Supporting Information: Automated Design of Pulse Sequences for Magnetic Resonance Fingerprinting using Physics-Inspired Optimization

Stephen P. Jordan[1], Siyuan Hu[2], Ignacio Rozada[3], Debra F. McGivney[2], Rasim Boyacioğlu[4], Darryl C. Jacob[5], Sherry Huang[2], Michael Beverland[1], Helmut G. Katzgraber*[6,1], Matthias Troyer[1], Mark A. Griswold[4], and Dan Ma[2,7]

[1]Microsoft Quantum, Redmond WA
[2]Biomedical Engineering, Case Western Reserve University, Cleveland OH
[3]1QBit, Vancouver BC
[4]Radiology Department, Case Western Reserve University, Cleveland OH
[5]Texas A & M University, College Station TX
[6]Professional Services, Amazon Web Services, Seattle WA
[7]To whom correspondence should be addressed: dan.ma@case.edu

1 Spin Dynamics

The dynamics of nuclear spins in a magnetic field are described phenomenologically by the Bloch equation [1]. Given the magnetic field as a function of time at a given location, an initial condition for the spin, and parameters T_1, T_2, the Bloch equation predicts the state of the spin at subsequent times. Here, we consider MRF pulse sequences for two-dimensional slices through brain tissue consisting of 256×256 voxels, each 1.2mm by 1.2mm. (The thickness is not explicitly modeled but in vivo is approximately 5mm.) Correspondingly, our mathematical model consists of a value of T_1, T_2, and m_0 (proton density) assigned to each voxel. The proton density only affects the magnetization of the voxel by acting as a time-independent multiplicative factor. For a given pulse sequence we solve the Bloch equations (in the hard-pulse approximation) to obtain magnetization vs. time for each (T_1, T_2) pair appearing within the voxels of the simulated tissue distribution.

Because the static B_0 field is not perfectly spatially homogeneous, different spins throughout the brain precess at slightly different rates. This induces unwanted artifacts in the resulting magnetic resonance images. The purpose of the spoiling gradient is to reduce sensitivity to B_0 inhomogeneity by effectively averaging away the x and y components of the magnetization at the end of each TR. For this to be effective, the spoiling gradient needs to be sufficiently strong that the difference in precession angle between different spins within the same voxel is at least 2π. In our computer model we assign $F = 400$ spins to each voxel, which get rotated by angles uniformly spaced between $-\pi$ and π during the spoiling gradient. At the measurement stage, the magnetization associated to a given voxel is obtained by averaging over these 400 spins, conventionally referred to as isochromats.

*The work of H. G. K. was performed before joining Amazon Web Services.

In the hard pulse approximation, the RF pulses that rotate the spins on the Bloch sphere are considered instantaneous. Consequently, the exponential decay dictated by T_1 and T_2 is not intermixed with these rotations. Let s denote the index of a given TR. The first step in a given TR is to apply an RF pulse implementing a rotation on the Bloch sphere according to polar angle α_s followed by azimuthal angle θ_s. That is, the magnetization vector $\vec{m}_{s,j}$ of the j^{th} isochromat at the s^{th} timestep undergoes the transformation

$$\vec{m}_{s,j}(x,y) \leftarrow R_s \; \vec{m}_{s,j}(x,y) \tag{1}$$

where R_s is the rotation matrix

$$R_s = \begin{bmatrix} \cos^2\theta_s + \cos\alpha_s \sin^2\theta_s & (\cos\alpha_s - 1)\cos\theta_s \sin\theta_s & \sin\alpha_s \cos\theta_s \\ (\cos\alpha_s - 1)\cos\theta_s \sin\theta_s & \cos\alpha_s \cos^2\theta_s + \sin^2\theta_s & \cos\theta_s \sin\alpha_s \\ -\sin\alpha_s \sin\theta_s & -\cos\theta_s \sin\alpha_s & \cos\alpha_s \end{bmatrix}. \tag{2}$$

The next step in the s^{th} TR is to wait for time TE_s. Left undisturbed for duration TE_s the magnetization will relax toward equilibrium according to

$$\vec{m}_{s,j}(x,y) \leftarrow D(\text{TE}_s) \; \vec{m}_{s,j}(x,y) + \vec{v}(\text{TE}_s). \tag{3}$$

Where

$$D(t) = \begin{bmatrix} e^{-t/T_2} & 0 & 0 \\ 0 & e^{-t/T_2} & 0 \\ 0 & 0 & e^{-t/T1} \end{bmatrix} \tag{4}$$

and

$$\vec{v}(t) = \begin{bmatrix} 0 \\ 0 \\ 1 - e^{-t/T_1} \end{bmatrix}. \tag{5}$$

Conventional magnetic resonance imaging hardware cannot measure the z-component of magnetization. Furthermore, the measurement cannot distinguish isochromats within a voxel but instead is sensitive only to their average magnetization. In keeping with widely used conventions in the magnetic resonance literature we express this average magnetization in the xy plane in a given voxel as a complex number whose real part is the x-component of the magnetization and whose imaginary part is the y-component, as follows.

$$m_s(x,y) \leftarrow \frac{1}{F} \sum_{j=0}^{F-1} \left([\vec{m}_{s,j}(x,y)]_x + i \; [\vec{m}_{s,j}(x,y)]_y \right). \tag{6}$$

Next, another idle waiting period is imposed for the remaining time $\text{TR}_s - \text{TE}_s$. Hence, relaxation dynamics again occurs in accordance with (3).

$$\vec{m}_{s,j}(x,y) \leftarrow D(\text{TR}_s - \text{TE}_s) \; \vec{m}_{s,j}(x,y) + \vec{v}(\text{TR}_s - \text{TE}_s). \tag{7}$$

Lastly, a spoiling gradient is applied, which rotates the different isochromats within the voxel by different angles, determined by their position along the gradient. That is,

$$\vec{m}_{s+1,j}(x,y) \leftarrow S_j \; \vec{m}_{s,j}(x,y), \tag{8}$$

$$
\begin{array}{rcll}
\vec{m}_{s,j}(x,y) & \leftarrow & R_s\ \vec{m}_{s,j}(x,y) & \text{apply rotation specified by } (\alpha_s, \theta_s) \\
\vec{m}_{s,j}(x,y) & \leftarrow & D(\mathrm{TE}_s)\ \vec{m}_{s,j}(x,y) + \vec{v}(\mathrm{TE}_s) & \text{wait for time } \mathrm{TE}_s \\
m_s(x,y) & \leftarrow & \frac{1}{F}\sum_{j=0}^{F-1}\left([\vec{m}_{s,j}(x,y)]_x + i\ [\vec{m}_{s,j}(x,y)]_y\right) & \text{measurement averages isochromats} \\
\vec{m}_{s,j}(x,y) & \leftarrow & D(\mathrm{TR}_s - \mathrm{TE}_s)\ \vec{m}_{s,j}(x,y) + \vec{v}(\mathrm{TR}_s - \mathrm{TE}_s) & \text{wait for remainder of } \mathrm{TR}_s \text{ duration} \\
\vec{m}_{s+1,j}(x,y) & \leftarrow & S_j\ \vec{m}_{s,j}(x,y) & \text{apply spoiling gradient}
\end{array}
$$

Figure 1: Summary of spin dynamics within a voxel of given T_1 and T_2 during the s^{th} TR of a FISP pulse sequence.

where

$$
S_j = \begin{bmatrix} \cos\phi_j & -\sin\phi_j & 0 \\ \sin\phi_j & \cos\phi_j & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{9}
$$

This mathematical model of a FISP pulse is summarized in figure 1.

For a sequence with an initial inversion pulse we take the initial state of $\vec{m}_j$ to be $(0, 0, -0.95)$ for all j.

2 Digital Phantom

For an MRF scan we wish to minimize T_1 error, T_2 error, and scan time. We use a weighted combination of the predicted values for these quantities as a cost function to minimize. Magnetic resonance scans are complicated processes involving many sources of random and systematic error. Modeling these to obtain a cost function that accurately predicts real-world in vivo performance of pulse sequences is highly nontrivial. Here, we use a model that incorporates random error due to thermal noise, and systematic errors due to Fourier undersampling and phase inhomogeneity. Such phase inhomogoneity is commonly observed and could be caused by B_0 or B_1 inhomogeneity or motion. (A preliminary report on our cost function appears in [2]).

At least ideally, the RF pulses in a magnetic resonance scan are uniform throughout the xy plane and induce the same rotation on the Bloch sphere to every spin throughout the targeted slice of tissue. Sensitivity to spatial variation is obtained during the measurement step through the use of magnetic field gradients, which allow the extraction of Fourier components of the magnetization in the xy-plane. Because spins at different locations have different values of T_1 and T_2 depending on tissue type, the magnetization vs time will vary spatially, and this variation can be used to map the distribution of different tissues. To obtain complete information about the magnetization after each pulse, one would need to measure a number of Fourier components equal to the number of voxels in the desired image. However, to obtain shorter scan durations it is typical in practice to measure only a much smaller number of Fourier components after each pulse.

In this study we use a high level of Fourier undersampling to achieve short scan times. Specifically, we use a variable density spiral readout in Fourier space [3] in the "one-shot" setting with undersampling factor $R = 48$. That is, after each RF pulse, we measure Fourier components along a single spiral, repeatedly cycling through a sequence of 48 spirals, which collectively provide full coverage of Fourier space. Superimposing all 48 spirals does not yield a set of points in Fourier space arranged according to a uniform grid. Rather, the density of samples in Fourier space varies slightly.

To infer spatial images, this non-uniform sample density is compensated for using a Non-Uniform Fast Fourier Transform (NUFFT) [4].

The mapping from actual xy-magnetization at a given time to the measured Fourier components is given by a Fourier transform, which is linear. The mapping from the measured Fourier components to the inferred xy-magnetization is given by a non-uniform Fourier Transform, which is also linear. Therefore, the inferred magnetization in position space is expressible as a linear combination of contributions from the actual magnetization in position space. That is,

$$I_s(x,y) = \sum_{x',y'} U_s(x,y,x',y') m_s(x',y'). \tag{10}$$

Here, $m_s(x',y')$ is the actual magnetization at location (x',y') at the time of s^{th} measurement, $I_s(x,y)$ is the magnetization at location (x,y) inferred based on the results of the s^{th} measurement, and $U_s(x,y,x',y')$ is the point spread function defined by the non-uniform Fourier transform applied to the set Fourier components measured in the s^{th} step. (In our case, the set of Fourier components measured in the s^{th} step are those lying within the j^{th} spiral trajectory, where j is given by s reduced modulo 48. Thus $U_s(x,y,x',y') = U_{s+48}(x,y,x',y')$.) Here we are taking $I_s(x,y)$ and $m_s(x',y')$ to be complex numbers as noted earlier.

In magnetic resonance fingerprinting, one discretizes the range of T_1 and T_2 that might be found in human tissues into a finite set of values. Given a pulse sequence, one then computes, for each (T_1,T_2) pair in this set, the corresponding magnetization vs. measurement index s. This list of potential magnetization vs. s curves is called a dictionary. After performing a magnetic resonance scan and applying non-uniform Fourier transforms, one obtains estimates of magnetization vs. measurement index for each voxel. Although the individual magnetization estimates for each measurement index have large error due to Fourier undersampling, the time-series across all measurement indices nevertheless contains useful information. The time series for a given voxel can be compared against the entries in the dictionary, and for each voxel one can assign (T_1,T_2) based on the dictionary entry that makes the closest match to the observed signal according to a suitably chosen metric of closeness. Here, following [5], we take the perspective that, for a sequence with n measurements, we can normalize $(I_1(x,y), I_2(x,y), \ldots, I_n(x,y))$ to obtain a unit vector in $\mathbb{C}^n$. The dictionary entries are also normalized to become unit vectors in $\mathbb{C}^n$. For a given voxel one infers (T_1,T_2) to be the values of the dictionary entry whose inner product with $(I_1(x,y), I_2(x,y), \ldots, I_n(x,y))$ has the largest magnitude.

In principle, given a pulse sequence, choice of Fourier-space trajectories, and a model tissue distribution assigning (T_1,T_1,m_0) values to each voxel, one can solve the Bloch equation to obtain $m_s(x',y')$ and solve (10) to obtain $I_s(x,y)$, thereby simulating the effect of Fourier undersampling errors. This data can then be matched against a dictionary of predicted signals to obtain inferred (T_1,T_2) values for each voxel. The inferred (T_1,T_2) values can then be compared against the model tissue distribution to evaluate error in inferred T_1 and T_2 induced by Fourier undersampling. Some pulse sequences will be more robust against Fourier undersampling error than others, and this metric of error can thus be used to construct a cost function to optimize.

Unfortunately, this is not very practical, as most optimization methods need to make a large number of queries to the cost function and evaluation of $I_s(x,t)$ via (10) is somewhat computationally intensive. Large computational savings can be made by taking a somewhat simplified model, as described in [2]. Instead of assigning each voxel in the model brain to a unique (T_1,T_2,m_0) value, we consider an idealized brain in which each voxel in our 256×256 array is one of four types: white

matter ($T_1 = 800\text{ms}, T_2 = 40\text{ms}, m_0 = 0.77$), grey matter ($T_1 = 1400\text{ms}, T_2 = 60\text{ms}, m_0 = 0.86$), cerebrospinal fluid ($T_1 = 3000\text{ms}, T_2 = 2000\text{ms}, m_0 = 1.0$), or air ($m_0 = 0$). Consequently, to compute $I_s(x, y)$, one need only to solve the Bloch equations for grey matter, white matter, and cerebrospinal fluid, and then for each measurement index s, take the corresponding complex linear combination of the three pre-summed point spread functions corresponding to the spatial distributions of these three tissues. That is,

$$I_s(x, y) = U_s^{(\text{GM})}(x, y)m_s^{(\text{GM})} + U_s^{(\text{WM})}(x, y)m_s^{(\text{WM})} + U_s^{(\text{CSF})}(x, y)m_s^{(\text{CSF})} \tag{11}$$

where:

$$U_s^{(\text{GM})}(x, y) = m_0^{(\text{GM})} \sum_{(x',y')\in\text{GM}} U_s(x, y, x', y') \tag{12}$$

and similarly for WM and CSF.

Although spoiling gradients, as used in FISP sequences, mitigate the effects of B_0 inhomogeneity, they do not eliminate spatial inhomogeneities entirely. It has been previously reported in highly undersampled MRF scans that systematic errors in the phase of $m_s(x, y)$ result in shading artifacts in the inferred T_1 and T_2 images, even in FISP sequences [6]. Examples of such shading artifacts are shown in figure 1 of the main text. Currently, this is usually dealt with in postprocessing, by employing methods such as iterative reconstruction [7, 8, 9, 10]. Here, we take a novel approach of optimizing pulse sequences to be intrinsically robust against these errors, thereby producing good quality images directly from inner-product maximizing dictionary matching, without the need for ad hoc corrections.

To optimize for robustness against phase errors we can simply incorporate a representative example of typically observed phase errors into the point spread functions $U_s^{(\text{GM})}(x, y)$, $U_s^{(\text{WM})}(x, y)$, and $U_s^{(\text{CSF})}(x, y)$. The phase errors observed experimentally vary from scan to scan even on the same machine. However, the phase errors tend to be smoothly varying across the field of view and differ between scans mainly in the direction across which they vary. Empirically, as discussed in the results section, we have found that optimizing against a representative example of a phase error tends to yield sequences that are also robust against other phase errors with different orientations. Furthermore, the sequences with smaller shading artifacts in simulation are observed to have smaller shading artifacts in vivo. Examples of simulated phase errors are given in figure 2 of the main text.

After the predicted magnetization time-series are predicted for each voxel, these are checked against a "dictionary" of simulated time-series for a list of possible T_1, T_2 pairs. The value of T_1 and T_2 from the dictionary entry that most closely matches the measured time signal from a given voxel are inferred as the most likely estimates of the ground truth T_1 and T_2 values for that voxel. In this manner a maps of T_1 and T_2 are extracted, in keeping with standard practice in Magnetic Resonance Fingerprinting [5]. At each query, the cost function is given a new pulse sequence, and thus must generate a new MRF dictionary. In our modeling we use a dictionary of 14,996 (T_1, T_2) pairs, discretizing the range

$$\begin{array}{rcccl} 2\text{ms} & \leq & T_1 & \leq & 3000\text{ms} \\ 2\text{ms} & \leq & T_2 & \leq & 2000\text{ms} \\ & & T_2 & \leq & T_1. \end{array} \tag{13}$$

Thus, at each query to the cost function, the Bloch equations must be solved for each of these 14,996 (T_1, T_2) values. Then, after the Fourier undersampling errors have been simulated according to (11), the inner products between the signals calculated for each of the 256×256 voxels (with

the exception of the air voxels) must be calculated with the signals calculated for each of the 14,996 library entries in order to perform the dictionary decoding. These two processes: dictionary generation, and dictionary decoding, are the dominant computational costs in the evaluation of the cost function, with the evaluation of (11) and the evaluation of of the predicted random errors being essentially negligible. Using an optimized multithreaded implementation running on a 24-vcpu virtual machine (Azure NC24) we find that the cost function, for a sequence with 1000 TRs, can be evaluated in 2.0 seconds.

It is unlikely that any mathematical model of a complicated system such as this one will ever be complete. In particular, our digital phantom does not include explicit modeling of slice profile corrections. It also does not directly model of spatial inhomogeneity of B_0 or B_1 magnetic fields or time-dependent effects due, for example, to eddy currents. Rather, these are incorporated via a simple phenomenological model in which we impose a phase that varies quadratically along an arbitrary direction in the xy-plane. The merits of this model are that it can be computed rapidly enough to incorporate into a cost function and it empirically does a good job of qualitatively reproducing the systematic errors observed in a series of in vivo scans that were carried out using a wide variety of pulse sequences on several volunteers. In future work, one could consider incorporating first-principles modeling of additional physical effects into the digital phantom.

3 Model of Random Errors

In addition to Fourier undersampling and phase errors, magnetic resonance scans are also affected by random error due to thermal fluctuations. These are typically modeled as independent identically distributed gaussian errors of mean zero and variance σ^2 added to each measured Fourier coefficient. Analytical formulas for the resulting errors in inferred T_1 and T_2 via dictionary matching are derived in [11, 12]. In addition to σ^2, these errors depend on the rate at which the dictionary entries vary with respect to T_1 and T_2. These formulas show that better robustness is achieved by pulse sequences such that the dictionary entries (thought of as vectors in $\mathbb{C}^n$) vary rapidly as T_1 and T_2 are changed. This is in agreement with general intuition. In fact, prior work has used small inner product between adjacent dictionary entries as a criterion for optimizing pulse sequences [13].

Using the formulas from [11, 12] and a value of σ^2, one can obtain predicted standard deviation on T_1 and T_2 for a given tissue. The value of σ^2 can either be inferred from experimental data or treated as effectively a tunable "weight" factor to adjust the importance of random error relative to systematic error in the optimization. From our simulation of Fourier undersampling artifacts we obtain predicted discrepancies between the theoretical and measured values of T_1 and T_2 associated with each voxel. Averaging over voxels of a given tissue type, we can obtain root-mean-square values of systematic error due to Fourier undersampling for each of the three tissue types in our model. These can be interpreted as standard deviations if one were to select a voxel uniformly at random among all voxels of the given tissue type. Correspondingly, we add these root-mean-squared undersampling errors to the predicted standard deviations to obtain a total predicted error. We thus obtain six numbers:

$$\sigma_t^{(p)} = \sqrt{\left(\nu_t^{(p)}\right)^2 + \left(\eta_t^{(p)}\right)^2} \quad p \in \{T_1, T_2\} \quad t \in \{\mathrm{GM}, \mathrm{WM}, \mathrm{CSF}\}, \tag{14}$$

where $\nu_t^{(p)}$ is the standard deviation in parameter p and tissue t predicted due to thermal noise,

and $\eta_t^{(p)}$ is the root-mean-square error in parameter p and tissue t predicted due to Fourier undersampling.

The sensitivity to thermal noise is affected by the choice of pulse sequence through several mechanisms. First, a sequence that yields larger magnetization in the xy-plane will yield stronger signals and hence better signal to noise ratio. More subtly, some sequences are better than others in terms of how rapidly the dictionary entry (which for a scan with n measurements can be thought of as a vector in $\mathbb{C}^n$) varies as a function of T_1 and T_1. If the dictionary entry varies rapidly as a function of these parameters then the inferred values of these parameters will be less affected by random error in the measured signal. A mathematical analysis of this effect is given in [11, 12].

Although we have motivated the above error model heuristically, it is worth highlighting that it can be derived from a minimal set of assumptions, which do not include any assumption about errors being Gaussian. Specifically, let $\delta_{j,t,p}$ be the (signed) discrepancy between the true value of $p \in \{T_1, T_2\}$ for voxel j of tissue $t \in \{\mathrm{GM}, \mathrm{WM}, \mathrm{CSF}\}$ and the value inferred by dictionary matching. Then, by definition, the mean and variance of $\delta_{j,t,p}$ are

$$\mu_{j,t,p} = \langle \delta_{j,t,p} \rangle \tag{15}$$

$$\nu_{j,t,p}^2 = \langle \delta_{j,t,p}^2 \rangle - \langle \delta_{j,t,p} \rangle^2, \tag{16}$$

respectively. Let N_p be the number of voxels of tissue type p. If we select a voxel uniformly at random among these voxels then the root-mean-squared error is[1]

$$\sigma_t^{(p)} = \sqrt{\frac{1}{N_p} \sum_{j=1}^{N_p} \langle \delta_j^2 \rangle} \tag{17}$$

By (15) and (16)

$$\sigma_t^{(p)} = \sqrt{\frac{1}{N_p} \sum_{j=1}^{N_p} \left(\nu_{j,t,p}^2 + \mu_{j,t,p}^2 \right)} \tag{18}$$

In our error model, $\mu_{j,t,p}$ is calculated separately for each voxel by explicitly modeling the Fourier undersampling and phase errors, applying dictionary matching to infer the value of parameter p for voxel j, and then subtracting from that the original ground truth value of parameter p for voxel j in the original model. In contrast, we estimate $\nu_{j,t,p}^2$ for each tissue type and parameter using the perturbative arguments of [11, 12]. Thus our estimated values of $\nu_{j,t,p}^2$ are in fact independent of j. Consequently, (18) simplifies to

$$\sigma_t^{(p)} = \sqrt{\nu_{t,p}^2 + \frac{1}{N_p} \sum_{j=1}^{N_p} \mu_{j,t,p}^2}. \tag{19}$$

Introducing the notation $\eta_t^{(p)} = \sqrt{\frac{1}{N_p} \sum_{j=1}^{N_p} \mu_{j,t,p}^2}$ for the root-mean-square systematic error and substituting into (19) yields (14). Note also that there is no assumption that error is of mean zero. Bias is incorporated into the error metric via the $\left(\eta_t^{(p)}\right)^2$ term.

[1] By linearity of expectation one could equivalently write $\sigma_t^{(p)} = \sqrt{\langle \frac{1}{N_p} \sum_{j=1}^{N_p} \delta_j^2 \rangle}$.

4 Magnitude Incentive

As shown in equation (1) of the main text, our cost function takes the form

$$C = C_{\text{main}} + w_{\text{mag}} C_{\text{mag}}, \tag{20}$$

where

$$C_{\text{main}} = \left(\sigma^{(T_1)} + w_2 \sigma^{(T_2)}\right) \sqrt{t} \tag{21}$$

and

$$C_{\text{mag}} = \frac{1}{\bar{m}_{\text{min}}}, \tag{22}$$

with $\bar{m}_{\text{min}}$ denoting the average signal magnitude of a tissue, minimized over tissues. Thus, C_{mag} penalizes pulse sequences yielding weak signals, and the strength of this penalty relative to the rest of the cost function is tuned by adjusting the coefficient w_{mag}.

In Figure 4 of the main text and tables 1-4 of this supporting material, we present fifteen optimized pulse sequences labelled a through o. Two of these, sequences i and j, are produced by optimizations in which the coefficient w_{mag} has been set non-zero. The other thirteen optimized sequences are all produced using $w_{\text{mag}} = 0$.

A stronger signal magnitude should result in a better signal to noise ratio. Thus, one may expect that the magnitude incentive is redundant with the model of random error (from [11, 12]) that is already incorporated into the calculation of $\sigma^{(T_1)}$ and $\sigma^{(T_2)}$. The C_{mag} term defined in (22) is not, however, manifestly equivalent to the magnitude incentive achieved indirectly through $\sigma^{(T_1)}$ and $\sigma^{(T_2)}$. Thus we chose to experimentally test some sequences optimized using such a term.

In figure 4 of the main text, one sees that in fact sequences i and j, which were produced with nonzero w_{mag}, achieve the best precision as measured by bootstrapping statistics applied to in vivo data. Two general classes of hypotheses might be proposed for why this is the case. One is that the incorporation of nonzero w_{mag} yields a cost function that achieves better modeling of the notion of precision that is measured by bootstrap statistics. The second is that nonzero $w_{\text{mag}} = 0$ simply yielded better convergence of the optimizer. In other words, with $w_{\text{mag}} = 0$ the optimizer converged poorly and reached solutions that were far from optimal, as defined by the cost function itself.

Cost functions like equation 20, containing a linear combination of two terms, are studied under the rubric of "multi-objective optimization." As the weighting coefficient is swept, the resulting set of global optima define a Pareto-optimal tradeoff frontier. It holds rigorously that the resulting tradeoff curve must be monotonically decreasing, as any decrease in one cost term must be accompanied by an increase in the other. (If this were not the case, one could improve one term without causing any deterioration in the other, thus the point being plotted is not actually optimal.) In an optimization which is well-converged but not hitting exact global optima, this property should hold approximately.

In figure 2, such a tradeoff frontier is shown, which displays this approximate monotonicity property, though with substantial scatter. This is consistent with the improved-modeling hypothesis but does not resolve the issue definitively. We thus leave further exploration of this issue to future work.

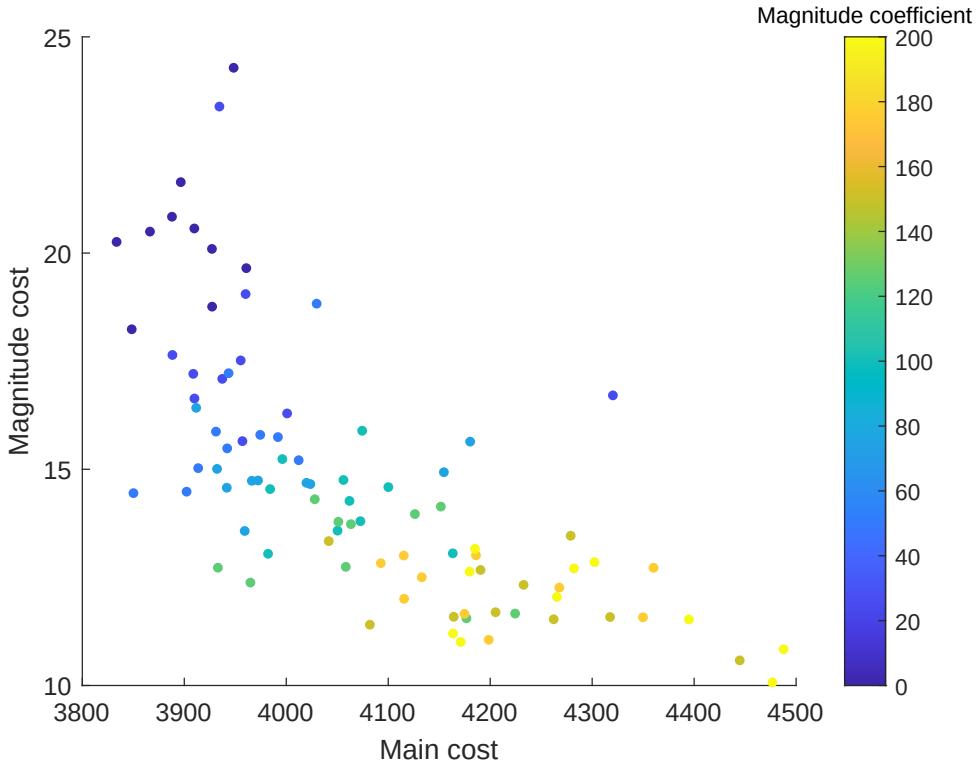

Figure 2: The cost function takes the form $C = C_{\text{main}} + w_{\text{mag}} C_{\text{mag}}$, where C_{main} is the main cost to be minimized, C_{mag} is a term that penalizes small signal magnitude, and w_{mag} is a coefficient that sets the relative weight of the magnitude term. In a well-converged optimization, any decrease in the magnitude penalty will come at the cost of an increase in the main cost. By changing the magnitude coefficient one can sweep across this tradeoff curve. The general trend of the points obtained from the 90 optimization results shown here is consistent with this. However, there is a substantial amount of scatter thus illustrating that the optimization is generally not reaching exact global optima.

5 Search Space Parameterization

For most of our optimizations we set $\theta_s = 0$ for all s. In this case, the number of parameters defining a pulse sequence is $2n$, where n is the number of TRs. In this work we consider sequences with $480 \leq n \leq 3000$, which have duration roughly 5 seconds to 35 seconds. MRF pulse sequence design is thus a continuous-variable optimization problem on a $2n$-dimensional search space parameterized by n flip angles $\alpha_1, \ldots, \alpha_n$ and n durations $\mathrm{TR}_1, \ldots, \mathrm{TR}_n$. The cost function also turns out to be highly non-convex, as illustrated in figure 5 of the main text. Finding global optima for such a high-dimensional non-convex optimization problem is likely out of reach for existing algorithms and computational hardware. Furthermore, even finding good local optima is challenging on such a high-dimensional and rugged optimization landscape.

To make a high dimensional optimization problem more manageable one can use prior knowledge to narrow the search to more promising regions of the search space. One way to do this is to initialize the optimization algorithm with a prior solution already known to be good. A different way is to use a parameterization of the search space that restricts the optimizer to explore some lower dimensional manifold of solutions thought to be promising. Here, we take this latter approach. Specifically, we restrict attention only to pulse sequences in which the flip angle and TR times vary smoothly from one TR to the next. This is motivated by the observation that such sequences typically have lower Fourier undersampling error than "rough" sequences [14, 15, 16, 17]. We achieve this by parameterizing the flip angle α vs. s and TR vs s curves using cubic splines[2] The spline is determined by a small number k of control points (typically $10 \leq k \leq 20$) over which the optimizer has control of vertical (*i.e.* α-axis or TR-axis) and horizontal (*i.e.* s-axis) position. Because the first and last control points of each spline are pinned to $s = 1$ and $s = n$, respectively, this yields a $(2k-2)$-dimensional search space. Within the resulting search space of more manageable dimension, we generate starting points for the optimizer uniformly at random and attempt to optimize more globally. This opens the possibility of finding novel pulse sequences unbiased by any human-designed starting point.

6 Optimization Algorithms

We formulate the design of MRF pulse sequences as a global optimization problem over continuous variables. The cost function is treated as a black box. There is no formula for the gradient of the cost function; strictly speaking, the cost function is not differentiable due to the discrete dictionary matching involved in computing Fourier undersampling errors. Due to the highly non-convex nature of the cost function, as illustrated in figure 3, we relied on optimization heuristics capable of escaping from local minima. The best performing of these, according to our experimentation, were simulated annealing and substochastic Monte Carlo. These algorithms are simplest to formulate in the context of discrete-variable optimization problems. The special considerations needed to adapt these to the continuous-variable problem of pulse-sequence optimization are outlined in this section.

[2]We have also tried other parameterizations: direct parameterization in terms of all $2n$ variables $(\alpha_s, \mathrm{TR}_s)$, piecewise linear, piecewise constant, and linear combination of Gaussians. However, all of the best sequences in terms of in vivo performance, and all of the sequences reported in this paper, come from spline parameterizations.

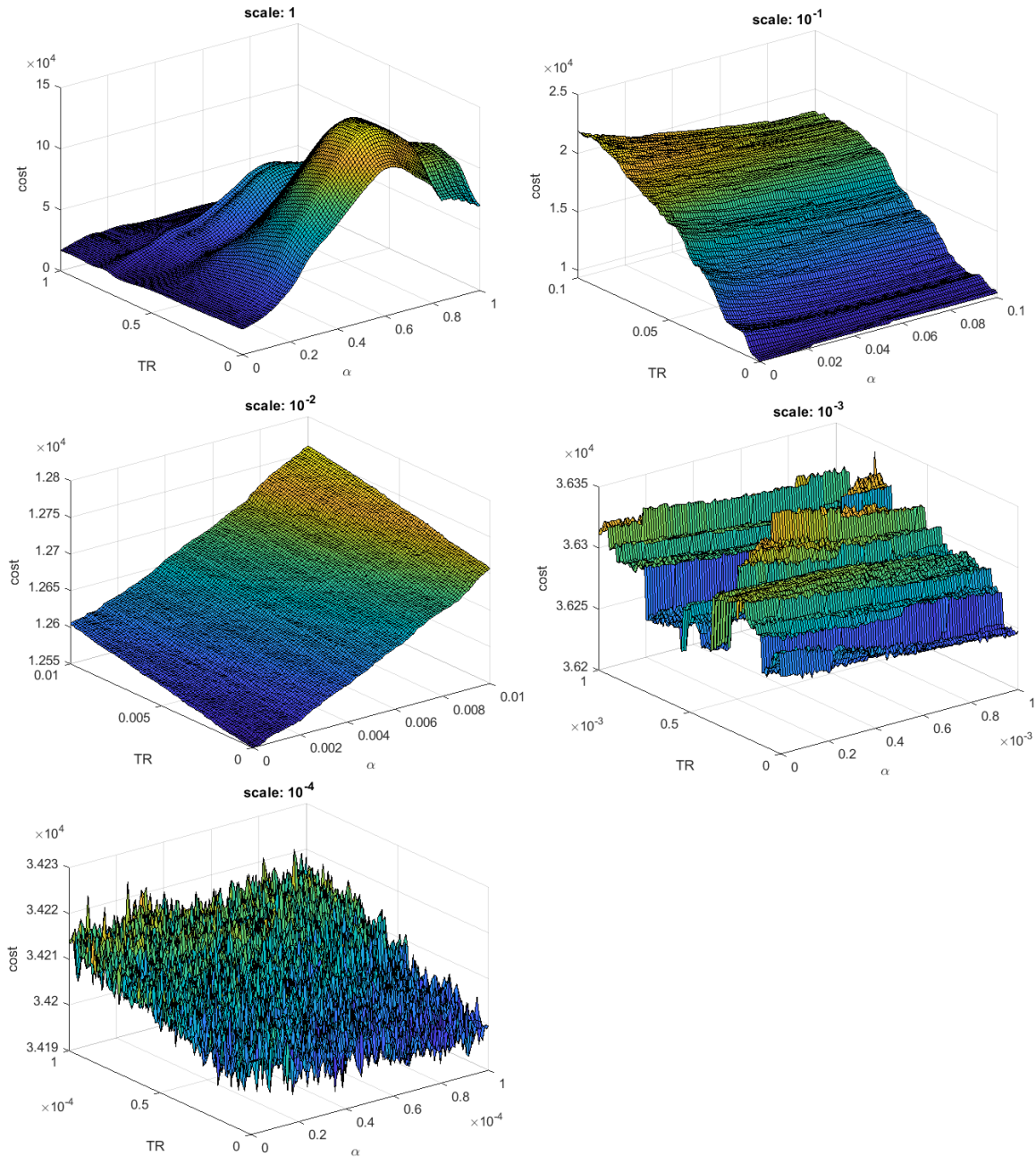

Figure 3: Samples of the cost function landscape for a pulse sequence of 480 TRs. Here, the optimization landscape is specified by a cubic spline with 18 degrees of freedom that dictates the flip angles (α) and a cubic spline with 18 degrees of freedom that specifies the TR times (TR). For each of the above plots, we take a random starting point in the resulting 36-dimensional space and a pair of random 18-dimensional unit vectors to determine directions of motion for the flip angle spline and TR spline. We then plot the cost function as a function of the distance moved along these two directions. The plots are zoomed in to five different scales, by factors of ten.

6.1 Continuous Substochastic Monte Carlo

Substochastic Monte Carlo is a quantum-inspired optimization method introduced in [18]. Substochastic Monte Carlo is inspired directly by adiabatic quantum algorithms for optimization [19]. In adiabatic computation, one starts with an initial Hamiltonian H_{init}, whose ground state is easy to prepare, and slowly interpolates to some final Hamiltonian H_{final}, whose ground state encodes the solution to the computational problem at hand. When executed on ideal quantum hardware (without decoherence) the resulting dynamics is that determined by Schrödinger's equation

$$\begin{aligned} \frac{d}{dt}|\psi\rangle &= -iH(t)|\psi\rangle \quad (23) \\ H(t) &= (1-s(t))H_{\text{init}} + s(t)H_{\text{final}}. \quad (24) \end{aligned}$$

Here, the function $s(t) \in [0,1]$ is the "annealing schedule" according to which the interpolation is performed. In the simplest case, one could proceed from H_{init} to H_{final} at a constant rate over a period of duration T by using $s(t) = t/T$. Quantum adiabatic theorems [20, 21] guarantee that, if the the interpolation is done sufficiently slowly, then the system will track the instantaneous ground state of $H(t)$ and thereby produce the ground state of H_{final}, as desired. Specifically, this can be achieved with $T = O(1/\gamma^2)$, where $\gamma = \min_{0\le t\le T}\gamma(t)$ and $\gamma(t)$ is the energy gap between the ground state and first excited state for $H(t)$.

Hamiltonians in which all off-diagonal matrix elements are non-positive are known as *stoquastic*. By the Perron-Frobenius theorem, the ground state of any stoquastic Hamiltonian can be expressed using only real nonnegative amplitudes. Complexity-theoretic evidence suggests that adiabatic quantum computation with stoquastic Hamiltonians cannot efficiently implement universal quantum computation [22]. More concretely, standard folklore in the computational physics community asserts that stoquastic adiabatic processes should be efficient to simulate on classical computers using path integral or diffusion Monte Carlo methods as they do not suffer from a "sign problem". On the other hand, some counterexamples have been constructed in which standard path integral and diffusion Monte Carlo methods fail to converge in polynomial time when simulating such Hamiltonians [23, 18] and there is complexity-theoretic evidence that polynomial-time classical simulation of general stoquastic Hamiltonians is impossible [24].

In [18] it was observed that, by applying a variant of diffusion Monte Carlo to simulate a quantum adiabatic optimization, one obtains a classical optimization heuristic which is competitive with state of the art solvers on a widely studied discrete optimization problem called MAXSAT. In diffusion Monte Carlo, one constructs a Markov Chain to mimic imaginary-time Schrödinger equation dynamics:

$$\frac{d}{dt}|\psi\rangle = -H(t)|\psi\rangle\,. \quad (25)$$

For timestep δt small compared to the variation of $H(t)$ one can approximately solve (25) by

$$|\psi(T)\rangle = \prod_{j=0}^{T/\delta t} e^{-H(j\delta t)\delta t}|\psi(0)\rangle\,, \quad (26)$$

which becomes exact in the limit $\delta t \to 0$.

As $|\psi(0)\rangle$ is a vector and $e^{-H(j\delta t)\delta t}|\psi(0)\rangle$ is a matrix, (26) looks much like a Markov chain. However, there remain two difficulties for simulating (25) using a Markov Chain. The first is that $e^{-H(j\delta t)\delta t}$ has matrix elements (which in a Markov chain become transition probabilities) that are

not easy to compute. The second is that $e^{-H(j\delta t)\delta t}$ is in general not a stochastic matrix, and therefore does not preserve the sum of the entries of the vector, which in a Markov chain represent probabilities that must sum to one. (Exponentially large dimension of $e^{-H(j\delta t)\delta t}$ and $|\psi(T)\rangle$ does not pose a problem for Markov Chain Monte Carlo methods because $|\psi(T)\rangle$ is not a list of numbers to be stored in memory but rather a probability distribution to be inhabited by following the specified transition probabilities.)

The solutions to these difficulties depends on the specific structure of the Hamiltonian (24). In a continuous variable optimization problem on n variables a natural choice is to take

$$H_{\text{init}} = -\nabla^2 \tag{27}$$

$$H_{\text{final}} = C(x_1, \ldots, x_n) \tag{28}$$

where ∇^2 is the Laplacian on $\mathbb{R}^n$ (*i.e.* a kinetic energy term for a single particle in n dimensions) and $C(x_1, \ldots, x_n)$ is a diagonal operator in the position basis (*i.e.* a potential energy term). This ensures that the ground state of H_{init} is the uniform superposition and the ground state of H_{final} is a delta function centered at the minimum of C. In this case, using a first order Trotter-Suzuki expansion[3], one obtains

$$e^{-H(t)\delta t} = e^{-(1-s(t))\nabla^2 \delta t} e^{-s(t)C\delta t} + O(\delta t^2). \tag{29}$$

The operator $e^{-(1-s(t))\nabla^2\delta t}$ has a direct interpretation in terms of random walks. By Fourier transform on finds that, in n dimensions, for any $\alpha > 0$

$$\langle \vec{y}| \, e^{\alpha \nabla^2} \, |\vec{x}\rangle = \left(\frac{1}{2\sqrt{\alpha\pi}}\right)^n \exp\left[-\frac{|\vec{x}-\vec{y}|^2}{4\alpha}\right]. \tag{30}$$

Thus, the corresponding stochastic dynamics is to perturb the position of the random walker by a gaussian random variable of variance 2α.

The operator $e^{-s(t)C\delta t}$ does not correspond directly to a stochastic process, since probability is not preserved. Since C is diagonal in the position basis, one has

$$e^{-s(t)C\delta t} \, |\vec{x}\rangle = e^{-s(t)C(\vec{x})\delta t} \, |\vec{x}\rangle \, , \tag{31}$$

where, on the lefthand side C is an operator, and on the righthand side, $C(\vec{x})$ is a number, namely the cost function evaluated at $\vec{x}$. Thus, for walkers at locations with cost less than zero, the probability must grow, and the for walkers at locations with cost greater than zero the probability must shrink. As in [18] we implement this via birth-death dynamics. For $C(\vec{x}) > 0$ we can assign the walker at $\vec{x}$ to "die" with some probability and be removed from the population. For $C(\vec{x}) < 0$ we assign the walker to "replicate" with some probability, yielding multiple walkers at $\vec{x}$. We choose these probabilities such that the expected number of walkers at site $\vec{x}$ gets multiplied by the desired factor $e^{-s(t)C(\vec{x})\delta t}$.

With this interpretation, (29) yields a prescription for an algorithm: alternate between steps where the population of walkers is perturbed according to gaussian-distributed moves, over length scales that gradually decrease over the course of the anneal, according to some schedule specified by $s(t)$, and steps where the birth-death dynamics kills off walkers at higher values of the cost

[3]For the discrete problems considered in [18] a Taylor expansion is used instead. However, this is not possible here since ∇^2 is an unbounded operator.

function and replicates walkers at lower values of the cost function. There are however some additional subtleties to address in order to turn this into a practical algorithm. Applied to an arbitrary cost function, such a procedure generically yields a population of walkers that either collapses to zero (if the population-average value of $C(\vec{x})$ is positive) or exponentially blows up (if the population-average value of $C(\vec{x})$ is negative). One can compensate for this by replacing C with $C - \langle C \rangle$, where $\langle C \rangle$ is the cost function averaged over the current distribution of walkers. Note that, in the context of Schrodinger's equation, subtracting a time-dependent constant term from the potential C would result only in an unobservable global phase. Similarly, in the imaginary-time Schrödinger equation, such a term only affects overall magnitude of the solution vector. Thus, this adjustment does not distort the underlying physics.

In practice, this is not quite sufficient to obtain highly stable population size, so one must add a feedback loop to stabilize it. For example, we have found it effective to replace $e^{-s(t)(C-\langle C \rangle)\delta t}$ with $fe^{-s(t)(C-\langle C \rangle)\delta t}$, where

$$f = \begin{cases} 0.96 & \text{if population exceeds target} \\ 1.05 & \text{otherwise} \end{cases} \tag{32}$$

The target population is then one of the hyperparameters of the optimization algorithm. In this work we have generally found it effective to set the target population at twenty walkers.

One also must choose a timestep δt. If δt is chosen too small then in the birth-death process very few walkers will die or replicate. Hence the tendency of the dynamics toward lower values of the cost function will be very weak and walkers will move according to almost pure diffusion. If δt is too large then almost the entire population will quickly get concentrated on the location of the walker that currently has the lowest value of the cost function. A choice of δt in the operator $e^{-s(t)(C-\langle C \rangle)\delta t}$ which achieves a good compromise between these extremes is to take

$$\delta t = \frac{C_{\max} - C_{\min}}{s(t)}. \tag{33}$$

If the distribution of C over the population of walkers is such the mean cost is halfway between the maximum and minimum then this ensures that the exponent $-s(t)(C - \langle C \rangle)\delta t$ lies between $-1/2$ and $1/2$. Thus the expected number of walkers on a given site will be adjusted by a factor in the range $[e^{-1/2}, e^{1/2}]$. For any distribution, it is still the case that the exponent will lie between -1 and 1, and thus the expected number of walkers on a given site will always be multiplied by a factor in the range $[e^{-1}, e]$. (By ignoring or, through choice of δt eliminating, the possibility of factor greater than 2 one can simplify the algorithm slightly by eliminating the need to ever replicate a walker into more than two walkers.)

If the goal were to achieve an accurate physical simulation of a process with pre-specified anneal schedule $s(t)$, then one would use the same timestep δt in both $e^{-s(t)C\delta t}$ and $e^{-(1-s(t))\nabla^2 \delta t}$ in (29). Here, however, our goal is instead to achieve effective optimization. Consequently, we take as user input an annealing schedule specifying the width of the gaussian update as a function of timestep. In pseudocode, one has algorithm 1.

Algorithm 1 Continuous-Variable Substochastic Monte Carlo

REQUIRED INPUTS:

$[x_v^{(\min)}(\vec{x}), x_v^{(\max)}(\vec{x})]$; functions that, given $\vec{x} \in \mathbb{R}^n$, specify allowed range of x_v for $v = 1 \ldots n$.
$T_{\max}$; number of timesteps
$s : \{1, \ldots, T_{\max}\} \to [0, 1]$; anneal schedule
P_{target}; target population size
$C : \mathbb{R}^n \to \mathbb{R}$; cost function

ALGORITHM:

Place P_{target} walkers uniformly at random in the search space
Let $C_{\text{winner}}, \vec{x}_{\text{winner}}$ equal the cost and location of lowest cost walker in population
for $t = 1$ to $T_{\max}$ **do**
  comment: First, simulate $e^{-s(t)(C-\langle C\rangle)\delta t}$
  Let $C_{\min}, C_{\max}, \langle C\rangle$ equal minimum, maximum, and average cost in current population
  for $w = 1$ to current population size **do**
    Let $\vec{x} \in \mathbb{R}^n$ be location of walker w
    Let $\hat{C} = (C(\vec{x}) - \langle C\rangle)/(C_{\max} - C_{\min})$
    Let f equal 0.96 if population exceeds P_{target}, 1.05 otherwise
    Let $q = f \times e^{-\hat{C}}$.
    if $q < 1$ **then**
      Keep walker as-is with probability q, remove walker with probability $1 - q$
    else
      if $q > 2$ **then**
        Let $q = 2$
        Print a warning (rare in practice)
      end if
      comment: Here we know $1 \le q \le 2$
      Duplicate walker with probability $q - 1$, keep walker as-is with probability $2 - q$
    end if
  end for
  comment: Second, simulate $e^{(1-s(t))\nabla^2\delta t}$
  for $w = 1$ to population size **do**
    Let $\vec{x} \in \mathbb{R}^n$ be location of walker w
    for $v = 1$ to n **do**
      Let $R = x_v^{(\max)}(\vec{x}) - x_v^{(\min)}(\vec{x})$
      Sample δ as gaussian random variable of $\mu = 0$ and $\sigma = (1 - s(t)) \times R \times 0.1$
      Add δ to coordinate x_v of walker w
      Truncate coordinate x_v of walker w back to range $[x_v^{(\min)}(\vec{x}), x_v^{(\max)}(\vec{x})]$, if necessary
    end for
  end for
  if lowest cost of a walker in current population is less than C_{winner} **then**
    Overwrite C_{winner} and $\vec{x}_{\text{winner}}$ with the cost and location of this walker
  end if
end for
output $\vec{x}_{\text{winner}}$

Some comments on algorithm 1:

- In algorithm 1 we allow for the possibility that the minimum is found at any timestep. However, it is typically found in the final timestep or a near-final timestep.

- At the initial time step we have the random walk perturb the variables by a constant fraction of the width of the search space. Here we have taken this fraction to be 0.1. This is a somewhat arbitrary value chosen so that the initial diffusion process take jumps large enough compared to the search space to ensure good mixing of the Markov chain but small enough to ensure that truncation at the boundaries is not too common. At the final timestep the variables are perturbed by much smaller distances according to the ratio $\frac{1-s(1)}{1-s(T_{\max})}$. To optimize the variables up to some desired precision ϵ we need these final perturbations to have magnitude on the order of ϵ. The ratio of the width of the search space to the desired precision ϵ is thus an important metric of the size of the search space and dictates the range over which s must be swept. For our pulse sequence optimizations we typically take this ratio to be 10^4. Minimization by exhaustive search over n variables would thus require 10^{4n} evaluations of the cost function. With our spline parameterization $n = 36$ is a typical value, and hence 10^{144} evaluations would be needed for exhaustive search. In our optimizations with substochastic Monte Carlo or simulated annealing we typically use on the order of 10^5 cost function evaluations. (However, unlike exhaustive search, these algorithms are not guaranteed to find the global minimum.)

- By examining algorithm 1 one can observe that, due to the adaptively chosen offset $\langle C \rangle$ and timestep δt, the dynamics of the walkers is invariant under the transformation $C(\vec{x}) \to aC(\vec{x}) + b$ for any constant b and any positive constant a. This is very useful in practice for optimizing pulse sequences because one must typically experiment with many different cost functions which may differ widely in scale. These changing scales require no adjustment to the algorithm or to the hyperparameters.

6.2 Adaptive Non-Isotropic Simulated Annealing

Simulated annealing is a widely used physics-inspired heuristic for non-convex optimization [25]. At each iteration a move in the search space is proposed, which is accepted with probability $\min\{1, e^{-\beta \Delta E}\}$, where β is interpreted as an inverse temperature, and ΔE is the change in cost function, interpreted as an energy. The rate of convergence of simulated annealing algorithms depends strongly on the acceptance rate of the proposed moves. For application to simulations in statistical physics, where the goal is sampling from a Boltzmann distribution, it can be proven under certain conditions that the optimal acceptance rate is 0.234 [26]. For continuous variable optimization it has been conjectured based on experimental results and heuristic arguments that an acceptance rate of 0.5 may yield optimal performance [27].

To achieve acceptance rates near half, the proposed moves in the search space should yield changes in the value of the cost function (here interpreted as energy) of roughly comparable magnitude to the temperature $T = 1/\beta$. (For convenience we here use units where Boltzmann's constant is unity.) For typical cost functions, making smaller magnitude changes to $\vec{x}$ will yield smaller magnitude changes to $C(\vec{x})$. Consequently, efficient simulated annealing algorithms for continuous variable optimization problems propose smaller moves in the search space as the temperature is decreased [27].

Here, we choose step size as a function of temperature using an approach tailored to the specific properties of our pulse sequence optimization problem. As discussed in §5, we have four distinct variable types: horizontal coordinates of spline control points for flip angle, vertical coordinates for spline control points for flip angle, horizontal coordinates for spline control points for TR time, and vertical coordinates for spline control points for TR time. The typical amount that the cost function is changed by making an adjustment of given magnitude to a variable can be expected to differ strongly between these four types of variables. Therefore, at the start of the optimization, for each of the four variable types, we separately perform random sampling to estimate the typical magnitude of change in cost as a function of magnitude of perturbation to those variables.

For our specific cost function, we find that the median absolute value of change in cost, as a function of the magnitude in the change of a variable, is well fit by assuming that the magnitude of change in cost is proportional to the magnitude of the move in the search space. We thus extract four proportionality constants, one for each variable type, from random sampling and linear fits, at the start of the anneal, and then generating proposed moves by perturbing a given variable by adding a gaussian random variable with mean zero and standard deviation proportional to the temperature, via the proportionality constant appropriate to the variable's type. Precise details of our implementation of adaptive non-isotropic simulated annealing are given in algorithms 2 through 5. Any questions can be directed to the corresponding author `dan.ma@case.edu`.

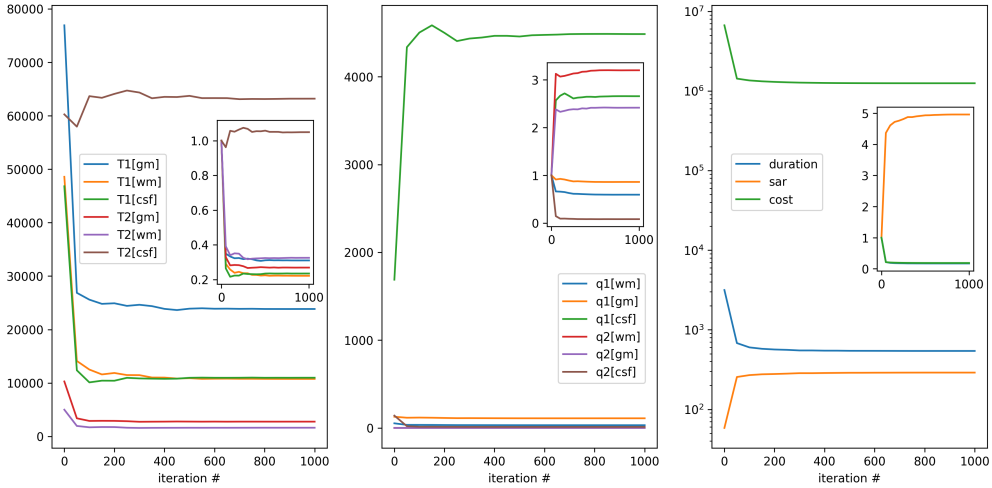

Figure 4: Evolution of the three tissue T_1 and T_2 undersampling errors (a), and random errors (b) over 2000 optimization steps. All errors were normalized to the value at the first step. The biggest gains are made in the initial iterations. At later iterations small gains in cost function are obtained by moving along tradeoff curves in which one form of error is improved at the expense of another source of error whose weight coefficient in the cost function is smaller.

Algorithm 2 PerturbVar($x, x_{\min}, x_{\max}, \text{scale}$)

COMMENT: this routine modifies x, e.g. via pass by reference
Let $\delta = \text{Gaussian}[\mu = 0, \sigma = 1] \times \text{scale} \times (x_{\max} - x_{\min})$
Let $x = x + \delta$
if $x < x_{\min}$ **then**
  Let $x = x_{\min}$
end if
if $x > x_{\max}$ **then**
  Let $x = x_{\max}$
end if

Algorithm 3 MedianDiff(scale, ℓ)

for $t = 1$ to 499 **do**
  Place $\vec{x}$ uniformly at random in the search space
  Let $E_{\text{before}} = C(\vec{x})$
  Choose v uniformly at random among variables of type ℓ
  PerturbVar($\vec{x}_v, x_v^{(\min)}, x_v^{(\max)}, \text{scale}$)
  COMMENT: E_{after} is evaluated at the perturbed value of $\vec{X}$.
  Let $E_{\text{after}} = C(\vec{x})$
  Let $\Delta_t = |E_{\text{after}} - E_{\text{before}}|$
end for
Return $\bar{\Delta}_\ell$ = median of $\Delta_{\{1,\ldots,499\}}$

Algorithm 4 FindScaleFactors

for $J = 1$ to 4 **do**
  Let $S_J = 0.03 \times e^{-2J}$
  for $\ell = 1$ to NumTypes **do**
    Let $\bar{\Delta}_J^{(\ell)} = \text{MedianDiff}(S_J, \ell)$
  end for
end for
for $\ell = 1$ to NumTypes **do**
  COMMENT: Least squares fit of $\bar{\Delta}_J^{(\ell)} = F_\ell S_J$ to $\{(\bar{\Delta}_J^{(\ell)}, S_J) : J = 1 \ldots 4\}$
  Let $F_\ell = \left(\sum_{J=0}^{4} S_J \bar{\Delta}_J^{(\ell)}\right) \Big/ \left(\sum_{J=0}^{4} S_J^2\right)$
end for
Return array $F_{\ell=1\ldots\text{NumTypes}}$

Algorithm 5 Adaptive Non-Isotropic Simulated Annealing (ANISA)

Let $F_{\ell=1\ldots\text{NumTypes}} = \text{FindScaleFactors}$
Let $kT = 0.1 \times F_0$
Let $r = R^{1/T_{\max}}$
Initialize $\vec{x}$ uniformly at random in the search space
for $t = 1$ to $T_{\max}$ **do**
  for $\ell = 1$ to NumTypes **do**
    for v in variables of type ℓ **do**
      PerturbVar$(\vec{x}_v, kT/F_\ell)$
      Accept or reject according to Metropolis rule at temperature kT
    end for
  end for
  Let $kT = kT \times r$
end for

7 Additional Data

The data in tables 1, 2, 3, and 4 is available for download in csv format at `https://github.com/madan6711/Automatic-MRF-seq-design`.

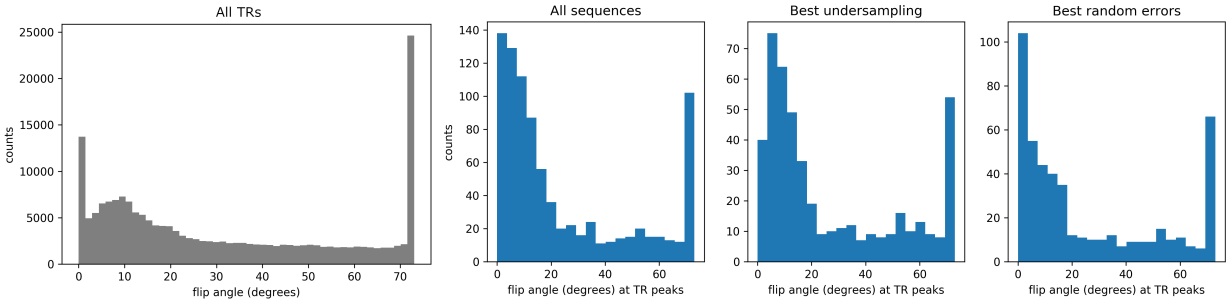

Figure 5: Histograms of flip angles aggregated across an ensemble of 388 optimized pulse sequences. The maximum flip angle of 73 degrees is determined from a Sinc pulse with a duration of 2000 us and time bandwidth product of 8, which is used to limit deviation from nominal flip angles and reduce bias in the resulting maps[28]. Left panel shows histograms of flip angles across all TRs of all pulse sequences. Right three panels show flip angles in TRs of peak duration. One can observe that low flip angles are much more prevalent at peak duration TRs. Furthermore, this favoring of low flip angles at TR duration peaks is most pronounced in the optimized pulse sequences in which random errors were more strongly optimized at the expense of undersampling errors.

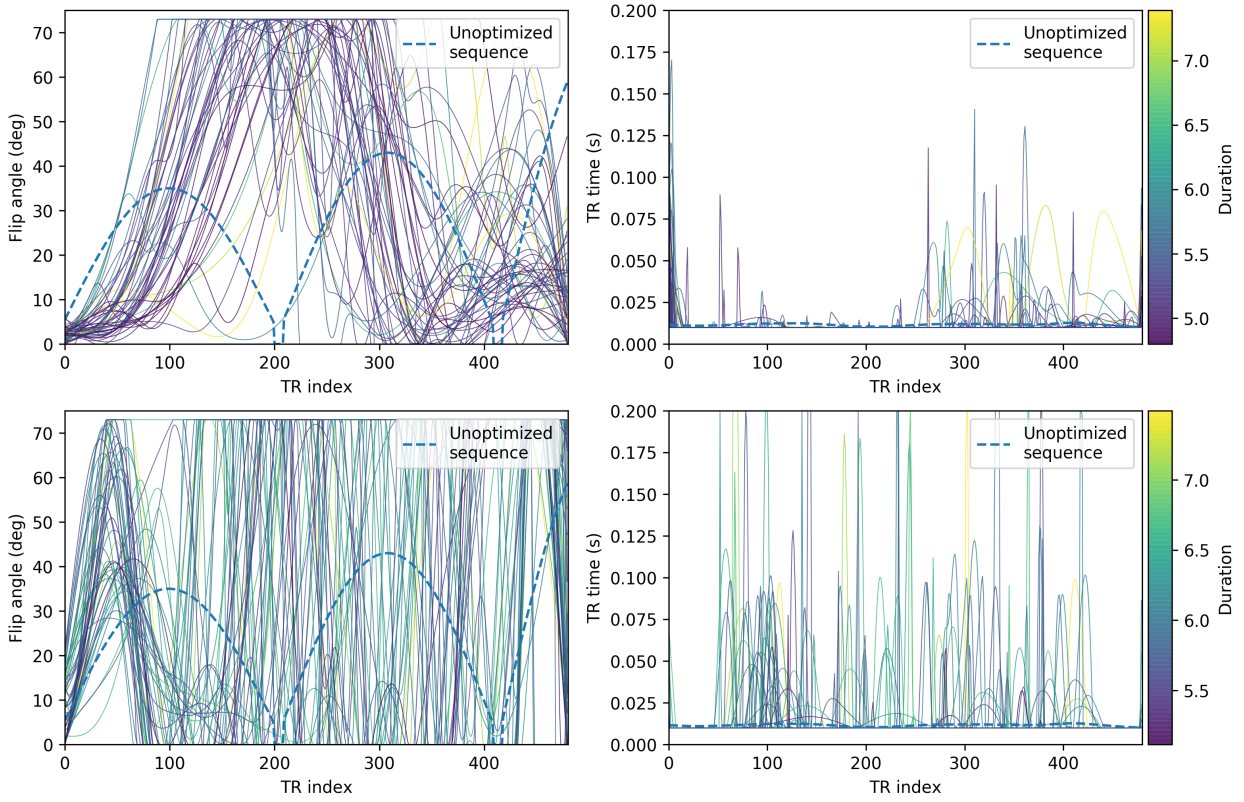

Figure 6: Flip angles and TR durations for the top 10% optimized sequences sorted by lowest undersampling errors (top row), and by lowest random errors (bottom row). A standard unoptimized sequence (dotted curve) is added for comparison.

Sequence	TRs	Duration (s)	Avg T_1 error (ms)	Avg T_2 error (ms)
standard480	480	5.57	25.1	3.51
standard672	672	7.75	24.6	2.04
standard864	864	9.93	24.3	1.85
standard1056	1056	12.21	24.1	1.76
standard1248	1248	14.50	23.6	1.64
standard1440	1440	16.69	23.2	1.51
standard1632	1632	18.99	23.1	1.43
standard2400	2400	28.05	22.3	1.20
standard2592	2592	30.30	22.2	1.17
standard2784	2784	32.52	21.9	1.16
standard3000	3000	34.95	21.7	1.13
optimized a	480	6.43	27.15	1.67
optimized b	1000	16.76	20.1	1.48
optimized c	1000	15.92	23.7	2.01
optimized d	480	11.40	29.7	2.45
optimized e	480	10.37	37.4	2.26
optimized f	480	14.11	20.3	1.96
optimized g	480	10.99	21.1	1.90
optimized h	480	8.89	25.1	1.81
optimized i	480	7.76	13.2	0.88
optimized j	960	13.66	13.2	0.73
optimized k	480	5.83	28.1	1.81
optimized l	480	5.00	28.7	1.81
optimized m	480	5.02	18.9	1.84
optimized n	480	4.88	27.1	1.32
optimized o	480	5.85	22.1	1.72

Table 1: Unoptimized sequences, c2p480–c2p3000, of different durations, are compared to optimized sequences. The average T_1 error and T_2 error are computed for four regions of interest in the white matter and then averaged. These are standard deviations under the influence of gaussian noise, as computed by applying the bootstrap method of [29] to in vivo data obtained from healthy volunteers.

sequence	σ_1(WM)	σ_2(WM)	σ_1(GM)	σ_2(GM)	σ_1(CSF)	σ_2(CSF)
standard480	94.4	47.3	153.7	69.5	78.3	873.1
standard672	98.3	12.0	166.1	27.2	75.8	451.6
standard864	113.8	8.5	166.7	26.5	71.8	381.1
standard1056	105.7	9.8	159.6	26.0	63.9	376.5
standard1248	90.1	11.7	148.8	26.7	57.6	386.4
standard1440	102.5	8.5	159.0	25.8	52.7	358.9
standard1632	108.1	8.1	162.5	24.9	49.5	363.9
standard2400	89.2	8.4	137.5	32.0	54.2	209.2
standard2592	79.9	7.5	128.4	30.8	51.9	206.7
standard2784	74.0	8.1	122.8	30.7	49.2	212.8
standard3000	92.9	7.2	138.3	30.2	50.4	233.6
optimized a	84.8	5.8	196.5	19.7	120.6	429.2
optimized b	84.3	6.3	139.5	24.2	97.8	208.8
optimized c	65.5	13.4	80.3	22.1	179.5	333.5
optimized d	181.5	5.8	340.3	63.5	263.4	341.5
optimized e	51.7	2.9	127.9	8.6	47.5	287.9
optimized f	47.7	18.1	84.0	28.1	60.6	149.9
optimized g	34.1	7.6	70.3	15.9	98.3	205.2
optimized h	79.2	6.4	128.2	21.6	118.8	416.7
optimized i	101.8	10.1	236.0	24.3	76.6	308.4
optimized j	322.3	9.3	376.8	27.0	74.0	301.6
optimized k	40.0	1.2	114.8	2.9	59.6	124.1
optimized l	38.1	1.3	107.4	2.4	53.0	176.7
optimized m	42.3	1.5	126.0	4.0	59.7	105.3
optimized n	26.1	1.9	81.3	7.0	136.8	691.0
optimized o	48.9	1.6	127.3	4.8	82.8	105.2

Table 2: Systematic errors as predicted by the three-tissue digital phantom. Each voxel in the simulated brain is assigned to be either white matter (WM), grey matter (GM), or cerebrospinal fluid (CSF). The bloch equations are solved, and the resulting measurement outcomes are computed as in equation 11 using point spread functions that incorporate phase errors. The inferred values of T_1 and T_2 are then computed from these signals for each voxel by dictionary matching. The root-mean-square deviation of the inferred value from the original value assigned to the voxels is tabulated for T_1 for T_2 for each of the three tissue types. $\sigma_j(T)$ is the RMS deviation in T_j for tissue type T, expressed in milliseconds.

sequence	σ_1(WM)	σ_2(WM)	σ_1(GM)	σ_2(GM)	σ_1(CSF)	σ_2(CSF)
standard480	57.3	7.3	148.1	18.7	77.5	615.6
standard672	54.3	4.3	136.7	26.3	79.6	337.6
standard864	54.0	4.4	131.5	27.1	74.5	284.0
standard1056	53.4	4.2	130.5	24.7	64.7	289.5
standard1248	51.2	4.4	127.9	24.9	58.7	306.0
standard1440	50.7	4.5	125.8	26.2	54.8	285.0
standard1632	50.1	4.5	123.4	25.6	50.8	277.4
standard2400	45.4	5.2	98.7	35.7	55.5	181.9
standard2592	43.8	5.2	96.6	34.5	52.1	192.8
standard2784	42.8	5.1	95.5	34.1	48.8	204.3
standard3000	43.0	4.9	93.7	33.5	47.4	208.4
optimized a	31.6	2.9	99.4	22.1	83.6	319.8
optimized b	42.5	4.5	78.1	27.0	92.6	127.2
optimized c	35.7	4.7	67.8	21.1	87.3	284.2
optimized d	49.8	1.6	116.0	6.0	131.7	60.2
optimized e	42.4	1.5	119.0	2.5	46.9	44.1
optimized f	32.3	4.6	58.0	23.7	49.5	136.0
optimized g	31.7	4.2	59.5	17.3	19.0	187.7
optimized h	32.4	5.0	60.2	23.5	38.4	281.5
optimized i	42.4	5.0	95.5	25.5	52.4	281.8
optimized j	48.6	8.1	110.5	33.5	68.6	207.6
optimized k	42.3	1.4	119.9	3.3	53.8	199.1
optimized l	42.2	1.4	111.0	2.3	46.8	328.4
optimized m	44.1	1.5	133.5	4.2	55.7	148.6
optimized n	29.0	1.9	81.1	7.8	130.3	781.2
optimized o	53.5	1.5	135.7	5.4	73.4	311.1

Table 3: Here, digital phantom predictions are tabulated as in table 2 except that this model assumes no phase errors.

sequence	min mag	q_1(WM)	q_2(WM)	q_1(GM)	q_2(GM)	q_1(CSF)	q_2(CSF)
standard480	0.0587	3.90e-01	2.73e-04	1.49e+00	6.05e-04	3.43e+01	3.83e-02
standard672	0.0547	3.99e-01	3.92e-04	1.53e+00	9.03e-04	4.69e+01	2.60e-01
standard864	0.0547	4.04e-01	4.81e-04	1.55e+00	1.05e-03	4.72e+01	3.28e-01
standard1056	0.0570	4.09e-01	5.31e-04	1.55e+00	1.12e-03	4.90e+01	3.30e-01
standard1248	0.0588	4.25e-01	6.46e-04	1.57e+00	1.37e-03	5.47e+01	3.54e-01
standard1440	0.0594	4.34e-01	7.63e-04	1.59e+00	1.58e-03	5.71e+01	4.11e-01
standard1632	0.0600	4.40e-01	8.66e-04	1.60e+00	1.75e-03	5.82e+01	4.31e-01
standard2400	0.0575	4.76e-01	1.36e-03	1.68e+00	2.67e-03	6.34e+01	2.12e+00
standard2592	0.0583	4.88e-01	1.42e-03	1.69e+00	2.78e-03	6.39e+01	2.17e+00
standard2784	0.0591	4.98e-01	1.48e-03	1.71e+00	2.87e-03	6.53e+01	2.30e+00
standard3000	0.0596	5.06e-01	1.57e-03	1.72e+00	3.04e-03	6.75e+01	2.47e+00
optimized a	0.0617	2.24e-01	5.12e-04	7.40e-01	1.01e-03	1.74e+01	2.31e-01
optimized b	0.0593	7.04e-01	9.16e-04	2.26e+00	2.16e-03	3.35e+01	1.52e+00
optimized c	0.0629	6.82e-01	1.21e-03	1.98e+00	3.03e-03	2.53e+01	6.61e-01
optimized d	0.0506	2.61e-01	2.66e-04	1.16e+00	2.82e-04	9.78e+00	1.82e-01
optimized e	0.0564	1.68e-01	3.58e-04	1.16e+00	4.63e-04	4.27e+01	2.73e-02
optimized f	0.0774	5.96e-01	8.60e-04	1.59e+00	2.37e-03	2.89e+01	2.79e+00
optimized g	0.0678	4.83e-01	7.94e-04	1.27e+00	1.92e-03	2.22e+01	7.85e-01
optimized h	0.0666	4.20e-01	8.06e-04	1.23e+00	1.93e-03	1.49e+01	4.78e-01
optimized i	0.0681	4.05e-01	8.25e-04	1.22e+00	2.03e-03	3.76e+01	7.38e-01
optimized j	0.0592	4.70e-01	1.40e-03	1.42e+00	3.24e-03	5.79e+01	1.89e+00
optimized k	0.0415	1.56e-01	2.97e-04	1.20e+00	3.47e-04	2.35e+01	1.19e-02
optimized l	0.0324	1.36e-01	2.77e-04	1.07e+00	3.45e-04	1.91e+01	6.17e-03
optimized m	0.0406	2.85e-01	2.88e-04	1.43e+00	3.80e-04	1.67e+01	1.85e-02
optimized n	0.0376	7.89e-02	3.67e-04	2.93e-01	6.18e-04	7.09e+00	6.55e-03
optimized o	0.0500	2.04e-01	3.22e-04	1.35e+00	3.88e-04	2.78e+01	1.65e-02

Table 4: For each of the three tissue types, the magnitude of the magnetization, as predicted by the Bloch equations, is averaged over the measurements (of which there is one for each TR). The minimum of these three numbers is recorded as "min mag". The quality factors for the three tissues are metrics of robustness against random error, which are estimated by a first order perturbative calculation in [11, 12]. The noise model is identical independently distributed complex gaussian noise of mean zero and standard deviation σ_η added to the data point associated with each point in k-space, at each measurement. In this approximation, the predicted standard deviation in the value of T_1 for grey matter due to random noise is given by $\sigma_\eta/\sqrt{q_1(GM)}$, and similarly for T_2 and for the other tissue types.

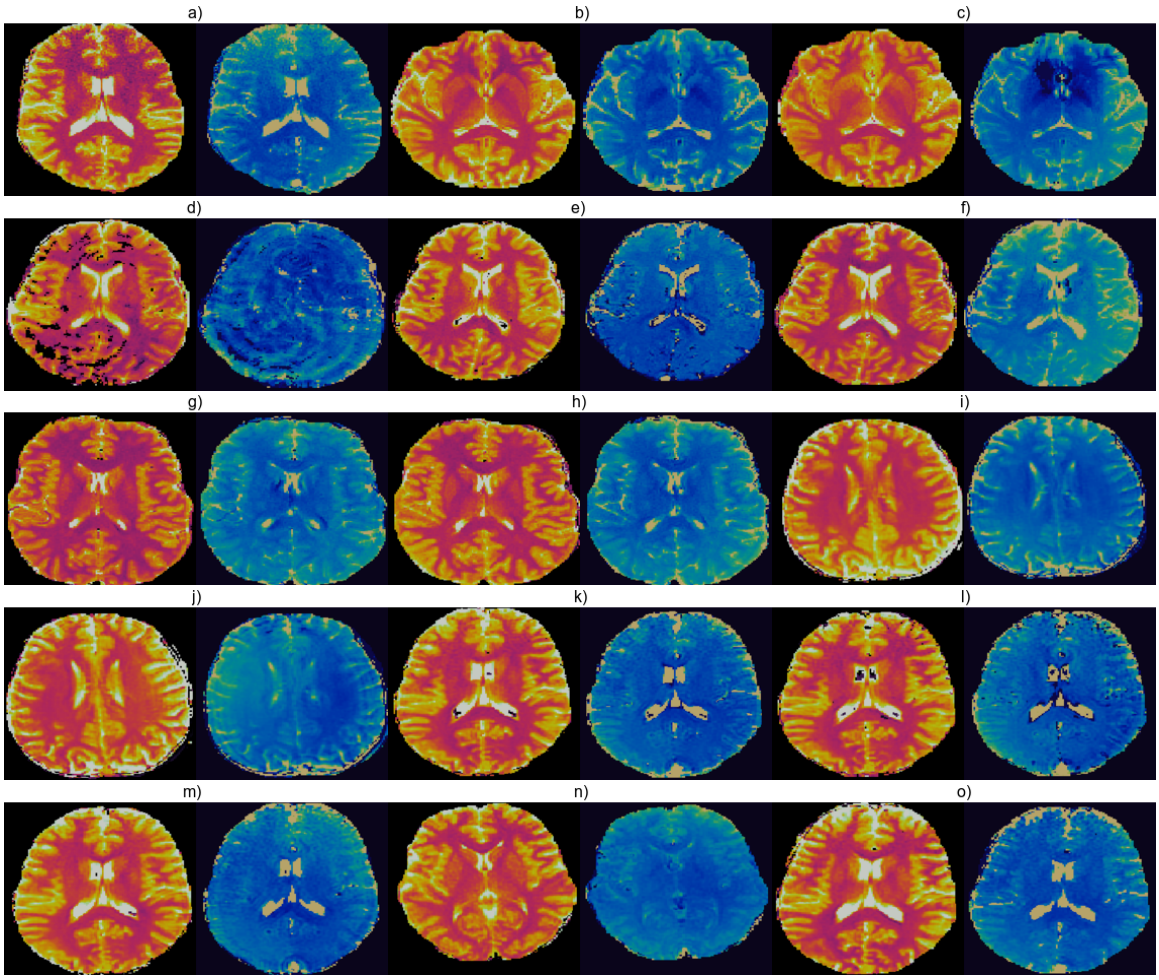

Figure 7: T_1 (red) and T_2 (blue) in vivo image pairs for each of the 15 optimized sequences. The labels match the data points in figure 3 of the main text. The sequence d was optimized with a direct parameterization rather than splines. This results in flip angles that vary less smoothly from one pulse to the next, and thereby produces stronger Fourier undersampling artifacts and inferior image quality relative the the other sequences, which are optimized using spline parameterizations.

References

[1] F. Bloch. Nuclear induction. *Physical Review*, 70:460–473, 1946. doi:10.1103/PhysRev.70.460.

[2] Debra McGivney, Rasim Rasim Boyacioğlu, Stephen Jordan, Ignacio Rozada, Sherry Huang, Siyuan Hu, Brad Lackey, Matthias Troyer, Mark Griswold, and Dan Ma. A fast approximation of undersampling artifacts in MR fingerprinting. In *The 28th Annual Meeting, ISMRM*, page 3747, 2020.

[3] Jin Hyung Lee, Brian A. Hargreaves, Bob S. Hu, and Dwight G. Nishimura. Fast 3D imaging using variable-density spiral trajectories with applications to limb perfusion. *Magnetic Resonance in Medicine*, 50:1276, 2003.

[4] Jeffrey A. Fessler and Bradley P. Sutton. Nonuniform fast Fourier transforms using min-max interpolation. *IEEE Transactions on Signal Processing*, 51(2):560, 2003.

[5] Dan Ma, Vikas Gulani, Nicole Seiberlich, Kecheng Liu, Jeffrey L. Sunshine, Jeffrey L. Duerk, and Mark A. Griswold. Magnetic resonance fingerprinting. *Nature*, 495(7440):187–192, 2013.

[6] G. Körzdörfer, J. Pfeuffer, T. Kluge, M. Gebhardt, B. Hansel, C. H. Meyer, and M. Nittka. Effect of spiral undersampling patterns on FISP MRF parameter maps. *Magnetic Resonance Imaging*, 62:174–180, 2019. doi:10.1016/j.mri.2019.01.011.

[7] D. F. McGivney, E. Pierre, D. Ma, Y. Jiang, H. Saybasili, V. Gulani, and M. A. Griswold. SVD compression for magnetic resonance fingerprinting in the time domain. *IEEE Transactions on Medical Imaging*, 33(12):2311–2322, 2014. doi:10.1109/tmi.2014.2337321.

[8] B. Zhao, K. Setsompop, E. Adalsteinsson, B. Gagoski, H. Ye, D. Ma, Y. Jiang, P. Ellen Grant, M. A. Griswold, and L. L Wald. Improved magnetic resonance fingerprinting reconstruction with low-rank and subspace modeling. *Magnetic Resonance in Medicine*, 79(2):933–942, 2018. doi:10.1002/mrm.26701.

[9] Jakob Assländer, Martijn A. Cloos, Florian Knoll, Daniel K. Sodickson, Jürgen Hennig, and Riccardo Lattanzi. Low rank alternating direction method of multipliers reconstruction for MR fingerprinting. *Magnetic Resonance in Medicine*, 79(1):83–96, 2017. doi:10.1002/mrm.26639.

[10] G. Mazor, L. Weizman, A. Tal, and Y. C. Eldar. Low-rank magnetic resonance fingerprinting. *Medical Physics*, 45(9):4066–4084, 2018. doi:10.1002/mp.13078.

[11] Danielle Kara, Mingdong Fan, Jesse Hamilton, Mark Griswold, Nicole Sieberlich, and Robert Brown. Parameter map error due to normal noise and aliasing artifacts in MR fingerprinting. *Magnetic Resonance in Medicine*, 8(5):3108, 2019. doi:10.1002/mrm.27638.

[12] Danielle Kara. *Understanding error in magnetic resonance fingerprinting.* PhD thesis, Case Western Reserve University, May 2018.

[13] K. Sommer, T. Amthor, M. Doneva, P. Koken, J. Mieneke, and P. Börnert. Towards predicting the encoding capability of MR fingerprinting sequences. *Magnetic Resonance Imaging*, 41:7–14, 2017. doi:10.1016/j.mri.2017.06.015.

[14] Bo Zhao, Justin P. Haldar, Congyu Liao, Dan Ma, Yun Jiang, Mark A. Griswold, Kawin Setsompop, and Lawrence L. Wald. Optimal experiment design for magnetic resonance fingerprinting: Cramér-Rao bound meets spin dynamics. *IEEE Transactions on Medical Imaging*, 38(3):844–861, 2019. doi:10.1109/tmi.2018.2873704.

[15] Jakob Assländer, Riccardo Lattanzi, Daniel K. Sodickson, and Martijn A. Cloos. Optimized quantification of spin relaxation times in the hybrid state. *Magnetic Resonance in Medicine*, 82(4):1385, 2019. doi:10.1002/mrm.27819.

[16] Alessandro Sbrizzi, Tom Bruijnen, Oscar van der Heide, Peter Luijten, and Cornelis A. T. van den Berg. Dictionary-free MR fingerprinting reconstruction of balanced-GRE sequences. *arXiv:1711.08905*, 2017.

[17] Christian C. Stolk and Alessandro Sbrizzi. Understanding the combined effect of k-space undersampling and transient states excitation in MR fingerprinting reconstructions. *IEEE Transactions in Medical Imaging*, 38(10):2445, 2019. doi:10.1109/tmi.2019.2900585.

[18] Michael Jarret, Stephen P. Jordan, and Brad Lackey. Adiabatic optimization versus diffusion Monte Carlo. *Physical Review A*, 94:042318, 2016. arXiv:1607.03389.

[19] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, Joshua Lapan, Andrew Lundgren, and Daniel Preda. A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem. *Science*, 20(5516):472–475, 2001. arXiv:quant-ph/0104129.

[20] Sabine Jansen, Mary-Beth Ruskai, and Ruedi Seiler. Bounds for the adiabatic approximation with applications to quantum computation. *Journal of Mathematical Physics*, 48:102111, 2007.

[21] Alexander Elgart and George A. Hagedorn. A note on the switching adiabatic theorem. *Journal of Mathematical Physics*, 52:102202, 2012.

[22] Sergey Bravyi, David P. DiVincenzo, Roberto I. Oliveira, and Barbara Terhal. The complexity of stoquastic local Hamiltonian problems. *Quantum Information and Computation*, 8:0361–0385, 2008. arXiv:quant-ph/0606140.

[23] Matthew B. Hastings. Obstructions to classically simulating the quantum adiabatic algorithm. *Quantum Information and Computation*, pages 1038–1076, 2013. arXiv:1302.5733.

[24] M. B. Hastings. The power of adiabatic quantum computation with no sign problem. *arXiv:2005.03791*, 2020.

[25] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983. doi:10.1126/science.220.4598.671.

[26] G. O. Roberts, A. Gelman, and W. R. Gilks. Weak convergence and optimal scaling of random walk Metropolis algorithms. *The Annals of Applied Probability*, 7(1):110–120, 1997.

[27] David Vanderbilt and Steven G. Louie. A Monte Carlo simulated annealing approach to optimization over continuous variables. *Journal of Computational Physics*, 56:259–271, 1984.

[28] D. Ma, S. Coppo, Y. Chen, D. F. McGivney, Y. Jiang, S. Pahwa, V. Gulani, and M. A. Griswold. Slice profile and b_1 corrections in 2D magnetic resonance fingerprinting. *Magnetic Resonance in Medicine*, 78(5):1781–1789, 2017. doi:10.1002/mrm.26580.

[29] M. J. Riffe, M. Blaimer, K. J. Barkausas, J. L. Duerk, and M. A. Griswold. SNR estimation in fast dynamic imaging using bootstrapped statistics. In *Proceedings of the 15th Scientific Meeting, International Society for Magnetic Resonance in Medicine*, page 1879, 2007.