Large-Scale Distributed GPU-Accelerated Respiratory Motion-Resolved Reconstruction of 3D non-Cartesian mGRE MRI

Chao Zhang[1], Xiangmin Jiao[1,2], Michelle Nash[3], Ryan Schaake[4], and Youngwook Kee[1,4]

[1] Applied Mathematics & Statistics, Stony Brook University, USA
[2] Institute for Advanced Computational Science, Stony Brook University, USA
[3] Pediatric Hematology/Oncology, Stony Brook Children's, USA
[4] Radiology, Stony Brook University, USA
{chao.zhang.1,xiangmin.jiao,youngwook.kee}@stonybrook.edu
{michelle.nash,ryan.schaake}@stonybrookmedicine.edu

Abstract. We propose a hierarchical multi-node/multi-GPU framework for respiratory motion-resolved reconstruction of 3D non-Cartesian multi-echo gradient-echo MRI. The core idea is a topology-aware, three-level parallelization across the motion, echo, and coil dimensions: 1) we partition motion states and echo times across nodes using a 2D process grid, enabling row-wise (motion-coupled) and column-wise (echo-coupled) communications; and 2) within each node, we further distribute coil computations across GPUs so that communication-intensive coil reductions are carried out intra-node over NVLink via NCCL, while inter-node traffic is limited to nearest-neighbor halo exchanges over CXI via MPI. This design removes the single-node memory ceiling that previously prevented enlarged-FoV reconstructions and reduces end-to-end runtime by scaling to many nodes/GPUs. Experimental results demonstrate that our approach has the potential to make multi-dimensional reconstruction of 3D non-Cartesian abdominal MRI more practical and scalable.

Keywords: Distributed Computing · Asynchronous Optimization · Large-Scale Reconstruction · 3D Non-Cartesian MRI · GPU Acceleration

1 Introduction

Recent advances in 3D non-Cartesian k-space sampling trajectories and respiratory motion-resolved reconstruction have enabled free-breathing multi-echo gradient-echo (mGRE) abdominal MRI and motion-resolved R2*, proton density fat fraction (PDFF) and quantitative susceptibility mapping (QSM) [9, 10]. Clinical applications include the assessment of hepatic iron overload [6] and the evaluation of fatty liver disease [13]. These techniques have been shown to mitigate respiratory motion-induced confounding factors in quantitative tissue parameter maps, benefiting pediatric patients and some adults unable to hold their breath, while achieving high spatial resolution and accelerated imaging [8].

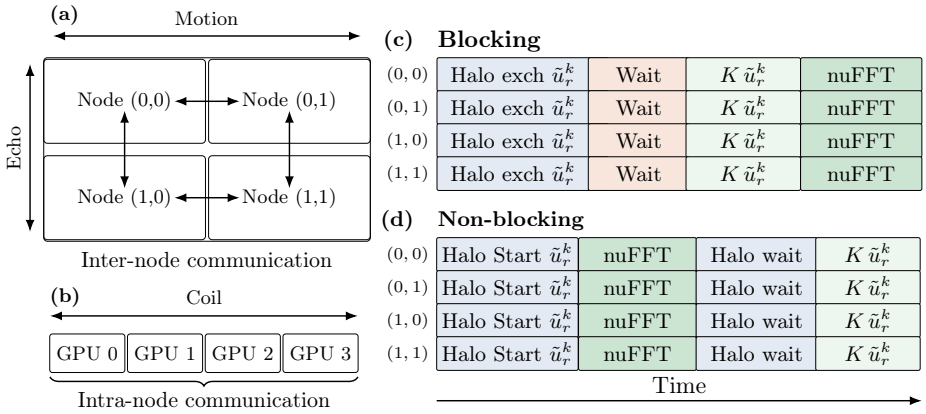


Fig. 1. Overview of the proposed hierarchical communication and execution scheme. (a) Echo-motion process grid with inter-node halo exchange. (b) Intra-node coil-parallel communication across GPUs. (c) Blocking execution. (d) Non-blocking execution that overlaps halo exchange with local nuFFT computation.

Despite the acquisition advantages of full 3D non-Cartesian trajectories (e.g., 3DPR, 3D Cones [4], FLORET [18], Yarnball [25]) over stacks of 2D non-Cartesian readouts [23, 26], reconstruction of 3D non-Cartesian data remains computationally demanding because the 3D encoding operator is non-separable. Runtimes are prohibitively long for just-in-time reconstruction, often exceeding the exam duration. Main computational bottlenecks include: 1) evaluating the 3D non-uniform Fourier transform (nuFFT) and its adjoint across all coils, echoes, and motion states per iteration, 2) reconstructing an enlarged field of view (FoV) to mitigate aliasing from signals excited outside the nominal FoV by nonselective excitations, and 3) subject-specific tuning of regularization parameters that requires a grid search. Prior implementations [21, 22] typically run on a single multi-GPU node (e.g., 4-8 GPUs [15]), limiting scalability. Consequently, pushing motion-resolved quantitative maps back to the scanner immediately after acquisition remains challenging.

In this paper, we propose a hierarchical, multi-node, multi-GPU framework for motion-resolved reconstruction of 3D non-Cartesian mGRE MRI. As illustrated in Fig. 1, our contributions are: *1) topology-aware 3D decomposition*—a 2D inter-node process grid partitions motion and echo dimensions (Fig. 1a) for structured nearest-neighbor exchanges, while coils are split across local GPUs (Fig. 1b); *2) cost-aware communication placement*—expensive coil-wise reductions are confined to high-bandwidth intra-node GPU collectives (Fig. 1b), whereas lighter boundary exchanges use inter-node MPI (Fig. 1a), minimizing cross-node traffic; *3) non-blocking asynchronous execution*—halo exchanges and collectives overlap with nuFFT computations, reducing explicit waiting (Fig. 1c–d); and *4) partition-aware I/O*—each GPU loads only local data, lowering memory footprint and enabling enlarged-FoV reconstructions. We demonstrate scalability up

to *36 compute nodes (144 GPUs).* Source code, scripts, and example datasets are available at `https://github.com/MRI-Research/MultiNodeMRI-MICCAI26`.

Related work. Most multi-GPU iterative methods accelerate reconstruction by decomposing workloads along coils and, for dynamic settings, time/frames, parallelizing nuFFT operations [7, 21, 22]. Gadgetron-based cloud reconstruction [5, 28], generic multi-node/multi-GPU frameworks [19], and BART [20, 27] provide infrastructure for modular reconstruction, online deployment, pipeline execution, and general MPI/multi-GPU support. In contrast, we partition a single tightly coupled 5D non-Cartesian optimization problem internally across echoes/motion-states while keeping coil reductions node-local. This topology-aware process grid is tailored to halo exchanges induced by motion/echo regularization and coil-summed adjoint nuFFT reductions, rather than dispatching independent reconstruction jobs or pipeline blocks.

2 Theory

Respiratory Motion-Resolved Reconstruction of mGRE Data. Let E, Q, and C denote the numbers of echoes, fully sampled (Nyquist-rate) k-space samples, and receive-coil elements, respectively. During free breathing, incoherently and continuously acquired non-Cartesian k-space samples are collected as vectors $g_{e,c} \in \mathbb{C}^Q$ for $e = 1, \dots, E$ and $c = 1, \dots, C$. Let T denote the number of latent respiratory motion states used to bin these samples. Let π be a permutation of $\{1, \dots, Q\}$, and let $P_\pi \in \{0,1\}^{Q \times Q}$ be the corresponding permutation matrix. Let $b_t = \lceil tQ/T \rceil$ for $t = 0, \dots, T$. We define $\tilde{g}_{e,c} := P_\pi g_{e,c} = (\tilde{g}^1_{e,c}, \dots, \tilde{g}^Q_{e,c})^\top$ such that samples assigned to motion state t form $\tilde{g}^{b_{t-1}+1}_{e,c}, \dots, \tilde{g}^{b_t}_{e,c}$ for $t = 1, \dots, T$, so each bin contains at most $\lceil Q/T \rceil$ samples. Let N be the image size, and let $u_1, \dots, u_E \in \mathbb{C}^{N \cdot T}$ denote the desired vectorized motion-state-resolved mGRE images. Kang et al. [9] proposed to minimize

$$J_{\text{TVM-L2}} = \sum_{e=1}^{E} \sum_{c=1}^{C} \|F S_c u_e - \tilde{g}_{e,c}\|_W^2 + \lambda_m \left(\sum_{e=1}^{E} \|\Delta_m^+ u_e\|_1^2 \right)^{1/2}, \quad (1)$$

where $S_c = \text{diag}\{(s_c^1, \dots, s_c^{N \cdot T})^\top\}$ is a diagonal matrix constructed from the coil sensitivity vector $(s_c^1, \dots, s_c^{N \cdot T})^\top$, and $F : \mathbb{C}^{N \cdot T} \to \mathbb{C}^Q$ is the nuFFT operator [1, 3]. The weighted least-squares fidelity $\|F S_c u_e - \tilde{g}_{e,c}\|_W^2$ is defined as $(F S_c u_e - \tilde{g}_{e,c})^H W (F S_c u_e - \tilde{g}_{e,c})$, where $W = \text{diag}(w_1, \dots, w_Q) \in \mathbb{R}^{Q \times Q}$ is the diagonal weighting matrix formed from the density compensation factors. The operator $\Delta_m^+ : \mathbb{C}^{N \cdot T} \to \mathbb{C}^{N \cdot T}$ denotes the forward difference along the motion-state dimension, and λ_m is a regularization parameter.

Let $\mathbf{u} := [u_1^\top, \dots, u_E^\top]^\top \in \mathbb{C}^{N \cdot T \cdot E}$ and $\tilde{\mathbf{g}}_c := [\tilde{g}_{1,c}^\top, \dots, \tilde{g}_{E,c}^\top]^\top \in \mathbb{C}^{Q \cdot E}$. Then, Kang et al. [8] proposed the following 5D reconstruction model (3D spatial + 1D motion + 1D echo signal evolution):

$$J_{\text{TVME-L2}} = \sum_{c=1}^{C} \|\mathbf{F S}_c \mathbf{u} - \tilde{\mathbf{g}}_c\|_{\mathbf{W}}^2 + \lambda_m \|\mathbf{\Delta}_m^+ \mathbf{u}\|_{1,2} + \lambda_e \|\mathbf{D \Delta}_e^+ \mathbf{u}\|_1, \quad (2)$$

Algorithm 1 Asynchronous PDHG for Node (i,j), GPU ℓ

Require: $\mathbf{u}^0_{i,j}, \bar{\mathbf{u}}^0_{i,j}, \{\mathbf{q}^0_{c,i,j}\}_{c\in\mathcal{C}_\ell}, \mathbf{p}^0_{i,j}$, tolerance $\epsilon, K_{\max}$

for $k = 0, 1, \ldots, K_{\max}-1$ **do**
  $\mathcal{R}_u \leftarrow \texttt{HaloStart}(\bar{\mathbf{u}}^k_{i,j})$ ▷ Async exchange along motion/echo
  $\mathbf{q}^{k+1}_{c,i,j} \leftarrow \mathrm{prox}_{\sigma f^*_c}\left(\mathbf{q}^k_{c,i,j} + \sigma\,\mathbf{F}\mathbf{S}_c\bar{\mathbf{u}}^k_{i,j}\right), \quad \forall c \in \mathcal{C}_\ell$ ▷ Eq. (4a), Parallelized
  $\texttt{HaloWait}(\mathcal{R}_u)$; $\mathbf{p}^{k+1}_{i,j} \leftarrow \mathrm{prox}_{\sigma g^*}\left(\mathbf{p}^k_{i,j} + \sigma\,\mathbf{K}\bar{\mathbf{u}}^k_{i,j}\right)$ ▷ Eq. (4b)
  $\mathcal{R}_p \leftarrow \texttt{HaloStart}(\mathbf{p}^{k+1}_{i,j})$ ▷ Async exchange for $\mathbf{K}^H$
  $\mathbf{z}^{(\ell)}_{i,j} \leftarrow \sum_{c\in\mathcal{C}_\ell}(\mathbf{F}\mathbf{S}_c)^H\mathbf{q}^{k+1}_{c,i,j}$ ▷ Local partial adjoint
  $\mathcal{R}_z \leftarrow \texttt{AllreduceStart}_{\text{intra}}\left(\mathbf{z}^{(\ell)}_{i,j}\right)$ ▷ Async Intra-node sum
  $\texttt{HaloWait}(\mathcal{R}_p)$; $\mathbf{w}_{i,j} \leftarrow \mathbf{K}^H\mathbf{p}^{k+1}_{i,j}$
  $\mathbf{z}_{i,j} \leftarrow \texttt{AllreduceWait}_{\text{intra}}(\mathcal{R}_z)$
  $\mathbf{u}^{k+1}_{i,j} \leftarrow \mathbf{u}^k_{i,j} - \tau(\mathbf{z}_{i,j} + \mathbf{w}_{i,j})$; $\bar{\mathbf{u}}^{k+1}_{i,j} \leftarrow 2\mathbf{u}^{k+1}_{i,j} - \mathbf{u}^k_{i,j}$ ▷ Eqs. (4c), (4d)
  if $\|\mathbf{u}^{k+1} - \mathbf{u}^k\|_2 / \|\mathbf{u}^k\|_2 < \epsilon$ **then break**
  end if
end for

where $\mathbf{S}_c := \mathrm{diag}(S_c, \ldots, S_c) : \mathbb{C}^{N\cdot T\cdot E} \to \mathbb{C}^{N\cdot T\cdot E}$, $\mathbf{F} := \mathrm{diag}(F, \ldots, F) : \mathbb{C}^{N\cdot T\cdot E} \to \mathbb{C}^{Q\cdot E}$, $\mathbf{\Delta}^+_m := \mathrm{diag}(\Delta^+_m, \ldots, \Delta^+_m) : \mathbb{C}^{N\cdot T\cdot E} \to \mathbb{C}^{N\cdot T\cdot E}$, and $\mathbf{D} = \mathrm{diag}(\nabla, \ldots, \nabla) : \mathbb{C}^{N\cdot T\cdot E} \to \mathbb{C}^{3\cdot N\cdot T\cdot E}$. In the weighted least-squares fidelity, $\mathbf{W} := \mathrm{diag}(W, \ldots, W) = I_E \otimes W : \mathbb{C}^{Q\cdot E} \to \mathbb{C}^{Q\cdot E}$. A similar 5D formulation using wavelet transforms was proposed in [10]:

$$J_{\text{TVMW-L1}} = \sum_{c=1}^{C} \|\mathbf{F}\mathbf{S}_c\mathbf{u} - \tilde{\mathbf{g}}_c\|^2_{\mathbf{W}} + \lambda_m\|\mathbf{\Delta}^+_m\mathbf{u}\|_1 + \lambda_e\|W_e\mathbf{u}\|_1 + \lambda_s\|W_s\mathbf{u}\|_1, \quad (3)$$

where W_e and W_s are echo and spatial wavelet transforms, and λ_e, λ_s are regularization parameters. These 5D reconstructions extend Eq. (1) and have demonstrated improved performance in prior work [8, 10].

When applied to Eqs. (1)–(3), the primal-dual hybrid gradient (PDHG) method [2] yields the following updates at iteration k:

$$\mathbf{q}^{k+1}_c = \mathrm{prox}_{\sigma f^*_c}\left(\mathbf{q}^k_c + \sigma\mathbf{F}\mathbf{S}_c\bar{\mathbf{u}}^k\right), \quad c = 1, \ldots, C, \quad (4a)$$

$$\mathbf{p}^{k+1} = \mathrm{prox}_{\sigma g^*}\left(\mathbf{p}^k + \sigma\mathbf{K}\bar{\mathbf{u}}^k\right), \quad (4b)$$

$$\mathbf{u}^{k+1} = \mathbf{u}^k - \tau\left(\sum_{c=1}^{C}\mathbf{S}^H_c\mathbf{F}^H\mathbf{q}^{k+1}_c + \mathbf{K}^H\mathbf{p}^{k+1}\right), \quad (4c)$$

$$\bar{\mathbf{u}}^{k+1} = 2\mathbf{u}^{k+1} - \mathbf{u}^k. \quad (4d)$$

Here, each $\mathbf{q}_c \in \mathbb{C}^{Q\cdot E}$ is a dual variable for the data-fidelity terms, and $\mathbf{p}$ is the dual variable for the regularizer. The operator $\mathbf{K}$ is $\mathbf{\Delta}^+_m$ for TVM-L2, $[\mathbf{\Delta}^+_m; \mathbf{D}\mathbf{\Delta}^+_e]$ for TVME-L2, and $[\mathbf{\Delta}^+_m; W_e; W_s]$ for TVMW-L1; σ and τ are step sizes.

Blocking and Non-blocking Multi-Node/Multi-GPU Implementations. We implement distributed PDHG on a 2D MPI process grid of size $G_e \times G_m$ that partitions the *echo* and *motion* dimensions across nodes. This hierarchy increases parallelism for fixed data dimensions (E, T, C) and reduces inter-node traffic by confining communication-heavy coil reductions to high-bandwidth intra-node

links (e.g., NVLink/NCCL), while using inter-node links (e.g., CXI) mainly for nearest-neighbor halo exchanges induced by coupling operators. Each node (i, j) owns a local image block $\mathbf{u}_{i,j}$ for a contiguous subset of echo and motion indices, containing all spatial points for that subset. Coil-dependent data and sensitivity-map operations are partitioned across GPUs within the node. Within node (i, j), let ℓ denote the local GPU index; GPU ℓ stores all spatial points for the local echo-motion block but computes only on a local subset of coils, denoted by $\mathcal{C}_\ell$. During the adjoint nuFFT computation, intra-node collectives (allreduce) are required to accumulate the partial coil-wise contributions across GPUs. Since the regularization operators in $\mathbf{K}$ are nearest-neighbor, evaluating $\mathbf{K}\bar{\mathbf{u}}^k$ and $\mathbf{K}^H\mathbf{p}^{k+1}$ requires only inter-node halo exchanges with adjacent nodes.

A *single-node* solver ($G_e = G_m = 1$) is often infeasible due to memory limits. A *synchronous multi-node* implementation alleviates this but performs blocking communication, causing GPU idling. To mitigate this, we propose an *asynchronous* pipeline (Algorithm 1) mapping Eqs. (4a)–(4d) to our architecture. The dual update (Eq. (4a)) is perfectly parallelized across local GPUs. The primal update (Eq. (4c)) requires the coil-summed data-term adjoint $\mathbf{z}_{i,j}$. Each GPU evaluates its partial adjoint $\mathbf{z}_{i,j}^{(\ell)} = \sum_{c\in\mathcal{C}_\ell}(\mathbf{F}\mathbf{S}_c)^H\mathbf{q}_{c,i,j}^{k+1}$ and initiates an asynchronous intra-node allreduce. This effectively hides the latency of this reduction and inter-node halo exchanges for $\mathbf{K}$ and $\mathbf{K}^H$ behind the dominant nuFFT computations.

3 Experimental Methods

With IRB approval, three healthy volunteers (HV01-02) and two patients with suspected iron overload (PT01-03) were recruited. Subjects were scanned on a 3 T scanner (SIGNA Architect, GE Healthcare) using a free-breathing 3D mGRE Cones sequence [4, 11]. Imaging parameters were: initial TE/ΔTE/TR = 0.036/1.1/10.4 ms; #TEs = 6 (ETL/#shots = 3/2); FA = 3 to 4°; in-plane resolution 2 × 2 mm^2, slice thickness = 2 mm, nominal FoV$_x$ = FoV$_y$ = 40 cm and FoV$_z$ = 20 to 24 cm, rBW = 1,250 Hz/Px, readout duration = 0.668 ms, #interleaves = 29,947 to 35,521, and scan time = 10 to 13 min. A non-selective hard pulse was employed.

Reconstructions ran on a Cray HPC with Grace-Hopper nodes (four 96 GB H100 GPUs/node), using NCCL over NVLink intra-node and MPI over CXI inter-node. Software included Slurm, CUDA 12.x, and Cray MPICH. Unless specified, experiments used an enlarged FoV doubling the nominal matrix size. R2*, PDFF, and QSM were estimated from $(2, 2)$ and $(6, 6)$ grid reconstructions using methods in [8–10].

Computational performance was evaluated using *reconstruction runtime*, measured as the wall-clock time of the iterative solver, and *speedup*, which quantifies the relative runtime reduction across algorithm configurations. Memory scalability was evaluated using the *peak per-GPU memory footprint (VRAM)*, which determines the feasibility of high-dimensional reconstructions. Numerical correctness was assessed using *relative MSEs* against reference reconstructions: Grid

Table 1. Runtime (s) and peak per-GPU VRAM (GiB) on HV01.

Multi-Node Grid		TVM-L2		TVME-L2		TVMW-L1	
(e, m)	#GPUs	Time	VRAM	Time	VRAM	Time	VRAM
(1,1)	1	–	OOM	–	OOM	–	OOM
(1,1)	4	–	OOM	–	OOM	–	OOM
(1,2)	8	772.32	86.7	–	OOM	–	OOM
(2,1)	8	788.16	85.7	–	OOM	–	OOM
(1,3)	12	526.77	61.3	575.10	85.0	–	OOM
(3,1)	12	568.60	59.4	655.78	87.4	–	OOM
(2,2)	16	404.43	48.1	449.49	67.6	1085.96	74.3
(1,6)	24	274.76	35.6	296.75	49.8	725.55	53.1
(2,3)	24	276.51	37.2	305.95	48.3	728.26	53.1
(3,2)	24	294.15	35.3	338.74	48.2	734.98	53.7
(6,1)	24	334.79	36.8	405.10	51.6	747.84	49.8
(3,3)	36	205.19	26.5	236.59	35.1	493.94	37.9
(2,6)	48	145.67	21.9	167.41	28.4	373.38	29.5
(6,2)	48	180.72	22.0	221.63	29.4	386.86	28.0
(3,6)	72	108.86	17.5	126.74	21.8	269.74	22.4
(6,3)	72	127.76	17.1	153.38	22.3	272.41	21.7
(6,6)	144	71.04	13.1	96.61	16.3	161.95	13.9

$(1, 1)$ with 4 GPUs for the nominal FoV and Grid $(2, 2)$ with 16 GPUs for the enlarged FoV. ROIs were placed inside the object in R2*, PDFF, and QSM maps to obtain measurements (mean ± sd) [8–10].

4 Experimental Results

Our 2D ($G_e \times G_m$) partitioning bypasses single-node memory ceilings and scales to many nodes/GPUs for 5D reconstructions. A single compute node (Grid $(1, 1)$) resulted in out-of-memory (OOM) failures due to the massive 5D operator memory footprint (Table 1). Echo-only (1D) partitioning imposes a hard scalability limit, capping the system at 6 nodes (24 GPUs; 4 GPUs per node). By simultaneously splitting motion, the 2D grid reaches 36 nodes (Grid $(6, 6)$; 144 GPUs). The overall improvements in reconstruction time and peak per-GPU VRAM from Grid $(2, 2)$ to $(6, 6)$ were $81.81 \pm 2.04\%$ and $76.55 \pm 3.74\%$, respectively.

Furthermore, for matched distributed and reference reconstructions, the relative MSE was 3.4×10^{-15}, corresponding to a relative RMSE of 5.8×10^{-8}. Mean absolute ROI-based element-wise differences in R2*, PDFF, and QSM maps were approximately 10^{-5}. Together with prior validation of the underlying 5D respiratory motion-resolved reconstruction models [8, 10], these results verify that the proposed distributed implementation preserves reconstruction and quantitative-map outputs to floating-point numerical precision.

Crucially, multi-node scaling substantially reduces reconstruction times (Tables 1 and 2). Expanding from 16 to 144 GPUs reduces the highly complex

Table 2. Runtime (s) and peak VRAM per GPU (GiB) on HV02 to PT03.

Subj	Multi-Node Grid		TVM-L2		TVME-L2		TVMW-L1	
	(e, m)	#GPUs	Time	VRAM	Time	VRAM	Time	VRAM
HV02	(2,2)	16	355.44	47.3	403.06	66.7	1037.71	73.8
	(3,3)	36	184.27	25.8	216.08	34.4	476.43	38.1
	(6,6)	144	67.76	12.5	86.84	15.7	176.92	13.6
PT01	(2,2)	16	395.12	45.7	462.23	64.6	1039.82	71.4
	(3,3)	36	206.65	25.0	227.22	33.4	470.95	36.9
	(6,6)	144	67.93	12.7	82.10	15.7	185.20	13.7
PT02	(2,2)	16	401.22	45.6	453.16	67.0	1097.37	73.8
	(3,3)	36	208.15	26.0	235.42	33.7	504.57	37.9
	(6,6)	144	70.09	13.1	88.40	16.3	180.98	13.9
PT03	(2,2)	16	312.86	39.2	347.76	55.4	888.83	61.8
	(3,3)	36	156.54	21.3	182.55	28.5	413.31	31.6
	(6,6)	144	59.50	10.7	72.38	13.3	135.25	11.6

TVMW-L1 runtime from 14.8–18.3 min to 2.3–3.1 min. TVME-L2 drops from 5.8–7.7 min to 1.2–1.6 min. This facilitates the implementation of practical, just-in-time, motion-resolved quantitative reconstruction workflows.

For a fixed node budget, the optimal (G_e, G_m) depends on the echo coupling in regularization (L2 or L1). With L1 coupling, nearest-neighbor halos dominate and runtime is largely grid-shape insensitive (Fig. 2). With L2 coupling, coupled-dimension allreduce cannot be fully hidden and costs grow with echo splitting, producing latency penalties: TVME-L2 increases from 296.8 s at $(1, 6)$ to 405.1 s at $(6, 1)$ (+36.5%), and from 167.4 s at $(2, 6)$ to 221.6 s at $(6, 2)$ (+32.4%) (Table 1). These results suggest *prioritizing motion splitting* (maximize G_m) and keeping the echo split G_e as small as memory allows.

To validate the effectiveness of our communication-hiding design, we evaluate the strong-scaling speedup of the asynchronous pipeline against a synchronous baseline using the TVME-L2 model. To explicitly establish a 1-GPU baseline for this strong-scaling study, which is infeasible under the default enlarged FoV due to memory limits, we scale the problem down to a nominal FoV.

TVME-L2 is a relevant stress test because it adds coupling-induced communication (e.g., halo exchanges and reductions) that can stall GPU streams at scale if executed synchronously. Fig. 3 compares synchronous vs. asynchronous speedup, computed as the 1-GPU runtime divided by the runtime at each GPU count under identical FoV and solver settings. Up to 16 GPUs, the curves are similar, indicating that compute (i.e., nuFFT/adjoint-nuFFT and proximal updates) still dominates. At larger GPU counts, communication and synchronization become stronger limiting factors, and communication–computation overlap becomes increasingly beneficial: asynchronous execution reaches 42X vs. 35X speedup at 72 GPUs and 52X vs. 41X at 144 GPUs, corresponding to approximately 20% and 26% higher speedup, respectively.

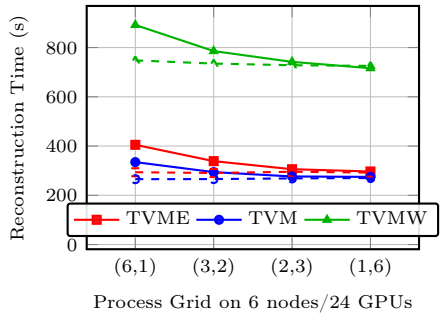


Fig. 2. Runtime under different grid shapes (G_e, G_m). Solid: L2 coupling; dashed: L1 coupling.

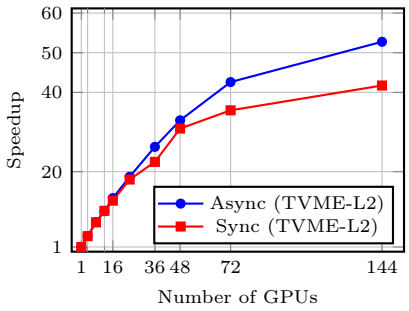
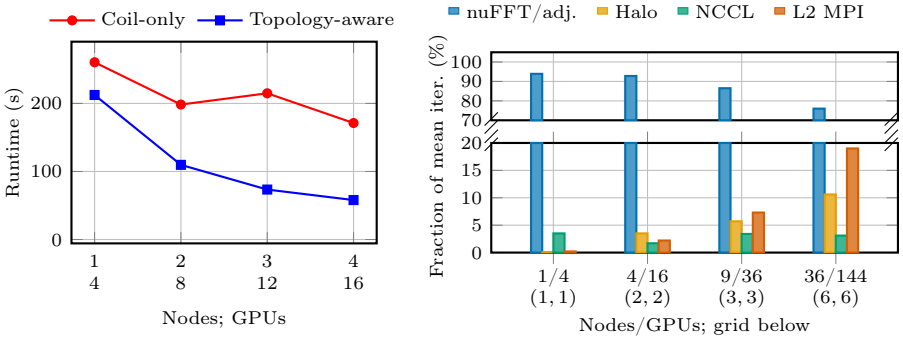


Fig. 3. Speedup of TVME-L2 for synchronous and asynchronous executions (nominal FoV).



(a) Runtime scaling

(b) Per-iteration profiling

Fig. 4. Topology-oblivious (coil-only) vs. topology-aware runtimes, and asynchronous per-iteration profiling for TVME-L2 at nominal FoV. (a) Runtime comparison between topology-oblivious (coil-only) and topology-aware configurations. (b) Per-iteration breakdown of topology-aware grids; percentages are relative to mean iteration time, and L2 MPI denotes MPI communication for the L2-coupled regularizer.

Fig. 4(a) shows that topology-aware configurations consistently outperform topology-oblivious coil-only splits at matched GPU counts (e.g., 57.97 s vs. 171.19 s on 16 GPUs). Note that the next possible coil-only/topology-oblivious configuration was 8 nodes/32 GPUs with a runtime of 176.84 s, whereas the closest corresponding topology-aware configuration was 9 nodes/36 GPUs with a runtime of 30.07 s. The 32-GPU coil-only point indicates saturation, lacking improvement over 16 GPUs. Fig. 4(b) shows nuFFT operations remain the dominant cost. Overlapped fractions are non-additive: at 144 GPUs, nuFFT accounts for 68 ms (76.0%) of the 89 ms iteration. The gap between L2 MPI (19.0%) and exposed halo wait (10.6%) indicates communication is successfully hidden, validating our strategy of restricting inter-node traffic to halo exchanges.

5 Discussion and Conclusion

The proposed framework is highly extensible. First, it adapts to other interconnects (e.g., InfiniBand) via MPI/NCCL backends. Second, it allows flexible hardware allocation to resolve coil-count load imbalances by reallocating a fixed GPU budget across more nodes, balancing the (G_e, G_m) grid. Third, while evaluated on 144 GPUs, the 2D topology intrinsically scales to nodes with more GPUs or larger echo/motion dimensions. Fourth, our implementation uses a SigPy/CuPy nuFFT backend [16, 17]; however, the proposed parallelization is backend-agnostic and can leverage gpuNUFFT [12], cuFINUFFT [24], or torchkbnufft [14]. Finally, the framework is trajectory- and platform-agnostic.

The framework exhibited implementation-dependent performance gaps across objectives. Notably, TVMW-L1 yielded consistently longer reconstruction times than TVM-L2 and TVME-L2. This occurs because our current wavelet implementation utilizes `ptwt` from PyTorch alongside a CuPy-optimized pipeline, leading to repeated framework conversions and synchronization overhead. Future work will implement a GPU-native wavelet operator within a unified array framework and pursue acceleration of complementary methods, such as low-rank/subspace and physics-driven learned reconstructions.

In conclusion, our distributed multi-GPU framework resolves the computational and memory bottlenecks of 5D respiratory motion-resolved MRI. By combining a 2D process grid with an asynchronous communication-hiding pipeline, this scalable approach renders previously prohibitive reconstructions feasible.

Acknowledgments. This work was supported in part by the National Institutes of Health under R21EB033985 and R21DK137146, and by SUNY System Administration using the SUNY AI Platform. Computational resources were provided by NCSA DeltaAI through ACCESS allocation MTH250015.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Beatty, P.J., Nishimura, D.G., Pauly, J.M.: Rapid gridding reconstruction with a minimal oversampling ratio. IEEE Trans. Med. Imaging **24**(6), 799–808 (2005)
2. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. J. Math. Imaging Vis. **40**(1), 120–145 (2011)
3. Fessler, J.A., Sutton, B.P.: Nonuniform fast Fourier transforms using min-max interpolation. IEEE Trans. Signal Process. **51**(2), 560–574 (2003)
4. Gurney, P.T., Hargreaves, B.A., Nishimura, D.G.: Design and analysis of a practical 3D cones trajectory. Magn. Reson. Med. **55**(3), 575–582 (2006)
5. Hansen, M.S., Sørensen, T.S.: Gadgetron: An open source framework for medical image reconstruction. Magn. Reson. Med. **69**(6), 1768–1776 (2013)
6. Hernando, D., Levin, Y.S., Sirlin, C.B., Reeder, S.B.: Quantification of liver iron with MRI: State of the art and remaining challenges. J. Magn. Reson. Imaging **40**(5), 1003–1021 (2014)

7. Hulet, J.P., Adluru, G., Facelli, J.C., DiBella, E., Parker, D.L.: Fast temporally constrained reconstruction on a GPU cluster. In: Proc. 20th Annu. Meet. Int. Soc. Magn. Reson. Med. (ISMRM). p. 2266. Melbourne, Australia (2012), abstract 2266
8. Kang, M., Alus, O., Kee, Y.: Accelerated free-breathing 5D multi-echo respiratory motion-resolved R2*, PDFF, and QSM using novel composite total variation. In: Med. Image Comput. Comput. Assist. Interv. vol. 15962, pp. 24–34 (2025)
9. Kang, M., Behr, G.G., Jafari, R., Gambarin, M., Otazo, R., Kee, Y.: Free-breathing high isotropic resolution quantitative susceptibility mapping (QSM) of liver using 3D multi-echo UTE cones acquisition and respiratory motion-resolved image reconstruction. Magn. Reson. Med. **90**(5), 1844–1858 (2023)
10. Kang, M., Otazo, R., Behr, G., Kee, Y.: 5D image reconstruction exploiting space-motion-echo sparsity for accelerated free-breathing quantitative liver MRI. Med. Image Anal. **102**, 103532 (2025)
11. Kee, Y., Sandino, C.M., Syed, A.B., Cheng, J.Y., Shimakawa, A., Colgan, T.J., Hernando, D., Vasanawala, S.S.: Free-breathing mapping of hepatic iron overload in children using 3D multi-echo UTE cones MRI. Magn. Reson. Med. **85**(5), 2608–2621 (2021)
12. Knoll, F., Schwarzl, A., Diwoky, C., Sodickson, D.K.: gpuNUFFT: An open-source GPU library for 3D gridding with direct MATLAB interface. In: Proc. Intl. Soc. Mag. Reson. Med. p. 4297 (2014)
13. Li, Q., Dhyani, M., Grajo, J.R., Sirlin, C., Samir, A.E.: Current status of imaging in nonalcoholic fatty liver disease. World J. Hepatol. **10**(8), 530–542 (2018)
14. Muckley, M.J., Stern, R., Murrell, T., Knoll, F.: TorchKbNufft: A high-level, hardware-agnostic non-uniform fast Fourier transform. In: ISMRM Workshop on Data Sampling & Image Reconstruction (2020)
15. NVIDIA Corporation: NVIDIA DGX H100/H200 User Guide (Jan 2026), `https://docs.nvidia.com/dgx/dgxh100-user-guide/dgxh100-user-guide.pdf`
16. Okuta, R., Unno, Y., Nishino, D., Hido, S., Loomis, C.: CuPy: A NumPy-compatible library for NVIDIA GPU calculations. In: Proc. LearnSys Workshop, NeurIPS (2017)
17. Ong, F., Lustig, M.: SigPy: A Python package for high-performance iterative reconstruction. In: Proc. Intl. Soc. Mag. Reson. Med. p. 4819 (2019)
18. Pipe, J.G., Zwart, N.R., Aboussouan, E.A., Robison, R.K., Devaraj, A., Johnson, K.O.: A new design and rationale for 3D orthogonally oversampled k-space trajectories. Magn. Reson. Med. **66**(5), 1303–1311 (2011)
19. Saybasili, H., Herzka, D.A., Barkauskas, K., Seiberlich, N., Griswold, M.A.: A generic, multi-node, multi-GPU reconstruction framework for online, real-time, low-latency MRI. In: Proc. Intl. Soc. Mag. Reson. Med. p. 3838 (2013)
20. Schaten, P., Blumenthal, M., Rapp, B., Unterberg-Buchwald, C., Uecker, M.: BART streams: Real-time reconstruction using a modular framework for pipeline processing (2025), arXiv:2512.02748
21. Schätz, S., Uecker, M.: A multi-GPU programming library for real-time applications. In: Proc. ICA3PP. vol. 7439, pp. 114–128 (2012)
22. Schätz, S., Voit, D., Frahm, J., Uecker, M.: Accelerated computing in magnetic resonance imaging: Real-time imaging using nonlinear inverse reconstruction. Comput. Math. Methods Med. **2017**, 3527269 (2017)
23. Schneider, M., Benkert, T., Solomon, E., Nickel, D., Fenchel, M., Kiefer, B., Maier, A., Chandarana, H., Block, K.T.: Free-breathing fat and R2* quantification in the liver using a stack-of-stars multi-echo acquisition with respiratory-resolved model-based reconstruction. Magn. Reson. Med. **84**(5), 2592–2605 (2020)

24. Shih, Y.H., Wright, G., Andén, J., Blaschke, J., Barnett, A.H.: cuFINUFFT: A load-balanced GPU library for general-purpose nonuniform FFTs. In: Proc. IEEE IPDPS Workshops. pp. 688–697 (2021)
25. Stobbe, R.W., Beaulieu, C.: Three-dimensional yarnball k-space acquisition for accelerated MRI. Magn. Reson. Med. **85**(4), 1840–1854 (2021)
26. Thedens, D.R., Irarrazaval, P., Sachs, T.S., Meyer, C.H., Nishimura, D.G.: Fast magnetic resonance coronary angiography with a three-dimensional stack of spirals trajectory. Magn. Reson. Med. **41**(6), 1170–1179 (1999)
27. Uecker, M., Ong, F., Tamir, J.I., Bahri, D., Virtue, P., Cheng, J.Y., Zhang, T., Lustig, M.: Berkeley Advanced Reconstruction Toolbox. In: Proc. Intl. Soc. Mag. Reson. Med. p. 2486 (2015)
28. Xue, H., Inati, S., Sørensen, T.S., Kellman, P., Hansen, M.S.: Distributed MRI reconstruction using gadgetron-based cloud computing. Magn. Reson. Med. **73**(3), 1015–1025 (2015)