Fast Non-Cartesian Reconstruction with Pruned Fast Fourier Transform

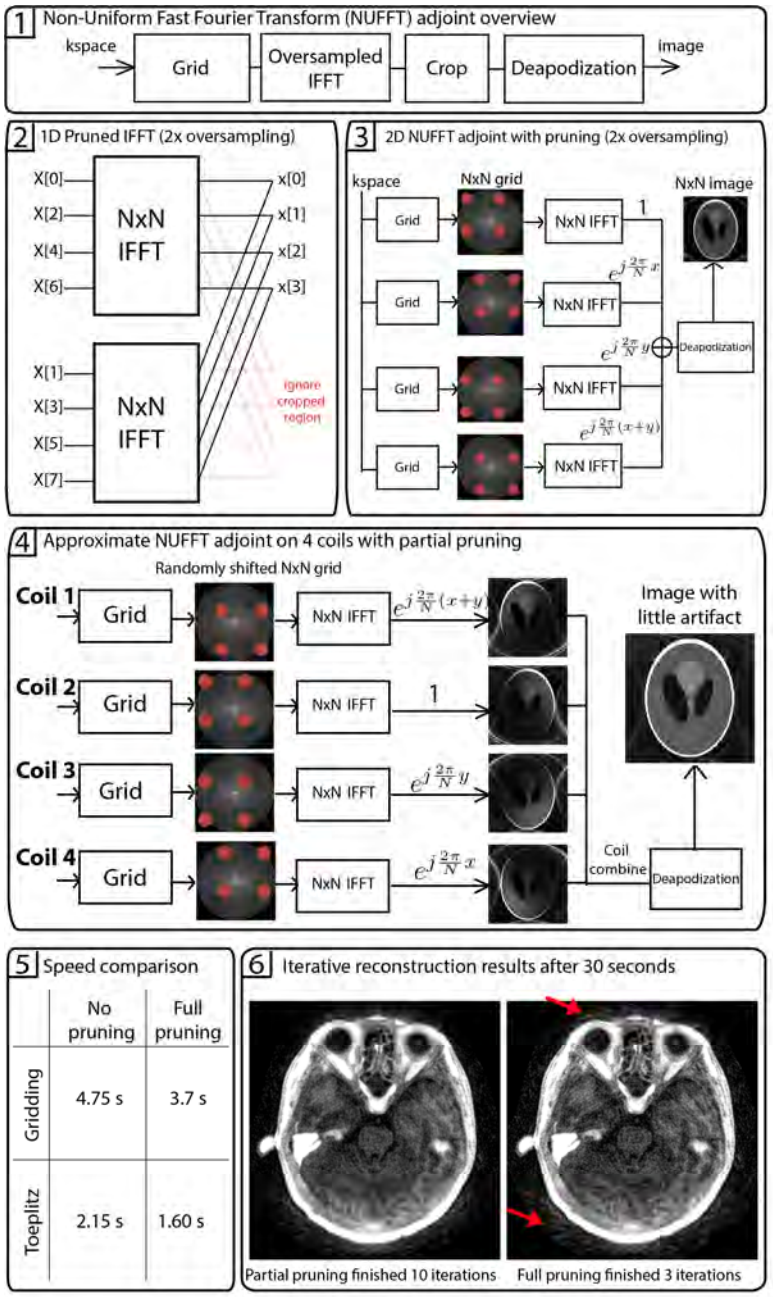
Frank Ong[1], Martin Uecker[1], Wenwen Jiang[2], and Michael Lustig[1]

[1]*Electrical Engineering and Computer Sciences, University of California, Berkeley, Berkeley, California, United States,* [2]*Bioengineering, UC Berkeley/UCSF, Berkeley, California, United States*

TARGET AUDIENCES: Those interested in fast non-Cartesian reconstruction.

PURPOSE: Non-Cartesian imaging is essential in applications such as ultra-short time-echo imaging and real-time cardiac imaging. However, reconstruction speed becomes a limitation when iterative reconstruction is desired for parallel imaging, compressed sensing or artifact correction. In this work, we propose using pruned Fast Fourier Transform (pruned FFT) to accelerate almost all fast non-Cartesian reconstruction methods (gridding, toeplitz-circulant embedding, iterative, non-iterative). For iterative methods, we also propose partial pruning to approximate the non-Cartesian Fourier Transform operator to speed up each iteration while guaranteeing convergence.

THEORY: To illustrate the concept of pruned FFT, we use the non-uniform FFT adjoint operation (NUFFT adjoint, Figure 1) as an example. Conventional NUFFT adjoint consists of an oversampled FFT followed by cropping, which requires ~2-8x more memory usage than the usual FFT. Instead of oversample then crop, pruned FFT simply prunes the computations on the cropped region, thus eliminating the need to initialize oversampled memory in the first place. Figure 2 shows an FFT butterfly flow diagram for 1D pruned FFT, illustrating how pruned FFT leverages the FFT butterfly structure to reduce memory and computation. Figure 3 demonstrates how pruned FFT works within NUFFT adjoint: instead of gridding on a 2x-oversampled grid and perform a 2N×2N IFFT, we can grid the kspace data on four shifted N×N grid, compute N×N FFTs and then compensate for the linear phase respectively.



Thus, using pruned FFT, **no memory overhead is required** when each shifted FFT is processed serially, **regardless of the oversampling factor**. Alternatively, significant speedup can be gained when each shifted FFT is parallelized. The pruned version is exactly equivalent to the original operation and can be applied to NUFFT forward, adjoint and toeplitz-circulant embedding.

To further accelerate iterative methods, we propose partial pruning. In each iteration, we randomly update only one shifted FFT for each coil. Random ghosting artifacts appear but are averaged over coils and iterations. When combined with gradient methods, partial pruning is guaranteed to converge in expectation as an instance of stochastic gradient method[2]. Since per-iteration speed is faster, operations other than Fourier transform can be done more frequently and overall convergence can be faster. Figure 4 illustrates adjoint NUFFT with partial pruning.

RESULTS AND DISCUSSIONS: Figure 5 shows the reconstruction speed comparison of L1-ESPIRiT[3] of a 128×128 test image with 200 iterations, 2x oversampled gridding and toeplitz-circulant embedding with and without (full) pruning. As shown, pruned FFT accelerates each method by ~1.3x. Figure 6 shows the reconstruction results with partial pruning and full pruning after 30 seconds with toeplitz-circulant embedding on a 292×264×74 sized brain dataset. Partial pruning was able to complete 10 iterations while full pruning was only able complete 3 iterations, thus more artifacts remain.

CONCLUSIONS: The proposed pruned FFT reduces computation time and memory overhead for both iterative and non-iterative methods. Partial pruning further accelerates iterative methods. The code is available online under BART.[4]

REFERENCES: [1] Markel, et al. IEEE Trans on Audio and Electroacoustics 1971;19:305-11. [2] Bertsekas, et al. Parallel and distributed computation 1989. [3] Uecker, et al. MRM 2014; 71: 990-1001 [4] BART. (2014) doi: 10.5281/zenodo.12495 https://www.eecs.berkeley.edu/~mlustig/Software.html