Stage-Aligned Knowledge Distillation for Fast and Memory-Efficient 4D Flow MRI Reconstruction

Yuliang Xiao[1,2(✉)], Kian Anvari Hamedani[1,2], Zach Vavasour[1,2], Simon J. Graham[1,2], and Mark Chiew[1,2]

[1] Physical Sciences Platform, Sunnybrook Research Institute, Toronto, ON, Canada
[2] Department of Medical Biophysics, Temerty Faculty of Medicine, University of Toronto, Toronto, ON, Canada
yl.xiao@mail.utoronto.ca

Abstract. Reducing the stage count of a physics-guided unrolled network accelerates four-dimensional (4D) flow magnetic resonance imaging (MRI) reconstruction but may degrade image and flow fidelity. We study stage-aligned knowledge distillation (KD) from a 16-stage FlowVN teacher (S16) to an eight-stage student (S8). S8 copies alternating teacher stages and receives final-output, trajectory, and two-stage-update supervision; a matched supervised control isolates KD. Across two seeds, 16 held-out cases, and accelerations 10–50, full-KD reduced normalized root-mean-square error, relative velocity error, angular error, and VENC-scaled velocity RMSE by 1.91%, 2.90%, 1.31%, and 2.48%, respectively, and increased structural similarity by 0.0037. In a nested ablation, trajectory matching reduced nRMSE by 0.00250 and VENC-scaled velocity RMSE by 0.763 cm/s, whereas output-only KD was inconclusive for nRMSE and update matching increased nRMSE by 0.00127. On 32 Task 2 validation volumes, full-KD S8 reduced the displayed Complex Difference Error from 0.129 to 0.127 relative to supervised S8, but neither met the organizer FlowVN reference, identified as S10 from the released implementation; S16 obtained 0.085. S8 halved parameter count and fixed-shape forward time (2.62 versus 5.24 s). Separately, execution optimization reduced S8 complete-case wall time from 60.02 to 30.44 s and peak GPU memory from 13.72 to 3.44 GiB, with a saved-output relative L_2 difference of at most 7.60×10^{-7}. KD therefore provides modest quality recovery at fixed S8 cost, but this recovery is small relative to the observed S8–S16 quality gap.

Keywords: 4D flow MRI · knowledge distillation · fast reconstruction · memory-efficient inference · unrolled networks

1 Introduction

Four-dimensional flow magnetic resonance imaging combines three-dimensional spatial encoding with cardiac-resolved phase-contrast measurements to estimate time-varying blood-flow velocity. The resulting data support cardiovascular flow

quantification, but repeated encoding over space, cardiac phase, receiver coil, and velocity direction creates both a time-consuming acquisition and a high-dimensional reconstruction problem [3]. Undersampling reduces acquisition time only if the resulting inverse problem can also be solved within a practical latency and memory budget.

The Ultra-Fast 4D Flow MRI Reconstruction Challenge (CMRx4DFlow2026) emphasizes accurate, efficient reconstruction, while broader CMR benchmarks and models target robustness across modalities, views, sampling schemes, and unseen domains [6, 12, 1, 2]. FlowVN is a physics-guided variational network for accelerated 4D flow MRI [7, 11]. It repeats learned regularization and multi-coil data consistency across stages, so reducing stage count lowers repeated work and parameters but truncates the learned reconstruction trajectory.

Knowledge distillation transfers information from a larger teacher to a smaller student [8]. Prior MRI reconstruction work has studied output-level and student-friendly teacher formulations [9, 10]. FlowVN provides additional structure: an eight-stage student and a 16-stage teacher share the same stage design, allowing intermediate states to be aligned by normalized iteration progress. We exploit this correspondence to supervise the final complex reconstruction, matched intermediate states, and the net update produced by each pair of teacher stages.

A valid comparison must separate the benefit of teacher-derived supervision from teacher initialization. We therefore compare distilled S8 with a teacher-initialized supervised S8 that receives the same copied weights, architecture, data order, optimizer schedule, precision, seed, and update budget. The only difference is the scheduled KD objective. At inference, both students contain only S8 parameters and execute the same graph.

We contribute a stage-aligned FlowVN KD objective; a matched two-seed, case-level nested ablation with a VENC-scaled velocity endpoint; and controlled evidence separating S16-to-S8 compression, execution optimization, and organizer Task 2 validation against the FlowVN reference identified as S10 from the released implementation.

2 Method

2.1 FlowVN and Stage Compression

Let $x \in \mathbb{C}^N$ denote the complex image series over velocity encodings, cardiac phases, and spatial dimensions. For coil sensitivities C, Fourier operator F, sampling mask M_R at nominal acceleration R, and noise or unmodeled effects ε, the acquired multi-coil k-space is

$$y_R = \mathcal{A}_R x + \varepsilon, \qquad \mathcal{A}_R = M_R F C. \tag{1}$$

FlowVN starts from an adjoint reconstruction. Each stage combines a learned regularization update with a learned transformation of the acquisition residual through the Hermitian adjoint $\mathcal{A}_R^H$ (H denotes conjugate transpose) [11]. We

denote a K-stage model by SK. S16 is the frozen teacher and descriptive quality reference; S8 is the compressed student. If stages have comparable cost, S8 approximately halves repeated regularization and data-consistency work. S8 is deliberately half-depth: it permits one uniform student stage for each consecutive pair of S16 stages. An S10 student would require a nonuniform 16-to-10 correspondence and would test a different alignment method; S8 was fixed before organizer validation scores were observed.

2.2 Stage-Aligned Distillation

Let $x_s^{(i)}$ and $x_t^{(j)}$ be the complex reconstructions after student stage $i \in \{1, \dots, 8\}$ and teacher stage $j \in \{1, \dots, 16\}$. Both models receive the same adjoint initialization $x^{(0)}$, and student stage i is aligned with teacher stage $2i$. All parameters of S8 are initialized from S16 stages $2, 4, \dots, 16$. The matched supervised control receives exactly the same initialization.

During teacher-active training, S16 is frozen, placed in evaluation mode, and executed without gradient tracking. It is excluded from the optimizer and student checkpoints. For complex tensors a and b, the teacher losses use the target-normalized distance

$$d(a, b) = \frac{\mathrm{mean}(|a - b|)}{\mathrm{mean}(|b|) + 10^{-6}}. \tag{2}$$

The ground-truth loss and three teacher-derived terms are

$$\begin{aligned} \mathcal{L}_{\mathrm{gt}} &= \mathrm{mean}(|x_s^{(8)} - x|), \\ \mathcal{L}_{\mathrm{final}} &= d(x_s^{(8)}, x_t^{(16)}), \\ \mathcal{L}_{\mathrm{traj}} &= \frac{1}{8} \sum_{i=1}^{8} d(x_s^{(i)}, x_t^{(2i)}), \\ \mathcal{L}_{\mathrm{update}} &= \frac{1}{8} \sum_{i=1}^{8} d\Big(x_s^{(i)} - x_s^{(i-1)}, x_t^{(2i)} - x_t^{(2i-2)} \Big). \end{aligned} \tag{3}$$

Thus one student stage is asked to approximate the net correction of the corresponding two teacher stages, rather than either individual teacher update. The combined training objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{gt}} + s_e(0.25\mathcal{L}_{\mathrm{final}} + 0.25\mathcal{L}_{\mathrm{traj}} + 0.10\mathcal{L}_{\mathrm{update}}). \tag{4}$$

The teacher scale is $s_e = (1.0, 1.0, 0.8, 0.4, 0.2)$ for epochs 1–5. Epochs 6–10 use $s_e = 0$ and optimize only the labeled target. This schedule first constrains the compressed trajectory, then removes direct teacher influence so that S8 can refine against ground truth.

To attribute the contribution of the teacher losses, we use a nested four-arm ablation. In addition to supervised S8, the three KD arms use weights $(\lambda_{\mathrm{final}}, \lambda_{\mathrm{traj}}, \lambda_{\mathrm{update}})$ of $(0.25, 0, 0)$, $(0.25, 0.25, 0)$, and $(0.25, 0.25, 0.10)$, respectively. All arms retain the same ground-truth term, optimizer schedule, and

update budget. The three KD arms share the teacher-scale schedule and become ground-truth-only in epochs 6–10; supervised S8 is ground-truth-only throughout. The ordered contrasts therefore estimate final-output KD over supervision, trajectory matching over final-output KD, and update matching over final-plus-trajectory KD. These are conditional increments in the stated nesting order, not independent component main effects or interactions.

2.3 Inference Execution

Distillation adds no inference operation or parameter: both S8 models execute the same eight-stage graph. We measure stage-count savings with synchronized `model.forward` calls on preallocated tensors.

We also benchmark an optimized complete-case path using fixed S8 and S16 weights. It combines vectorized interpolation, selective compilation, overlapped loading, GPU adjoint preprocessing, and an algebraically equivalent replacement of three-dimensional transposed convolution by convolution with exchanged channels and flipped kernels. Because floating-point accumulation order can differ, we check numerical agreement on saved complex outputs.

3 Experimental Setup

3.1 Data and Matched Training

We use the 138 released aortic training cases [4]. Detailed metric analysis uses a fixed center-scoped split of 122 training and 16 internal validation cases with no exact-path or center-scoped patient-identifier overlap. Organizer references remain hidden, but the challenge evaluator returns aggregate Task 2 validation scores; we use these only as external technical validation, not for model selection. The archived organizer mask generator uses one deterministic seed at $R \in \{10, 20, 30, 40, 50\}$. Each case is evaluated at all five accelerations; each case–acceleration group combines four velocity encodings, yielding 80 groups per checkpoint. Stable pseudonymous identifiers are used throughout.

The joint-recipe comparison is full-KD S8 versus teacher-initialized supervised S8 for seeds 12345 and 23456. The nested ablation additionally evaluates final-output-only and final-plus-trajectory KD at both seeds. Within each seed, all four arms share initialization, architecture, data order, Adam settings, cosine learning-rate schedule, 32-bit floating-point (FP32) precision, and update budget. They use eight stages, 24 output features, $5 \times 5 \times 5$ learned kernels, batch size 1, an initial learning rate of 10^{-4}, and gradient clipping at norm 1.0. Training lasts ten epochs. The primary checkpoint is fixed at epoch index 9 after 137,380 updates and is not selected on the reporting cohort. Training jobs use separate 24-GB Multi-Instance GPU (MIG) profiles of an NVIDIA RTX PRO 6000 Blackwell Server Edition; full-cohort validation uses one 48-GB MIG profile of the same model.

3.2 Quality Evaluation and Statistics

Standalone validation uses the archived challenge mask implementation and FP32. An accepted checkpoint must contain 80 finite groups from 16 cases, zero incomplete groups, and verified source, configuration, checkpoint, and mask provenance. S16 is evaluated on the same cohort as a descriptive capacity reference.

We report normalized complex-image L_1, magnitude normalized root-mean-square error (nRMSE), structural similarity index measure (SSIM) [13], phase-derived relative velocity error (RelErr), and angular error (AngErr). RelErr and AngErr preserve the project phase-difference convention without velocity-encoding scaling and are not physical velocity errors in centimeters per second.

We report VENC-scaled velocity-vector RMSE (VENC RMSE) over flow-encoding directions $q \in \{1, 2, 3\}$. With scanner-provided VENC_q in cm/s, the adopted phase convention gives $v_q = (\mathrm{VENC}_q/\pi)\phi_q$. Using the principal phase $\mathrm{Arg}(z) \in (-\pi, \pi]$, circular phase correction gives

$$\begin{aligned} \widehat{\phi}_q &= \mathrm{Arg}\left(\widehat{x}_q \overline{\widehat{x}_0}\right), & \phi_q &= \mathrm{Arg}(x_q \overline{x_0}), \\ \delta v_q &= \frac{\mathrm{VENC}_q}{\pi} \mathrm{Arg}\left[\mathrm{e}^{i(\widehat{\phi}_q - \phi_q)}\right], & \mathrm{RMSE}_{\mathrm{VENC}} &= \left\langle \|\delta \boldsymbol{v}\|_2^2 \right\rangle_{\Omega}^{1/2}. \end{aligned} \tag{5}$$

Here $\delta \boldsymbol{v} = (\delta v_1, \delta v_2, \delta v_3)$, and $\langle \cdot \rangle_{\Omega}$ averages over the established ROI voxels and cardiac phases. VENC RMSE is therefore a wrapped, VENC-limited root-mean-square Euclidean velocity-vector error; it is not phase-unwrapped velocity or volumetric flow. Aggregate values appear in Supplementary Table S3, seed-specific values in Table 1, and acceleration-specific values in Supplementary Tables S5–S6.

The independent unit is the internal validation case. For each seed and ordered arm contrast, we compute paired effects within case and average the five accelerations. We then average the two seed-specific case effects and obtain a 10,000-sample percentile bootstrap 95% confidence interval by resampling the 16 cases. We report both seed-specific effects as well as their average and interval. The interval describes case heterogeneity, not a population over training seeds. Magnitude nRMSE and VENC RMSE are co-primary revision endpoints. Because the historical supervised and full-KD validations predated the physical metric, we re-evaluated their fixed epoch-index-9 checkpoints under the same protocol. All eight arm–seed validations passed the prespecified 80-group completeness and provenance audits; no physical values are inferred from RelErr or AngErr. Acceleration-specific analyses are secondary, descriptive, and unadjusted for multiplicity. S16 is not part of a matched KD contrast; it is a descriptive stage-capacity reference.

3.3 Organizer Task 2 External Validation

Before organizer scoring, we froze S16, supervised S8, and full-KD S8 (seed 12345) submissions for 32 Task 2 validation volumes. The released FlowVN entry

points and checkpoint identify the quality reference as S10 [5]. This is an external qualification comparison, not a matched estimate of removing two stages; the fixed-depth KD estimate remains full-KD S8 versus supervised S8.

3.4 Inference Benchmarks

The stage-count benchmark runs fresh-process S8 and S16 on a 48-GB NVIDIA RTX PRO 6000 MIG in FP32, with five warm-ups and 30 timed calls. Preallocated inputs have shape $1 \times 1 \times 22 \times 33 \times 119 \times 144$, 10 coils, and $R = 50$. CUDA events time only `model.forward`; compiled and eager outputs agree within absolute and relative tolerances of 10^{-4}.

The complete-case benchmark profiles one four-encoding $R = 10$ case in FP32 for full-KD S8 seed 12345 and S16 on an NVIDIA RTX 3090 24-GB GPU. S8 reports the median of three fresh-process eager/optimized pairs; S16 uses one accepted pair. Each optimized run uses a fresh compilation cache, and wall time spans initialization through saving. On a separate labeled S8 case at $R = 10$–50, the optimized path changes nRMSE, SSIM, RelErr, and AngErr by at most 4.6×10^{-6} in absolute value. The NVIDIA RTX A6000 specified for the official Task 2 ranking [6] was unavailable locally; the RTX 3090 measurements are therefore hardware-specific project benchmarks, not estimates of organizer-side runtime.

4 Results

4.1 Component Attribution

Table 1 reports four arms and six metrics by seed; Supplementary Table S4 gives ordered contrasts with confidence intervals.

Table 1. Seed-specific means (16 cases; $R = 10$–50). Rows add output, trajectory, and update matching; arrows indicate the favorable direction; VENC RMSE is in cm/s.

(a) Seed 12345

Arm	Norm. $L_1 \downarrow$	nRMSE $\downarrow$	SSIM $\uparrow$	RelErr $\downarrow$	AngErr (°) $\downarrow$	VENC RMSE $\downarrow$
Supervised	0.2101	0.0720	0.9002	0.7826	49.640	23.654
Output KD	0.2102	0.0719	0.9002	0.7825	49.653	23.659
+Trajectory	0.2014	0.0695	0.9056	0.7473	49.224	22.956
Full KD	0.2081	0.0709	0.9037	0.7532	49.190	23.102

(b) Seed 23456

Arm	Norm. $L_1 \downarrow$	nRMSE $\downarrow$	SSIM $\uparrow$	RelErr $\downarrow$	AngErr (°) $\downarrow$	VENC RMSE $\downarrow$
Supervised	0.2128	0.0723	0.8997	0.7837	49.694	23.699
Output KD	0.2120	0.0724	0.8994	0.7853	49.700	23.729
+Trajectory	0.2035	0.0699	0.9052	0.7441	49.218	22.906
Full KD	0.2104	0.0711	0.9036	0.7510	49.159	23.073

Final-output matching did not change nRMSE, with opposite seed directions. Its secondary effects were mixed: seed-averaged normalized L_1 decreased slightly despite opposing seed directions, SSIM worsened slightly, and RelErr and AngErr were inconclusive. Adding trajectory matching reduced nRMSE by 0.002497 (95% CI, 0.001829–0.003258) and VENC-scaled velocity-vector RMSE by 0.763 cm/s (95% CI, 0.485–1.061), with favorable effects in both seeds; all four secondary-metric intervals were also favorable. Final-output matching changed VENC RMSE by +0.017 cm/s (95% CI, −0.003–0.037) and was inconclusive. Adding update matching increased nRMSE by 0.001274 (95% CI, 0.000623–0.002019) and changed VENC RMSE by +0.156 cm/s (95% CI, −0.048–0.384); it also worsened normalized L_1 and SSIM. Thus trajectory alignment is supported under the evaluated order, weights, and schedule, whereas output-only KD is inconclusive and update matching is not supported. Effects were strongest at $R = 30$–50; Supplementary Tables S3–S6 report the two-seed aggregates, ordered contrasts, raw per-R values, and paired 95% CIs.

4.2 Joint-Recipe Quality Recovery at Fixed S8 Cost

Both seeds favored full-KD for all six aggregate metrics. After seed averaging, nRMSE, RelErr, and AngErr decreased by 1.91%, 2.90%, and 1.31%, respectively, and SSIM increased by 0.0037; their 95% CIs excluded zero. Normalized complex L_1 decreased by 1.69%, but its interval crossed zero. VENC RMSE decreased from 23.677 to 23.087 cm/s ($\Delta = -0.589$ cm/s; 95% CI, −0.964 to −0.211), a paired 2.48% reduction. Supplementary Table S2 reports the matched absolute and relative effects with case-bootstrap confidence intervals.

S16 remained substantially better on all five shared metrics; full-KD closed only 2.3–8.6% of the matched control-to-S16 gap. This supports modest recovery at fixed S8 cost, not S8–S16 parity.

Supplementary Table S7 provides the distinct joint full-KD-versus-supervised acceleration analysis and Supplementary Fig. S1 a control-selected $R = 40$ example. The full recipe remains the prespecified proposed method and external submission, but its gain should not be attributed to every teacher loss.

4.3 Organizer Task 2 Validation

The frozen organizer validation returned displayed Complex Difference Errors of 0.085, 0.129, and 0.127 for S16, supervised S8, and full-KD S8, respectively. Thus KD improved the fixed-S8 score by 0.002 (about 1.6%), but neither S8 model met the organizer FlowVN threshold on any of 32 volumes. Its released implementation is S10, whereas S16 met the threshold on all 32 volumes. This external result exposes the remaining compression gap but does not isolate S8-to-S10 stage reduction because the reference checkpoint and training are not matched. Supplementary Tables S8 and S9 report the overall and center-level scores, respectively.

4.4 Inference Efficiency

Table 2 separates the inference scopes. S8 versus S16 used 0.152 versus 0.304 million parameters and 2.618 versus 5.237 s fixed-shape forward time, reducing both by 50.0%. Both peaked at 6.07 GiB because sequential inference retains no stage list; S16 adds only 0.61 MB of FP32 parameters. Distilled and supervised S8 share the same inference graph and nominal cost.

Table 2. Controlled inference comparisons. Panel A isolates stage count using FP32 model-forward calls on a 48-GB RTX PRO 6000 MIG. Panel B isolates the execution path on one RTX 3090 complete case; S8 reports three paired runs and S16 one. The timing scopes are not interchangeable.

Configuration	Params (M) ↓	Time (s) ↓	Peak (GiB) ↓	Speed ↑	Rel. L_2 ↓
A. Fixed-shape model-forward benchmark					
S8 architecture	**0.152**	**2.618**	6.07	**2.00**×	–
S16 teacher	0.304	5.237	6.07	1.00×	–
B. Checkpoint-backed complete-case wall time					
S8 eager	0.152	60.018	13.72	1.00×	–
S8 optimized	0.152	**30.437**	**3.44**	**1.97**×	7.60×10^{-7}
S16 eager	0.304	105.341	13.72	1.00×	–
S16 optimized	0.304	**38.554**	**3.44**	**2.73**×	2.75×10^{-6}

At fixed full-KD S8 weights, the optimized path reduced median wall time from 60.018 s (range 59.888–60.712) to 30.437 s (30.101–30.925) and peak GPU memory from 13.72 to 3.44 GiB; saved-output relative L_2 was at most 7.60×10^{-7}. Under the same case, hardware, precision, and stack, S16 decreased from 105.341 to 38.554 s with the same memory reduction and a 2.75×10^{-6} relative L_2 difference. These 1.97× and 2.73× complete-case speedups must not be multiplied by the fixed-shape stage-count result because the shapes and timing scopes differ. Supplementary Table S1 provides the complete execution-path decomposition.

5 Discussion and Conclusion

Stage-aligned KD produced modest gains at unchanged S8 cost. The nested, nonfactorial ablation supports trajectory matching, leaves output-only KD inconclusive, and disfavors update matching because of its nRMSE penalty. The high-acceleration update penalty may reflect an over-constrained two-stage target, motivating targeted reweighting experiments.

S8 was chosen before organizer scoring because it gives exact half-depth compression and uniform alignment of each student stage with two successive S16 teacher stages. An S10 student would require irregular 16-to-10 alignment, confounding the intended stage-pair transfer. The released organizer S10 reference

is therefore an external quality threshold rather than a matched control: its depth, checkpoint, and training differ. Neither S8 model cleared that threshold; KD yielded a small fixed-S8 recovery, but the depth-associated quality gap was larger. Separately, S16-to-S8 compression halves parameters and fixed-shape forward time, whereas execution optimization reduces complete-case latency and memory at either depth.

Limitations are two seeds and 16 cases from one internal split; the bootstrap captures case, not seed, heterogeneity. Organizer validation is neither final-test nor clinical evidence, and its rounded summaries preclude paired inference. VENC RMSE is wrapped and VENC-limited, not phase-unwrapped velocity or hemodynamics. Efficiency covers one fixed shape and one RTX 3090 case, not the official RTX A6000. Overall, trajectory distillation recovers a measurable but incomplete part of the S8–S16 quality gap; update redesign and matched intermediate-depth studies remain future work.

Acknowledgments. The research reported in this paper was supported by the Canadian Institutes of Health Research (PJT-186038) and the Natural Sciences and Engineering Research Council of Canada (RGPIN-2023-04408 and RGPIN-2023-03410). M.C. is supported by the Canada Research Chairs program.

Disclosure of Interests. The authors have no competing interests to declare.

References

1. Anvari Hamedani, K., Razizadeh, N., Nabavi, S., Ebrahimi Moghaddam, M.: An all-in-one approach for accelerated cardiac MRI reconstruction. In: Statistical Atlases and Computational Models of the Heart. Workshop, CMRxRecon and MBAS Challenge Papers. Lecture Notes in Computer Science, vol. 15448, pp. 464–475. Springer, Cham (2025). https://doi.org/10.1007/978-3-031-87756-8_45
2. Anvari Hamedani, K., Razizadeh, N., Nabavi, S., Ebrahimi Moghaddam, M.: GENRE-CMR: Generalizable deep learning for diverse multi-domain cardiac MRI reconstruction. In: Statistical Atlases and Computational Models of the Heart. Regular and CMRxRecon Challenge Papers. Lecture Notes in Computer Science, vol. 16459, pp. 287–298. Springer, Cham (2026). https://doi.org/10.1007/978-3-032-17734-6_27
3. Bissell, M.M., Raimondi, F., Ait Ali, L., et al.: 4D flow cardiovascular magnetic resonance consensus statement: 2023 update. Journal of Cardiovascular Magnetic Resonance **25**(1), 40 (2023). https://doi.org/10.1186/s12968-023-00942-z
4. CMRx4DFlow2026 Organizing Committee: CMRx4DFlow2026: Official data specification. `https://cmrxrecon.github.io/2026/data.html` (2026), last accessed 2026/07/27
5. CMRx4DFlow2026 Organizing Committee: CMRx4DFlow2026: Official reconstruction and evaluation code. `https://github.com/CmrxRecon/CMRx4DFlow2026` (2026), released FlowVN entry point, batch wrapper, and checkpoint; last accessed 2026/08/15
6. CMRx4DFlow2026 Organizing Committee: CMRx4DFlow2026: Official task definitions and evaluation protocol. `https://cmrxrecon.github.io/2026/tasks.html` (2026), last accessed 2026/07/27

7. Hammernik, K., Klatzer, T., Kobler, E., Recht, M.P., Sodickson, D.K., Pock, T., Knoll, F.: Learning a variational network for reconstruction of accelerated MRI data. Magnetic Resonance in Medicine **79**(6), 3055–3071 (2018). https://doi.org/10.1002/mrm.26977
8. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network (2015). https://doi.org/10.48550/arXiv.1503.02531, arXiv preprint arXiv:1503.02531
9. Matcha, N., Ramanarayanan, S., Fahim, M.A., S, R.G., Ram, K., Sivaprakasam, M.: SFT-KD-Recon: Learning a student-friendly teacher for knowledge distillation in magnetic resonance image reconstruction. In: Medical Imaging with Deep Learning. Proceedings of Machine Learning Research, vol. 227, pp. 1423–1440. PMLR (2024), `https://proceedings.mlr.press/v227/matcha24a.html`
10. Murugesan, B., Vijayarangan, S., Sarveswaran, K., Ram, K., Sivaprakasam, M.: KD-MRI: A knowledge distillation framework for image reconstruction and image restoration in MRI workflow. In: Proceedings of the Third Conference on Medical Imaging with Deep Learning. Proceedings of Machine Learning Research, vol. 121, pp. 515–526. PMLR (2020), `https://proceedings.mlr.press/v121/murugesan20a.html`
11. Vishnevskiy, V., Walheim, J., Kozerke, S.: Deep variational network for rapid 4D flow MRI reconstruction. Nature Machine Intelligence **2**(4), 228–235 (2020). https://doi.org/10.1038/s42256-020-0165-6
12. Wang, F., Wang, Z., Li, Y., Lyu, J., Qin, C., Wang, S., et al.: Toward modality- and sampling-universal learning strategies for accelerating cardiovascular imaging: Summary of the CMRxRecon2024 challenge. IEEE Transactions on Medical Imaging **45**(5), 1872–1887 (2026). https://doi.org/10.1109/TMI.2025.3641610
13. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: From error visibility to structural similarity. IEEE Transactions on Image Processing **13**(4), 600–612 (2004). https://doi.org/10.1109/TIP.2003.819861