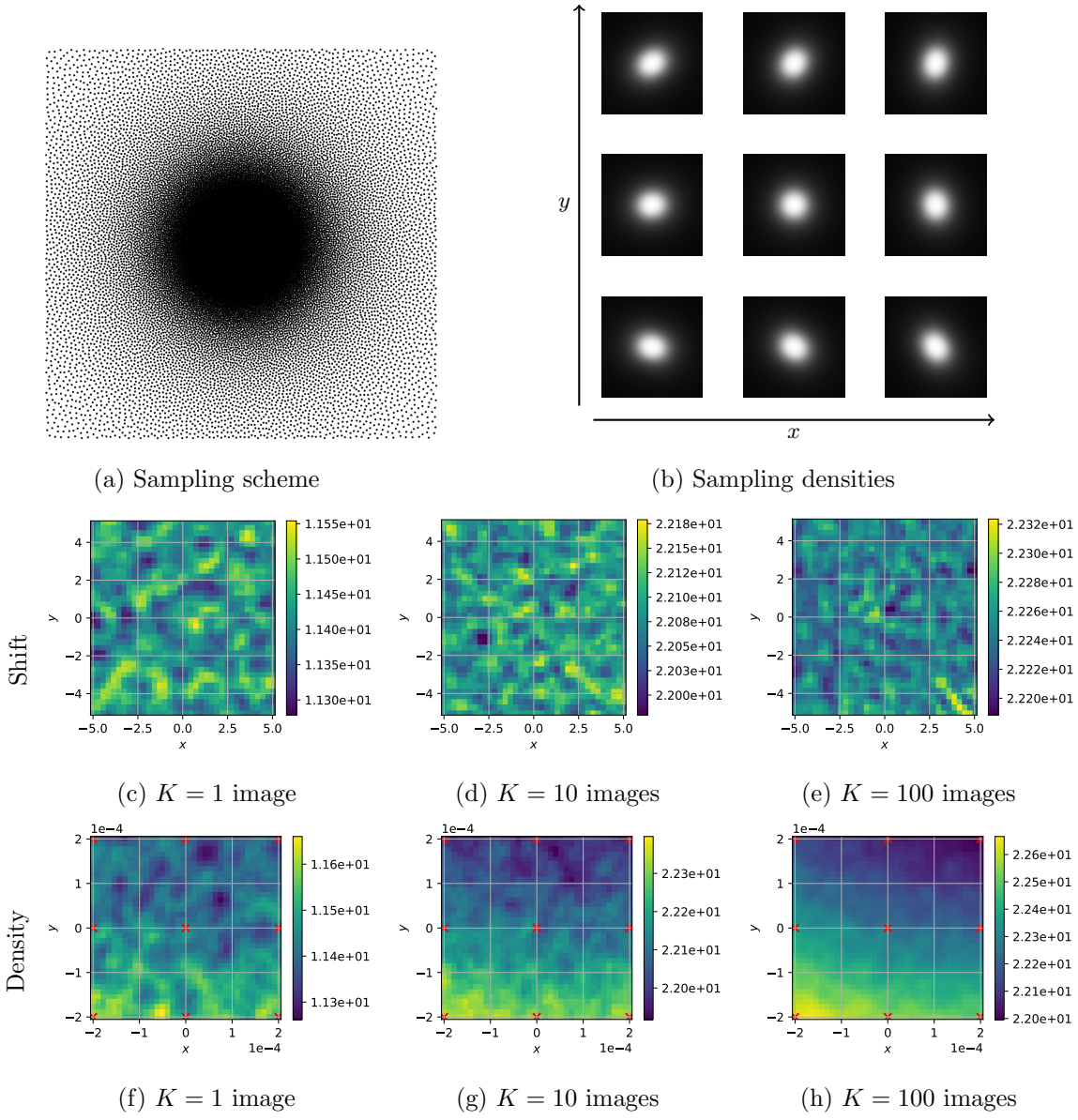
As can be seen from the energy profiles in Fig. 4, the cost function seems to be decomposable as a smooth function plus an oscillatory one of low amplitude. This calls for the use of algorithms that i) sample the function at a few scattered points, ii) construct a smooth surrogate approximation, iii) find a minimizer of the surrogate and add it to the explored samples, iv) go back to ii).

Bayesian Optimization (BO) [23] is a principled approach that follows these steps. It seems particularly adequate since it models uncertainty on the function evaluations and comes with advanced solvers [7]. Its application is nonetheless nontrivial and requires some care in our setting. We describe some technical details hereafter.

Consider an objective function of the form

$$\inf_{z \in \mathcal{C}} \mathbb{E}\left(F(z, V)\right),$$

where V is a random vector. In our setting, V models both the noise n and the input images x_k. We consider V to be a random vector taken uniformly inside a database. In that setting, Bayesian Optimization requires the following ingredients:



(a) Sampling scheme

(b) Sampling densities

(c) $K = 1$ image

(d) $K = 10$ images

(e) $K = 100$ images

(f) $K = 1$ image

(g) $K = 10$ images

(h) $K = 100$ images

Figure 4: Illustration of the spurious minimizers. Here, we consider a total variation reconstructor and 25% under-sampling. Second row: cost function when the sampling scheme on the top-left is shifted along the x and y axes (grid size = 1 pixel). Last row: energy profiles when sampling using interpolation of the densities on the top-right. The 3×3 red dots correspond to the densities on the top-right. Observe that the oscillation amplitude decays with the number of images, but spurious minimizers are present whatever the number of images.

1. An initial sampling set.
2. A black-box evaluation routine of $F(z, V)$.
3. A family of interpolation functions together with a regression routine.
4. A solver that minimizes the regression function.

Hereafter, each choice made in this work is described.

The initial sampling set To initialize the algorithm, we need the convex set $\mathcal{C}$ to be covered as uniformly as possible in order to achieve a good uniform approximation of the energy profile. In this work, we used a maximin space covering design [49]. The idea is to construct a discrete set $\mathcal{Z} = \{z_1, \ldots, z_P\}$ that solves approximately

$$\max_{\mathcal{Z} \in \mathcal{C}^P} \min_{p' \neq p} \|x_p - x_{p'}\|_2. \tag{12}$$

In words, we want the minimal distance between pairs of points in $\mathcal{Z}$ to be as large as possible. This problem is known to be hard. In this work we used the recent solver proposed in [20] together with the Faiss library [34].

The evaluation routine Evaluating the cost function (6) is not an easy task. For just one realization of the noise n and image x_k, we need a fast reconstruction method and a fast way to evaluate the non uniform Fourier transform. The technical details are provided in the appendix A. Second, K might be very large. For instance, the fastMRI knee training database contains more than 30 000 slices of size 320×320. Hence, it is impossible to compute the complete function and it is necessary to either pick a random, but otherwise fixed subset of the images, or to consider random batches that would vary from one iteration to the next. A similar comment holds for the noise term n.

While Bayesian optimization allows the use of random functions, it requires evaluating integrals with Monte-Carlo methods, which is computationally costly . Hence, in all the forthcoming experiments, we will fix a subset of K images. In practice, we observed that using random batches increases the computational load without offering perceptible advantages.

The interpolation process In Bayesian optimization, a Gaussian process is used to model the underlying unknown function. This random process models both the function and the uncertainty associated with each prediction. This uncertainty is related to the fact that the function F is evaluated only at a finite number of points hence leading to an unknown behavior when getting distant from the samples. It is also related to the fact that the function evaluations might be noisy. Every sampled point has a zero variance when using a fixed realization or a low variance when using random noise and batches. The variance increases with the distance between the sampled points.

In our experiments, the Gaussian process is constructed using a Matern kernel of parameter 5/2, which is a popular choice for dimensions in the range $[5, 20]$. It is defined as

$$\Phi(z_1, z_2) = \left(1 + \frac{\sqrt{5}\|z_1 - z_2\|_2}{\nu} + \frac{5\|z_1 - z_2\|_2^2}{3\nu^2}\right) \exp\left(\frac{-\sqrt{5}\|z_1 - z_2\|_2}{\nu}\right),$$

where ν is a scaling parameter that controls the smoothness of the interpolant and its pointwise variance. In practice, the value of ν is a parameter that is optimized at each iteration when fitting the Gaussian process to the sampled data.

The interpolant mean and its variance are then constructed by solving a linear system constructed using the kernel Φ and the sampled points $z_1, \dots, z_P$. We refer to [23] for more details.

Sampling new points Bayesian optimization works by iteratively sampling new points. The point in the sampling set with lowest function value, is an approximation of the minimizer. To choose a new point, there is a trade-off between finding a better minimizer in the neighborhood of this point and space exploration. Indeed, big gaps in between the samples could hide a better minimizer. This trade-off is managed through a so-called utility function. In this work, we chose the expected improvement [23], resulting in a new function $\mathcal{L}(z)$. The new sampled point is found by solving a constrained non-convex problem:

$$\inf_{z \in \mathcal{C}} \mathcal{L}(z)$$

Since the function $\mathcal{L}$ is non-convex, we use a multi-start strategy. We first sample 1000 points evenly in $\mathcal{C}$ using a maximin design. Then, we launch many projected gradient descents on $\mathcal{L}$ in parallel, starting from those points. The best critical point is chosen and added as a new sample.

This process requires projecting z on $\mathcal{C}$ defined in (10). To this end, we designed an efficient first order solver.

3 Numerical experiments and results

3.1 The experimental setting

Database and computing power Throughout this section, we used the fastMRI database [61]. It contains MRI images of size 320×320. We focused on the single coil and fully sampled knee images. The training set is composed of 973 3D volumes, which represents a total of 34 742 slices. The validation set has 199 volumes and 7135 slices.

Some images in the dataset have a significant amount of noise. This presents two significant drawbacks: i) the signal-to-noise-ratio of the reconstructed images is increased artificially and ii) we have shown that noise can dramatically impact the convergence of off-the-grid Fourier sampling optimization [26]. To mitigate these effects, we pre-processed all the slices using a non-local mean denoising algorithm [13].

The experiments are conducted on the Jean-Zay HPC facility. For each task we use 10 cores and an Nvidia Tesla V100 with 16GB of memory.

Sampling The bounds of the constraint sets in (3) are given by:

$$\alpha = \Delta t \gamma \frac{G_{max}}{K_{max}} \quad \text{and} \quad \beta = \Delta t^2 \gamma \frac{S_{max}}{K_{max}}, \tag{13}$$

where Δt is the sampling step of the scanner. Following [16], we used the following realistic hardware constraints: $G_{max} = 40\text{mT/m}$, $S_{max} = 180\text{T/m/s}$, $K_{max} = 2\pi$ and $\gamma = 42.57\text{MHz/T}$. The value of Δt is fixed to ensure that at maximal speed, the distance between two consecutive points equals the Shannon-Nyquist rate [39].

We consider two different scenarii: 25% and 10% undersampling. Each shot consists of 646 acquisition points and we use $N_s = 40$ shots and $N_s = 16$ shots respectively for the 25% and the 10% undersampling. Each shot is constrained to start at the center of the k-space. The first few points of each trajectory are fixed to be radial, see Appendix A.5.4 for the technical details.

The family of densities is generated using the process described in Section 2.3.3 with 10^4 densities generated at random.

Sampling baseline All the optimized schemes are compared to a state-of-the-art handcrafted baseline: the SPARKLING method described in [40]. There, the attraction-repulsion problem (11) is solved with a radial density ρ. Its value at the center has been optimized to yield the best possible signal-to-noise ratio on the validation set in a way similar to [15]. The corresponding point configuration is given in Fig. 6a and Fig. 6b for the 25% and 10% undersampling rates respectively. It provides a 7dB improvement compared to the usual radial lines commonly found in the literature (see the first two rows of Table 2).

Image reconstruction The experiments are conducted with two reconstruction models:

- a total varation reconstruction method with 120 iterations of Algorithm 1 in Appendix A.1 and with a regularization parameter $\lambda = 10^2$ and,
- an unrolled network (NN) with 6 iterations of ADMM and a DruNet as the denoising step [62], 30 iterations of the CG algorithm that initializes the ADMM and 10 iterations of CG to solve the data-consistency equations at each iteration.

3.2 Choosing a kernel for the discrepancy

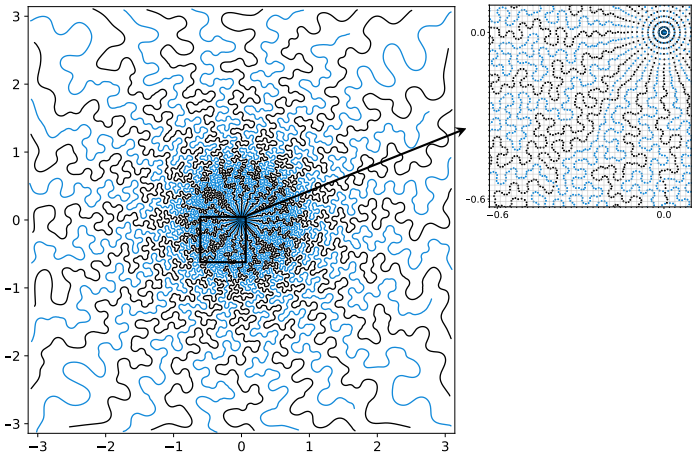
In all the previous "SPARKLING" papers [40, 18], the kernel function $h(x) = \|x\|_2$ was used. This choice seems like the most natural alternative since it is the only one which is scale invariant in the unconstrained setting. This means that if $\Xi = (\mathbb{R}^D)^M$ and if a density ρ is dilated by a certain factor, then so is the optimal sampling scheme. However, this property is not true anymore when constraints are added. In that case, the choice of kernel turns out to be of importance.

To illustrate this fact, we considered the three different radial kernels $h(x) = \|x\|_2$, $h(x) = \sqrt{\|x\|_2}$ and $h(x) = \log(\|x\|_2)$. As can be seen on Fig. 5, performance variations of more than 0.2dB are obtained depending on the kernel. The reason is that contiguous points on the trajectories are spaced more or less depending on this choice. For instance, observe that the points on the zoom of Fig. 5a are more packed along the trajectories that on Fig. 5c. To compensate for this higher longitudinal density, the sampler then increases the distance between adjacent trajectories, thereby creating holes in the sampling set. This is detrimental, since low frequency information is lost in the process. This effect can be quantified by evaluating the distances between contiguous points in the k-space center. As can be seen, it goes from 0.52 for the usual kernel $h(x) = \|x\|_2$ to a significantly higher value 0.72 for the logarithmic kernel. The latter kernel creates a higher repulsion for neighboring points.

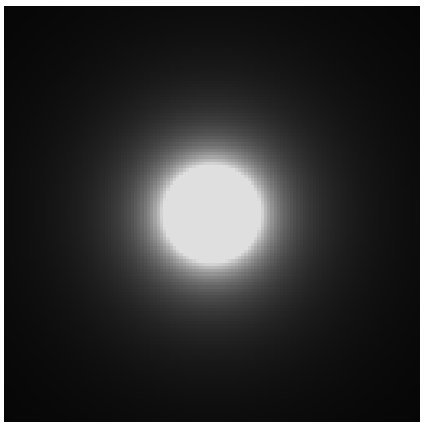
3.3 Bayesian optimization: database size and numerical complexity

In this section, we aim at evaluating the computational complexity of the Bayesian optimization routine. To this end, we study the impact of the number of images K in the dataset, the size of the initial sampling set and the number of iterations, which are governing the algorithm's complexity. Table 1 summarizes our main findings for the total variation reconstruction and unrolled neural network.

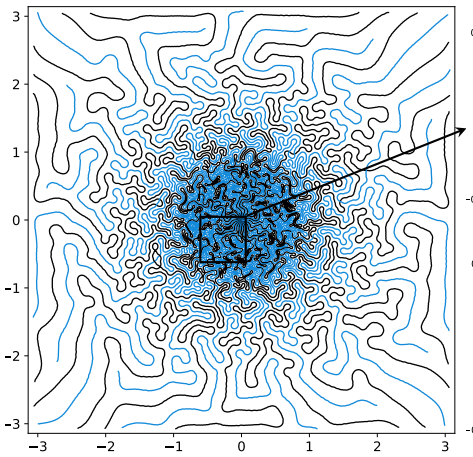
There, we see that the number of images K in the dataset has nearly no influence on the quality of the final sampling density. Taking $K = 32$ or $K = 512$ images yields an identical PSNR on the validation set. This holds both for the 25% and 10% undersampling rates. As can be seen in the Tables, reconstructing as little as 200×32 images is enough to reach the best possible density in the family. The same conclusion holds for the 10% undersampling rate. This represents 18% of a single epoch.



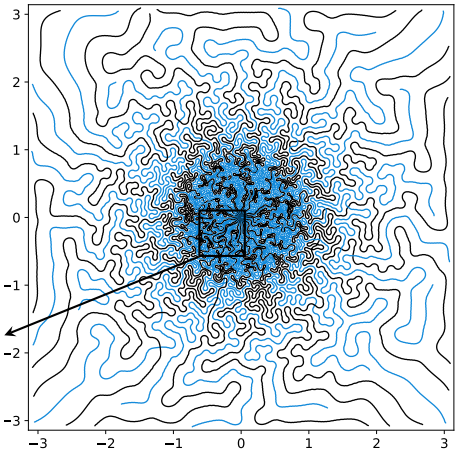
(a) $h(x) = \|x\|_2$ — 35.10 ± 3.70dB
$d(\xi) = 0.52$



(b) Target radial density



(c) $h(x) = \log(\|x\|_2)$ — 35.35 ± 3.78dB
$d(\xi) = 0.72$



(d) $h(x) = \sqrt{\|x\|_2}$ — 35.32 ± 3.78dB
$d(\xi) = 0.69$

Figure 5: On the importance of the discrepancy's kernel $h(x)$. The same density is sampled with different kernels. The average PSNR of the reconstructed images on the validation set is displayed with its standard deviation. The average distance between contiguous points on the trajectories is displayed as $d(\xi)$.

We also see that the initial sampling set of the convex $\mathcal{C}$ plays a marginal role on the quality of the final result. In addition, taking a small number of initial points allows to reduce the overall complexity of the algorithm to reach a given PSNR.

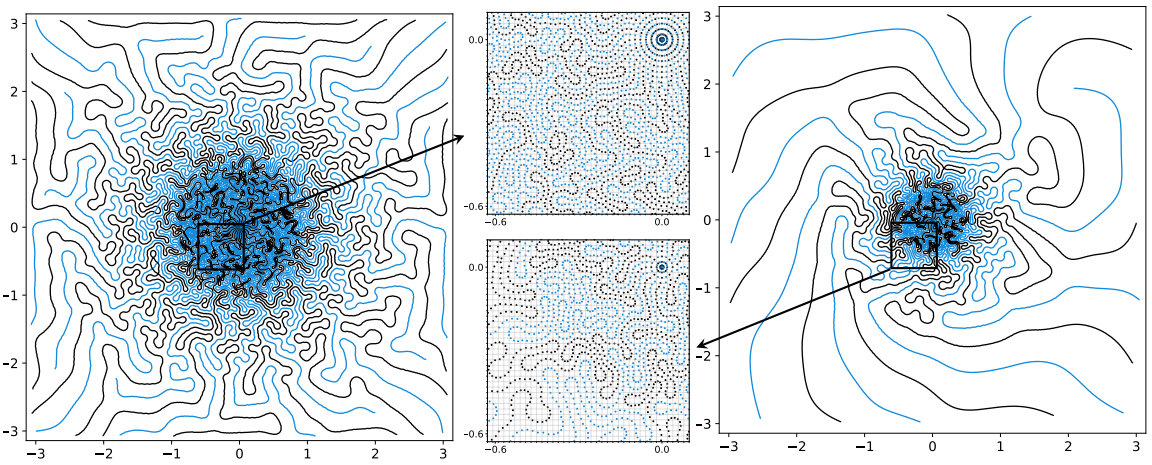
3.4 Comparing optimization routines for the total variation reconstructor

In what follows, we aim at comparing two different sampling optimization approaches:

Traj. optim. The minimization of (4) in the space of trajectories. We use a modified version of the multi-scale approach in [58], see Appendix A.5.1.

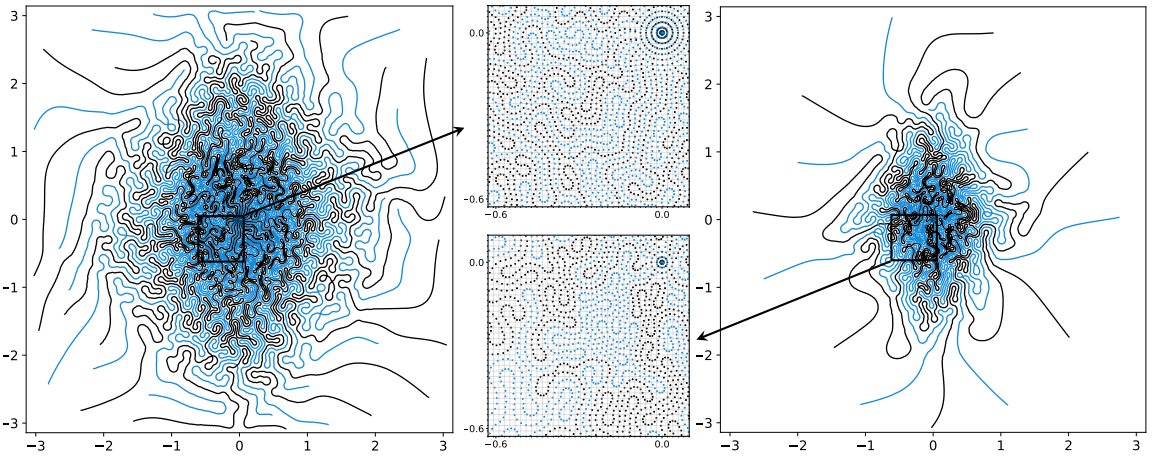
BO density The Bayesian approach to minimize (6) globally.

To compare these approaches, we conduct various experiments. The corresponding results are shown in Table 3, Table 2 and Fig 6. Below, we summarize our main findings.



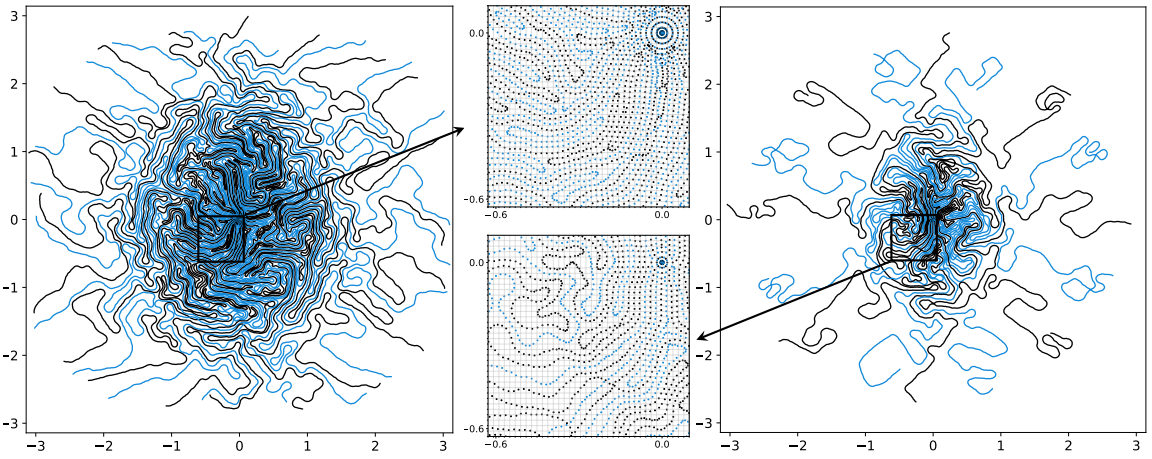
(a) Baseline radial 25%

(b) Baseline radial 10%



(c) Bayesian density optimization 25%

(d) Bayesian density optimization 10%



(e) Trajectory optimization 25%

(f) Trajectory optimization 10%

Figure 6: Optimized sampling schemes with various optimization approaches. A total variation reconstruction algorithm is used.

Method	# init. points	# evaluations	$K = 32$ images	$K = 128$ images	$K = 512$ images
TV	20	200	35.64 ± 3.82	35.65 ± 3.82	35.65 ± 3.82
TV	100	300	35.63 ± 3.81	35.66 ± 3.82	35.66 ± 3.82
TV	200	300	35.65 ± 3.81	35.66 ± 3.82	35.66 ± 3.82
NN	20	200	38.14 ± 4.77	38.10 ± 4.75	38.09 ± 4.76
NN	100	300	38.17 ± 4.79	38.05 ± 4.73	38.08 ± 4.75
NN	200	300	38.20 ± 4.80	38.08 ± 4.75	38.10 ± 4.76

Table 1: Bayesian optimization on a convex set $\mathcal{C}$ of dimension $L = 20$ using a total variation reconstruction algorithm and an unrolled network for 25% undersampling. The PSNR is evaluated for the optimized density on the validation dataset containing 7135 images. The total number of cost function evaluations is given in the second column.

Qualitative comparison of the sampling schemes In this paragraph, we compare our method with existing works [58, 59]. The optimized sampling schemes are shown in Fig. 6 for the TV reconstructor. In Fig. 6, we see the results of the different optimization routines.

The two optimization methods yield anisotropic sampling schemes with a higher density along the vertical axis. However the trajectories present significant differences.

The Bayesian optimization yields a sampling scheme which covers the space more uniformly. The trajectories have a significantly higher curvature at the k-space center. These features are somehow hard-coded within the sampling generator $\mathcal{S}_M$ described in Section 2.3.4.

The trajectory optimization yields trajectories which are locally linear and aligned at a distance of about a pixel. This suggests that the trajectory optimization favors Shannon's sampling rate at the center of the k-space. A potential explanation is as follows. When the sampling points are close to a subgrid [26], the adjoint of the forward operator $A(\xi)^*$ is roughly the pseudo-inverse. Using a points configuration close to a subgrid therefore helps iterative reconstruction algorithms to converge.

Finally, at the bottom-left of the zoomed region on the 25% undersampling rate, it seems that Bayesian optimization (Fig. 6c) yields a density slightly higher than trajectory optimization (Fig. 6e). This density is critical for the reconstruction quality and might explain a part of the quantitative differences observed in the next section.

Performance comparison Table 2 reveals that the trajectory optimization yields better performance than the Bayesian optimization approach both for the 25% (+0.26dB) and 10% (+0.07dB) undersampling rates. This was to be expected since the density optimization is much more constrained. The difference is however mild.

We also report some of the best (resp. worst) PSNR increase (resp. decrease) in Fig. 7. For each case, we selected 3 images that are representative among the top 10 best (resp. worst) images of the test dataset. For both the trajectory optimization and the Bayesian optimization method, the images that have the largest PSNR increase have large vertical structures. This increase might be due to the anisotropy of the optimized schemes that are more adapted to the center slices of the 3D knee images. On the contrary, the images having the largest PSNR decrease are outliers that are not prevalent in the dataset such as extreme slices. Notice that images that have the best (resp. worst) PSNR increase (resp. decrease) are the same for the Bayesian optimization and for the trajectory optimization.

Method	25%	10%
Radial scheme	27.87 ± 2.75dB 0.66 ± 0.12	24.28 ± 2.67dB 0.57 ± 0.12
Sparkling radial (baseline)	35.35 ± 3.78dB 0.85 ± 0.11	32.94 ± 3.20dB 0.79 ± 0.14
Bayesian optim. $K = 32$	35.66 ± 3.82dB (+0.31dB) 0.86 ± 0.11	33.41 ± 3.26dB (+0.47dB) 0.80 ± 0.14
Trajectory optim. $K = 34742$	35.92 ± 3.89dB (+0.57dB) 0.87 ± 0.11	33.48 ± 3.31dB (+0.54dB) 0.80 ± 0.14
Trajectory optim. $K = 32$	35.67 ± 3.88dB (+0.32dB) 0.86 ± 0.11	32.84 ± 3.19dB (-0.10dB) 0.79 ± 0.14
Trajectory optim. $K = 128$	35.67 ± 3.85dB (+0.32dB) 0.86 ± 0.11	32.89 ± 3.17dB (-0.05dB) 0.79 ± 0.14

Table 2: Comparison of different optimization procedures for the TV reconstructor with different numbers of images K in the training set. For each test case, the first line is the PSNR and the second line is the SSIM. The number after $\pm$ indicates the standard deviation.

Computing times Table 3 gives the computation times for each method with the total variation reconstruction method. The proposed approach has the significant advantage of giving an optimized sampling scheme with guarantees on the underlying density with a reduced computational budget and with a reduced number of images. As can be seen, our approach requires only 32 images and 3 hours. This has to be compared to the 85 hours (3 days and a half) needed by the trajectory optimization routine.

This feature is a significant advantage of our approach. It could be key element when targeting high resolution images or 3D data.

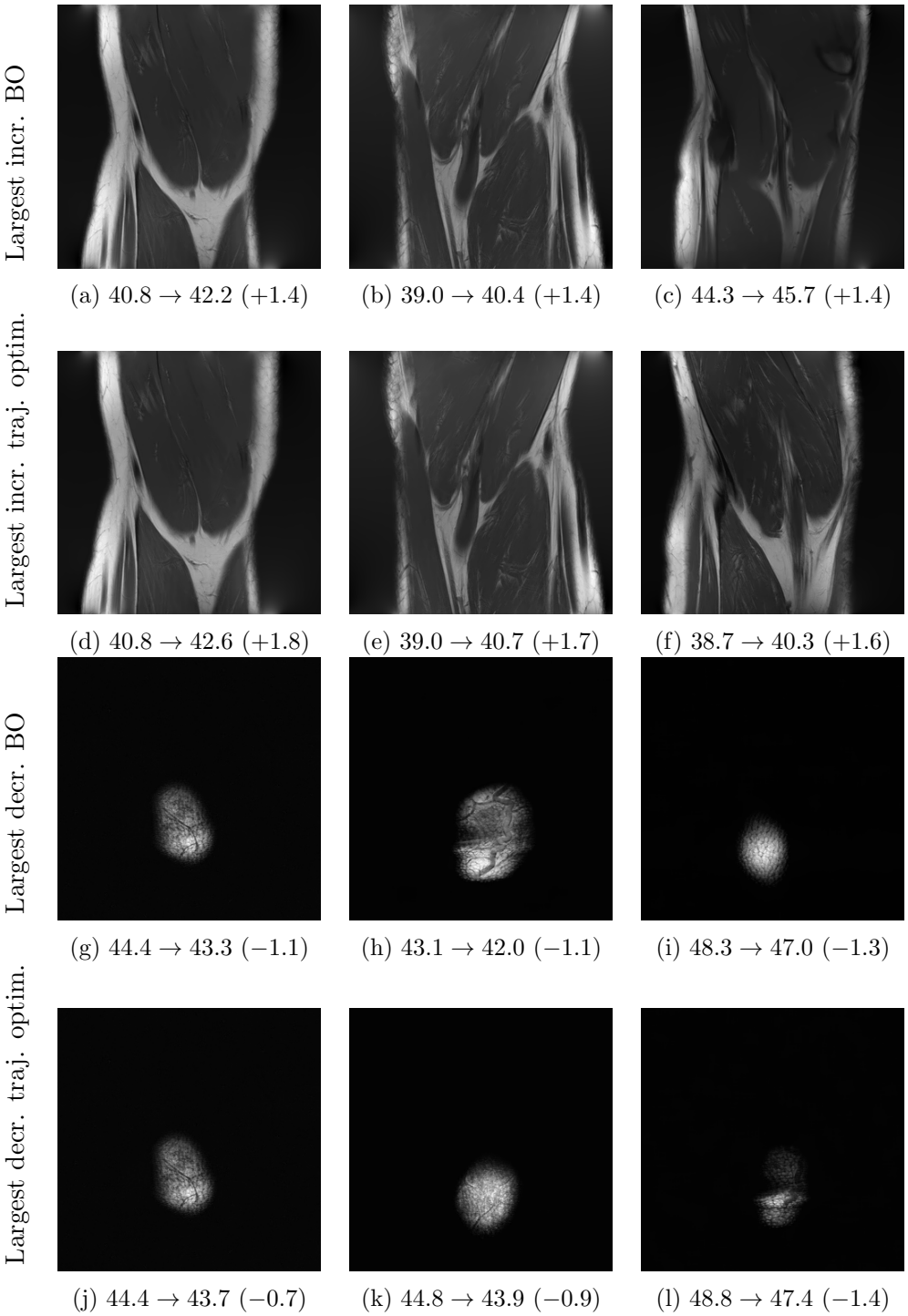
Method	Computational time
Trajectory optimization	85h
Bayesian optimization	
Optimization $K = 32$	3h
Optimization $K = 128$	4h

Table 3: Computational of the different optimization procedures (25% undersampling and TV reconstructor) with an NVIDIA Quadro RTX 5000 GPU.

Size of the training set As advertised, the Bayesian optimization approach works even for small datasets. The trajectory optimization routine also provides competitive results with only 32 images in the training set. However, the performance collapses for the 10% undersampling rate. Increasing the size of the training set to $K = 128$ does not improve the situation. This feature is in strong favor of our approach, when having access to a limited dataset.

3.5 Comparing optimization routines for a neural network reconstructor

The aim of this section, is to compare three different sampling optimizers:



(a) 40.8 → 42.2 (+1.4) (b) 39.0 → 40.4 (+1.4) (c) 44.3 → 45.7 (+1.4)

(d) 40.8 → 42.6 (+1.8) (e) 39.0 → 40.7 (+1.7) (f) 38.7 → 40.3 (+1.6)

(g) 44.4 → 43.3 (−1.1) (h) 43.1 → 42.0 (−1.1) (i) 48.3 → 47.0 (−1.3)

(j) 44.4 → 43.7 (−0.7) (k) 44.8 → 43.9 (−0.9) (l) 48.8 → 47.4 (−1.4)

Figure 7: Sample of images that have the largest increase (resp. decrease) of the PSNR for the different optimization methods with the TV reconstructor and 25% undersampling. The numbers below the images are the PSNR using the baseline sampling scheme and the PSNR using optimized trajectories.

Method	25%	10%
Baseline with unrolled net	37.26 ± 4.57dB 0.89 ± 0.09	34.49 ± 3.71dB 0.83 ± 0.13
BO scheme $K = 128$ with unrolled net	38.20 ± 4.80dB (+0.94dB) 0.91 ± 0.08	35.13 ± 3.92dB (+0.64dB) 0.86 ± 0.13
Traj. optim. with fixed unrolled net	39.09 ± 5.06dB (+1.83dB) 0.92 ± 0.07	35.65 ± 4.18dB (+1.16dB) 0.85 ± 0.12
Joint optim. multi-scale	39.03 ± 4.87dB (+1.77dB) 0.92 ± 0.07	35.53 ± 4.05dB (+1.04dB) 0.85 ± 0.12

Table 4: Comparison of different optimization procedures for the unrolled ADMM reconstructor. For each test case, the first line is the PSNR and the second line is the SSIM. The increase compared to the baseline scheme is shown in parentheses.

- The Bayesian density optimization solver proposed in this paper.
- The trajectory optimization solver with a fixed unrolled neural network trained on a family of sampling schemes, see Appendix A.3. This is a novelty of this paper.
- An optimization routine minimizing the trajectories and the unrolled network weights simultaneously, as proposed in [58, 59].

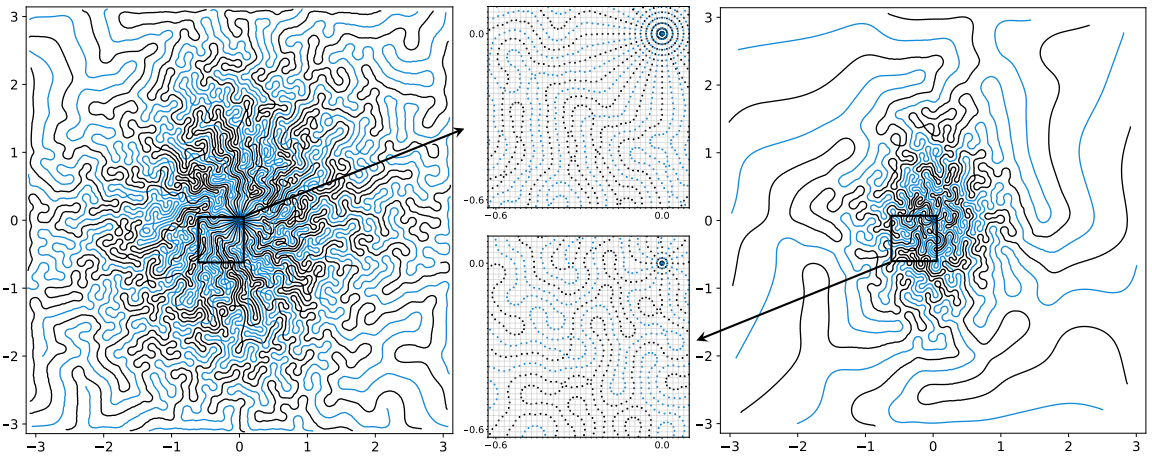
Qualitative comparisons The differences between the density optimization and the trajectory optimization can be observed on Fig. 8. They are much more pronounced that for the total variation reconstructor. Surprisingly, the trajectory optimized sampling schemes leave large portions of the low frequencies unexplored. Hence, it seems that the unrolled network is able to infer low frequency information better than the traditional total variation prior. This suggests that the existing compressed sampling theories designed for the Fourier-Wavelet system have to be revised significantly to account for the progress in neural network reconstructions. The optimization of a trajectory for a fixed sampling scheme or the joint optimization yield qualitatively similar trajectories, with perhaps larger unexplored parts of the k-space for the fixed reconstruction method.

Quantitative comparisons Table 4 allows comparing the different methods quantitatively. BO yields a PSNR increase almost twice lower than the multi-scale optimization (+0.94dB VS +1.83dB for 25% and +0.64dB VS +1.16dB for 10%). This can likely be explained by the fact that the chosen family of densities (dimension 20) is unable to reproduce the complexity of the optimized trajectories. It is possible that richer sampling densities could reduce the gap between both approaches. However, Bayesian optimization is known to work only in small dimension and it is currently unclear how to extend the method to this setting.

Interestingly, the trajectory optimized with a fixed unrolled neural network trained on a family provides slightly better results ($\approx$ +0.1dB) than the joint optimization. This suggests that the joint optimization gets trapped in a local minimizer since it can only be better if optimized jointly with the reconstructor.

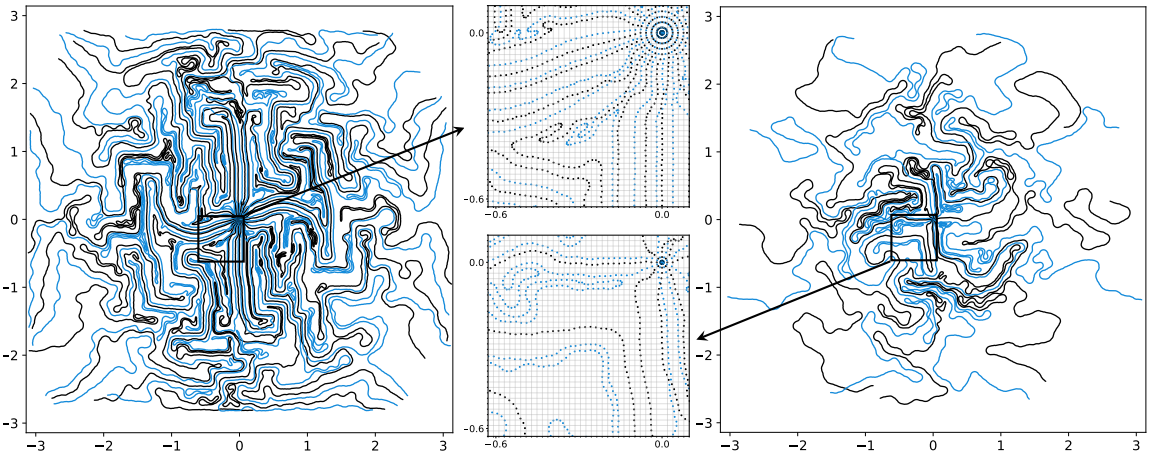
4 Conclusion

In this work, we designed efficient optimization algorithms that either optimize trajectories directly or learn a sampling density and an associated sampling pattern in MRI. Overall,



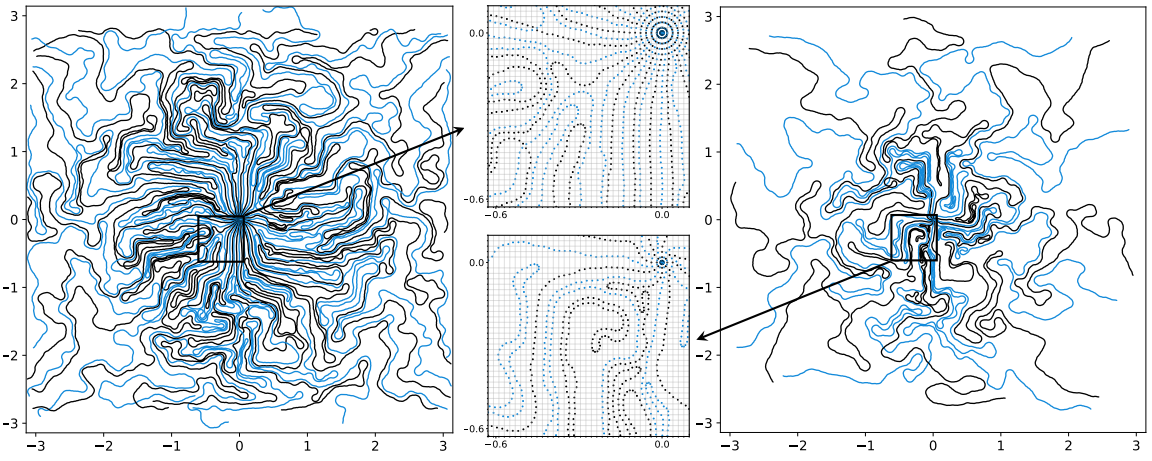
(a) Bayesian density optim. 25%

(b) Bayesian density optim. 10%



(c) Joint traj. optim. 25%

(d) Joint traj. optim. 10%



(e) Traj. optim. with fixed net 25%

(f) Traj. optim. with fixed net 10%

Figure 8: Optimized sampling schemes with the various optimization approaches for a neural network reconstruction.

the main highlights of this work are:

- The compressed sensing theories designed for the Fourier-Wavelet system with ℓ^1 reconstruction (e.g. [2]) seem nearly optimal from an experimental point of view. Sampling schemes can be designed based on a density that is close to Shannon's rate at the k-space center and that decay towards the high frequencies. The precise shape of the density depends on the images structure.
- In that context, the Bayesian optimization of densities is an attractive method to design sampling schemes. It works with small datasets, ensures the convergence to a global minimizer. Its performance is close to much heavier trajectory optimizers and is from one to two orders of magnitude faster.
- In the case of unrolled neural network reconstructions, the proposed Bayesian optimization framework is still interesting with gains of up to 1dB in average on the fastMRI knee validation set. However, the gain can be nearly doubled with a direct optimization of the trajectories. A possible explanation for this fact is that the family of densities is too poor to describe the best convoluted trajectories.
- We also improved the Sparkling trajectories [40], by changing the discrepancies.
- We also provided various improvements to the direct optimization of trajectories by using the Extra-Adam algorithm to handle hard constraints and by training reconstruction networks on families of operators.

Acknowledgments

P. Weiss and F. de Gournay were supported by the ANR JCJC Optimization on Measures Spaces ANR-17-CE23-0013-01 and the ANR-3IA Artificial and Natural Intelligence Toulouse Institute. This work was granted access to the HPC resources of IDRIS under the allocation AD011012210 made by GENCI.

References

[1] Ben Adcock, Claire Boyer, and Simone Brugiapaglia. On oracle-type local recovery guarantees in compressed sensing. Information and Inference: A Journal of the IMA, 2020.

[2] Ben Adcock and Anders C Hansen. Compressive Imaging: Structure, Sampling, Learning. Cambridge University Press, 2021.

[3] Ben Adcock, Anders C Hansen, Clarice Poon, and Bogdan Roman. Breaking the coherence barrier: A new theory for compressed sensing. In Forum of Mathematics, Sigma, volume 5. Cambridge University Press, 2017.

[4] Hemant Kumar Aggarwal and Mathews Jacob. J-modl: Joint model-based deep learning for optimized sampling and reconstruction. IEEE Journal of Selected Topics in Signal Processing, 2020.

[5] CB Ahn, JH Kim, and ZH Cho. High-speed spiral-scan echo planar NMR imaging-I. IEEE Transactions on Medical Imaging, 5(1):2–7, 1986.

[6] Cagla Bahadir, Alan Wang, Adrian Dalca, and Mert R Sabuncu. Deep-learning-based optimization of the under-sampling pattern in mri. IEEE Transactions on Computational Imaging, 2020.

[7] Maximilian Balandat, Brian Karrer, Daniel R. Jiang, Samuel Daulton, Benjamin Letham, Andrew Gordon Wilson, and Eytan Bakshy. BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization. In Advances in Neural Information Processing Systems 33, 2020.

[8] Luca Baldassarre, Yen-Huan Li, Jonathan Scarlett, Baran Gözcü, Ilija Bogunovic, and Volkan Cevher. Learning-based compressive subsampling. IEEE Journal of Selected Topics in Signal Processing, 10(4):809–822, 2016.

[9] Matt A Bernstein, Kevin F King, and Xiaohong Joe Zhou. Handbook of MRI pulse sequences. Elsevier, 2004.

[10] Martin Blaimer, Felix Breuer, Matthias Mueller, Robin M Heidemann, Mark A Griswold, and Peter M Jakob. Smash, sense, pils, grappa: how to choose the optimal method. Topics in Magnetic Resonance Imaging, 15(4):223–236, 2004.

[11] Claire Boyer, Jérémie Bigot, and Pierre Weiss. Compressed sensing with structured sparsity and structured acquisition. Applied and Computational Harmonic Analysis, 46(2):312–350, 2019.

[12] Claire Boyer, Nicolas Chauffert, Philippe Ciuciu, Jonas Kahn, and Pierre Weiss. On the generation of sampling schemes for magnetic resonance imaging. SIAM Journal on Imaging Sciences, 9(4):2039–2072, 2016.

[13] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. Non-local means denoising. Image Processing On Line, 1:208–212, 2011.

[14] Emmanuel J Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. IEEE Transactions on information theory, 52(2):489–509, 2006.

[15] GR Chaithya, Zaccharie Ramzi, and Philippe Ciuciu. Learning the sampling density in 2d sparkling mri acquisition for optimized image reconstruction. In 2021 29th European Signal Processing Conference (EUSIPCO), pages 960–964. IEEE, 2021.

[16] GR Chaithya, Pierre Weiss, Guillaume Daval-Frérot, Aurélien Massire, Alexandre Vignaud, and Philippe Ciuciu. Optimizing full 3d sparkling trajectories for high-resolution magnetic resonance imaging. IEEE Transactions on Medical Imaging, 2022.

[17] Benjamin Charlier, Jean Feydy, Joan Glaunès, François-David Collin, and Ghislain Durif. Kernel operations on the gpu, with autodiff, without memory overflows. Journal of Machine Learning Research, 22(74):1–6, 2021.

[18] Nicolas Chauffert, Philippe Ciuciu, Jonas Kahn, and Pierre Weiss. A projection method on measures sets. Constructive Approximation, 45(1):83–111, 2017.

[19] Nicolas Chauffert, Pierre Weiss, Jonas Kahn, and Philippe Ciuciu. A projection algorithm for gradient waveforms design in magnetic resonance imaging. IEEE transactions on medical imaging, 35(9):2026–2039, 2016.

[20] Valentin Debarnot and Pierre Weiss. Deep-blur: Blind identification and deblurring with convolutional neural networks. 2022.

[21] Hans G Feichtinger and Karlheinz Gröchenig. Theory and practice of irregular sampling. Wavelets: mathematics and applications, 1994:305–363, 1994.

[22] Jeffrey A Fessler and Bradley P Sutton. Nonuniform fast fourier transforms using min-max interpolation. IEEE transactions on signal processing, 51(2):560–574, 2003.

[23] Peter I Frazier. A tutorial on bayesian optimization. arXiv preprint arXiv:1807.02811, 2018.

[24] Martin Genzel, Ingo Gühring, Jan Macdonald, and Maximilian März. Near-exact recovery for tomographic inverse problems via deep learning. In International Conference on Machine Learning, pages 7368–7381. PMLR, 2022.

[25] Gauthier Gidel, Hugo Berard, Gaëtan Vignoud, Pascal Vincent, and Simon Lacoste-Julien. A variational inequality perspective on generative adversarial networks. arXiv preprint arXiv:1802.10551, 2018.

[26] Alban Gossard, Frédéric de Gournay, and Pierre Weiss. Spurious minimizers in non uniform fourier sampling optimization. Inverse Problems, 2022.

[27] Alban Gossard and Pierre Weiss. Training adaptive reconstruction networks for inverse problems. arXiv preprint arXiv:2202.11342, 2022.

[28] Baran Gözcü, Rabeeh Karimi Mahabadi, Yen-Huan Li, Efe Ilıcak, Tolga Çukur, Jonathan Scarlett, and Volkan Cevher. Learning-based compressive mri. IEEE transactions on medical imaging, 37(6):1394–1406, 2018.

[29] Manuel Gräf, Daniel Potts, and Gabriele Steidl. Quadrature errors, discrepancies, and their relations to halftoning on the torus and the sphere. SIAM Journal on Scientific Computing, 34(5):A2760–A2791, 2012.

[30] Kerstin Hammernik, Teresa Klatzer, Erich Kobler, Michael P Recht, Daniel K Sodickson, Thomas Pock, and Florian Knoll. Learning a variational network for reconstruction of accelerated mri data. Magnetic resonance in medicine, 79(6):3055–3071, 2018.

[31] John I Jackson, Dwight G Nishimura, and Albert Macovski. Twisting radial lines with application to robust magnetic resonance imaging of irregular flow. Magnetic Resonance in Medicine, 25(1):128–139, 1992.

[32] Mathews Jacob, Jong Chul Ye, Leslie Ying, and Mariya Doneva. Computational mri: Compressive sensing and beyond [from the guest editors]. IEEE Signal Processing Magazine, 37(1):21–23, 2020.

[33] Kyong Hwan Jin, Michael Unser, and Kwang Moo Yi. Self-supervised deep active accelerated mri. arXiv preprint arXiv:1901.04547, 2019.

[34] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. IEEE Transactions on Big Data, 7(3):535–547, 2019.

[35] Anatoli Juditsky, Arkadi Nemirovski, and Claire Tauvel. Solving variational inequalities with stochastic mirror-prox algorithm. Stochastic Systems, 1(1):17–58, 2011.

[36] Jens Keiner, Stefan Kunis, and Daniel Potts. Using nfft 3—a software library for various nonequispaced fast fourier transforms. ACM Transactions on Mathematical Software (TOMS), 36(4):1–30, 2009.

[37] Florian Knoll, Christian Clason, Clemens Diwoky, and Rudolf Stollberger. Adapted random sampling patterns for accelerated mri. Magnetic resonance materials in physics, biology and medicine, 24(1):43–50, 2011.

[38] Florian Knoll, Tullie Murrell, Anuroop Sriram, Nafissa Yakubova, Jure Zbontar, Michael Rabbat, Aaron Defazio, Matthew J Muckley, Daniel K Sodickson, C Lawrence Zitnick, et al. Advancing machine learning for mr image reconstruction with an open competition: Overview of the 2019 fastmri challenge. Magnetic Resonance in Medicine, 2020.

[39] Carole Lazarus, Maximilian März, and Pierre Weiss. Correcting the side effects of adc filtering in mr image reconstruction. Journal of Mathematical Imaging and Vision, pages 1–14, 2020.

[40] Carole Lazarus, Pierre Weiss, Nicolas Chauffert, Franck Mauconduit, Loubna El Gueddari, Christophe Destrieux, Ilyess Zemmoura, Alexandre Vignaud, and Philippe Ciuciu. Sparkling: variable-density k-space filling curves for accelerated t2*-weighted mri. Magnetic resonance in medicine, 81(6):3643–3661, 2019.

[41] Léo Lebrat, Frédéric de Gournay, Jonas Kahn, and Pierre Weiss. Optimal transport approximation of 2-dimensional measures. SIAM Journal on Imaging Sciences, 12(2):762–787, 2019.

[42] Michael Lustig, David L Donoho, Juan M Santos, and John M Pauly. Compressed sensing mri. IEEE signal processing magazine, 25(2):72–82, 2008.

[43] Michael Lustig, Jin Hyung Lee, David L Donoho, and John M Pauly. Faster imaging with randomly perturbed, under-sampled spirals and ℓ_1 reconstruction. In Proceedings of the 13th annual meeting of ISMRM, page 685, Miami Beach, FL, USA, 2005.

[44] M. J. Muckley, R. Stern, T. Murrell, and F. Knoll. TorchKbNufft: A high-level, hardware-agnostic non-uniform fast fourier transform. In ISMRM Workshop on Data Sampling & Image Reconstruction, 2020.

[45] Matthew J Muckley, Bruno Riemenschneider, Alireza Radmanesh, Sunwoo Kim, Geunu Jeong, Jingyu Ko, Yohan Jun, Hyungseob Shin, Dosik Hwang, Mahmoud Mostapha, et al. State-of-the-art machine learning mri reconstruction in 2020: Results of the second fastmri challenge. arXiv preprint arXiv:2012.06318, 2020.

[46] Yurii E Nesterov. A method for solving the convex programming problem with convergence rate o (1/kˆ 2). In Dokl. akad. nauk Sssr, volume 269, pages 543–547, 1983.

[47] Michael K Ng, Pierre Weiss, and Xiaoming Yuan. Solving constrained total-variation image restoration and reconstruction problems via alternating direction methods. SIAM journal on Scientific Computing, 32(5):2710–2736, 2010.

[48] Peter Ochs, René Ranftl, Thomas Brox, and Thomas Pock. Bilevel optimization with non-smooth lower level problems. In International Conference on Scale Space and Variational Methods in Computer Vision, pages 654–665. Springer, 2015.

[49] Luc Pronzato. Minimax and maximin space-filling designs: some properties and methods for construction. Journal de la Société Française de Statistique, 158(1):7–36, 2017.

[50] Peter B Roemer, William A Edelstein, Cecil E Hayes, Steven P Souza, and Otward M Mueller. The nmr phased array. Magnetic resonance in medicine, 16(2):192–225, 1990.

[51] Thomas Sanchez, Baran Gözcü, Ruud B van Heeswijk, Armin Eftekhari, Efe Ilıcak, Tolga Çukur, and Volkan Cevher. Scalable learning-based sampling optimization for compressive dynamic mri. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 8584–8588. IEEE, 2020.

[52] Christian Schmaltz, Pascal Gwosdek, Andrés Bruhn, and Joachim Weickert. Electrostatic halftoning. In Computer Graphics Forum, volume 29, pages 2313–2327. Wiley Online Library, 2010.

[53] Franz Schmitt, Michael K Stehling, and Robert Turner. Echo-planar imaging: theory, technique and application. Springer Science & Business Media, 2012.

[54] Ferdia Sherry, Martin Benning, Juan Carlos De los Reyes, Martin J Graves, Georg Maierhofer, Guy Williams, Carola-Bibiane Schönlieb, and Matthias J Ehrhardt. Learning the sampling pattern for mri. IEEE Transactions on Medical Imaging, 2020.

[55] Yu-hsuan Shih, Garrett Wright, Joakim Andén, Johannes Blaschke, and Alex H Barnett. cufinufft: a load-balanced gpu library for general-purpose nonuniform ffts. In 2021 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), pages 688–697. IEEE, 2021.

[56] SS Vasanawala, MJ Murphy, Marcus T Alley, P Lai, Kurt Keutzer, John M Pauly, and Michael Lustig. Practical parallel imaging compressed sensing mri: Summary of two years of experience in accelerating body mri of pediatric patients. In 2011 IEEE international symposium on biomedical imaging: From nano to macro, pages 1039–1043. IEEE, 2011.

[57] Guanhua Wang and Jeffrey A Fessler. Efficient approximation of jacobian matrices involving a non-uniform fast fourier transform (nufft). arXiv preprint arXiv:2111.02912, 2021.

[58] Guanhua Wang, Tianrui Luo, Jon-Fredrik Nielsen, Douglas C Noll, and Jeffrey A Fessler. B-spline parameterized joint optimization of reconstruction and k-space trajectories (bjork) for accelerated 2d mri. IEEE Transactions on Medical Imaging, 2022.

[59] Tomer Weiss, Ortal Senouf, Sanketh Vedula, Oleg Michailovich, Michael Zibulevsky, and Alex Bronstein. Pilot: Physics-informed learned optimal trajectories for accelerated mri. Journal of Machine Learning for Biomedical Imaging (MELBA), pages 1–23, 2021.

[60] Songhyun Yu, Bumjun Park, and Jechang Jeong. Deep iterative down-up cnn for image denoising. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pages 0–0, 2019.

[61] Jure Zbontar, Florian Knoll, Anuroop Sriram, Tullie Murrell, Zhengnan Huang, Matthew J Muckley, Aaron Defazio, Ruben Stern, Patricia Johnson, Mary Bruno, et al. fastmri: An open dataset and benchmarks for accelerated mri. arXiv preprint arXiv:1811.08839, 2018.

[62] Kai Zhang, Yawei Li, Wangmeng Zuo, Lei Zhang, Luc Van Gool, and Radu Timofte. Plug-and-play image restoration with deep denoiser prior. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021.

[63] Yudong Zhang, Bradley S Peterson, Genlin Ji, and Zhengchao Dong. Energy preserved sampling for compressed sensing mri. Computational and mathematical methods in medicine, 2014, 2014.

[64] Marcelo VW Zibetti, Gabor T Herman, and Ravinder R Regatte. Fast data-driven learning of parallel mri sampling patterns for large scale problems. Scientific Reports, 11(1):1–19, 2021.

A Implementation details

A.1 TV reconstruction algorithm

In this part we detail the TV iterative reconstruction algorithm that is used in this paper. We consider a regularized version of the total variation of the form

$$TV_\epsilon(x) = \sum_{n=1}^{N} \sqrt{\|(\nabla x)[n]\|_2^2 + \epsilon^2}.$$

Given $y \in \mathbb{C}^M$, the solver of problem (2) is given in Algorithm 1. The parameter α drives the acceleration and D is the dimension, here $D = 2$. It corresponds to a Nesterov accelerated gradient descent [46] with a regularized version of the ℓ^1 norm. A critical point is the

Algorithm 1 A TV minimization algorithm

Require: Number of iterations Q.
  Set $z^{(0)} = x^{(0)} = 0$, $\tau = \frac{1}{\|A(\xi)\|_{2\to 2}^2 + 4D\lambda/\epsilon}$.
  for all $q = 0$ to $Q - 1$ **do**
    $r^{(q)} = A(\xi)^*(A(\xi)z^{(q)} - y)$
    $x^{(q+1)} = z^{(q)} - \tau \left[r^{(q)} + \lambda \nabla TV_\epsilon(z^{(q)}) \right]$
    $z^{(q+1)} = x^{(q+1)} + \alpha(x^{(q+1)} - x^{(q)})$
  end for
  return $x^{(Q)}$.

choice of the step τ in Algorithm 1. This step is computed using the spectral norm of the data fidelity term which can be computed using a power iteration method for each point configuration ξ. The resulting step is taken into account in the computation of the gradient with respect to the locations ξ of the cost function in (4).

A.2 The unrolled neural network

The neural network based reconstruction is an unrolled network. The one used in this work is based on the ADMM (Alternative Descent Method of Multipliers) [47]. It consists in alternating a regularized inverse followed by a denoising step with a neural network. If $\mathcal{D}_{\lambda^{(p)}}$ denotes the denoiser used at iteration p, the unrolled ADMM can be expressed through the sequence:

$$\begin{cases} x^{(p+1)} = (A(\xi)^* A(\xi) + \beta \mathrm{Id})^{-1} \left(A(\xi)^* y + \beta z^{(p)} - \mu^{(p)} \right) \\ z^{(p+1)} = \mathcal{D}_{\lambda^{(p)}} \left(x^{(p+1)} + \dfrac{\mu^{(p)}}{\beta} \right) \\ \mu^{(p+1)} = \mu^{(p)} + \beta \left(x^{(p+1)} - z^{(p+1)} \right) \end{cases}$$

with a pseudo-inverse initialization $z^{(0)} = A(\xi)^\dagger y$.

In this work, we use the DruNet network [62] to define the denoising mappings $\mathcal{D}_{\lambda^{(p)}}$. We choose an ADMM algorithm for the following reasons:

1. for well-spread sampling schemes, the matrix $A(\xi)^* A(\xi)$ has a good conditioning and the linear system that has to be inverted can be solved in less than a dozen iterations,

2. it has demonstrated great performance to solve linear inverse problems in imaging, including image reconstruction from Fourier samples [58].

We opted for a different network at each iteration instead of a network that share its weights accross all iterations. This leads to slightly higher performance at the price of a slightly harder to interpret architecture (see e.g. [24] for a similar discussion in CT reconstruction).

A.3 Training the reconstruction network for a family of operators

Following [27], we trained our network in a non usual way. Instead of training the denoising networks $\mathcal{D}_{\lambda^{(p)}}$ for a single operator $A(\xi_0)$, we actually trained it for a whole family of operators $\mathcal{A} = \{A(\xi), \xi \in \mathcal{F}\}$, where $\mathcal{F}$ is a large family of sampling schemes. We showed in [27], that this simple approach yields a much more robust network, which is adaptive to the forward operator.

In our experiments, the network is trained on a family of 10^3 sampling schemes that are generated using the attraction-repulsion minimization problem (11). These schemes are parameterized by densities that are within $\mathcal{C}$. This pretraining step consists of 32 epochs with a batch of 8 images using the Adam optimizer with default parameters ($\beta_1 = 0.9$ and $\beta_2 = 0.999$). The step for the CNN weights is set to 10^{-4} with a multiplicative update of 0.95 after each epoch. The measurements are perturbed by an additive white noise (see n in (4)).

A.4 Joint optimization

Instead of optimizing the sampling scheme for a fixed network, we can also optimize jointly the sampling locations together with the network weights. This approach was proposed in [58, 59]. Due to memory requirements, we set the batch size to 7 for the unrolled network in our training procedure. The step size for the CNN weights in this experiment is also set to 10^{-4} with the default Adam parameters.

A.5 Computational details

In this paragraph, we describe the main technical tools used to optimize the reconstruction process.

A.5.1 Solving the particle problem (4)

Problem (4) is a highly non-trivial problem. Two different computational solutions were proposed in [59, 58]. In this work, we re-implemented a solver with some differences outlined below.

First, the optimization problem (4) involves a nontrivial constraint set Ξ. While the mentioned works use a penalization over the constraints, we enforce the constraints by using a projection at each iteration. Handling constraints in stochastic optimization was first dealt with stochastic mirror-prox algorithms [35]. This approach turned out to be inefficient in practice. We therefore resorted to an extension of Adam in the constrained case called

Extra-Adam [25]. The step size was set to 10^{-3} and the default Adam parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We observed no significant difference by tuning these last two parameters. We also use a step decay of 0.9 each fourth of epoch and batch size of 13, which is the largest achievable by our GPU.

Similarly to [59, 58], we use a multi-scale strategy. The trajectories are defined through a small number of control points, that progressively increases across iterations. We simply use a piecewise linear discretization (contrarily to higher order splines in [58]). The initial decimation factor is 2^7 and is divided by two every two epochs. This results in a total number of epochs equal to 14 and takes about 86 hours for a total variation solver. In comparison [58], reports a total of 40 epochs.

A.5.2 Implementing the Non-uniform Fourier Transform (NUFT)

Various fast implementations of the Non-uniform Fourier Transform (1) are now available [36, 22, 55, 44]. In this work, we need a pyTorch library capable of backward differentiation. Evaluating the gradient of the cost function in (4) or in (6) indeed requires computing the differential of the forward operator $A(\xi)$ with respect to ξ. This can be done by computing D non-uniform Fourier transforms (see [58, 26, 57]). Different packages were tested and we finally opted for the cuFINUFFT implementation [55]. The bindings for different kind of NUFT are available at https://github.com/albangossard/Bindings-NUFFT-pytorch/.

A.5.3 Minimizing the discrepancy

The minimization of the discrepancy (11) is achieved with a gradient descent, as was proposed in the original paper [52], see Algorithm 2. The input parameters are the initial sampling set ξ^{ini}, the target density ρ and a step-size $\tau > 0$. The step size needs to be carefully chosen to ensure a fast convergence. The optimal choice can be shown to be related to the minimal distance between adjacent points. In our experiments, it was tuned by hand and fixed respectively to 2×10^4 and 5×10^3 for the 25% and 10% undersampling schemes.

Computing the gradient requires to compute pairwise interactions between all particles: in our codes, it is achieved using PyKeOps [17]. This approach presents the advantages of being fast, adapting to arbitrary kernels h and to natively allow backward differentiation within PyTorch. For a number of particles M above 10^6, fast multipole methods might become preferable [16].

Algorithm 2 Gradient descent to minimize (11).

Set $\zeta^{(0)} = \xi^{\text{ini}}$
for $j = 1 \dots J$ **do**
  $\zeta^{(j)} = \Pi_{\Xi}\left(\zeta^{(j-1)} - \tau \nabla_1 \text{dist}(\zeta^{(j-1)}, \rho)\right)$
end for
Set $\xi^{(n)} = \zeta^{(J)}$

A.5.4 Handling the mass at 0

An important issue is related to the fact that all trajectories start at the k-space origin. This creates a large mass for the sampling scheme at 0. When minimizing a discrepancy between the sampling scheme and a target density, the sampling points are therefore repulsed from the origin, creating large holes at the center. To avoid this detrimental effect, we fix rectilinear radial trajectories at the origin at maximal acceleration until a distance of 0.5 pixel between adjacent trajectories samples is reached. This creates a fixed pattern in the

k-space center, which can easily be seen in the zoom of Fig. 6a and Fig. 6b. We compute the discrepancy only at the exterior of a disk centered at the origin containing this fixed pattern.

A.5.5 Projection onto the constraint set

The projector onto the constraint set is used twice in this work. First the Extra-Adam algorithm requires a Euclidean projector on the constraint set Ξ to solve (4). This projector is also needed to compute one evaluation of the sampler in (11). In this project, we used the dual approach proposed in [19], implemented on a GPU. This algorithm can be implemented in PyTorch, and can be differentiated. This allows computing the gradient of the overall function in (6).