Globally optimized 3D SPARKLING trajectories for high-resolution T2*-weighted Magnetic Resonance Imaging

Chaithya Giliyar Radhkrishna, Pierre Weiss, Aurélien Massire, Alexandre Vignaud, Philippe Ciuciu

Globally optimized 3D SPARKLING trajectories for high-resolution T2*-weighted Magnetic Resonance Imaging

Chaithya GR, Pierre Weiss, Aurélien Massire, Alexandre Vignaud and Philippe Ciuciu *Senior Member, IEEE*

***Abstract*— The Spreading Projection Algorithm for Rapid K-space samplING, or SPARKLING, is an optimization-driven method that has been recently introduced for accelerated 2D T2* weighted magnetic resonance imaging (MRI) using compressed sensing. It has then been extended to address 3D imaging using either stacks of 2D sampling patterns or a local 3D strategy that optimizes a single sampling trajectory at a time. 2D SPARKLING actually performs variable density sampling (VDS) along a prescribed target density while maximizing sampling efficiency and meeting the gradient-based hardware constraints. However, 3D SPARKLING has remained limited in terms of acceleration factors along the third dimension if one wants to preserve a peaky point spread function (PSF) and thus good image quality. In this paper, in order to achieve higher acceleration factors in 3D imaging while preserving image quality, we propose a new efficient algorithm that performs globally optimized 3D SPARKLING. Its computational complexity scales up with the number of k-space samples p at the rate of $O(p \log(p))$ and allows for the optimization of all k-space trajectories simultaneously, thus yielding a full 3D VDS pattern. We compare multi-CPU and GPU implementations and demonstrate that the latter is optimal for 3D imaging in the high-resolution acquisition regime (600 µm isotropic). Finally, we show that this novel globally optimized 3D SPARKLING approach outperforms stacking strategies through retrospective and prospective studies on NIST phantom and in vivo brain scans at 3 Tesla, respectively. Overall the proposed method allows for 5x shorter scan times compared to GRAPPA-4 parallel imaging acquisition at 3 Tesla.**

***Index Terms*— 3D MRI, optimization, non-Cartesian, compressed sensing, acceleration.**

I. Introduction

The theory of compressed sensing (CS) boosted the search for efficient sampling patterns, resulting in faster MRI scan times [1]. It was proved and observed empirically that an efficient implementation relies on non-Cartesian trajectories with a variable density [2]–[5]. Non-Cartesian k-space trajectories (e.g. spiral, radial, rosette, etc.) [6]–[10] have been proposed for accelerated and robust-to-motion 2D imaging, prior to the existence of theoretical foundations. Specific and realistic adaptations, i.e. compliant with the hardware gradient constraints, of these patterns have then emerged to follow the insights provided by the theory, notably variable density sampling (VDS) [11]–[13]. More recently, the SPARKLING algorithm [14], has been shown to automatically generate optimized non-Cartesian sampling patterns compatible with MR hardware constraints on maximum gradient amplitude and slew rate as well as with key criteria for optimal sampling, namely a controlled distribution of samples (e.g., variable density) and a locally uniform k-space coverage.

However, for the sake of signal-to-noise ratio (SNR), 3 dimensional (3D) imaging is preferred to reach isotropic high-resolution imaging (e.g. 600µm isotropic). In this regard, multiple approaches have been utilized to efficiently down-sample 3D k-space. Some of these involve a combination of a readout in the z-direction with a 2D under-sampled mask based on Poisson disk sampling [15]. Additional attempts on full 3D readouts were proposed like 3D radial trajectory [16], 3D cones [17], twisted projections [18] and hybrid radial-cones [19]. However, these trajectories were primarily based on parameterizing a k-space sampling curve, and the final sampling pattern was not optimized with respect to the reconstruction quality. Some recent studies explored how to optimize the sampling pattern [20]–[22], but did not include a clear sampling criterion in order to maximize the image reconstruction quality.

Other methodologies in the literature involved stacking a 2D under-sampled trajectory like stack of stars [23], [24], stack of spirals [17], [25] and stack of 2D SPARKLING [26]. Uniform (i.e.cylindrical) stacking is usually implemented even though a spherical strategy, with a number of shots varying as a function of the latitude plan, was shown to be beneficial on image quality for SPARKLING trajectories [26]. Further in [26], we made attempts at solving for a local 3D SPARKLING trajectory which was limited to a cone obtained from a parcellation of the 3D spherical k-space. Then we filled up all the cones covering a given elevation plane with the replication of the resulting trajectory. However, this method did not ensure a locally uniform sampling pattern at the boundaries of cones as the problem was solved locally.

Chaithya G R and P. Ciuciu are both with CEA, NeuroSpin, F-91191 Gif-sur-Yvette, cedex, France and Inria, Parietal, Université Paris-Saclay, F-91120 Palaiseau, France, whereas A. Vignaud is only affiliated with CEA, NeuroSpin (Email: philippe.ciuciu@cea.fr). Pierre Weiss is with CNRS, IMT (UMR 5219) and ITAV (USR 3505), F-31106, Toulouse. Aurélien Massire is with Siemens Healthcare SAS, F-93527 Saint-Denis, cedex, France.

The recent rise of machine and deep learning has also impacted the literature on MRI sampling [27]–[32]. All approaches that have proposed in this field are supervised learning techniques, which means that they need ground truth corresponding to fully sampled data, to learn an optimal under-sampling pattern. While the initial contributions focused on Cartesian sampling along lines [27], [28], more recent approaches made some advance on learning the target density off the Cartesian grid [29], [30]. Additionally, to the best of our knowledge, the only works that have learned hardware-compliant real trajectories are PILOT [31] and 3D FLAT [32] for 2D and 3D imaging, respectively. Nonetheless it is worth noting that none of these approaches has been prospectively validated on real acquisitions. For that reason, they won't be discussed any further in this paper.

In this paper, for the first time, we solve the SPARKLING optimization globally in 3 dimensions. First, in Sec II, we remind the optimization problem to be solved for generating SPARKLING trajectories. Then we focus on major computational bottlenecks that prevented us from scaling the original solution to 3D and provide detail on our main contributions. One key ingredient in SPARKLING lies in setting the right target sampling density. The latter may vary as a function of the resolution and acceleration factor. For that purpose, we parameterize radially decaying densities by two parameters (cut-off, decay) and find the optimal density through grid search over pairs of parameters. This study can be conducted on retrospective analysis and then the sought optimal density can be used further in prospective acquisitions. In Sec. IV, we present the experimental data set on which the numerical studies are performed later on for validation and comparison purposes. In this regard, we carry out retrospective analysis on NIST phantom data collected at 3 Tesla. Then we run prospective in vivo brain imaging acquisitions on a healthy adult volunteer still at 3 Tesla and compare the proposed full 3D SPARKLING with the existing spherical stack of 2D SPARKLING.

II. Theory

In this section we briefly introduce the SPARKLING algorithm as described in [14]. We detail the particular steps involved in the optimization process. We point to some computational bottlenecks in each of these steps. Later, we describe the methods used to overcome these computational challenges, thereby allowing us to scale the problem to 3 dimensions. Most of the theoretical aspects are directly based on earlier works in [33]–[35], which can be consulted for the problem description and derivations of (4).

A. 3D K-space sampling

A 3D k-space sampling pattern $\mathbf{K}$ is usually composed of several shots or curves, say N_c, $\mathbf{K} = (\mathbf{k}_i)_{i=1}^{N_c}$, where each 3D shot $\mathbf{k}_i(t) = (k_{i,x}(t), k_{i,y}(t), k_{i,z}(t))$, is controlled by magnetic field gradients $\mathbf{G}_i(t) = (G_{i,x}(t), G_{i,y}(t), G_{i,z}(t))$ as follows:

$$\mathbf{k}_i(t) = \frac{\gamma}{2\pi} \int_0^t \mathbf{G}_i(\tau) d\tau \,, \tag{1}$$

with γ the gyro-magnetic ratio ($\gamma = 42.57$MHz/T for proton imaging). In practice, each shot $\mathbf{k}_i(t)$ is sampled at a pace of Δt, the gradient raster time, during the readout T_{obs} (e.g. $T_{\text{obs}} \simeq 20$ms for our T_2^*-weighted imaging). Then the number of gradient time steps is given by $N_s = \left\lfloor \frac{T_{\text{obs}}}{\Delta t} \right\rfloor$ and the global sampling pattern $\mathbf{K}$ finally consists of $p = N_c \times N_s$ points. The so-defined trajectory is then interpolated by the hardware and sampled with an analog-to-digital converter at a different pace δt called dwell-time. In practice the dwell-time δt is a fraction of the raster time Δt.

The k-space domain for a 3D MR volume of size $N_x \times N_y \times N_z$ over a field of view $\text{FOV}_x \times \text{FOV}_y \times \text{FOV}_z$, is defined within $[-K^x_{\max}, K^x_{\max}] \times [-K^y_{\max}, K^y_{\max}] \times [-K^z_{\max}, K^z_{\max}]$, with $K^\ell_{\max} = \frac{N_\ell}{2FOV_\ell}$ and $\ell = x, y, z$. For the sake of simplicity, in what follows we assume the same spatial resolution along the three dimensions so $K^x_{\max} = K^y_{\max} = K^z_{\max} = K_{\max}$ even though we meet different matrix and FOV dimensions ($N^z \neq (N^x = N^y)$ and $\text{FOV}^z \neq (\text{FOV}^x = \text{FOV}^y)$). Hereafter, the 3D k-space domain will be normalized to $\Omega = [-1, 1]^3$.

Hardware constraints on the maximum gradient amplitude ($G_{\max}$) and slew rate ($S_{\max}$) induce limitations in trajectory speed and acceleration. These limits can be expressed as box constraints on the amplitude of the discrete derivatives of the k-space trajectory $(\mathbf{k}_i[n])_{n=0}^{N_s-1}$. Typically, rotation invariant speed and acceleration constraint can be expressed as follows:[1]

$$\mathcal{Q}^{N_c}_{\alpha,\beta} = \left\{ \begin{array}{c} \forall i = 1, \dots, N_c, \quad \mathbf{k}_i \in \mathbb{R}^{3\times N_s}, \\ \mathbf{A}\mathbf{k}_i = \mathbf{v}, \\ \|\mathbf{k}_i\|_\infty \leq 1, \ \|\dot{\mathbf{k}}_i\|_{2,\infty} \leq \alpha, \ \|\ddot{\mathbf{k}}_i\|_{2,\infty} \leq \beta, \end{array} \right\} \tag{2}$$

where

$$\begin{aligned} \dot{\mathbf{k}}_i[n] &= \frac{\mathbf{k}_i[n] - \mathbf{k}_i[n-1]}{\Delta t} \\ \ddot{\mathbf{k}}_i[n] &= \frac{\mathbf{k}_i[n+1] - 2\mathbf{k}_i[n] + \mathbf{k}_i[n-1]}{\Delta t^2} \\ \|\mathbf{c}\|_{2,\infty} &= \sup_{0\leq n\leq N_s-1} \left(|c_x[n]|^2 + |c_y[n]|^2 + |c_z[n]|^2 \right)^{1/2}, \end{aligned}$$

for all $\mathbf{c} \in \Omega^{N_s}$ and (α, β) are obtained by normalizing hardware and Nyquist constraints to the sampling domain Ω (from [14]):

$$\left\{ \begin{array}{rcl} \alpha & = & \frac{1}{K_{\max}} \min \left(\frac{\gamma G_{\max}}{2\pi}, \frac{1}{FOV \cdot \delta t} \right) \\ \beta & = & \frac{\gamma S_{\max}}{2\pi K_{\max}} \end{array} \right. \tag{3}$$

The purpose of $\mathbf{A}$ and $\mathbf{v}$ are to model linear constraints

[1] In [33], we have also dealt with the case of rotation variant constraints where the ℓ_∞-norm replaces the mixed $\ell_{2,\infty}$-norm here.

on the trajectory, like the TE point constraint, which ensures that each trajectory passes through center of k-space at echo time (TE). More sophisticated linear constraints can be modeled too, see details in [33]. The normalized constraint $\alpha\delta t \leq \frac{1}{FOV \times K_{\max}}$ ensures that the distance of the k-space locations associated with two consecutive measurements is lower than the Nyquist rate, which is essential to discard some undesired filtering effects [36].

B. 3D SPARKLING formulation

Let $\rho : \Omega \to \mathbb{R}$ denote a target sampling density, with $\rho(x) \geq 0$ for all x and $\int \rho(x)\, dx = 1$. Following previous works [14], [37]–[39], we obtain $\mathbf{K} \in \Omega^p$ by solving:

$$\widehat{\mathbf{K}} = \arg \min_{\mathbf{K} \in \mathcal{Q}_{\alpha,\beta}^{N_c}} F_p(\mathbf{K}) = \left[F_p^{\mathrm{a}}(\mathbf{K}) - F_p^{\mathrm{r}}(\mathbf{K})\right] \tag{4}$$

with $\mathcal{Q}_{\alpha,\beta}^{N_c}$ being the constraint set for the N_c shots. Here we remind that p refers to the total number of k-space samples (or particles), so $p = N_c \times N_s$.

The term $F_p^{\mathrm{a}}(\mathbf{K})$ corresponds to an attraction term which ensures that the final distribution of the k-space sampling points follows the target density ρ and $F_p^{\mathrm{r}}(\mathbf{K})$ is the repulsion term to ensure that the sampling is locally uniform and that we don't have any local clusters. These terms are defined as:

$$F_p^{\mathrm{a}}(\mathbf{K}) = \frac{1}{p} \sum_{i=1}^{p} \int_{\Omega} H(x - \mathbf{K}[i]) \rho(x)\, dx \tag{5a}$$

$$F_p^{\mathrm{r}}(\mathbf{K}) = \frac{1}{2p^2} \sum_{1 \leq i,j \leq p} H(\mathbf{K}[i] - \mathbf{K}[j]) \tag{5b}$$

where $\mathbf{K}[i] \in \mathbb{R}^3$ describe the locations of k-space samples in a shot-based lexicographical order $[\mathbf{k}_1, \dots \mathbf{k}_{N_c}]$. The function H is a well chosen kernel, typically $H(x) = \|x\|_2$. Note that alternative choices such as $H(x) = \log(x)$ have been also investigated in [40]. The minimization problem (4) can be attacked by various nonlinear programming procedures. In this work, we propose to use a projected gradient descent as described below:

$$\mathbf{K}^{(t+1)} = \Pi_{\mathcal{Q}_{\alpha,\beta}^{N_c}} \left(\mathbf{K}^{(t)} - \eta^{(t)} \nabla F_p(\mathbf{K}^{(t)}) \right) \tag{6}$$

The computational bottlenecks in (4) involve the calculation of $\nabla F_p(\mathbf{K}) = \nabla F_p^{\mathrm{a}}(\mathbf{K}) - \nabla F_p^{\mathrm{r}}(\mathbf{K})$, and the projection of each shot onto the constraint set $\mathcal{Q}_{\alpha,\beta}^{N_c}$.

C. Gradient Descent Step

In what follows, we provide details about the calculation of F_p and ∇F_p.

1) Evaluating $\mathsf{F}_{\mathsf{p}}^{\mathsf{a}}$ and its gradient: To calculate the attraction term and its gradient, we can re-write (5a) as:

$$F_p^{\mathrm{a}}(\mathbf{K}) = \frac{1}{p} \sum_{i=1}^{p} (H \star \rho)(\mathbf{K}[i]) \tag{7}$$

where $\star$ denotes the convolution-product in the continuous setting. The main difficulty is thus to quickly evaluate $(H \star \rho)(x)$ (optional, if we want to compute the cost function) and its derivatives. To this end, we discretize the target sampling distribution ρ as follows:

$$\boldsymbol{\rho}[i,j,k] = \rho(i/N, j/N, k/N) \tag{8}$$

where $i,j,k \in [-N, N]$, and $N \in \mathbb{N}$ describes the number of discretization points. We typically take N twice as large as $\max(N_x, N_y, N_z)$ to define the density at a better resolution than the image size. Similarly we compute a discrete version of the filter H as:

$$\mathbf{H}[i,j,k] = H(i/N, j/N, k/N) \tag{9}$$

Letting $*$ denote the discrete convolution-product, we use the following approximation

$$(H \star \rho)(\mathbf{K}[i]) \simeq \mathcal{I}(\mathbf{H} * \boldsymbol{\rho})(\mathbf{K}[i]), \tag{10}$$

where $\mathcal{I} : R^{(2N+1)^3} \to C^0(\Omega)$ denotes a tri-linear interpolant function. Hence, the computation of $F_p^{\mathrm{a}}(\mathbf{K})$ requires to precompute $\mathbf{H} * \boldsymbol{\rho}$ on a discrete grid with fast Fourier transforms once for all. The computation of the sum in (7) then has a complexity $O(p)$, which is linear in the number of particles.

Similarly, the computation of $\nabla F_p^{\mathrm{a}}(\mathbf{K})$ involves the calculation of the partial derivatives $\partial_{i,l} F_p^{\mathrm{a}}(\mathbf{K})$ where $1 \leq i \leq p$ is the index of a particle and $1 \leq \ell \leq 3$ the index of a dimension. According to (7), the partial derivative is:

$$\partial_{i,\ell} F_p^{\mathrm{a}}(\mathbf{K}) = \frac{1}{p} (\partial_\ell H \star \rho)(\mathbf{K}[i]) \tag{11}$$

Thus, letting $\nabla \mathbf{H} \in \mathbb{R}^{(2N+1)^3 \times 3}$ denote a discretization of ∇H, we can precompute the discrete vector field $\nabla \mathbf{H} * \boldsymbol{\rho}$ on a Cartesian grid using fast Fourier transforms and then use a tri-linear interpolant to evaluate it off the grid.

2) Evaluating F^{r} and its gradient: The problem addressed here is to compute F_p^{r} and $\nabla F_p^{\mathrm{r}}(\mathbf{K})$. For purposes of simplification, we introduce $r_{ij} = \|\mathbf{K}[i] - \mathbf{K}[j]\|_2$ and consider H to be a radial function depending on r_{ij} only. Letting K_ℓ denote the spatial components of $\mathbf{K} = [K_1, K_2, K_3]$, we get:

$$F_p^{\mathrm{r}}(\mathbf{K}) = \sum_{1 \leq i,j \leq p} H(r_{ij}) \tag{12a}$$

$$\partial_{i,\ell} F_p^{\mathrm{r}}(\mathbf{K}) = \frac{1}{p^2} \sum_{j \neq i} \left(\frac{K_\ell[i] - K_\ell[j]}{r_{ij}} \right) \partial_\ell H(r_{ij}) \tag{12b}$$

The evaluation of all the components of the gradient require $O(p^2)$ computations, where p can reach 10^8 for high resolution imaging. An efficient implementation is therefore critical. In this work, we explored two possibilities.

a) Brute force calculation using PyKeops: The computation of (12a) and (12b) can be highly parallelized, which is amenable to efficient GPU implementations. Carrying out such computations on array centric frameworks like PyTorch and Tensorflow would require the use of huge $p \times p$-dimensional matrices. This would result in

a large memory footprint, much larger than what is typically available on current modern GPUs. For the sake of efficient memory usage, we used `PyKeops`, a library that permits low cost calculations of large kernel operations [41]. `PyKeops` carries out the naive and direct computations using online map reduce schemes from CUDA routines for summations. Due to this, the whole matrices are not stored in the GPU memory, but rather just the final results.

b) Fast Multipole Methods: Sums of the form (12a) and (12b) appear in many n-body problems and can be computed efficiently using Fast Multipole Methods (FMM) [42]. Given a set of positions $\mathbf{K}[i]$, a kernel $\Psi : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ and a set of weights $\mathbf{w} \in \mathbb{R}^p$, the FMM method allows for the efficient computation of vector $\mathbf{v}$ of the form

$$\mathbf{v}[i] = \sum_{j=1}^{p} \Psi(\mathbf{K}[i], \mathbf{K}[j])\mathbf{w}[j]\,.$$

The FMM utilizes a multipole expansion of the kernel Ψ, which allows for a hierarchical grouping of closely spaced k-space points and treat them as a single source. This results in a massive acceleration of the above computation with a complexity $O(p \log p/\epsilon)$, where ϵ is a user-prescribed precision. For our implementations, we used the Parallel Black box FMM [43], [44] in 3D (`PBBFMM3D`[2]), which can be run with any arbitrary kernel Ψ.

To evaluate the cost function F_p^{r}, we only need to set

$$\Psi(\mathbf{K}[i], \mathbf{K}[j]) = H(r_{ij}) \text{ and } \mathbf{w}[j] = 1, \forall j.$$

To evaluate the gradient $(\partial_{i,\ell} F_p^{\mathrm{r}}(\mathbf{K}))_i$, we set $\mathbf{w}[j] = 1$ and

$$\Psi(\mathbf{K}[i], \mathbf{K}[j]) = \left(\frac{K_\ell[i] - K_\ell[j]}{r_{ij}}\right) \partial_\ell H(r_{ij})\,.$$

c) Comparisons: From Fig 1, we see that naive GPU implementations on `PyKeops` outperforms the `PBBFMM3D` implementation for $p < 5 \times 10^6$. Beyond this value, `PBBFMM3D` gets faster. It is likely that faster computations with the FMM would be obtained with a GPU implementation. Unfortunately, we did not find any robust and efficient GPU implementation of FMM.

3) Choice of step size $\gamma^{(t)}$: In our implementation, we use a combination of two step sizes. In the first 100 iterations, we use a fixed step size. As analyzed in [35], this strategy provides a convergence guarantee to a local minimizer of the cost function given that:

- The kernel H has a L-Lipschitz continuous gradient.
- The step size is inversely proportional to the Lipschitz constant.

These conditions are satisfied with a regularized norm of the form $H(r) = \sqrt{r^2 + \epsilon^2}$. We can then set $\eta^{(t)}$ proportional to ϵ (i.e. 6.25^{-2}). The value of ϵ can be chosen as a fraction of the minimal distance between two points at a stationary point.

A constant step size is too conservative and a faster convergence can be obtained using a second-order dynamics close to the minimizer. This justifies switching to a Barzilai−Borwein [45] in the last iterations. Few theoretical guarantees are available for this technique, but it significantly accelerates the convergence empirically.

[2]see `https://github.com/ruoxi-wang/PBBFMM3D`.

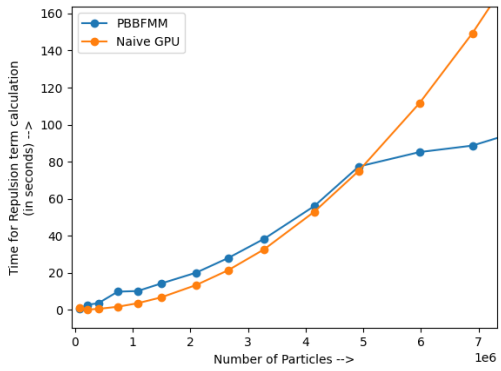

Fig. 1: Computation times for the repulsion term F^{r} as a function of the number of particles p.

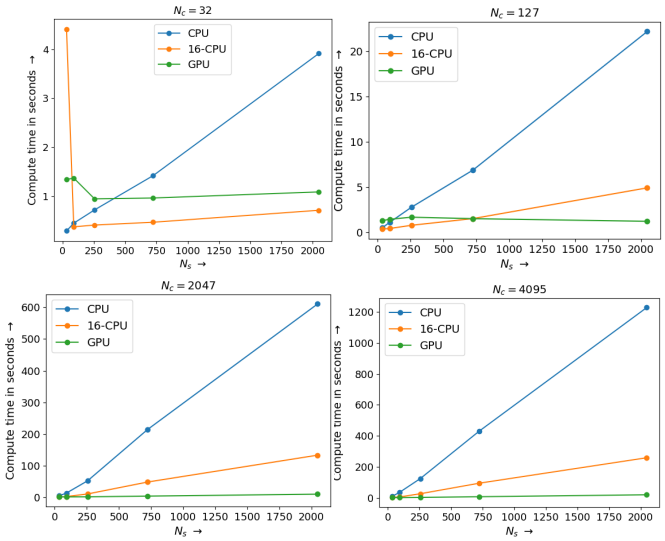

Fig. 2: Computation times for varying N_s and N_c for the projection step $\Pi_{\mathcal{Q}_{\alpha,\beta}^{N_c}}(\cdot, n_{pit})$ that was run over $n_{pit} = 200$ iterations with $G_{\max} = 40\mathrm{mT/m}$ and $S_{\max} = 180\mathrm{T/m/s}$.

D. Projection step

The projection step in (6) for a general single k-space shot onto a given constraint set parameterized by (α, β) has been explored in [33], [35]. For our implementations, we needed to extend the single shot iterative procedure called Algorithm 1 in [33] to projecting N_c shots. Note that the actual projection of a k-space shot is independent of other shots, and thus the computation can be done in parallel. Hence, we have implemented this step both on multi-CPU and GPU. To efficiently utilize a

GPU, we used the CuPy module [46]. The computation times with different implementations for varying N_c and N_s are shown in Fig. 2. We found that the computation times vary linearly with N_s and are drastically reduced for the GPU implementation compared to the CPU versions (single and multicore).

E. Multi-resolution strategy for faster convergence

In order to allow for the algorithm to reach faster convergence and lead to a better approximation of the target density, a multi-resolution approach as described in [47] was implemented. Under this methodology, the optimization of the sampling pattern was carried out on down-sampled curves. The interpolated solution was later used as a warm restart for the up-sampled problem. Our implementations involved dyadic scaling and up-scaling through simple linear interpolation of k-space shots. Let the linear interpolator be of the form $\mathcal{L}_{2d} : \Omega^d \rightarrow \Omega^{2d}$. We define the parameter N_d as the number of decimation steps in the algorithm. Note that the constraint space needs to be equally scaled with the problem, which results in scaling the α and β constraints mentioned in (3) to:

$$\alpha = \frac{\gamma G_{\max} 2^{N_d}}{2\pi K_{\max}}, \quad \beta = \frac{\gamma S_{\max} 2^{N_d}}{2\pi K_{\max}} \tag{13}$$

As we move through the dyadic decimation steps and up-sample the curve, these constraints are halved.

Algorithm 1: Multi-resolution implementation of SPARKLING

Inputs: ρ, G_{max}, S_{max}, N_c, N_s, N_d, n_{git}, n_{pit}
Output: $\mathbf{K}$, the k-space sampling pattern
Initializations: $\mathbf{K}^{(0)} \in \Omega^{\frac{N_c \times N_s}{2^{N_d}}} = \Omega^p$
$\alpha \leftarrow \frac{\gamma G_{\max} 2^{N_d}}{2\pi K_{\max}}$, $\beta \leftarrow \frac{\gamma S_{\max} 2^{N_d}}{2\pi K_{\max}}$
while $N_d > 0$ **do**
  $p \leftarrow \frac{N_c \times N_s}{2^{N_d}}$
  for $t = 1 \dots n_{git}$ **do**
    $\mathbf{K}^{(t-1/2)} = \mathbf{K}^{(t-1)} - \eta^{(t)} \nabla F_p(\mathbf{K}^{(t-1)})$
    $\mathbf{K}^{(t)} = \Pi_{\mathcal{Q}^{N_c}_{\alpha,\beta}} \left(\mathbf{K}^{(t-1/2)}, n_{pit} \right)$
  end
  // Warm restart next decimation step with linear interpolation
  // The dimension of k is doubled at each decimation step
  for $s = 1 \dots N_c$ **do**
    $\mathbf{k}_s^{(0)} \leftarrow \mathcal{L}_{\frac{N_s}{2^{N_d-1}}} \left(\mathbf{k}_s^{(n_{git})} \right)$
  end
  $\mathbf{K}^{(0)} \leftarrow \left[\mathbf{k}_1^{(0)}, \dots, \mathbf{k}_{N_c}^{(0)} \right]$
  // Scale constraints
  $\alpha \leftarrow \frac{\alpha}{2}, \beta \leftarrow \frac{\beta}{2}$
  $N_d \leftarrow N_d - 1$
end

F. Overall algorithm

Algorithm 1 summarizes how to concretely compute the SPARKLING solution along with multi-resolution steps described in subsection II-E. Further, for the sake of completeness, we have briefly summarized the multi-shot projection step $\Pi_{\mathcal{Q}^{N_c}_{\alpha,\beta}}$ in Sec. S1 of supplementary material. However, for more details on the iterative procedure involved in the projection step, the reader can refer to [33].

III. Numerical experiments and data acquisition

The sampling patterns were obtained by carrying out projected gradient descent as described above. With the above described improvements, the SPARKLING Generation time was just 10 minutes for 2D and nearly 18 hours for 3D on Intel Xeon (256GB RAM), with most code running on a NVIDIA Quadro P6000 with 3840 CUDA cores and 24GB DDR5X memory.

A. SPARKLING: a Python package

In the ethos of reproducible research and to move forward into better optimized patterns for MRI acquisition, all the implementations as described above is present in a Python package at the private repository[3]. All codes in the package scale to 2 and 3 dimensions directly, and most codes are agnostic and can be run on CPU or GPU with some change in run parameters. All the scanner constants and trajectory specification can be provided through a configuration file, and most of the codes are modular in nature. Interested researchers are requested to contact the authors for obtaining access to this package[4].

B. Acquisition parameters

With a goal of 600µm isotropic resolution in 3D MRI acquisitions, we planned to obtain a volume of $(N_x \times N_y \times N_z) = (384 \times 384 \times 208)$ size in order to cover the whole brain. For the sake of consistency, we used the same matrix size and resolution for our acquisitions on the NIST phantom[5]. The trajectories were generated for a Siemens Prisma 3T Scanner with maximum gradient strength $G_{\max} = 40\text{mT/m}$ and peak slew rate $S_{\max} = 180\text{T/m/s}$. The number of samples per shot was set to $N_s = 2048$, while the data was acquired with a dwell time $\delta t = 2\mu\text{s}$ ($\Delta t = 10\mu\text{s}$) at the analog-to-digital-converter level. The number of shots N_c was varied based on the study described hereafter. For our in vivo studies, the k-space data was acquired on a Siemens 64 channel Head/Neck coil, while using 44 channels

[3] https://gitlab.com/cea-cosmic/CSMRI_sparkling

[4] It cannot be made open source given patent application.

[5] https://www.nist.gov/programs-projects/quantitative-mri

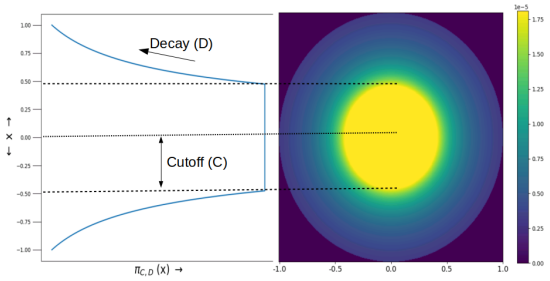

Fig. 3: Parameterization of variable density with cutoff C and decay D.

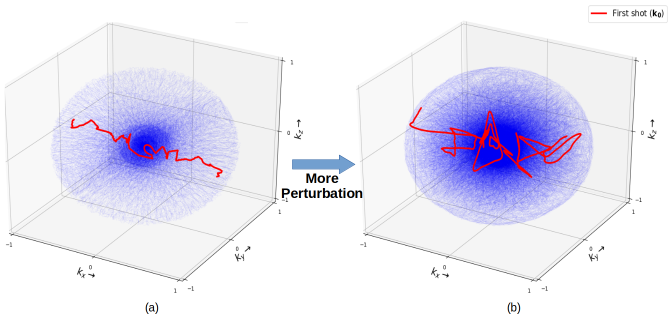

Fig. 4: Effect of adding a perturbation to the initial k-space trajectory in $\Omega = [-1, 1]^3$ as zero mean uniform random noise. Trajectories are generated with maximum displacement of k-space point to (a) 0.10 and (b) 0.75 in the initialization.

around head during acquisition. The echo time (TE) was 20ms and repetition time (TR) was 37ms. The flip angle was set to 15°, and the slice excitation was slab selective. We also obtained a reference volume collected using a 4-fold accelerated Cartesian acquisition (acquisition time or TA=15min13s) based on GRAPPA parallel imaging technique [48]. The projected gradient descent was carried out with multi-resolution decimation steps $N_d = 6$ for faster convergence. The algorithm was run for $n_{git} = 200$ outer gradient descent iterations with $n_{pit} = 200$ steps in the inner projection loop.

1) Choice of target sampling density: The target sampling density was chosen to be radially isotropic, which decays as an inverse polynomial with a constant plateau in the center of k-space. The density was defined with C, the cutoff frequency in k-space center having a constant density and D, the rate of decay for higher frequencies. Mathematically, we define the target density $\pi_{C,D}(x) : \Omega^d \to \mathbb{R}_+$ as follows:

$$\pi_{C,D}(x) = \begin{cases} \kappa & |x| < C \\ \kappa \left(\frac{C}{|x|}\right)^D & |x| > C \end{cases} \tag{14}$$

where κ is a constant obtained through normalization as $\kappa = \frac{1-D}{2C(C^{D-1}-D)}$. The resulting density is radially symmetric and is of the form described in Fig. 3. The choice of a radial density was motivated by the wish to provide rotation invariant reconstruction results. Notice that the recent learning based approaches [29], [30] result in non-symmetric densities. This is probably due to the fact that brains or knees databases such as fastMRI [49] used for training have boundaries which are dominantly vertical or horizontal. However, the fine details may be in arbitrary orientations.

The correct choice of density was carried out by grid searching for the optimal parameters $\hat{C}$ and $\hat{D}$ on the target sampling distribution as defined in (14). We performed retrospective reconstruction on raw Cartesian data acquired on a NIST Phantom. For this study, the number of shots was set to $N_c = 4096$. We found that $\hat{C} = 15\%$ and $\hat{D} = 3$ provided the best SSIM score of 0.98. The SSIMs for varying values of C and D are presented in supplementary material (see Fig. S1).

2) Initialization: As the problem being solved is non-convex, different choices of initialization would lead to different solutions. In [14] for 2D imaging we observed that radial initialization performed the best for exploring the k-space. Hence, here for 3D imaging we stick to 3D radial initialization too. Additionally, to ensure that each shot maximally explores the k-space, we perturb each sample point in k-space by adding zero mean uniform random noise in each dimension. Particularly, we compare the resulting optimized trajectory obtained after a perturbation as a random motion of each k-space point with maximum amplitude set at 0.1 and 0.75 (we remind that the sampling domain is normalized to $\Omega = [-1, 1]^3$). The optimized sampling patterns are presented in Fig. 4. We clearly show that with more perturbation, the k-space trajectory tends to explore a broader part of k-space giving a better coverage overall. Further, we would like to emphasize that these trajectories are particularly useful in cases of high receiver sampling rates, as they would then sample more of the k-space per shot and would overall prevent any hole in the sampling pattern.

IV. Results

A. Trajectories and point spread function

We present the full 3D SPARKLING, obtained with $N_c = 4096$ in Fig. 5. We visualize the trajectory along the mid-planes of 3 orthogonal orientations and provide an approximate sampling mask in these planes. Further, to understand why these trajectories are expected to yield good image reconstructions, we measure the 3D point spread function (PSF). In Fig. 6 we compare the PSFs with respect to earlier generated spherical stack of 2D SPARKLING (`SpSOS`) trajectories. We do not include any comparison with 3D radial sampling scheme or stack of spirals as this was already done in [26]. Particularly, we emphasize the reduction in sidelobes along the z axis. Further, for the purpose of numerical comparison, we computed the full width at half maximum (FWHM) and peak-to-sidelobe level (PSL) in Tab. I. The global 3D pattern provides us with reduced FWHM and larger PSL,

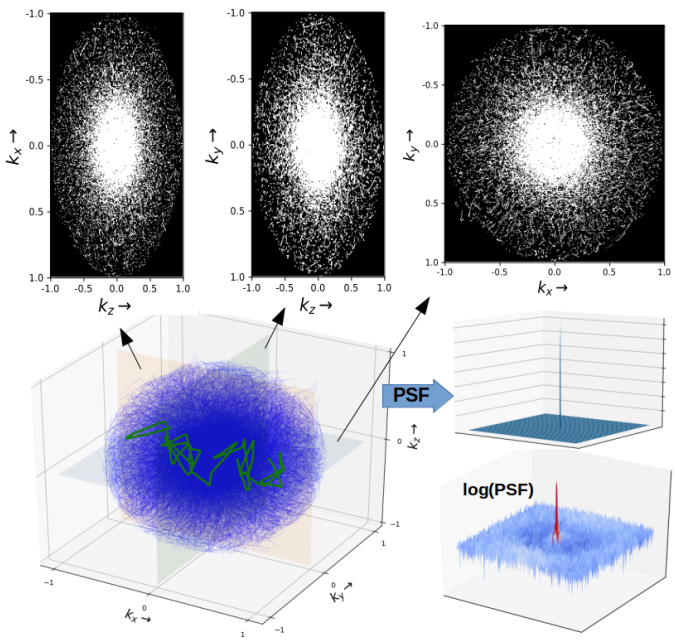

Fig. 5: Full 3D SPARKLING Trajectory for $N_c = 4096$, $N_s = 2048$ and the point spread function along the mid z-plane

TABLE I: Comparing metrics of PSF with FWHM (lower is better) and PSL (higher is better).

Trajectory	FWHM			PSL
	x	y	z	
Global 3D	2.7	2.7	**2.9**	**15.34**
SpSOS	2.9	2.9	3.3	10.48

two indicators that demonstrate the full 3D SPARKLING methodology outperforms the spherical stack version.

B. Phantom

We proceed by carrying out retrospective study to assess the quality of reconstructed images. We varied the acceleration factor (AF $= \frac{N_x \times N_z}{N_c}$ for 3D MR imaging, i.e. computed with respect to fully sampled data) from 20 (TA=2min30s) to 70 (TA=45s) compared to a fully sampled scenario and thus from 5 to 15 compared to the reference Cartesian p4 (i.e. AF=4) undersampled acquisition. Our motivation was to understand the degradation in image quality while decreasing the number of collected spokes. The results are presented in Tab. II. They clearly show that the globally optimized 3D SPARKLING strategy is robust to high acceleration factors in terms of image quality as reflected by the higher SSIM scores. In contrast, the performances of the SpSOS approach start to get worse already for AF=40. The reconstructed images are presented in the supplementary material (see Fig. S2).

TABLE II: Comparing SSIM metrics of retrospective reconstruction

Trajectory	AF20	AF40	AF70
Global 3D	**0.98**	**0.964**	**0.9452**
SpSOS	0.955	0.9026	0.817

C. In vivo

We collected in vivo data with globally optimized 3D SPARKLING and SpSOS trajectories for brain imaging on a healthy volunteer (male, 25 y.o.). This study was approved by a national ethics committee (CPP 100048). The volunteer signed a written informed consent form. The MR image associated with this k-space data was reconstructed by the method [50]–[53] as described in supplementary material (see Sec. S2).

1) Retrospective Acquisition: To understand how the trajectories perform for in vivo brain data, we repeat the earlier retrospective study on Cartesian p4 scans acquired on the volunteer. The results of the scans are presented in Fig. 7. We see that global 3D SPARKLING trajectories outperforms the SpSOS trajectory both visually and quantitatively in SSIM metrics with maximum SSIM of 0.92 (AF=20). Additionally, the SSIM metrics follow the similar trend as seen for phantom data, with the SSIMs for SpSOS dropping off more rapidly from 0.86 (AF=20) to 0.72 (AF=70). In contrast, the global 3D SPARKLING trajectories tend to preserve the structures and show some blurring artifacts only at AF=70 where SSIM drops to 0.83.

2) Prospective Acquisition: Finally, we proceed to prospectively acquire data on the scanner for the same trajectories and on the same individual. We present the reconstructed results in Fig. 8. They clearly show superiority of the full 3D pattern compared to SpSOS. Image quality is well preserved for AF=20 in full 3D strategy, while it tends to get noisy at AF=40 and severely impaired at AF=70 (top row in Fig. 8). Moreover, we observed that the quality of AF=40 and AF=70 in full 3D strategy is comparable to AF=20 and AF=40 in SpSOS pattern respectively, allowing for an additional 2x shorter scan time. Finally, by comparing Fig. 8 to Fig. 7, we see a larger degradation in image quality for a given acceleration factor in prospective studies. This confirms that in vivo acquisitions are more challenging.

V. Discussions

One key aspect of globally optimized 3D SPARKLING trajectories is that it results in a sampling pattern that enforces variable density sampling in all the 3 dimensions. We hypothesized that this allows us to efficiently under-sample the k-space acquisitions, thus making it possible to push the acceleration factor to a larger value than what was achieved earlier, while still maintaining a good image quality.

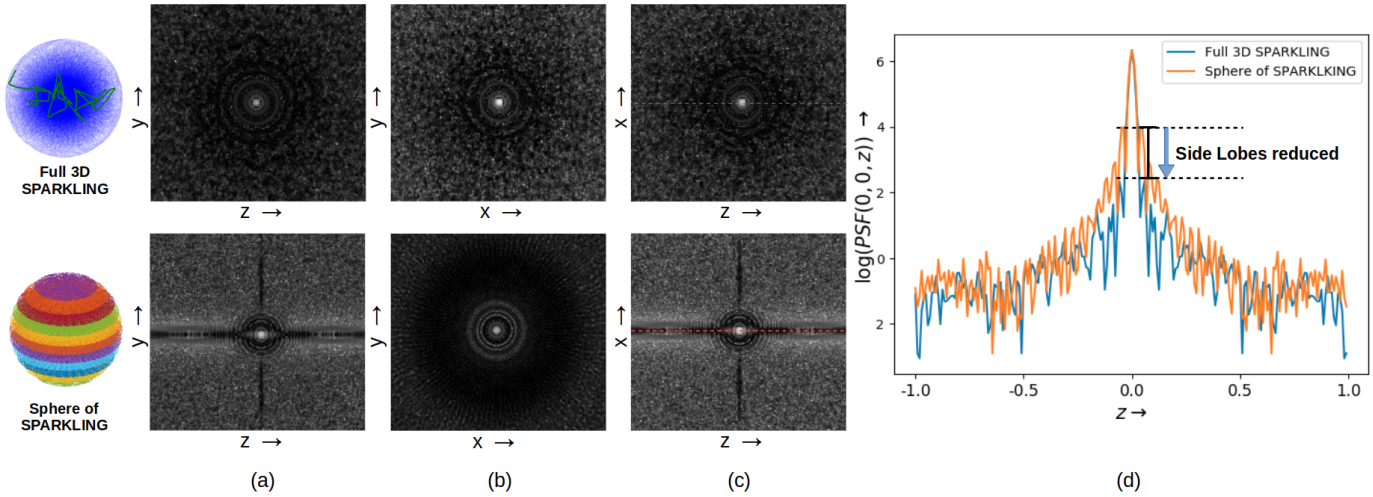

Fig. 6: Comparison of PSFs between Global 3D SPARKLING and spherical stack of 2D SPARKLING (SpSOS) trajectories. The logarithm of 3D PSFs are viewed along the mid-slices in **(a)** coronal plane (0, y, z), **(b)** axial plane (x, y, 0) and **(c)** sagittal plane (x, 0, z). **(d)** The PSFs are compared in logarithmic scale along the z direction.

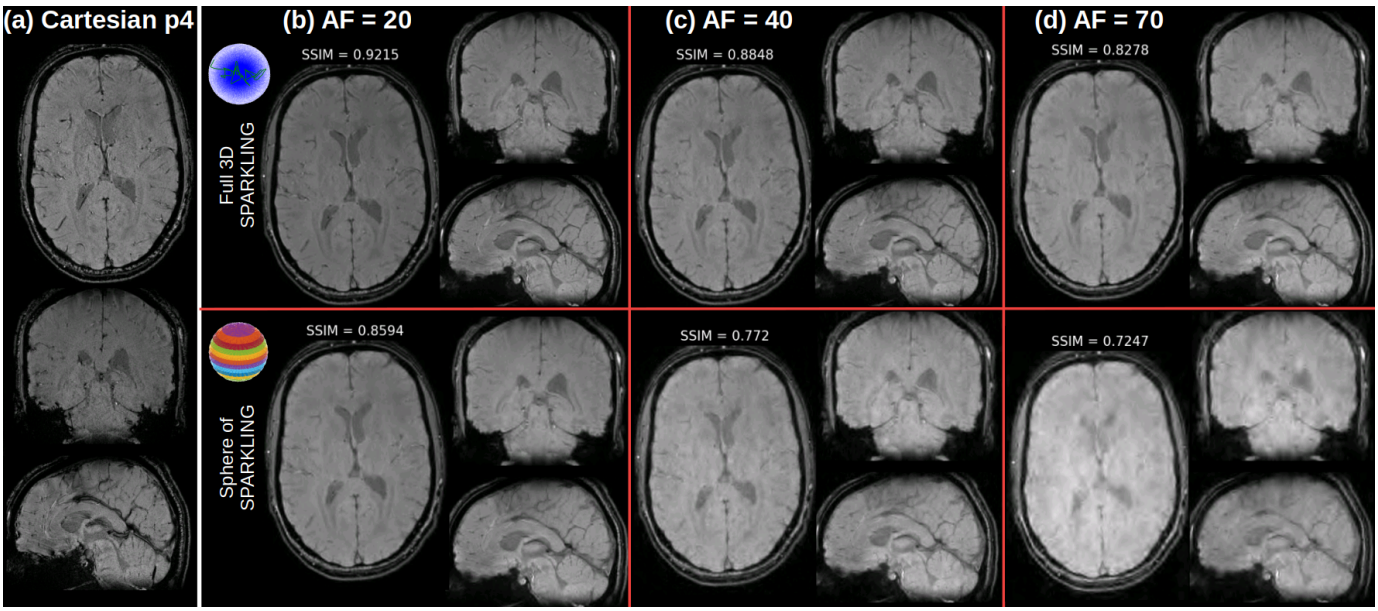

Fig. 7: Comparison of retrospective results for globally optimized 3D SPARKLING (top row) and SpSOS (bottom row) for varying acceleration factors (from left to right, AF=20 (b), 40 (c) and 70 (d)) on in vivo human brain scans. SSIM scores are reported for each setup.

We see that in globally optimized 3D SPARKLING trajectories, the PSF shape is preserved as we varied the number of shots (N_c), while maintaining the same target sampling distribution. However, as the number of shots is getting smaller, the peak amplitude of the PSF diminishes, thereby translating to just lower SNR in the resulting reconstructed images.

Further, a key aspect of a full 3D trajectory is that each shot is constrained to pass through the center of k-space at echo time. This ensures that we obtain the lower frequency image content repeatedly, due to which we can potentially use these trajectories for motion correction. Also, as the center of k-space is visited repeatedly at different time intervals in scan, this allows for easy adaptability of this trajectory for dynamic imaging like functional MRI. Such a trajectory can also be used for correcting certain artifacts due to some physiological signals (like breathing).

Also, in Fig. 4 we observed that a larger perturbation results in a much better k-space coverage, allowing us to reach a better minimizer to the original optimization problem. This happens mainly because the original problem described in (4) is non-convex and we used a locally convergent optimization algorithm. Hence the final solution heavily depends on its initialization.

One limitation of the resulting reconstructed MR im-

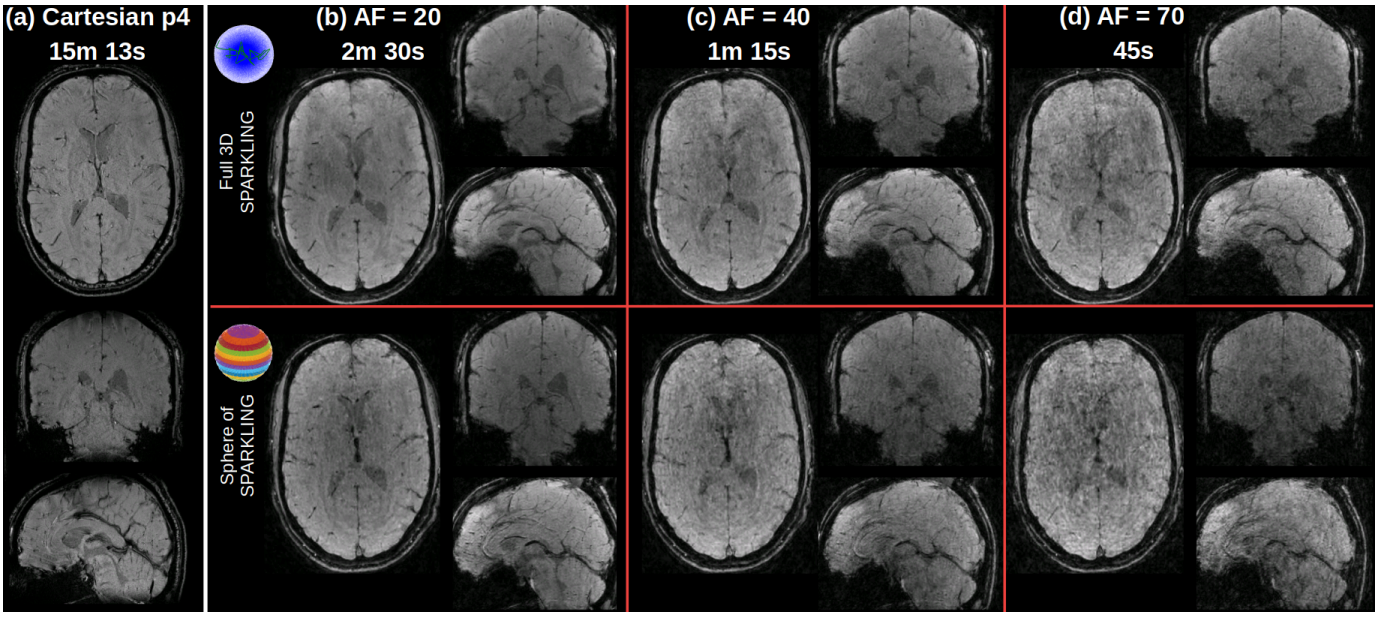

Fig. 8: Comparison of prospective results for globally optimized 3D SPARKLING (top row) and `SpSOS` (bottom row) for varying acceleration factors (from left to right, AF=20 (b), 40 (c) and 70 (d)) on in vivo human brain scans. The reference Cartesian image (AF=4) is shown in (a). Acquisition time is reported for each setup.

ages is that they heavily depend on the target sampling distribution. We obtained our results by parameterizing this distribution, thereby optimizing for its parameters using a grid search on phantom data. However, optimal parameters for the phantom are not necessarily the best for in vivo human acquisitions. To overcome this limitation, future work intends to couple SPARKLING with the learning of the target sampling density from the magnitude spectrum of human brain MR images as proposed in [30]. Akin to this work, we could also jointly optimize for the acquisition (sampling pattern) and reconstruction schemes (regularization parameters) under MR hardware and imaging contrast constraints, either in a bilevel optimization variational framework [30] or using deep learning approaches [31], [32], [54].

VI. Conclusion

In this paper, we proposed the globally optimized 3D SPARKLING k-space trajectories for accelerated high resolution 3D T2*-weighted imaging and demonstrated its superiority over the previously proposed stacking strategies on phantom and in vivo human brain data at 3 Tesla. We discussed the major computational bottlenecks that prevented us earlier from proceeding towards these full 3D trajectories. We then derived some implementations (GPU and multi-CPU) that helped us in massively accelerating the original algorithm. Our results showed that a 600µm isotropic scan on the NIST phantom is achievable in 45s, whereas 2m30s is required to reach image quality comparable to GRAPPA-4 parallel imaging. Overall, this is a significant step forward for CS acquisitions in MRI. Future work will be devoted to the extension to 4D imaging, namely for fMRI.

Acknowledgements

We acknowledge the financial support of the Cross-Disciplinary Program on Numerical Simulation of CEA (SILICOSMIC project, PI: P. Ciuciu), the French Alternative Energies and Atomic Energy Commission. This work was granted access to the HPC resources of TGCC in France under the allocation 2019-GCH0424 made by GENCI. Pierre Weiss was supported by the ANR JCJC Optimization on Measures Spaces ANR-17-CE23-0013-01 and the ANR-3IA Artificial and Natural Intelligence Toulouse Institute.

References

[1] M. Lustig, D. Donoho, and J. M. Pauly, "Sparse MRI: The application of compressed sensing for rapid MR imaging," *Magn. Reson. Med.*, vol. 58, no. 6, pp. 1182–1195, 2007.

[2] G. Puy, P. Vandergheynst, and Y. Wiaux, "On variable density compressive sampling," *IEEE Signal Process. Lett.*, vol. 18, no. 10, pp. 595–598, 2011.

[3] N. Chauffert, P. Ciuciu, J. Kahn, and P. Weiss, "Variable density sampling with continuous trajectories. Application to MRI," *SIAM J. Imag. Sci.*, vol. 7, no. 4, pp. 1962–1992, Nov. 2014.

[4] B. Adcock, A. C. Hansen, C. Poon, and B. Roman, "Breaking the coherence barrier: A new theory for compressed sensing," in *Forum of Math., Sigma*, vol. 5. Cambridge University Press, 2017.

[5] C. Boyer, J. Bigot, and P. Weiss, "Compressed sensing with structured sparsity and structured acquisition," *Appl. Comput. Harmon. Anal.*, vol. 46, no. 2, pp. 312–350, 2019.

[6] C. Ahn, J. Kim, and Z. Cho, "High-speed spiral-scan echo planar NMR imaging-I," *IEEE Trans. Med. Imag.*, vol. 5, no. 1, pp. 2–7, 1986.

[7] C. H. Meyer, B. S. Hu, D. G. Nishimura, and A. Macovski, "Fast spiral coronary artery imaging," *Magn. Reson. Med.*, vol. 28, no. 2, pp. 202–213, 1992.

[8] J. I. Jackson, D. G. Nishimura, and A. Macovski, "Twisting radial lines with application to robust magnetic resonance imaging of irregular flow," *Magn. Reson. Med.*, vol. 25, no. 1, pp. 128–139, 1992.

[9] D. C. Noll, "Multishot rosette trajectories for spectrally selective MR imaging," *IEEE Trans. Med. Imag.*, vol. 16, no. 4, pp. 372–377, 1997.

[10] C. S. Law and G. H. Glover, "Interleaved spiral-in/out with application to functional MRI (fMRI)," *Magn. Reson. Med.*, vol. 62, no. 3, pp. 829–834, 2009.

[11] M. Lustig, J. H. Lee, D. L. Donoho, and J. M. Pauly, "Faster imaging with randomly perturbed, under-sampled spirals and ℓ_1 reconstruction," in *Proc. 13th ISMRM*, Miami Beach, FL, USA, 2005, p. 685.

[12] A. Bilgin, T. Trouard, A. Gmitro, and M. Altbach, "Randomly perturbed radial trajectories for compressed sensing MRI," in *Proc. 16th ISMRM*, Toronto, Canada, 2008, p. 3152.

[13] H. Wang, X. Wang, Y. Zhou, Y. Chang, and Y. Wang, "Smoothed random-like trajectory for compressed sensing MRI," in *Conf. Proc. IEEE Eng. Med. Biol. Soc.* IEEE, 2012, pp. 404–407.

[14] C. Lazarus, P. Weiss, N. Chauffert, F. Mauconduit, L. El Gueddari, C. Destrieux, I. Zemmoura, A. Vignaud, and P. Ciuciu, "SPARKLING: variable-density k-space filling curves for accelerated T2*-weighted MRI," *Magn. Reson. Med.*, vol. 81, no. 6, pp. 3643–3661, 2019.

[15] S. S. Vasanawala, M. T. Alley, B. A. Hargreaves, R. A. Barth, J. M. Pauly, and M. Lustig, "Improved pediatric MR imaging with compressed sensing," *Radiology*, vol. 256, no. 2, pp. 607–616, 2010.

[16] P. E. Larson, P. T. Gurney, and D. G. Nishimura, "Anisotropic field-of-views in radial imaging," *IEEE Trans. Med. Imag.*, vol. 27, no. 1, pp. 47–57, 2008.

[17] P. Irarrazabal and D. G. Nishimura, "Fast three dimensional magnetic resonance imaging," *Magn. Reson. Med.*, vol. 33, no. 5, pp. 656–662, 1995.

[18] F. E. Boada, J. S. Gillen, G. X. Shen, S. Y. Chang, and K. R. Thulborn, "Fast three dimensional sodium imaging," *Magn. Reson. Med.*, vol. 37, no. 5, pp. 706–715, 1997.

[19] K. M. Johnson, "Hybrid radial-cones trajectory for accelerated MRI," *Magn. Reson. Med.*, vol. 77, no. 3, pp. 1068–1081, 2017.

[20] B. M. Dale, J. S. Lewin, and J. L. Duerk, "Optimal design of k-space trajectories using a multi-objective genetic algorithm," *Magn. Reson. Med.*, vol. 52, no. 4, pp. 831–841, 2004.

[21] R. Mir, A. Guesalaga, J. Spiniak, M. Guarini, and P. Irarrazaval, "Fast three-dimensional k-space trajectory design using missile guidance ideas," *Magn. Reson. Med.*, vol. 52, no. 2, pp. 329–336, 2004.

[22] C. Kumar Anand, A. Thomas Curtis, and R. Kumar, "Durga: A heuristically-optimized data collection strategy for volumetric magnetic resonance imaging," *Eng. Optim.*, vol. 40, no. 2, pp. 117–136, 2008.

[23] H. K. Song and L. Dougherty, "Dynamic MRI with projection reconstruction and KWIC processing for simultaneous high spatial and temporal resolution," *Magn. Reson. Med.*, vol. 52, no. 4, pp. 815–824, 2004.

[24] W. Lin, J. Guo, M. A. Rosen, and H. K. Song, "Respiratory motion-compensated radial dynamic contrast-enhanced (DCE)-MRI of chest and abdominal lesions," *Magn. Reson. Med.*, vol. 60, no. 5, pp. 1135–1146, 2008.

[25] D. R. Thedens, P. Irarrazaval, T. S. Sachs, C. H. Meyer, and D. G. Nishimura, "Fast magnetic resonance coronary angiography with a three-dimensional stack of spirals trajectory," *Magn. Reson. Med.*, vol. 41, no. 6, pp. 1170–1179, 1999.

[26] C. Lazarus, P. Weiss, L. El Gueddari, F. Mauconduit, A. Massire, M. Ripart, A. Vignaud, and P. Ciuciu, "3D variable-density SPARKLING trajectories for high-resolution T2*-weighted magnetic resonance imaging," *NMR Biomed.*, p. e4349, 2020.

[27] L. Baldassarre, Y.-H. Li, J. Scarlett, B. Gözcü, I. Bogunovic, and V. Cevher, "Learning-based compressive subsampling," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 4, pp. 809–822, 2016.

[28] B. Gözcü, R. K. Mahabadi, Y.-H. Li, E. Ilıcak, T. Çukur, J. Scarlett, and V. Cevher, "Learning-based compressive MRI," *IEEE Trans. Med. Imag.*, vol. 37, no. 6, pp. 1394–1406, 2018.

[29] C. D. Bahadir, A. Q. Wang, A. V. Dalca, and M. R. Sabuncu, "Deep-learning-based optimization of the under-sampling pattern in MRI," *IEEE Trans. Comput. Imaging*, vol. 6, pp. 1139–1152, 2020.

[30] F. Sherry, M. Benning, J. C. De los Reyes, M. J. Graves, G. Maierhofer, G. Williams, C.-B. Schönlieb, and M. J. Ehrhardt, "Learning the sampling pattern for MRI," *IEEE Trans. Med. Imag.*, 2020.

[31] T. Weiss, O. Senouf, S. Vedula, O. Michailovich, M. Zibulevsky, and A. Bronstein, "PILOT: Physics-informed learned optimal trajectories for accelerated MRI," *arXiv:1909.05773v4*, Aug. 2020.

[32] S. Vedula, O. Senouf, and A. Bronstein, "3D FLAT: Feasible Learned Acquisition Trajectories for Accelerated MRI," in *Machine Learning for Medical Image Reconstruction: 3rd Intern. WS MLMIR 2020, Held in Conjunction with MICCAI 2020*. Lima, Peru: Springer Nature, Oct. 2020, p. 3.

[33] N. Chauffert, P. Weiss, J. Kahn, and P. Ciuciu, "A projection algorithm for gradient waveforms design in MRI," *IEEE Trans. Med. Imag.*, vol. 35, no. 9, pp. 2026–2039, Sep. 2016.

[34] C. Boyer, N. Chauffert, P. Ciuciu, J. Kahn, and P. Weiss, "On the generation of sampling schemes for MRI," *SIAM J. Imag. Sci.*, vol. 9, no. 4, pp. 2039–2072, 2016.

[35] N. Chauffert, P. Ciuciu, J. Kahn, and P. Weiss, "A projection method on measures sets," *Constr. Approx.*, vol. 45, no. 1, pp. 83–111, 2017.

[36] C. Lazarus, M. März, and P. Weiss, "Correcting the side effects of ADC filtering in MR image reconstruction," *J. Math. Imaging Vis.*, pp. 1–14, 2020.

[37] M. Gräf, D. Potts, and G. Steidl, "Quadrature errors, discrepancies, and their relations to halftoning on the torus and the sphere," *SIAM J. Sci. Comput.*, vol. 34, no. 5, pp. A2760–A2791, 2012.

[38] C. Schmaltz, P. Gwosdek, A. Bruhn, and J. Weickert, "Electrostatic halftoning," in *Computer Graphics Forum*, vol. 29, no. 8. Wiley Online Library, 2010, pp. 2313–2327.

[39] M. Ehler, M. Gräf, S. Neumayer, and G. Steidl, "Curve based approximation of measures on manifolds by discrepancy minimization," *arXiv:1910.06124*, 2019.

[40] T. Teuber, G. Steidl, P. Gwosdek, C. Schmaltz, and J. Weickert, "Dithering by differences of convex functions," *SIAM J. Imag. Sci.*, vol. 4, no. 1, pp. 79–108, 2011.

[41] B. Charlier, J. Feydy, J. A. Glaunès, F.-D. Collin, and G. Durif, "Kernel operations on the GPU, with autodiff, without memory overflows," *arXiv:2004.11127*, 2020.

[42] W. Fong and E. Darve, "The black-box fast multipole method," *J. Comput. Phys.*, vol. 228, no. 23, pp. 8712–8725, 2009.

[43] R. Wang, C. Chen, J. Lee, and E. Darve, "PBBFMM3D: a parallel black-box fast multipole method for non-oscillatory kernels," *CoRR*, vol. abs/1903.02153, 2019. [Online]. Available: http://arxiv.org/abs/1903.02153

[44] W. Fong and E. Darve, "The black-box fast multipole method," *J. Comput. Phys.*, no. 228, 2009.

[45] J. Barzilai and J. M. Borwein, "Two-Point Step Size Gradient Methods," *IMA J. Numer. Anal.*, vol. 8, no. 1, pp. 141–148, 1988.

[46] R. Okuta, Y. Unno, D. Nishino, S. Hido, and Crissman, "Cupy : A numpy-compatible library for nvidia gpu calculations," 2017.

[47] L. Lebrat, F. de Gournay, J. Kahn, and P. Weiss, "Optimal Transport Approximation of 2-Dimensional Measures," *SIAM J. Imag. Sci.*, vol. 12, no. 2, pp. 762–787, Jan. 2019.

[48] M. A. Griswold, P. M. Jakob, R. M. Heidemann, M. Nittka, V. Jellus, J. Wang, B. Kiefer, and A. Haase, "Generalized autocalibrating partially parallel acquisitions (GRAPPA)," *Magn. Reson. Med.*, vol. 47, no. 6, pp. 1202–1210, 2002.

[49] J. Zbontar, F. Knoll, A. Sriram, T. Murrell, Z. Huang, M. J. Muckley, A. Defazio, R. Stern, P. Johnson, M. Bruno *et al.*, "fastMRI: An open dataset and benchmarks for accelerated MRI," *arXiv:1811.08839*, 2018.

[50] J. G. Pipe and P. Menon, "Sampling density compensation in MRI: Rationale and an iterative numerical solution," *Magn. Reson. Med.*, vol. 41, no. 1, pp. 179–186, 1999.

[51] F. Knoll, A. Schwarzl, C. Diwoky, and D. K. Sodickson, "gpuNUFFT-An open source GPU library for 3D regridding with direct Matlab interface," in *Proc. 22nd ISMRM*, 2014, p. 4297.

[52] L. E. Gueddari, C. GR, Z. Ramzi, S. Farrens, S. Starck, A. Grigis, J.-L. Starck, and P. Ciuciu, "PySAP-MRI: a Python Package for MR Image Reconstruction," in *ISMRM WS on Data Sampling and Image Reconstruction*, Sedona, AZ, United States, Jan. 2020. [Online]. Available: https://hal.inria.fr/hal-02399267

[53] L. El Gueddari, C. Lazarus, H. Carrié, A. Vignaud, and P. Ciuciu, "Self-calibrating nonlinear reconstruction algorithms for variable density sampling and parallel reception MRI," in *10th IEEE SAM Signal Processing WS*, Sheffield, UK, Jul. 2018, pp. 415–419.

[54] Z. Ramzi, P. Ciuciu, and J.-L. Starck, "Benchmarking MRI Reconstruction Neural Networks on Large Public Datasets," *Applied Sciences*, vol. 10, no. 5, p. 1816, 2020.

Supplementary Material – Globally optimized 3D SPARKLING trajectories for high-resolution T2*-weighted Magnetic Resonance Imaging

Chaithya GR, Pierre Weiss, Aurélien Massire, Alexandre Vignaud, and Philippe Ciuciu *Senior Member, IEEE*

S1. Multishot projection Step

We briefly extend the single shot projection step to N_c shots. Particularly, we would like to project the complete sampling pattern $\mathbf{K} = [\mathbf{k}_1, \mathbf{k}_2, \ldots, \mathbf{k}_{N_c}]$ such that each shot $\mathbf{k}_i$ complies with the hardware constraints, $\mathcal{Q}_{\alpha,\beta}$ defined in (2). From [33], the projection step for a single shot $\mathbf{k}$ is given by:

$$\mathbf{k}^* = \Pi_{\mathcal{Q}_{\alpha,\beta}}(\mathbf{k}) = \arg\min_{\boldsymbol{s}\in\mathcal{Q}_{\alpha,\beta}} \frac{1}{2}\|\boldsymbol{s}-\boldsymbol{k}\|_2^2$$

We can now further generalize above equation to project all shots as:

$$\mathbf{K}^* = \Pi_{\mathcal{Q}_{\alpha,\beta}^{N_c}}(\mathbf{K}) = \left[\Pi_{\mathcal{Q}_{\alpha,\beta}}(\mathbf{k}_1), \ldots, \Pi_{\mathcal{Q}_{\alpha,\beta}}(\mathbf{k}_{N_c})\right]$$

where, we can use accelerated proximal gradient descent algorithm as described in [33] to carry out individual projections of each shot in parallel.

S2. Image reconstruction

The MR image reconstruction of 3D multi-channel data acquired from phased array receiver coils was carried out using Self Calibrating MR Reconstruction [53] in the synthesis formulation by solving for the wavelet coefficients $\mathbf{z}$ in (S1).

$$\widehat{\mathbf{z}} = \underset{\mathbf{z}\in\mathbb{C}^{N_x\times N_y\times N_z}}{\operatorname{argmin}} \frac{1}{2}\sum_{\ell=1}^{L}\|F_\Omega \mathbf{S}_\ell \boldsymbol{\Psi}^*\mathbf{z} - \mathbf{y}_\ell\|_2^2 + \lambda\|\mathbf{z}\|_1 \quad \text{(S1)}$$

where the number of channels was $L = 44$ and $N_x = N_y = 384$ and $N_z = 208$. Here the data fidelity is enforced with SENSE operator $F_\Omega \mathbf{S}_\ell$, where F_Ω is the NUFFT operator and $\mathbf{S}_\ell$ is sensitivity map for ℓ^{th} channel estimated by density compensated adjoint of the 20 percent of acquired k-space center (see details in [53]). $\lambda > 0$ is the regularization parameter for ℓ_1-sparsity was promoted on the image in the wavelet domain $\boldsymbol{\Psi}$. For our reconstructions, we used Symlet 8 wavelet with 4 scales for $\boldsymbol{\Psi}$. The regularization parameter λ was grid searched between $(10^{-10}, 10^0)$ while maximizing for the reconstruction quality using SSIM score in retrospective reconstruction. As the sampling is Non-Cartesian in 3D, the reconstruction model is severely ill-posed with the forward operator $F_\Omega \mathbf{S}_\mathbf{l} \boldsymbol{\Psi}^*$ having a high condition number, thereby impacting the convergence speed. In order to accelerate convergence, we preconditioned the k-space using density compensation. This translates to adding a preconditioner D in the classical proximal gradient descent algorithm (here we used FISTA):

$$\mathbf{z}^{(k+1)} = \text{soft}_{\lambda\tau}\left(\mathbf{z}^{(k)} - \tau\sum_{\ell=1}^{L}\boldsymbol{\Psi}\mathbf{S}_\ell^* F_\Omega^H D\left(F_\Omega \mathbf{S}_\ell \boldsymbol{\Psi}^*\mathbf{z}^{(k)} - \mathbf{y}_\ell\right)\right)$$

where $\text{soft}_{\lambda\tau}$ is the soft threshold operator and τ is the step size. The density compensators D were obtained by 10 iterations of method described in [50]. The final MR image is $\widehat{\mathbf{x}} \in \mathbb{C}^{N_x\times N_y\times N_z}$ given by $\boldsymbol{\Psi}^*\widehat{\mathbf{z}}$. As the raw data is large (for AF=20, $p = 8,388,608$ k-space points), we needed to utilize memory efficient methods to carry out the SENSE operation. For this, we implemented python wrappers for gpuNUFFT [51] which implements the NUFFT operator in CUDA and utilizes cuBLAS and cuFFT libraries to be efficient in speed and memory. The implementation of the reconstruction was completely done using PySAP-MRI[1] [52] an MR plugin to PySAP. Despite being a 3D reconstruction problem, the computation time was just 15-30 minutes on a machine with same specifications as described earlier in Section III.

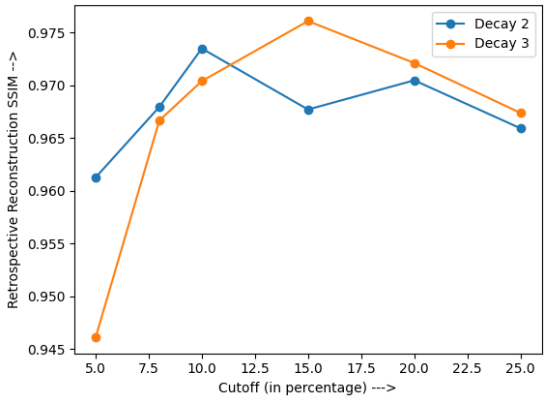

Fig. S1: SSIM score for retrospective reconstructions on the NIST phantom as a function of varying density parameterizations (cutoff C in x-axis, decay D in y-axis).

[1] https://github.com/CEA-COSMIC/pysap-mri

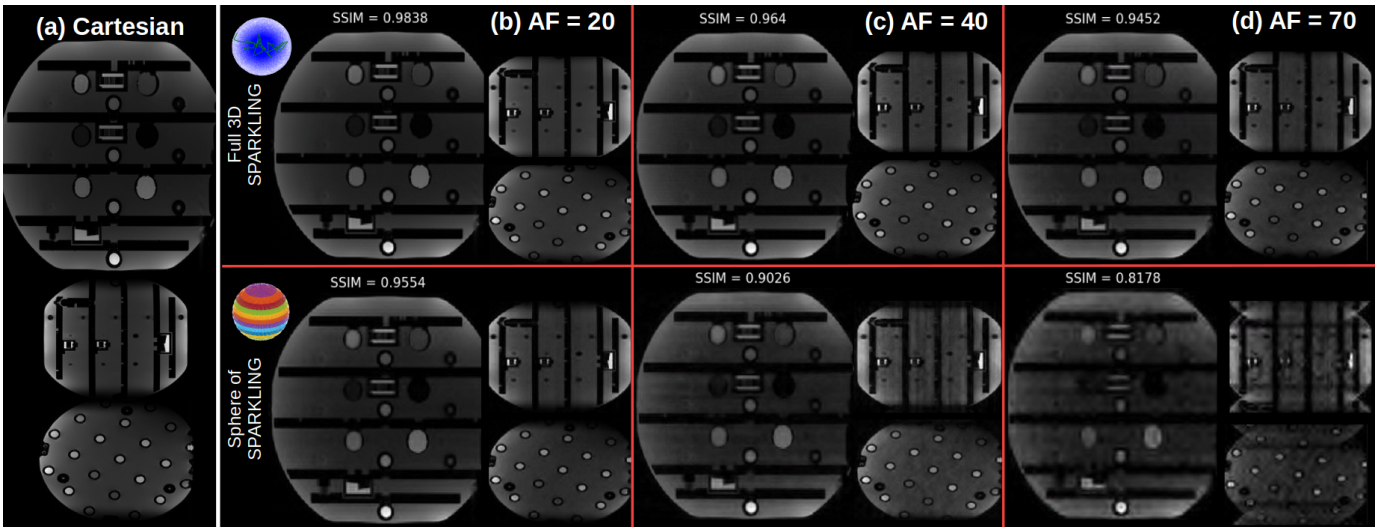

Fig. S2: Comparison of retrospective results for globally optimized 3D SPARKLING (top row) and `SpSOS` (bottom row) for varying acceleration factors (from left to right, AF=20 (b), 40 (c) and 70 (d)) on the NIST phantom. The reference Cartesian image (AF=4) is shown in (a). Acquisition time and SSIM score are reported for each setup. Global 3D SPARKLING gives superior results than `SpSOS` approach which starts to get worse at AF=40 with some blurring and at AF=70 we see strong folding artifacts in the sagittal plane (top right view in each box).