Supporting Information

The following supporting information accompanies the manuscript, "T_2 Shuffling: Sharp, Multi-Contrast, Volumetric Fast Spin-Echo Imaging." Additional information and MATLAB demonstration code is available at `http://eecs.berkeley.edu/~mlustig/Software.html`.

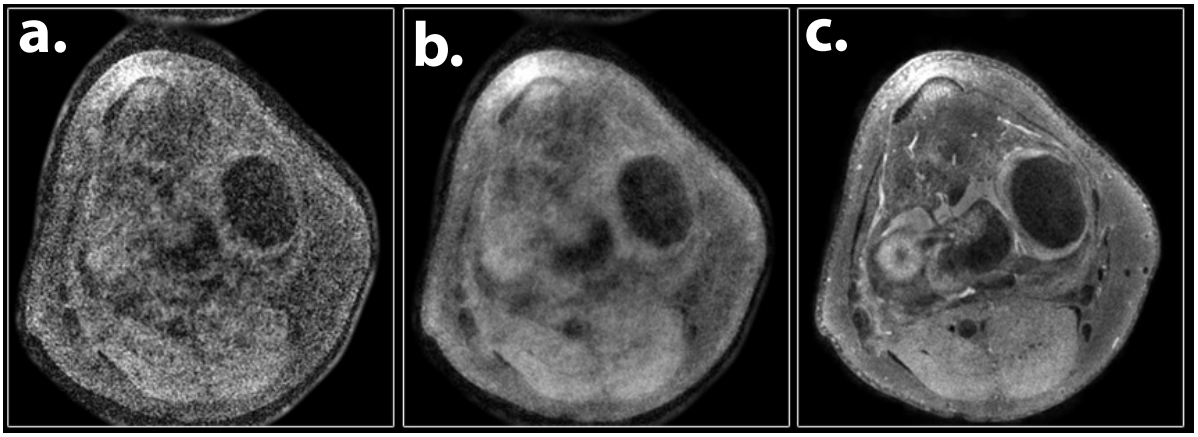
Additional In Vivo Results

Supporting Figure S-1 compares Poisson disc CUBE with L1-ESPIRiT to the 5^{th} and 30^{th} virtual TE of T_2 Shuffling on an adult volunteer (Table 1, Scan 1b and Scan 2b, respectively). In this comparison, a short ETL of 28 was used for the first scan to reduce intrinsic blur. The short ETL was compensated for by very high Poisson disc sampling acceleration with compressed sensing. At this scan time and ETL, the deblurring is negligible; however, both proton-density and T_2 contrast are recovered by T_2 Shuffling in the same scan time.

Supporting Figure S-1: Reconstruction comparison of volunteer scans (see Table 1, Scan 1b and Scan 2b for acquisition parameters). **a.** Poisson disc CUBE acquisition and L1-ESPIRiT reconstruction. The short ETL of 28 was offset by a high Relative Acceleration of 6.6. **b.** T_2 Shuffling at the 5^{th} virtual echo time ($\text{TE}_{\text{eff}} = 21$ ms). **c.** T_2 Shuffling at the 30^{th} virtual echo time ($\text{TE}_{\text{eff}} = 90$ ms).

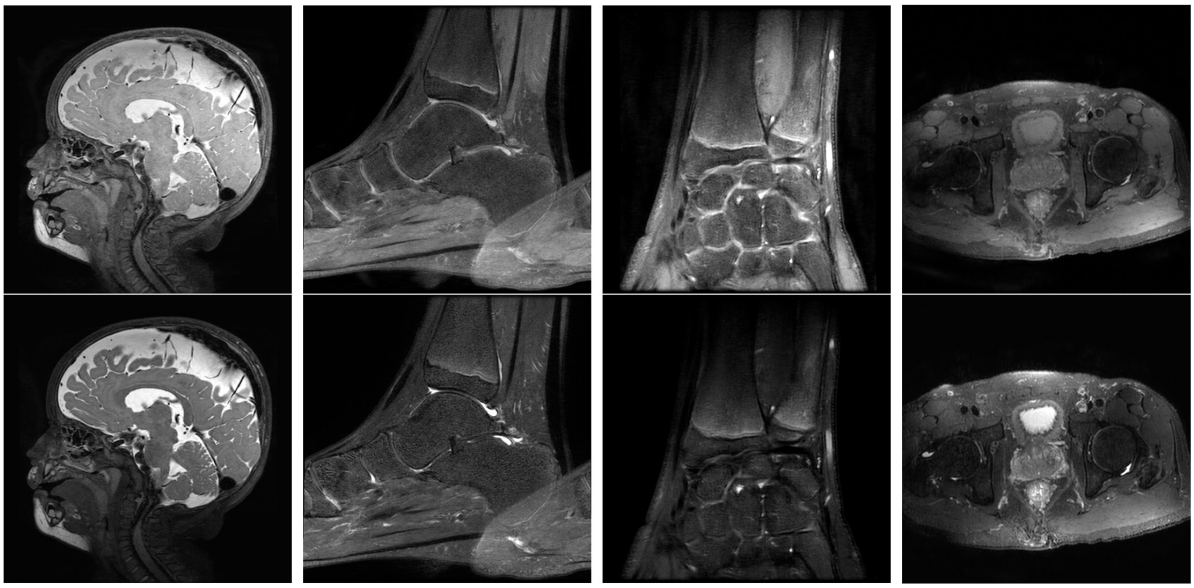
Supporting Figure S-2 compares linear reconstructions without and with Tikhonov regularization, i.e. solving Eq. (15) without LLR, to T_2 Shuffling with LLR regularization at the first virtual TE. From the figure, it is evident that the linear subspace constraint alone is not sufficient to fully recover the object. The LLR constraint reduces the sample complexity and

creates a more well-posed reconstruction.



Supporting Figure S-2: Comparison of linear reconstructions **(a)** without and **(b)** with Tikhonov regularization to **(c)** T_2 Shuffling with LLR regularization at the first virtual TE. The linear reconstruction without regularization suffers from noise amplification. Tikhonov regularization reduces noise amplification, but detailed structure is lost. The LLR-regularized reconstruction recovers the image, indicating the reduced sample complexity. Scan parameters are given in Table 1, Scan 2b.

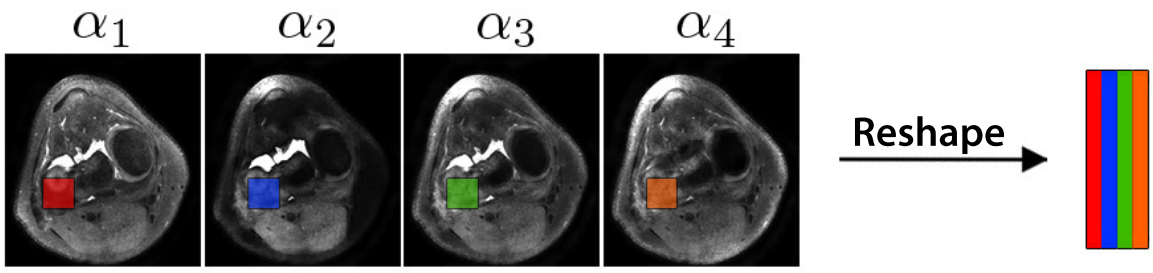
The primary focus of this work was the application of T_2 Shuffling to pediatric knee imaging; however, we have begun to explore the broader use of T_2 Shuffling in other applications. These include other musculoskeletal anatomy such as ankle, shoulder, and wrist imaging, as well as brain and prostate imaging. Supporting Figure S-3 shows T_2 Shuffling reconstructions at an early and a late virtual echo time for brain, ankle, wrist, and prostate scans. The brain and prostate reconstructions are sharp and show proton and T_2 contrast. The ankle and wrist reconstructions suffer from residual aliasing artifacts and ghosting, potentially due to motion. The use of T_2 Shuffling in motion-corrupted environments is a topic of future exploration.



Supporting Figure S-3: T_2 Shuffling reconstructions of brain, ankle, wrist, and prostate scans at two virtual echo times.

Transform Point Spread Function

The transform point spread function (TPSF) can be used to empirically assess the interference caused by a single transform coefficient (42). For T_2 Shuffling, locally low rank (LLR) matrices are used as a sparsifying transform (33). The LLR transform operates on temporal coefficient image blocks, as shown in Supporting Figure S-4. A block centered at position $\mathbf{r}$ is extracted from each temporal coefficient image and reshaped into a column of a small matrix. The transform components are the singular vectors of this matrix, and the coefficient amplitudes are the singular values. Since there are K temporal coefficient images, the maximum number of coefficients (rank) for each matrix is equal to K.

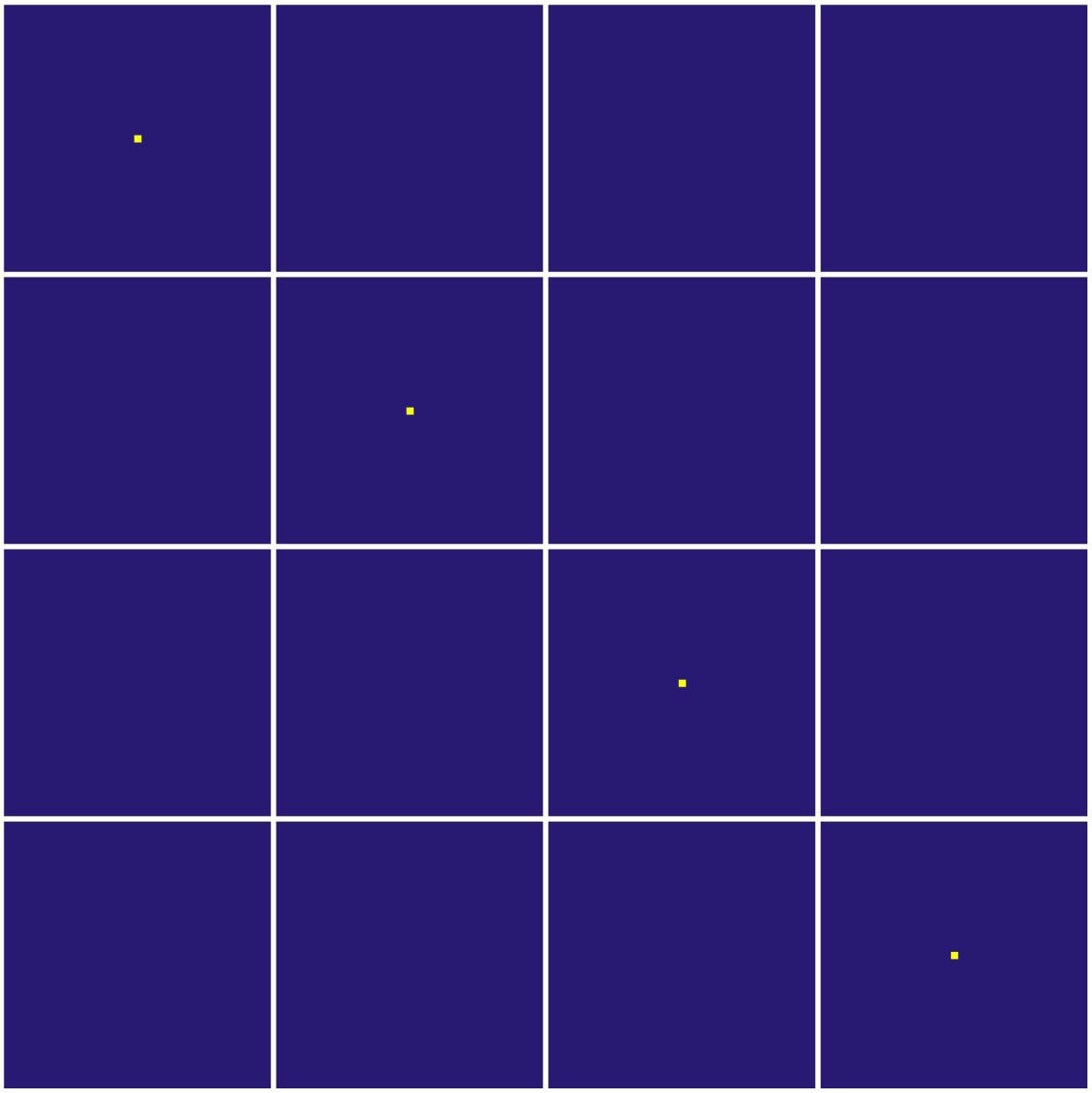


Supporting Figure S-4: Visualization of the block-wise matrix operator for LLR. Non-overlapping blocks from each temporal coefficient image are reshaped into columns to form small matrices. The transform components are the singular vectors of the matrices.

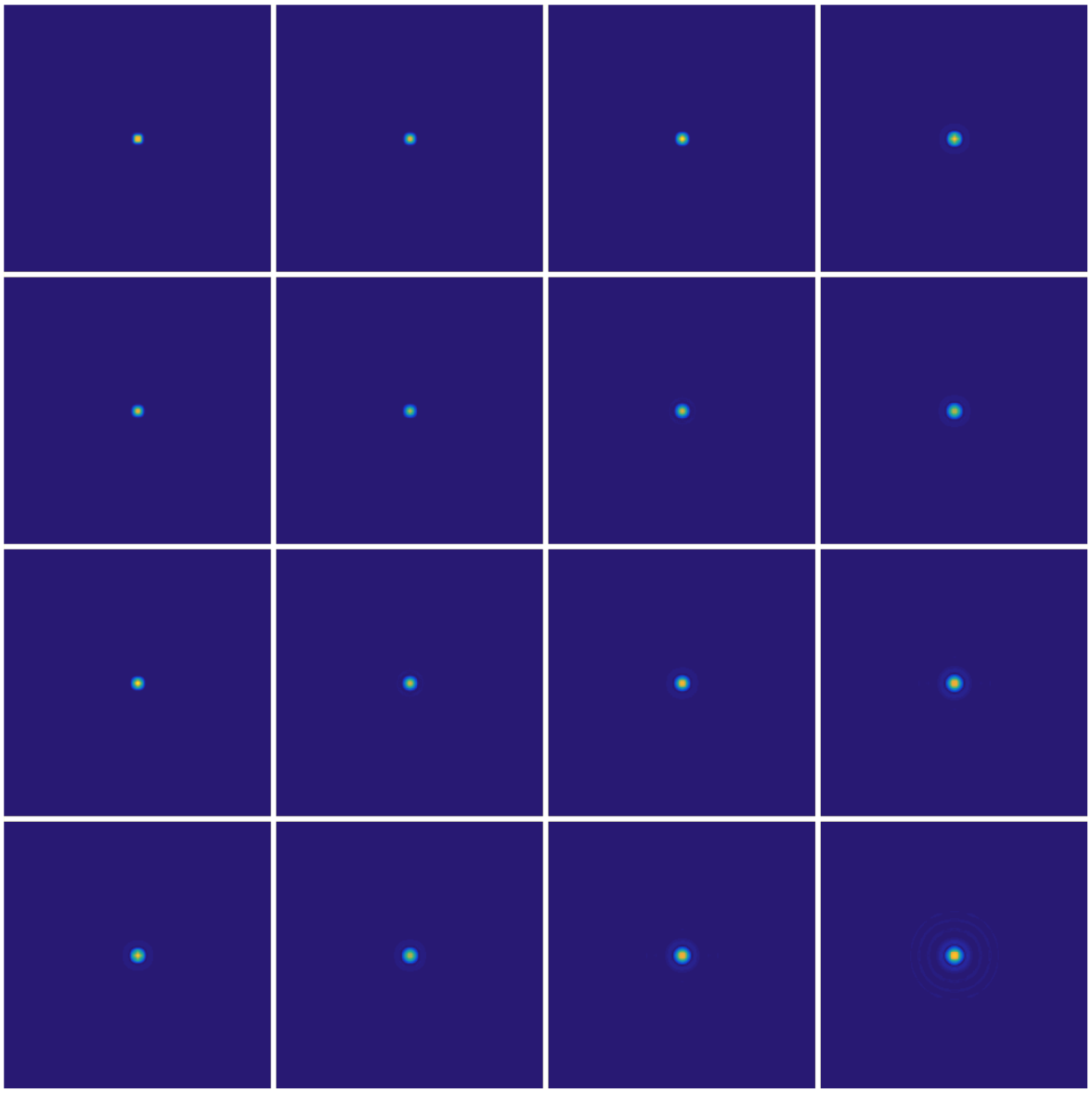
We can analyze the effect of a single coefficient in the LLR domain by constructing a system input with a uniform patch in the center of a single temporal coefficient image, and zero elsewhere. The TPSF is then computed by passing the system input through the forward operator and its adjoint:

$$\alpha_{\text{TPSF}} = \mathbf{\Phi}_K^H \mathbf{S}^H \mathbf{F}^H \mathbf{P} \mathbf{F} \mathbf{S} \mathbf{\Phi}_K \alpha_{\text{input}}. \tag{33}$$

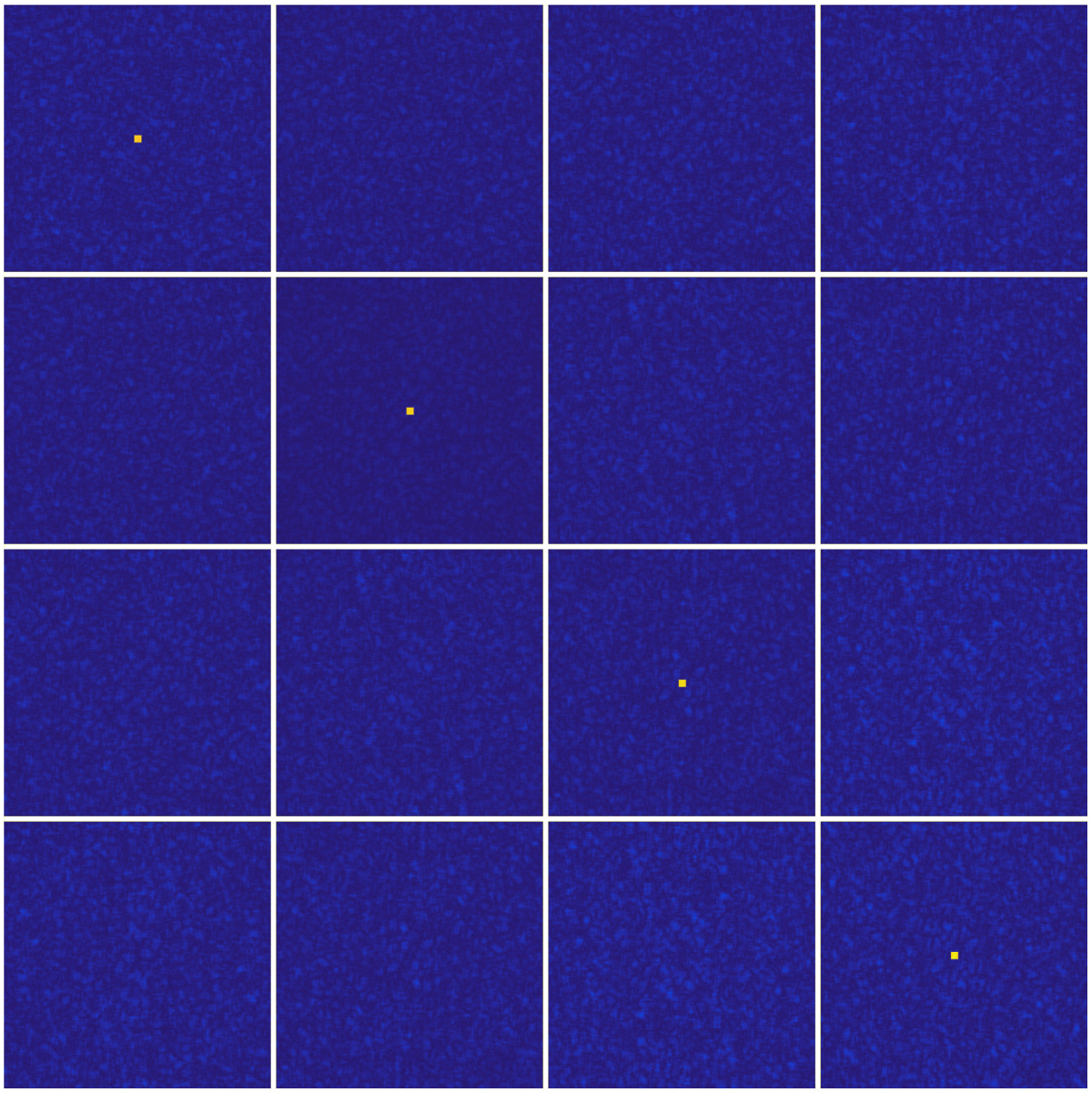
The procedure is repeated K times; each repetition places the uniform patch in the subsequent temporal coefficient image. The final TPSF is a K-by-K grid of temporal coefficient images. The ideal system input is shown in Supporting Figure S-5. The TPSF for a center-out view ordering is shown in Supporting Figure S-6, and the TPSF for a randomly shuffled view ordering is shown in Supporting Figure S-7. Both the center-out and the randomly shuffled sampling patterns were constructed such that every phase encode was sampled exactly once (i.e. no parallel imaging acceleration or repeated phase encode re-sampling). The center-out TPSF shows coherent interference and blurring, and the coefficients are coupled. This demonstrates the difficulty of deconvolution. The randomly shuffled TPSF shows incoherent interference and maintains a sharp central patch. The coefficients are decoupled, and the interference manifests as benign noise-like artifacts.



Supporting Figure S-5: System input and ideal system response for TPSF calculation. Each row represents a system input. The center of the active temporal coefficient image contains a 7-by-7 uniform magnitude-one patch.



Supporting Figure S-6: The TPSF for center-out echo train ordering shows coherent interference. The patch in the active coefficient leaks to adjacent coefficients and is blurred.



Supporting Figure S-7: TPSF for randomly shuffled echo train ordering. The main patch remains sharp, and interference is spread incoherently to all transform coefficients.

Sampling Pattern and Echo Train Formation

For large ETLs, it is not practical to generate a unique variable density Poisson disc sampling pattern for each TE separately. Instead, we outline an alternative technique described in Algorithm 1. The echo times are segmented into M batches and a unique variable-density

Poisson disc sampling pattern is generated for each of the M batches. The acceleration factor for each pattern is chosen so that there are at least

$$N_{\text{points}} = \frac{T}{M} \times N_{\text{trains}} \tag{34}$$

samples in each mask, where T is the ETL (minus initial skipped echoes) and N_{trains} is the number of echo trains, given by Eq. (17). The acceleration factor is computed as

$$R = \sqrt{\frac{\pi}{4} \frac{N_y \times N_z}{T \times N_{\text{trains}} \times \tau}}, \tag{35}$$

where (N_y, N_z) are the k-space phase encode dimensions and $\tau \approx 1.1$ is a "fudge factor" so that the number of actual points is greater than or equal to N_{points}. The masks are then randomly pruned until each mask contains exactly N_{points} samples. The $\pi/4$ factor is included because the corners of k-space are not acquired (6).

For the m^{th} sampling pattern, a small sliding window is moved until T/M points are selected. These points are randomly ordered to form the m^{th} segment of one echo train. The m^{th} segment of the subsequent echo train is formed by sliding the window until T/M new points are selected. This is repeated until all points are covered, and the procedure continues with the next sampling pattern. Since the echo trains are formed by grouping phase encodes locally, eddy current effects are potentially reduced.

Several user-defined parameters affect the generated sampling distribution in k-t space. The number of batches, M, controls the level of phase encode re-sampling. For large M, low frequency phase encodes are more likely to be re-acquired due to the variable density weighting. Note that T (excluding skipped echoes) must be divisible by M. Each sampling pattern can specify a fully-sampled central region and the degree of variable-density weighting. Finally, the sliding window size, (W_y, W_z), affects the degree of spatial locality for the echo train in k-space.

Algorithm 1 Randomly shuffled sampling pattern and echo train formation.

Inputs: N_y, N_z, N_{trains}, T, M, W_y, W_z, τ

Outputs: `trains [`N_{trains}`, T]` – array of phase encodes to acquire

`trains` $\leftarrow$ `zeros(`N_{trains}`, T)`

$N_{\text{points}} \leftarrow$ Eq. (34)

$R \leftarrow$ Eq. (35)

for $m = 1 : M$ **do**

    `mask` $\leftarrow$ `genVDPoissonMask(`N_y`, `N_z`, `R`, `τ`)`

    `mask` $\leftarrow$ `mask.randprune(`N_{points}`)`

    Initialize sliding window of size (W_y, W_z) in `mask`

    for $n = 1 : N_{\text{trains}}$ **do**

        `arr` $\leftarrow$ `[]`

        while `arr.size()` $< T/M$ **do**

            Add active (k_y, k_z) points within the sliding window to `arr`

            Move sliding window to next region in `mask`

        end while

        `arr` $\leftarrow$ `randperm(arr)`

        `trains[`n`].append(arr)`

    end for

end for