FOUNDATION MODEL ALIGNMENT REGULARIZES TRANSFORMER-BASED UNROLLED MRI RECONSTRUCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep unrolled reconstruction networks alternate data consistency steps with learned refinement networks and are typically supervised only at the final reconstruction. More expressive refinement networks have produced limited improvements over the convolutional E2E-VarNet on fastMRI, partly, we hypothesize, because their training converges to suboptimal solutions. Across single-coil knee and multi-coil brain reconstruction, we find that the bottlenecks of unregularized transformer cascades collapse early to a small fraction of their available rank and do not recover, whereas the convolutional model partially recovers during training.

We address this problem by aligning the refinement network's bottleneck of every cascade with dense representations extracted from the fully sampled ground truth by a frozen diffusion or self-supervised foundation model. A learned 3×3 convolution in each cascade maps the bottleneck to the corresponding foundation model features. The foundation model and the alignment convolutions are removed after training, leaving the deployed reconstruction network and its inference cost unchanged.

We evaluate this approach on single- and multi-coil datasets using multiple foundation models and convolutional and transformer-based reconstruction architectures. At longer training budgets, the aligned transformers surpass aligned E2E-VarNet, whereas their unregularized counterparts remain below unregularized E2E-VarNet. On knee data, the gains increase with acceleration and are largest for the transformer architectures. Alignment maintains high-rank bottlenecks across cascades and improves transformer training when the unregularized optimization stalls or deteriorates. These gains arise from foundation model alignment rather than increased rank alone: directly regularizing the bottleneck rank raises it beyond that of every aligned model without producing a lasting improvement in reconstruction. Where tested, the improvement over unregularized training is significant at $p < 0.001$ in one-sided paired Wilcoxon signed-rank tests over evaluation slices, with Holm correction across the seven reported tests.

1 INTRODUCTION

Unrolled networks are a leading supervised approach to accelerated MRI reconstruction. They map the iterations of a variational solver onto a sequence of T cascades, each alternating between a data consistency update using the measured k-space and a learned refinement network (Hammernik et al., 2018; Aggarwal et al., 2019; Sriram et al., 2020). Much of the subsequent progress has come from redesigning the network. On fastMRI knee data (Zbontar et al., 2018), such redesigns have yielded modest improvements over the convolutional E2E-VarNet, despite using several times as many parameters (Fabian et al., 2022; Xin et al., 2023). This suggests that increasing the capacity of the refinement network does not automatically translate into corresponding improvements in reconstruction quality. Although limited training data are often viewed as the main obstacle, recent studies reported diminishing returns from additional in-distribution examples and from physics-aware augmentation once large reconstruction datasets are available (Fabian et al., 2021; Klug & Heckel, 2023). We hypothesize that translating additional model capacity into improved reconstruction be-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

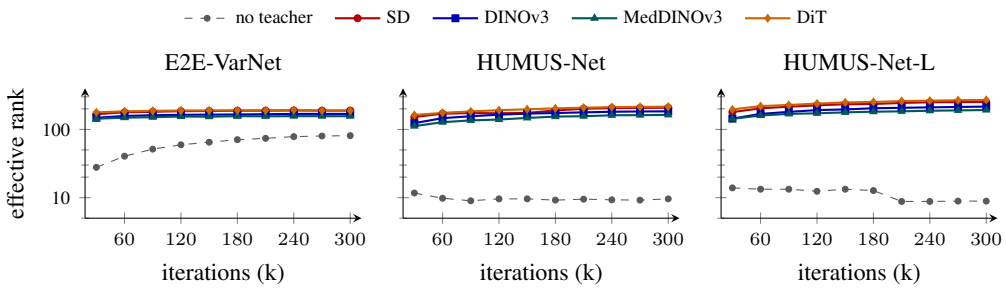

Figure 1: Effective rank of the cascade bottleneck versus training iteration for the configurations of Table 1, averaged over the eight cascades. Dashed lines denote unregularized runs. The transformer-based refinement networks rapidly collapse to a small fraction of their available dimensionality and remain low rank throughout training, whereas E2E-VarNet partially recovers. Alignment with a foundation model prevents this collapse and maintains substantially higher rank across training, particularly for the transformer-based refinement networks.

comes difficult as severe undersampling weakens the constraints supplied by data consistency and leaves more of the reconstruction to the learned refinement networks.

We argue that a part of the difficulty is that an unrolled network chain is deep. We study one convolutional and two transformer-based architectures, each with eight cascades whose refinement networks are U-Nets or Swin-U-Nets. They are trained end to end with a loss on the final image alone. In this setup, we observed that representations generated by the bottlenecks of the refinement networks collapse to a low effective rank (Figure 1), and the collapse is not confined to cascades farthest from the loss, i.e., earlier cascades in the chain (Section 4.2.3). The collapse is most severe in the transformer architectures, whose bottlenecks remain at the lowest effective rank throughout training, while the convolutional architecture partially recovers. This behavior is consistent with prior observations that representation rank can decrease with network depth (Feng et al., 2022) and that repeated self-attention can induce rank collapse (Dong et al., 2021). We therefore propose to give every cascade a local objective, following the broader principle of local supervision in deep networks (Belilovsky et al., 2019; Nøkland & Eidnes, 2019; Lee et al., 2015).

We construct this local objective from the dense representations of a frozen vision foundation model, which serves as a "teacher" to the refinement networks, to which we refer as the "student". Representation alignment has recently been shown to accelerate the training of generative models by matching their internal features, through a lightweight projection head, to spatial features from a frozen pretrained encoder (Yu et al., 2025). MRI reconstruction datasets are small relative to the corpora used to train modern vision encoders and generative models, and the representations of these models may therefore provide an external training signal that regularizes the features learned by the reconstruction network, particularly for expressive refinement networks and highly accelerated acquisitions. Recent works either place frozen vision encoders inside the reconstruction network at training and inference or apply foundation model guidance to its output (Hashmi et al., 2026; Feng et al., 2026). These approaches have shown benefits in cross-domain robustness or perceptual quality, but have not improved in-distribution reconstruction accuracy over E2E-VarNet. We instead use the foundation model (FM) only as a training time teacher.

The rank collapse is observed to be stronger in the transformer architectures, which have greater capacity. Therefore, we expect the value of local teacher alignment to be higher in higher capacity architectures, and grow with the capacity. We place the alignment where the collapse is measured, at the bottleneck of every cascade, a location that has also proven effective for representation alignment in generative U-Nets (Tian et al., 2025). At each cascade, a single, learned 3×3 convolution projects the student bottleneck toward the dense features that the teacher extracts from the fully sampled ground truth. This projection gives the alignment flexibility: allows the bottleneck to follow the structure of the teacher's representation without having to reproduce it directly. The bottleneck remains free to represent the information required for reconstruction. Unlike prior generative alignment methods, our aligning model is a supervised unrolled reconstruction network chain, the teacher

observes the fully sampled ground truth, and the alignment is applied at every cascade of the chain. The teacher and projection heads are discarded after training.

2 Related Work

Representation alignment. Yu et al. (2025) align the hidden states of a diffusion transformer, through a projection head, with the patch tokens of a frozen encoder and report faster training and better samples. Later works found the bottleneck of U-shaped backbones to be the best place for the loss (Tian et al., 2025) and that switching the loss off partway through training, where its gradient stops agreeing with the generative one, helps (Wang et al., 2025b). What the teacher contributes is the spatial structure of its features rather than its linear probe accuracy, so stronger teachers do not reliably help (Singh et al., 2026; Chefer et al., 2026). Objectives that do not use an external teacher recover part of the benefit of representation alignment. Examples include encouraging the model's own features to occupy a more diverse set of directions (Wang & He, 2025) or aligning them with a momentum-averaged copy of the network (Jiang et al., 2026). Their success suggests that some of the gains arise from regularizing the student's representations rather than from knowledge specific to a pretrained teacher. These works train generative backbones and measure sample quality. We study a supervised, training only loss inside an unrolled network chain and measure fidelity to the ground truth, by PSNR, the structural similarity index (SSIM) (Wang et al., 2004) and the normalized mean squared error (NMSE). Like Tian et al. (2025), we place the loss at the bottleneck of each refinement network, and adopt the spatial normalization of Singh et al. (2026) for the diffusion transformer teacher.

Rank collapse and local objectives. The effective rank (Roy & Vetterli, 2007) measures how many directions a representation uses, and a low value signals dimensional collapse, documented in self-supervised learning (Jing et al., 2022), where rank also predicts downstream performance (Garrido et al., 2023). The rank of deep representations diminishes with depth (Feng et al., 2022), and pure self-attention loses rank doubly exponentially with depth unless skip connections and feed-forward layers counteract it (Dong et al., 2021). Deep networks can also be trained with objectives local to each block, greedily layer by layer (Belilovsky et al., 2019), with local error signals (Nøkland & Eidnes, 2019) or with companion losses on hidden layers (Lee et al., 2015), and a layer-wise alignment to a guide network makes otherwise untrainable architectures trainable (Subramaniam et al., 2025). In unrolled networks for MRI reconstruction, FLAT supervises each cascade in pixel space along a path toward the ground truth, without a teacher or a feature space (Qi et al., 2025). We use the effective rank to diagnose the cascades of an unrolled network and a per-cascade alignment in feature space as the local objective.

Distillation as regularization. Regressing a student's intermediate features onto a teacher's through a learned head goes back to hint learning (Romero et al., 2015). Training with a teacher that sees information the student will not have at test time is learning using privileged information (Vapnik & Vashist, 2009; Lopez-Paz et al., 2016), used for super-resolution with a teacher trained on the task that sees the high-resolution image (Lee et al., 2020). Distillation also acts as a regularizer, so that even a poorly trained teacher can help (Yuan et al., 2020). A larger teacher need not make a better student, and on a hard task distillation can help early in training but hurt later (Cho & Hariharan, 2019). In MRI, distillation has so far compressed large reconstruction networks into smaller ones with a teacher trained on the same task (Murugesan et al., 2020). Our teachers see the ground truth as privileged information, were never trained on MRI reconstruction, and act as a regularizer on a network of unchanged size.

Unrolled architectures. Unrolled networks turn the iterations of a variational solver into a chain of cascades, each combining the measurement model with a learned refinement network (Hammernik et al., 2018; Aggarwal et al., 2019; Monga et al., 2021). E2E-VarNet uses a U-Net per cascade and learns the coil sensitivities end to end (Sriram et al., 2020). Later designs make the refinement network more expressive, with a hybrid Swin transformer in HUMUS-Net (Fabian et al., 2022) or with learned prompts (Xin et al., 2023). Others change what passes between cascades, through a feature-domain path (Giannakopoulos et al., 2024), a recurrent state (Yiasemis et al., 2022) or features aggregated across cascades (Xin et al., 2024), or add cascades (Wang et al., 2025a). Our

objective leaves the architecture unchanged and instead gives every cascade a local training signal of its own.

Foundation models in MRI reconstruction. Natural image pretraining helps a vision transformer more than a U-Net, mainly when MRI data are scarce (Lin & Heckel, 2022). Hashmi et al. (2026) run a frozen DINOv2 or CLIP encoder inside every cascade at inference and trail E2E-VarNet in-distribution on cardiac data. Feng et al. (2026) replace total variation regularization in a scan-specific, self-supervised reconstruction with a contrastive loss on vision-language embeddings of the output, which lowers PSNR in most of their settings. Chen et al. (2026) align a pooled DINO embedding to the latent of a three-dimensional autoencoder for contrast synthesis, without undersampling or unrolling, and adding the alignment to their encoder lowers autoencoding PSNR. Restoration methods use frozen pretrained networks on the restored output during training, as a loss (Tan et al., 2024; Lin et al., 2023) or as a semantic prior distilled back into the network (Zhang et al., 2024), or on the degraded input at inference (Lin et al., 2023). Unlike all of these, our teacher sees the fully sampled ground truth rather than the input or the output, constrains the dense feature grid inside every cascade rather than the final image or a pooled embedding, and is absent at inference.

3 METHOD

Given undersampled k-space measurements y, MRI reconstruction is commonly formulated as

$$\min_x \ \tfrac{1}{2}\|Ax - y\|_2^2 + \Psi(x), \tag{1}$$

where A is the undersampled Fourier encoding operator, including coil sensitivity maps in the multi-coil setting, and Ψ is a regularizer. Starting from the zero-filled image $\hat{x}_0 = A^*y$, an unrolled network approximates an iterative solver with T cascades,

$$\hat{x}_t = \hat{x}_{t-1} - \eta_t A^*(A\hat{x}_{t-1} - y) - R_t(\hat{x}_{t-1}), \qquad t = 1, \dots, T, \tag{2}$$

where η_t is a learned data consistency weight and the cascade-specific network R_t replaces the regularization gradient step. Each R_t is implemented as a U-Net in E2E-VarNet (Sriram et al., 2020) and as a hybrid convolutional and Swin-transformer U-Net in HUMUS-Net and HUMUS-Net-L (Fabian et al., 2022).

In the single-coil setting, the final reconstruction is the magnitude image $\hat{x} = |\hat{x}_T|$. In the multi-coil setting, $\hat{x}$ is obtained by root-sum-of-squares combination of the reconstructed coils after the final cascade. The training target x is the corresponding centered crop of the fully sampled reference image. With a standard end-to-end loss applied only to $\hat{x}$, the bottleneck representation h_t of cascade t receives supervision only indirectly through its decoder and the subsequent cascades.

The refinement networks should ideally contribute more to the reconstruction when they are more expressive, yet in a deep unrolled network trained through its output alone, their bottlenecks collapse to a low effective rank (Figure 1). We tackle this by regularizing refinement networks in every cascade with the prior of a frozen foundation model, and the rank collapse is avoided (Section 4.2.3).

We give every bottleneck h_t a local objective of its own by aligning it to the features of a frozen teacher. A frozen teacher encoder E processes the fully sampled ground truth and produces a dense spatial feature map $Z = E(x) \in \mathbb{R}^{H_E \times W_E \times D}$, where D is the teacher feature dimension. The teacher representation is computed once per training step and shared across all cascades. Each cascade has an independent projection head g_{ϕ_t}, implemented as a single 3×3 convolution, that maps its bottleneck features to the teacher feature dimension, $U_t = g_{\phi_t}(h_t)$.

The projected student features U_t and teacher features Z generally differ in spatial resolution and field of view. We map both onto a common grid defined over the fully sampled ground truth, preserving its aspect ratio and pairing features that correspond to the same anatomical regions. The input to the network itself is never cropped and it reconstructs the full field of view. Therefore, alignment uses only the part of each bottleneck that covers the cropped area corresponding to the ground truth. We denote the resulting student and teacher features at location n by $u_{t,n} \in \mathbb{R}^D$ and $z_n \in \mathbb{R}^D$, respectively. The exact coordinate mapping and dataset-specific preprocessing are described in Appendix A.2.

The alignment loss is the cosine distance between corresponding student and teacher features, averaged over spatial locations and cascades:

$$\mathcal{L}_{\text{align}} = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{N}\sum_{n=1}^{N}\left(1 - \frac{\langle u_{t,n}, z_n\rangle}{\max(\|u_{t,n}\|_2, \epsilon)\max(\|z_n\|_2, \epsilon)}\right), \tag{3}$$

where N is the number of locations on the common grid and $\epsilon > 0$ ensures numerical stability. The term of cascade t in this loss depends directly on h_t. Its gradient reaches that bottleneck directly and, through standard backpropagation, also updates the parameters of the preceding cascades that contribute to h_t.

The complete training objective is

$$\mathcal{L} = \mathcal{L}_{\text{recon}}(\hat{x}, x) + \lambda\mathcal{L}_{\text{align}}, \qquad \mathcal{L}_{\text{recon}}(\hat{x}, x) = 1 - \text{SSIM}(\hat{x}, x), \tag{4}$$

where SSIM is computed over the ground truth field of view and λ controls the strength of alignment.

Because the projection head maps local neighborhoods of the bottleneck into the teacher feature space while preserving their spatial arrangement, the bottleneck itself need not replicate the teacher representation, and the alignment acts as a soft, spatially structured constraint shaped jointly with the reconstruction loss. The teacher remains frozen, gradients from $\mathcal{L}_{\text{align}}$ update the projection heads and the reconstruction network, and after training, the teacher and heads are removed, leaving the only student, therefore the inference cost of the chain is unchanged.

4 Experiments

4.1 Setup

Data and sampling. Single-coil experiments use fastMRI knee data and multi-coil experiments fastMRI brain data (Zbontar et al., 2018), and every model is trained on the fastMRI training split. Unless a table states otherwise, sampling uses random Cartesian masks at $8\times$ with a fully sampled center of 4% of the lines (the autocalibration region). At $4\times$ and $16\times$ acceleration factors we keep 8% and 2% of all lines respectively.

Models and training. The models we experimented with are E2E-VarNet (Sriram et al., 2020) with 62.1M parameters, HUMUS-Net (Fabian et al., 2022) with 107.3M and HUMUS-Net-L with 226.6M. Each has $T = 8$ cascades and is trained with the SSIM loss, Adam (Kingma & Ba, 2015) and full single precision (Appendix A.1). The refinement network for the first one is convolutional while the latter two have transformer-based networks. Each refinement network downsamples its input image by 16 along each axis to reach its bottleneck, which has 512 channels in the convolutional architecture and 528 and 768 in the two transformer architectures. On knee data we run for 300k iterations with a tenfold learning rate drop at iteration 240k. On the brain data we run for 420k iterations.

Teachers. As teachers, we experimented with large-scale pre-trained diffusion as well as self-supervised encoders. The diffusion models were Stable Diffusion 1.4 (SD) (Rombach et al., 2022), trained on about two billion captioned web images (Schuhmann et al., 2022), and DiT-XL/2 (DiT) (Peebles & Xie, 2023), trained on the 1.3 million images of ImageNet, and the self-supervised encoders were DINOv3 (Siméoni et al., 2025), trained on 1.7 billion web images, and MedDINOv3 (Li et al., 2025) (DINOv3 further pretrained on 3.9 million computed tomography slices). Each teacher receives the ground truth resized to 512×512 pixels for the diffusion models and 640×640 for DINOv3 and MedDINOv3. The diffusion-based teachers produce their features in a single denoiser pass at $t = 0$, adapting the feature extraction of Tang et al. (2023); Appendix A.1 specifies the internal activations used as feature maps for the alignment. Student and teacher features are spatially matched so that paired locations correspond to the same region of the ground truth.

Alignment weight and evaluation. Each combination of student, teacher, anatomy and acceleration receives its own alignment weight λ. For both knee and brain experiments, λ is selected from a 30k-iteration sweep consisting of three candidate weights.

Table 1: Reconstruction performance of three student architectures with and without teacher alignment on single-coil knee MRI at $8\times$ acceleration. We report PSNR_{hi} (dB) at different training iterations, with the improvement over the corresponding same-seed no-teacher baseline shown as a subscript. Results are reported for two seeds (first / second). On the first seed, the HUMUS-Net-L control collapses between 180k and 300k iterations. Marks denote one-sided paired Wilcoxon signed-rank tests over the 1066 evaluation slices after 300k iterations: $*$ compares DiT with no teacher on the second seed, and † compares HUMUS-Net-L with E2E-VarNet, both with DiT, on their first seed. One, two and three marks denote $p < 0.05$, 0.01 and 0.001 after Holm correction over the seven tests of Tables 1 and 3.

Student	teacher	30k iter.	180k iter.	300k iter.
E2E-VarNet	none	28.212 / 28.335	29.772 / 29.853	30.204 / 30.259
	rank regularizer	$28.546_{+0.334}$ / $28.471_{+0.136}$	$29.876_{+0.104}$ / $29.729_{-0.124}$	$30.247_{+0.042}$ / $30.185_{-0.074}$
	deep supervision	$28.360_{+0.149}$ / $28.073_{-0.262}$	$29.741_{-0.030}$ / $29.722_{-0.131}$	$30.272_{+0.067}$ / $30.215_{-0.044}$
	SD	$29.073_{+0.862}$ / $29.088_{+0.753}$	$30.149_{+0.377}$ / $30.108_{+0.255}$	$30.477_{+0.273}$ / $30.531_{+0.272}$
	DINOv3	$28.867_{+0.656}$ / $28.925_{+0.590}$	$30.024_{+0.252}$ / $30.246_{+0.393}$	$30.458_{+0.254}$ / $30.555_{+0.296}$
	MedDINOv3	$28.950_{+0.739}$ / $28.882_{+0.547}$	$29.950_{+0.179}$ / $30.170_{+0.316}$	$30.420_{+0.216}$ / $30.504_{+0.245}$
	DiT	$29.227_{+1.015}$ / $29.205_{+0.870}$	$30.141_{+0.370}$ / $30.412_{+0.558}$	$30.528_{+0.323}$ / $30.544^{***}_{+0.285}$
HUMUS-Net	none	27.385 / 28.336	29.420 / 29.397	29.736 / 29.995
	SD	$29.054_{+1.669}$ / $29.136_{+0.800}$	$30.298_{+0.878}$ / $30.337_{+0.940}$	$30.587_{+0.851}$ / $30.719_{+0.723}$
	DINOv3	$28.841_{+1.456}$ / $29.032_{+0.696}$	$30.114_{+0.694}$ / $30.251_{+0.855}$	$30.579_{+0.843}$ / $30.678_{+0.683}$
	MedDINOv3	$28.710_{+1.326}$ / $28.938_{+0.602}$	$30.283_{+0.863}$ / $30.261_{+0.865}$	$30.623_{+0.887}$ / $30.562_{+0.566}$
	DiT	$29.199_{+1.815}$ / $29.225_{+0.888}$	$30.207_{+0.787}$ / $30.181_{+0.785}$	$30.668_{+0.932}$ / $30.668^{***}_{+0.673}$
HUMUS-Net-L	none	27.385 / 27.637	29.036 / 29.351	25.165 / 29.972
	SD	$29.088_{+1.703}$ / $29.301_{+1.665}$	$30.470_{+1.434}$ / $30.445_{+1.094}$	$30.690_{+5.525}$ / $30.755_{+0.782}$
	DINOv3	$29.158_{+1.773}$ / $28.965_{+1.328}$	$30.159_{+1.123}$ / $30.409_{+1.059}$	$30.733_{+5.568}$ / $30.729_{+0.757}$
	MedDINOv3	$29.020_{+1.635}$ / $29.037_{+1.401}$	$30.389_{+1.353}$ / $30.388_{+1.037}$	$30.680_{+5.515}$ / $30.659_{+0.687}$
	DiT	$29.327_{+1.943}$ / $29.314_{+1.677}$	$30.391_{+1.355}$ / $30.505_{+1.155}$	$30.816^{\dagger\dagger\dagger}_{+5.651}$ / $30.772^{***}_{+0.800}$

Because many fastMRI knee slices have a high noise floor, where metrics are dominated by noise rather than by the design choices under study, we evaluate on a high-SNR cohort of 1155 validation slices (Appendix A.3). Each weight sweep selects its weight on 40 of these slices, drawn at a fixed stride through the cohort in volume and slice order (Appendix A.4). After excluding the slices used for weight selection, the evaluation cohort contains 1066 slices. Because the fastMRI knee test split does not provide public ground truth, all knee reconstruction metrics in the tables are computed on this validation cohort. Appendix C.1 additionally reports the volume-wise metrics over all knee volumes in the validation split.

For brain reconstruction, alignment weights are selected in the same fashion in the validation split, while all reported metrics are computed on the fully sampled fastMRI brain test split. We report PSNR, SSIM and NMSE on 2382 slices from 150 volumes randomly drawn, with a fixed seed, in proportion to the five contrasts. Qualitative examples are provided in Appendix D.

Unless stated otherwise, every teacher-aligned model comparison is against the corresponding unregularized model, i.e., the control, and the same random seed. Tables 1 and 3 mark in total seven one-sided paired Wilcoxon signed-rank tests over the evaluation slices. We keep the number of tests small to avoid a large number of comparisons. All statistical tests reported in the paper are corrected jointly using the method of Holm (1979).

4.2 Results

4.2.1 Quantitative results on knee data

Table 1 crosses four teachers with three models, and Figure 2 follows the runs without a teacher and with SD over training.

Every teacher improved every model on which it was evaluated. The gain was largest on the transformer-based models, whose unregularized runs were also the weakest. After 300k iterations the gain was $+0.22$ to $+0.32$ dB on E2E-VarNet and $+0.57$ to $+0.93$ dB on the transformer-based models whose control stayed stable, and above $+5$ dB where it collapsed (Table 1). In the statistical significance tests of Table 1, DiT improved on the unregularized run at $p < 0.001$ at this training budget.

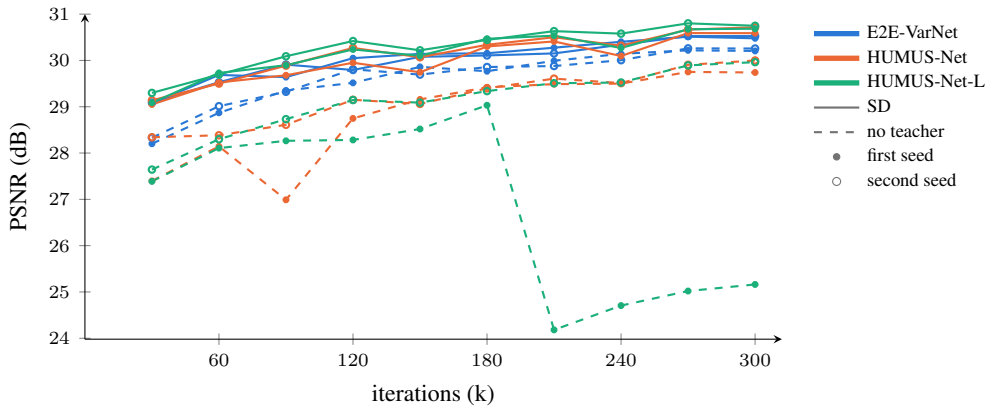

Figure 2: PSNR_{hi} on 554 slices (Appendix A.3) against training iterations for the model runs in Table 1, without a teacher (dashed) and aligned with SD (solid)

Table 2: PSNR_{hi} against acceleration, single-coil knee, gain over the same student's no-teacher run as a subscript.

Student	teacher	4×	8×	16×
E2E-VarNet, 30k iter.	none	32.478	28.212	23.209
	SD	$32.783_{+0.305}$	$29.073_{+0.862}$	$24.441_{+1.233}$
	DINOv3	$32.880_{+0.401}$	$28.867_{+0.656}$	$24.267_{+1.059}$
E2E-VarNet, 240k iter.	none	33.262	30.001	25.688
	SD	$33.352_{+0.090}$	$30.298_{+0.297}$	$26.376_{+0.688}$
	DINOv3	$33.344_{+0.081}$	$30.196_{+0.195}$	$26.259_{+0.571}$
HUMUS-Net, 30k iter.	none	32.169	27.385	21.638
	SD	$32.409_{+0.240}$	$29.054_{+1.669}$	$24.615_{+2.977}$
	DINOv3	$32.483_{+0.314}$	$28.841_{+1.456}$	$24.629_{+2.991}$
HUMUS-Net, 240k iter.	none	32.920	29.488	23.553
	SD	$33.230_{+0.310}$	$30.104_{+0.616}$	$26.510_{+2.956}$
	DINOv3	$33.167_{+0.246}$	$30.324_{+0.836}$	$26.779_{+3.226}$

Increasing rank alone does not reproduce these lasting gains. A teacher-free rank regularizer, formed by applying the redundancy reduction term of Barlow Twins (Zbontar et al., 2021) to the standardized channels of every E2E-VarNet bottleneck (more details in Appendix A.1), raised the mean effective rank (Section 4.2.3) to between 459 and 485 out of 512 channels, higher than for any aligned run. Its initial reconstruction gain, however, fell to between -0.07 and $+0.04$ dB after 300k iterations. High effective rank is therefore insufficient to explain the lasting reconstruction gains of alignment, indicating that the structure supplied by the teacher features also matters.

A local objective alone based on the fully sampled ground truth is not sufficient either to reproduce the gains from foundation model supervision. Deep supervision of every cascade output (Vishnevskiy et al., 2019), detailed in Appendix A.1, changed the PSNR by between -0.04 and $+0.07$ dB after 300k iterations, whereas every foundation model alignment gained at least $+0.22$ dB. Its bottleneck rank stayed close to that of the unregularized run and far below the aligned runs (Table 9).

Table 2 shows that the benefit of alignment increases monotonically with acceleration. Across the tested students and teachers, the improvement at 16× was $+2.96$ to $+3.23$ dB on HUMUS-Net and $+0.57$ to $+0.69$ dB on E2E-VarNet. The larger gains at higher acceleration suggest that alignment becomes more useful as the acquired measurements used in data consistency steps provide weaker constraints and the reconstruction relies more strongly on the learned refinement networks.

Table 3: Multi-coil brain at $8\times$, SSIM loss over the whole ground truth, all 2382 slices of 150 test volumes, PSNR gain over the no-teacher run of the same student and budget as a subscript. One seed per student. The 150k-iteration runs drop the learning rate tenfold at iteration 96k, and the 420k-iteration runs continue them at that reduced rate (Appendix C.2). The weights are listed in Table 6. Marks on the 420k-iteration DiT PSNR cells give one-sided paired Wilcoxon signed-rank tests over the 2382 slices against no teacher, Holm-corrected together with those of Table 1 and with the same notation; the corresponding test of HUMUS-Net against E2E-VarNet, both with DiT, is not significant.

		30k iter.			150k iter.			420k iter.		
Student	teacher	PSNR	SSIM	NMSE	PSNR	SSIM	NMSE	PSNR	SSIM	NMSE
HUMUS-Net	none	27.57	0.8044	0.0540	28.08	0.8163	0.0466	29.77	0.8494	0.0308
	SD	$31.46_{+3.88}$	0.8687	0.0207	$33.14_{+5.06}$	0.8853	0.0143	$33.31_{+3.54}$	0.8875	0.0137
	DINOv3	$31.10_{+3.52}$	0.8693	0.0223	$33.35_{+5.27}$	0.8875	0.0138	$33.57_{+3.80}$	0.8898	0.0131
	DiT	$31.10_{+3.53}$	0.8651	0.0235	$33.39_{+5.32}$	0.8878	0.0136	$33.62^{***}_{+3.85}$	0.8900	0.0129
E2E-VarNet	none	31.55	0.8718	0.0205	33.10	0.8858	0.0144	33.35	0.8880	0.0137
	SD	$31.92_{+0.37}$	0.8736	0.0188	$33.39_{+0.29}$	0.8882	0.0136	$33.58_{+0.23}$	0.8901	0.0130
	DINOv3	$31.78_{+0.23}$	0.8715	0.0205	$33.37_{+0.27}$	0.8884	0.0139	$33.65_{+0.30}$	0.8906	0.0130
	DiT	$31.72_{+0.17}$	0.8749	0.0197	$33.43_{+0.33}$	0.8887	0.0135	$33.64^{***}_{+0.29}$	0.8906	0.0129

4.2.2 Quantitative results on brain data

We further evaluated alignment on multi-coil brain data, where the student additionally estimates coil sensitivities and the ground truth images are larger and vary in size. Alignment improved the PSNR of E2E-VarNet with every teacher and at every reported training budget (Table 3). From 150k iterations onward, the gains ranged from $+0.23$ to $+0.33$ dB. Alignment also improved HUMUS-Net, whose unregularized control stayed more than 3.5 dB below the E2E-VarNet control at every budget, and it maintained substantially higher bottleneck rank than unregularized training.

Additional results across different training checkpoints and the corresponding rank measurements are shown in Appendix B.1 and Appendix B.1 respectively.

4.2.3 Regularization of the bottleneck rank

We characterize the dimensionality of each cascade bottleneck using its effective rank (Roy & Vetterli, 2007). For cascade t, we arrange the centered bottleneck features as $H_t \in \mathbb{R}^{M\times C}$, where the M rows correspond to bottleneck locations and the C columns to channels. Given its r nonzero singular values s_i and the normalized spectrum $p_i = s_i / \sum_j s_j$, the effective rank is

$$\operatorname{erank}(H_t) = \exp\Big(-\sum_{i=1}^{r} p_i \log p_i\Big). \tag{5}$$

A value close to one indicates that the spatial variation is concentrated in few feature directions, whereas a larger value indicates a more broadly distributed representation.

Figure 1 shows a clear difference between the effective rank of aligned and unregularized bottlenecks throughout training. Without alignment, the transformer bottlenecks remained strongly low rank across training, with most of their variation concentrated in a small number of directions. Furthermore, we observed that the convolutional E2E-VarNet partially recovered a higher-rank representation, but its bottleneck remained less distributed than those learned with alignment. In contrast, alignment maintained considerably higher effective rank throughout all measured cascades and across both convolutional and transformer students. These observations are illustrated in Figure 3 in the appendix. These results suggest that the local teacher objective discourages the representational collapse observed under supervision from the final reconstruction loss alone. Detailed training trajectories, cascade-wise measurements, leading singular direction fraction of variation, as well as extensions to multi-coil brain measurements are provided in Appendix B.1. We also show bottleneck visualizations for some representative slices in Appendix B.2.

5 DISCUSSION

When trained only through the final reconstruction loss, the cascade bottlenecks use a small fraction of their available dimensionality. This cannot be explained solely by the dimensionality of the training images, since alignment produces substantially higher-rank representations on the same data. Rank alone, however, does not account for the reconstruction gains. On E2E-VarNet, a teacher-free rank regularizer raises the bottleneck rank beyond that of the aligned models without producing a lasting improvement. Moreover, E2E-VarNet partially recovers its rank without a teacher but still benefits from alignment at our current training budgets. Together, these results indicate that high rank accompanies alignment but is not sufficient for its success. What distinguishes our FM alignment is the local supervision provided by the spatially dense teacher representation at every cascade bottleneck.

The low rank bottlenecks may also help explain why more expressive refinement networks have yielded limited improvements in unrolled reconstruction. Without local supervision, the additional capacity of the transformer students appears to remain underused. This effect is less pronounced for the convolutional E2E-VarNet, whose bottleneck partially recovers during training, but it persists in both transformer architectures. These observations are consistent with knowledge distillation acting as a regularizer (Yuan et al., 2020) and with representation alignment improving the training of architectures that otherwise optimize poorly (Subramaniam et al., 2025). They also agree with the larger benefit observed for the larger transformer model, similar to the model scaling behavior reported for representation alignment in generative models (Yu et al., 2025). Alignment may therefore help expressive refinement networks make more effective use of their capacity, rather than merely accelerating their convergence.

Teachers trained with different objectives and data sources provide broadly similar improvements (Table 1), suggesting that the benefit does not depend on knowledge specific to a single teacher or pretraining task. Representations learned by large models with different architectures and objectives can converge toward similar structures (Huh et al., 2024), and diffusion denoisers develop semantically organized features despite being trained without labels (Kadkhodaie et al., 2025). Alignment may therefore transfer properties shared across these representations, particularly their spatial organization, rather than task-specific semantics. The main difference between teachers appears early in training, when a diffusion teacher gives the largest gain for every student and dataset. One possible explanation is that denoising and MRI reconstruction both require recovering image structure from corrupted observations, making diffusion features especially useful during the early stages of optimization.

6 CONCLUSION

Unrolled reconstruction networks trained only through the final image can develop low-rank cascade bottlenecks, particularly when using transformer-based refinement networks. Aligning these bottlenecks with dense features from a frozen foundation model improves reconstruction across all evaluated architectures, with the largest gains at higher acceleration and for the transformer-based models,, without changing the deployed network or its inference cost. Future work will investigate alternative alignment objectives and locations, as well as more expressive reconstruction backbones.

REPRODUCIBILITY STATEMENT

Every experiment uses the public fastMRI knee and brain data (Zbontar et al., 2018). Appendix A.1 specifies the models, the teachers and the layers their features are taken from, the projection heads, the optimizer, the arithmetic precision, the training schedules, and Appendix A.2 the preprocessing and the spatial correspondence between student and teacher. The alignment weight of every cell and how it was selected are given in Appendix A.4 and Table 6, the reporting cohort and scoring in Appendix A.3.

AI USE STATEMENT

In this work, we used generative AI tools for research execution, namely to write and debug the code for training, evaluation and analysis and the scripts that ran the experiments on our compute clusters.

Additionally, we used generative AI tools to aid and polish the writing. We have reviewed all AI-assisted work. Changes to the training code were tested before use, including checks that short runs reproduce the losses of the unchanged code, and the reported numbers were checked against the logs of the jobs that produced them. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Hemant Kumar Aggarwal, Merry P. Mani, and Mathews Jacob. MoDL: Model-based deep learning architecture for inverse problems. *IEEE Transactions on Medical Imaging*, 38(2):394–405, 2019.

Eugene Belilovsky, Michael Eickenberg, and Edouard Oyallon. Greedy layerwise learning can scale to ImageNet. In *International Conference on Machine Learning*, 2019.

Hila Chefer, Patrick Esser, Dominik Lorenz, Dustin Podell, Vikash Raja, Vinh Tong, Antonio Torralba, and Robin Rombach. Self-supervised flow matching for scalable multi-modal synthesis. In *International Conference on Machine Learning (ICML)*, 2026.

Yonghao Chen, Sicheng Yang, Rui Tang, and Lei Zhu. Recover semantics first, generate better: Improved latent modeling for 3D MRI reconstruction and cross-contrast synthesis. In *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, Lecture Notes in Computer Science, pp. 418–428. Springer, 2026.

Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4793–4801. IEEE, 2019.

Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. Attention is not all you need: Pure attention loses rank doubly exponentially with depth. In *International Conference on Machine Learning*, 2021.

Zalan Fabian, Reinhard Heckel, and Mahdi Soltanolkotabi. Data augmentation for deep learning based accelerated MRI reconstruction with limited data. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 3057–3067. PMLR, 2021.

Zalan Fabian, Berk Tinaz, and Mahdi Soltanolkotabi. HUMUS-Net: Hybrid unrolled multi-scale network architecture for accelerated MRI reconstruction. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 25306–25319, 2022.

Ruili Feng, Kecheng Zheng, Yukun Huang, Deli Zhao, Michael Jordan, and Zheng-Jun Zha. Rank diminishing in deep neural networks. In *Advances in Neural Information Processing Systems*, 2022.

Ruimin Feng, Xingxin He, Ronald Mercer, Zachary Stewart, and Fang Liu. On the utility of foundation models for fast MRI: Vision-language-guided image reconstruction. *Magnetic Resonance in Medicine*, 96(2):943–959, 2026.

Chaofan Gan, Yuanpeng Tu, Xi Chen, Tieyuan Chen, Yuxi Li, Mehrtash Harandi, and Weiyao Lin. Unleashing diffusion transformers for visual correspondence by modulating massive activations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Quentin Garrido, Randall Balestriero, Laurent Najman, and Yann LeCun. RankMe: Assessing the downstream performance of pretrained self-supervised representations by their rank. In *International Conference on Machine Learning*, 2023.

Ilias I. Giannakopoulos, Matthew J. Muckley, Jesi Kim, Matthew Breen, Patricia M. Johnson, Yvonne W. Lui, and Riccardo Lattanzi. Accelerated MRI reconstructions via variational network and feature domain learning. *Scientific Reports*, 14(1):10991, 2024.

Kerstin Hammernik, Teresa Klatzer, Erich Kobler, Michael P Recht, Daniel K Sodickson, Thomas Pock, and Florian Knoll. Learning a variational network for reconstruction of accelerated MRI data. *Magnetic Resonance in Medicine*, 79(6):3055–3071, 2018.

Anam Hashmi, Mayug Maniparambil, Julia Dietlmeier, Kathleen M. Curran, and Noel E. O'Connor. Are natural-domain foundation models effective for accelerated cardiac MRI reconstruction? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 6202–6211, 2026.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. Position: The Platonic representation hypothesis. In *International Conference on Machine Learning (ICML)*, 2024.

Dengyang Jiang, Mengmeng Wang, Liuzhuozheng Li, Lei Zhang, Haoyu Wang, Wei Wei, Guang Dai, Yanning Zhang, and Jingdong Wang. Representation alignment for diffusion transformers without external components. In *International Conference on Learning Representations (ICLR)*, 2026.

Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. In *International Conference on Learning Representations*, 2022.

Zahra Kadkhodaie, Stéphane Mallat, and Eero P. Simoncelli. Unconditional CNN denoisers contain sparse semantic representation of images. *arXiv preprint arXiv:2506.01912*, 2025.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Tobit Klug and Reinhard Heckel. Scaling laws for deep learning based image reconstruction. In *International Conference on Learning Representations*, 2023.

Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu. Deeply-supervised nets. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 38 of *Proceedings of Machine Learning Research*, pp. 562–570, 2015.

Wonkyung Lee, Junghyup Lee, Dohyung Kim, and Bumsub Ham. Learning with privileged information for efficient image super-resolution. In *Computer Vision – ECCV 2020*, Lecture Notes in Computer Science, pp. 465–482. Springer, 2020.

Yuheng Li, Yizhou Wu, Yuxiang Lai, Mingzhe Hu, and Xiaofeng Yang. MedDINOv3: How to adapt vision foundation models for medical image segmentation? *arXiv preprint arXiv:2509.02379*, 2025.

Kang Lin and Reinhard Heckel. Vision transformers enable fast and robust accelerated MRI. In *Proceedings of The 5th International Conference on Medical Imaging with Deep Learning*, volume 172 of *Proceedings of Machine Learning Research*, pp. 774–795. PMLR, 2022.

Xin Lin, Jingtong Yue, Kelvin C. K. Chan, Lu Qi, Chao Ren, Jinshan Pan, and Ming-Hsuan Yang. Multi-task image restoration guided by robust DINO features. *arXiv preprint arXiv:2312.01677*, 2023.

David Lopez-Paz, Léon Bottou, Bernhard Schölkopf, and Vladimir Vapnik. Unifying distillation and privileged information. In *International Conference on Learning Representations (ICLR)*, 2016.

Vishal Monga, Yuelong Li, and Yonina C Eldar. Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. *IEEE Signal Processing Magazine*, 38(2):18–44, 2021.

Balamurali Murugesan, Sricharan Vijayarangan, Kaushik Sarveswaran, Keerthi Ram, and Mohanasankar Sivaprakasam. 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*, volume 121 of *Proceedings of Machine Learning Research*, pp. 515–526. PMLR, 2020.

Arild Nøkland and Lars Hiller Eidnes. Training neural networks with local error signals. In *International Conference on Machine Learning*, 2019.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Kehan Qi, Saumya Gupta, Xiaoling Hu, Qingqiao Hu, Weimin Lyu, Yicun Wang, and Chao Chen. FLAT: FLow-aligned training of unrolled networks for MRI reconstruction. *arXiv preprint arXiv:2512.03020*, 2025. v3, August 2026; v1 titled "Unrolled Networks are Conditional Probability Flows in MRI Reconstruction".

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. FitNets: Hints for thin deep nets. In *International Conference on Learning Representations (ICLR)*, 2015.

Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *European Signal Processing Conference (EUSIPCO)*, 2007.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5B: An open large-scale dataset for training next generation image-text models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 25278–25294, 2022.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, et al. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025.

Jaskirat Singh, Xingjian Leng, Zongze Wu, Liang Zheng, Richard Zhang, Eli Shechtman, and Saining Xie. What matters for representation alignment: Global information or spatial structure? In *International Conference on Learning Representations (ICLR)*, 2026.

Anuroop Sriram, Jure Zbontar, Tullie Murrell, Aaron Defazio, C Lawrence Zitnick, Nafissa Yakubova, Florian Knoll, and Patricia Johnson. End-to-end variational networks for accelerated MRI reconstruction. In *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pp. 64–73, 2020.

Vighnesh Subramaniam, David Mayo, Colin Conwell, Tomaso Poggio, Boris Katz, Brian Cheung, and Andrei Barbu. Training the untrainable: Introducing inductive bias via representational alignment. In *Advances in Neural Information Processing Systems*, volume 38, pp. 107305–107343, 2025.

Mingjie Sun, Xinlei Chen, J. Zico Kolter, and Zhuang Liu. Massive activations in large language models. In *Conference on Language Modeling (COLM)*, 2024.

Jiangtong Tan, Hu Yu, Jie Huang, Zizheng Yang, and Feng Zhao. DiffLoss: Unleashing diffusion model as constraint for training image restoration network. In *Computer Vision – ACCV 2024*, Lecture Notes in Computer Science, pp. 105–123. Springer Nature Singapore, 2024.

Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent correspondence from image diffusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Yuchuan Tian, Hanting Chen, Mengyu Zheng, Yuchen Liang, Chao Xu, and Yunhe Wang. U-REPA: Aligning diffusion U-Nets to ViTs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Vladimir Vapnik and Akshay Vashist. A new learning paradigm: Learning using privileged information. *Neural Networks*, 22(5-6):544–557, 2009.

Valery Vishnevskiy, Richard Rau, and Orcun Goksel. Deep variational networks with exponential weighting for learning computed tomography. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 310–318. Springer, 2019.

Puyang Wang, Pengfei Guo, Keyi Chai, Jinyuan Zhou, Daguang Xu, and Shanshan Jiang. SDUM: A scalable deep unrolled model for universal MRI reconstruction. *arXiv preprint arXiv:2512.17137*, 2025a.

Runqian Wang and Kaiming He. Diffuse and disperse: Image generation with representation regularization. *arXiv preprint arXiv:2506.09027*, 2025.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004.

Ziqiao Wang, Wangbo Zhao, Yuhao Zhou, Zekai Li, Zhiyuan Liang, Mingjia Shi, Xuanlei Zhao, Pengfei Zhou, Kaipeng Zhang, Zhangyang Wang, Kai Wang, and Yang You. REPA works until it doesn't: Early-stopped, holistic alignment supercharges diffusion training. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025b.

Bingyu Xin, Meng Ye, Leon Axel, and Dimitris N Metaxas. Fill the k-space and refine the image: Prompting for dynamic and multi-contrast MRI reconstruction. In *Statistical Atlases and Computational Models of the Heart (STACOM)*, Lecture Notes in Computer Science, pp. 261–273. Springer, 2023.

Bingyu Xin, Meng Ye, Leon Axel, and Dimitris N Metaxas. Rethinking deep unrolled model for accelerated MRI reconstruction. In *European Conference on Computer Vision*, pp. 164–181. Springer, 2024.

George Yiasemis, Jan-Jakob Sonke, Clarisa Sánchez, and Jonas Teuwen. Recurrent variational network: A deep learning inverse problem solver applied to the task of accelerated MRI reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 732–741, June 2022.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *International Conference on Learning Representations (ICLR)*, 2025.

Li Yuan, Francis EH Tay, Guilin Li, Tao Wang, and Jiashi Feng. Revisiting knowledge distillation via label smoothing regularization. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3902–3910. IEEE, 2020.

Junno Yun, Yaşar Utku Alçalar, and Mehmet Akçakaya. UDT: Reconciling U-Nets and diffusion transformers with data-adaptive token reduction. *arXiv preprint arXiv:2608.01298*, 2026.

Jure Zbontar, Florian Knoll, Anuroop Sriram, Tullie Murrell, Zhengnan Huang, Matthew J Muckley, Aaron Defazio, Ruben Stern, Patricia Johnson, Mary Bruno, et al. fastMRI: An open dataset and benchmarks for accelerated MRI. *arXiv preprint arXiv:1811.08839*, 2018.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow Twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning (ICML)*, 2021.

Quan Zhang, Xiaoyu Liu, Wei Li, Hanting Chen, Junchao Liu, Jie Hu, Zhiwei Xiong, Chun Yuan, and Yunhe Wang. Distilling semantic priors from SAM to efficient image restoration models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 25409–25419, 2024.

A Implementation and protocol

This part describes the students, the teachers and their training, how student and teacher locations are paired, which slices are scored and how each alignment weight is selected.

A.1 STUDENTS, TEACHERS, AND TRAINING

Table 4: The three students. Parameters are those of the deployed network, without the projection heads.

Student	refinement network	parameters	bottleneck	learning rate
E2E-VarNet	U-Net, 32 channels	62.1M	512 ch., stride 16	3×10^{-4}
HUMUS-Net	Swin, embedding 66	107.3M	528 ch., stride 16	1×10^{-4}
HUMUS-Net-L	Swin, embedding 96	226.6M	768 ch., stride 16	7×10^{-5}

Students. Our E2E-VarNet (Sriram et al., 2020) has 8 cascades, each with a U-Net refinement network of 32 base channels and four pooling stages. Its bottleneck is therefore $32 \cdot 2^4 = 512$ channels wide and covers the knee ground truth with 20×20 cells. HUMUS-Net (Fabian et al., 2022) retains its original block configuration: convolutional downsampling followed by three stages of two Swin layers with 3, 6, and 12 heads and window size 8, and a bottleneck of two layers with 24 heads. The default model has embedding dimension 66, while HUMUS-Net-L has embedding dimension 96; this is the only architectural difference between them. The two Swin transformers operate at a fixed size of 640×384 on knee data and 384×512 on brain data, with each image centered by reflection padding. In the multi-coil setting, a small U-Net estimates coil sensitivities from the calibration region before the cascades and is trained end to end, with 16 base channels for HUMUS-Net and 8 for E2E-VarNet.

Teachers. Every teacher is frozen and evaluated without gradients, so it adds one forward pass per training step and no parameters to the optimizer. The ground truth magnitude is normalized per image to $[0, 1]$ by clipping at its 2nd and 98th percentiles, replicated to three channels, padded to a square with zeros, and resized to the input size of choice. The ImageNet channel statistics are additionally applied for the self-supervised encoders.

For DINOv3 (Siméoni et al., 2025) and MedDINOv3 (Li et al., 2025), which is further pretrained on computed tomography slices, we use the ViT-B/16 backbone and concatenate the patch tokens of its last four blocks, so $D = 4 \times 768 = 3072$. The DINOv3 family includes one class token and four register tokens, which are discarded before the spatial grid is formed. The diffusion transformer has neither class nor register tokens.

SD (Rombach et al., 2022) is used in the dense feature manner of Tang et al. (2023): the prepared image is encoded into the latent space and one denoiser pass is run at $t = 0$ with a cached null-prompt embedding, without a sampling loop. We use the 1280-channel activation of decoder up-block 1, which forms a 32×32 feature grid at an input size of 512×512 pixels. DiT (Peebles & Xie, 2023) is used analogously, conditioned on the null class label it reserves for unconditional generation, with features extracted from transformer block 8 and spatially standardized as described below. In Tables 1 and 3, DINOv3 and MedDINOv3 use 640 pixel inputs, producing 40×40 feature grids, while SD and DiT use 512×512 pixel inputs, producing 32×32 grids. In Table 2, E2E-VarNet uses DINOv3 with 320×320 pixel inputs.

Projection heads. Each cascade has an independent 3×3 convolution with padding 1 that maps the bottleneck width to the teacher dimension D. For the 512 channel convolutional bottleneck and the 1280 dimensional SD features, each head has approximately 5.9M parameters. The heads are optimized with the backbone optimizer and learning rate. They are used only to compute the alignment loss and do not lie on the reconstruction forward path, so they are discarded with the teacher after training and add no capacity to the deployed network. A head with no learnable capacity would constrain the native bottleneck directly to the teacher space, whereas an excessively expressive head could satisfy the alignment objective while leaving the bottleneck weakly constrained. A single 3×3 convolution per cascade provides an intermediate choice that accommodates differences between the two feature spaces while preserving their spatial organization.

Spatial standardization of DiT features. The alignment loss is a per-location cosine, but diffusion transformer features contain massive activations (Gan et al., 2025), a phenomenon first described in language models and vision transformers (Sun et al., 2024). At block 8, the five largest of

DiT's 1152 channels account for 19.9% of the total feature magnitude, whereas the corresponding share is only 1.9% in the diffusion U-Net. The largest DiT channel is 420× the median channel magnitude, compared with 8× in the diffusion U-Net. Moreover, the five dominant DiT channels vary little across spatial locations, with pairwise spatial correlations between 0.79 and 0.97. After unit normalization, they account for 93.8% of the energy, compared with 5.8% for the diffusion U-Net, so the cosine is dominated by a direction that varies little over space. Consequently, over the first 18k iterations, the mean per-location cosine distance to the raw DiT target is only 0.0209, compared with 0.2170 for the diffusion U-Net. The student therefore appears nearly aligned with DiT from the start, even though the cosine is dominated by an almost spatially constant direction. A much larger nominal alignment weight is then needed for the loss to provide a meaningful training signal.

We therefore standardize the representation over the spatial axes independently for each channel. Following Singh et al. (2026), who subtract a fraction $\gamma \in [0.6, 0.8]$ of the spatial mean before dividing by the spatial standard deviation, we subtract the mean in full and set $\gamma = 1$. This removes a spatially constant channel exactly and raises the mean cosine distance by a factor of 24, from 0.0209 to 0.5041. Alternative treatments remove or normalize the dominant channels directly (Gan et al., 2025). We instead use spatial standardization because our objective compares features at corresponding locations. We extract DiT features from block 8 of 28, which lies before the late blocks, as early to mid layers provide better representation quality in diffusion transformers (Yu et al., 2025; Yun et al., 2026).

Every DiT run in the paper applies this standardization to the teacher features.

Training and computational cost. Every student is trained with the SSIM loss and Adam at batch size one on single slice inputs, using the learning rate in Table 4. Training uses full single precision, with the reduced-precision matrix mode enabled by default on current accelerators switched off, because matched tests changed the reported values by up to 0.5 dB. One iteration denotes one single slice optimization step. For Table 1, the 300k-iteration schedule reduces the learning rate tenfold at iteration 240k. The brain schedule is specified in Appendix C.2.

The teacher adds one forward pass per training step and the heads add one convolution per cascade, all during training only. In full precision, 6000 iterations take approximately 2520 versus 1415 s on E2E-VarNet, 11400 versus 10200 s on HUMUS-Net, and 9050 versus 7900 s on HUMUS-Net-L, with and without SD at an input size of 512 pixels (Table 5). On an RTX A6000, a 180k-iteration E2E-VarNet run takes approximately 21 device hours with SD at 512 pixels and 12 device hours without a teacher.

Deep supervision. The deep supervision row of Table 1 uses no teacher. It adds to the reconstruction loss the weighted SSIM loss of every cascade output, $\sum_{k=1}^{8} w_k \ell_k$, with $w_k \propto \exp(-\tau(8-k))$ normalized to sum to one and τ growing from 0 by $1/3$ every 6000 iterations. The eight outputs are weighted equally at initialization, and from 30k iterations on the final output carries at least 81% of the weight.

Rank regularizer. The rank regularizer row of Table 1 also uses no teacher. For every cascade and slice, the bottleneck is read as a matrix of N locations by $C = 512$ channels, centered over its locations, and each channel is divided by its standard deviation over the locations. With R the resulting $C \times C$ channel correlation matrix, the regularizer is $\frac{1}{C}\sum_{i \neq j} R_{ij}^2$, averaged over the eight cascades, which is the redundancy reduction term of Barlow Twins (Zbontar et al., 2021) applied to a single view. Driving R toward the identity decorrelates the channels and flattens the spectrum of the standardized bottleneck, which raises its effective rank directly. It is added to the reconstruction loss with weight 0.008 (Table 6) and has no parameters.

Initialization and reproducibility. A seeded run fixes the student initialization, data order, and data loader generator. The two knee seeds are 2 and 3 for E2E-VarNet and 0 and 1 for HUMUS-Net and HUMUS-Net-L; the brain runs use seed 0 unless stated otherwise.

Table 5: Training time per 6000 iterations on one RTX A6000 in full precision for the runs of Table 1, as the range of the per-run medians. HUMUS-Net-L is the faster of the two transformers although it has twice the parameters: the two differ only in embedding width, 96 against 66, which gives head dimensions of 32 against 22 and stage widths of 96 to 768 against 66 to 528, and only the wider model's dimensions are multiples of eight.

Student	teacher	seconds per 6000 iterations
E2E-VarNet	none	1410 to 1420
	rank regularizer	1480 to 1500
	deep supervision	1500 to 1520
	SD	2510 to 2520
	DINOv3	2170 to 2190
	MedDINOv3	2180 to 2340
	DiT	2510 to 2530
HUMUS-Net	none	10160 to 10310
	SD	11370 to 11490
	DINOv3	11120 to 11260
	MedDINOv3	11050 to 11190
	DiT	11380 to 11650
HUMUS-Net-L	none	7840 to 8280
	SD	8990 to 9110
	DINOv3	8870 to 9000
	MedDINOv3	8750 to 9140
	DiT	9070 to 9240

A.2 Spatial correspondence and preprocessing

Eq. 3 compares student and teacher features at corresponding spatial locations, so each pair must represent the same region of the ground truth. Establishing this correspondence is non-trivial for three reasons. First, the ground truth is a centered crop of size $h \times w$ from the full $H \times W$ image, whereas the student operates on a larger working image. Second, the convolutional and transformer students pad or crop that image differently before downsampling. Third, the ground truth is not always square, and its dimensions need not be integer multiples of the bottleneck stride. We therefore identify the ground truth region in both feature maps and resample the two regions onto a common spatial grid. The reconstruction network continues to process its full working image and the ground truth crop is used only to define the region included in the alignment objective.

The ground truth in the student bottleneck. Three frames are involved, and each step below maps one into the next. First, the fully sampled image has size $H \times W$, for example 640×368 on knee data, and fastMRI defines the ground truth as its centered $h \times w$ crop, here 320×320. The ground truth therefore starts at row and column

$$o_y = \left\lfloor \frac{H-h}{2} \right\rfloor, \qquad o_x = \left\lfloor \frac{W-w}{2} \right\rfloor$$

of the full image. Second, the student reconstructs the full image, but its refinement network first places it on a working canvas of size $H' \times W'$ whose sides are multiples of $s = 16$. The convolutional student pads each axis to the next multiple of 16. The transformer student has a fixed canvas and pads each axis that is shorter than the canvas and crops each axis that is longer. Each axis is thus either padded or cropped, never both; the image stays centered on the canvas, and when the difference is odd the extra pixel goes after the image. The first row of the image then lies at row

$$a_y = \left\lfloor \frac{H'-H}{2} \right\rfloor \text{ if the axis is padded}, \qquad a_y = -\left\lfloor \frac{H-H'}{2} \right\rfloor \text{ if it is cropped}$$

of the canvas, and a_x is defined in the same way. Third, the bottleneck divides the canvas by s, so the ground truth occupies the student box

$$B_{\mathrm{s}} = \left[\frac{a_y + o_y}{s}, \frac{a_y + o_y + h}{s} \right] \times \left[\frac{a_x + o_x}{s}, \frac{a_x + o_x + w}{s} \right] \tag{6}$$

in bottleneck cells. In the knee example the convolutional student needs no padding, since 640 and 368 are multiples of 16, and the ground truth spans rows 10 to 30 and columns 1.5 to 21.5 of its bottleneck, so its left and right edges fall in the middle of cells. The transformer pads the width to 384, which gives $a_x = 8$ and moves the ground truth to columns 2 to 22.

The ground truth in the teacher grid. The teacher input is always square. An $h \times w$ ground truth is first padded with background, symmetrically along its shorter side, to an $S \times S$ square with $S = \max(h, w)$, and this square is then resized to the teacher input size, 640×640 for DINOv3 and MedDINOv3 and 512×512 for SD and DiT, so the anatomy is never stretched. The teacher grid has one location per 16 input pixels, 40×40 at 640 pixels and 32×32 at 512 (Appendix A.1), and 20×20 for DINOv3 at 320 pixels in Table 2. A square ground truth, such as every 320×320 knee ground truth, fills the whole grid. A non-square one occupies the centered teacher box

$$B_{\mathrm{t}} = \left[\frac{\lfloor (S-h)/2 \rfloor}{S}, \frac{\lfloor (S-h)/2 \rfloor + h}{S}\right] \times \left[\frac{\lfloor (S-w)/2 \rfloor}{S}, \frac{\lfloor (S-w)/2 \rfloor + w}{S}\right]$$

as a fraction of the grid; a 320×260 brain ground truth, for example, is padded to 320×320 and covers all 40 rows and columns 3.75 to 36.25 of the DINOv3 grid. About 4% of the brain ground truths are non-square, with sizes such as 320×260, 384×312 and 256×224.

A common grid. The two boxes are resampled onto one grid of $\mathrm{round}(h/s) \times \mathrm{round}(w/s)$ bins over the ground truth, 20×20 on knee data, by bilinear interpolation with each feature value placed at the center of its cell. The teacher is sampled once at the center of each bin. For DINOv3 and MedDINOv3 at 640 pixels, whose grid is twice as fine as the common grid, this equals the mean of the four teacher cells in the bin, and for SD and DiT at 512 pixels it interpolates the nearest teacher cells.

Using the whole bottleneck without this crop would compare a teacher representation of the ground truth with student features that additionally cover anatomy and background outside it.

The knee and brain data come in a limited number of shapes: the fastMRI file headers list 64 combinations of acquisition and reconstruction size, which give 63 distinct pairs of full image size $H \times W$ and ground truth size $h \times w$. We check the correspondence on one slice of every such pair and for both students. In every case the student box, mapped back to pixels, lands exactly on the ground truth, the teacher box covers exactly the square-padded ground truth, and the stored ground truth equals the centered crop of the image reconstructed from the fully sampled k-space, before and after the brain readout crop.

Dataset-specific preprocessing. Brain images are acquired with a fully sampled, twofold-oversampled readout. Before undersampling, every brain slice is reduced to 384 readout rows by applying an inverse Fourier transform along that axis, taking a centered crop, and transforming back to k-space. Because the undersampling mask acts only on the phase encode columns, this operation commutes with masking and preserves the measurements supplied to the student. Every brain ground truth remains centered and lies entirely inside the cropped readout, while columns are padded only when necessary. The transformer's 384×512 working size therefore contains every brain slice. Every brain run evaluates the SSIM loss over the complete ground-truth field of view after cropping the reconstruction to it, so the reconstruction loss, alignment region, and reported metrics cover the same pixels.

On knee data the transformer's working size is 640×384. Knee images are 640 pixels tall but between 320 and about 400 pixels wide, so the canvas always holds the 320×320 ground truth and, for about 98% of slices, the whole image. On the remaining slices, which are wider than 384 pixels, the outermost columns are cropped away and the refinement network passes its input through unchanged there. These columns lie outside the ground truth, so the alignment, the reconstruction loss and the evaluation are unaffected.

A.3 The high-SNR validation cohort

The fastMRI validation split contains many slices with a high noise floor, including but not limited to those near the ends of volumes. Metrics on these slices are often dominated by noise and therefore

vary for reasons largely unrelated to the design choices under study, whereas many of the differences considered in this paper are small. We therefore define a high signal-to-noise ratio (SNR) cohort in advance and report the mean peak signal-to-noise ratio over it, denoted by $\mathrm{PSNR}_{\mathrm{hi}}$.

Cohort definition. The SNR of each slice is estimated once from its fully sampled k-space, independently of any reconstruction model. Signal power is the mean of $|k|^2$ in a 32×32 square centered on the direct-current component. Noise power is the mean of $|k|^2$ over four 32×32 corner squares placed within the acquired phase-encoding range, so that zero padding cannot create an artificial zero noise floor. The SNR is

$$10 \log_{10} \left(\frac{\text{signal power}}{\text{noise power}} \right),$$

and the cohort contains all slices above 40 dB unless stated otherwise.

Selection and evaluation. Alignment weights are chosen on a dedicated selection subset (Appendix A.4). The 89 slices used by any selection run are removed from the high-SNR cohort of 1155 slices, leaving 1066 evaluation slices for the knee tables. The figures and the bottleneck analyses use the 554 odd-indexed slices within this evaluation set. No slice used for weight selection therefore contributes to the reported knee results or representation analyses.

Restricting evaluation to high-SNR slices reduces variation from noisy slices, but an aggregate cohort metric may obscure rare or spatially localized errors. Appendix C.1 therefore repeats Tables 1, 2 and 3 using the standard fastMRI volume metrics. Evaluation on annotated pathological regions is outside the scope of this work.

A.4 Choosing the alignment weight

Each combination of student, teacher, anatomy and acceleration receives its own weight λ, because the magnitude of the alignment loss differs between teachers (Appendix A.1). A sweep trains a no-teacher control and three weights for 30k iterations, spaced by a factor of two, or of three on brain data. Each run is scored every 6000 iterations on the selection subset, 40 slices taken at a fixed stride through the high-SNR validation cohort (Appendix A.3), and the weight with the highest mean over the 30k iterations is kept. A mean is used because single checkpoints of the transformer students can differ by more than 1 dB. If the best weight lies at the edge of the sweep, one further weight beyond it is trained, and if the sweep varies by less than 0.30 dB on the transformers or 0.15 dB on E2E-VarNet, the center weight is kept. The selection slices are excluded from every reported number. The selected weight and the control then continue from their own 30k-iteration checkpoints. Table 6 lists every grid and selection.

Table 6: Weight sweeps behind the aligned cells of Tables 1, 2, 3 and 15, selected on the 30k-iteration mean over the selection subset. A flat sweep, whose scores span less than 0.15 dB on E2E-VarNet or 0.30 dB on the transformers, keeps its center weight, and an edge maximum receives one further weight beyond it. A tie between the center and an edge keeps the center, and an edge weight that matches its neighbor within 0.01 dB ends the sweep on that plateau. The E2E-VarNet acceleration cells use SD at 512 pixels and DINOv3 at 320 pixels, with weights 0.78 and 0.777 at 8×. The rank regularizer of Table 1 uses a weight of 0.008, selected from 0.002, 0.008 and 0.032 on the first seed.

Student	teacher	data	grid	selected	shape	seed
Single-coil knee						
HUMUS-Net	SD	knee 8×	0.8, 1.6, 3.2	1.6	interior	0
HUMUS-Net	DINOv3	knee 8×	1.2, 2.4, 4.8	2.4	flat, center	0
HUMUS-Net	MedDINOv3	knee 8×	0.5, 1.0, 2.0	1.0	flat, center	0
HUMUS-Net	DiT	knee 8×	0.225, 0.45, 0.9	0.45	flat, center	0
HUMUS-Net	SD	knee 4×	0.8, 1.6, 3.2	1.6	flat, center	0
HUMUS-Net	SD	knee 16×	0.8, 1.6, 3.2	1.6	flat, center	0
HUMUS-Net	DINOv3	knee 4×	1.2, 2.4, 4.8	2.4	flat, center	0
HUMUS-Net	DINOv3	knee 16×	1.2, 2.4, 4.8	2.4	flat, center	0
HUMUS-Net-L	SD	knee 8×	0.8, 1.6, 3.2	1.6	flat, center	0
HUMUS-Net-L	DINOv3	knee 8×	1.2, 2.4, 4.8	2.4	flat, center	0
HUMUS-Net-L	DiT	knee 8×	0.225, 0.45, 0.9	0.45	flat, center	0
HUMUS-Net-L	MedDINOv3	knee 8×	0.5, 1.0, 2.0	1.0	flat, center	0
E2E-VarNet	SD	knee 8×	0.39, 0.78, 1.56	0.78	flat, center	2
E2E-VarNet	DINOv3	knee 8×	0.39, 0.777, 1.55	0.777	interior	2
E2E-VarNet	MedDINOv3	knee 8×	0.35, 0.7, 1.4	0.7	flat, center	2
E2E-VarNet	DiT	knee 8×	0.9, 1.8, 3.6	1.8	tie, center	2
E2E-VarNet	SD	knee 4×	0.2, 0.4, 0.78, 1.5	0.4	interior	2
E2E-VarNet	DINOv3	knee 4×	0.4, 0.777, 1.5	0.777	flat, center	2
E2E-VarNet	SD	knee 16×	0.4, 0.78, 1.5	0.78	interior	2
E2E-VarNet	DINOv3	knee 16×	0.4, 0.777, 1.5	0.777	flat, center	2
Multi-coil brain						
HUMUS-Net	SD	brain 8×	1.2, 3.6, 10.8	3.6	plateau	0
HUMUS-Net	DINOv3	brain 8×	0.9, 2.7, 8.1	2.7	interior	0
HUMUS-Net	DiT	brain 8×	0.45, 1.35, 4.05	1.35	flat, center	0
HUMUS-Net	DINOv3	brain 16×	0.9, 2.7, 8.1	2.7	interior	0
E2E-VarNet	SD	brain 8×	0.13, 0.4, 1.2	0.4	flat, center	0
E2E-VarNet	DINOv3	brain 8×	0.1, 0.3, 0.9, 2.7	0.9	interior	0
E2E-VarNet	DiT	brain 8×	0.25, 0.75, 2.25	0.75	flat, center	0
E2E-VarNet	DINOv3	brain 16×	0.1, 0.3, 0.9, 2.7	0.9	plateau	0

B THE BOTTLENECK IN DETAIL

This part gives the rank measurement behind Section 4.2.3, its main findings and visualizations of the bottleneck features.

B.1 BOTTLENECK RANK

Measurement. For every cascade we take the bottleneck, the tensor the alignment head reads, on 64 fixed slices taken at a stride of eight through the 554 odd-indexed evaluation slices. On one slice it is a matrix of spatial locations by channels. We report the effective rank (Eq. 5) and the top-1 share $s_1^2 / \sum_i s_i^2$ of this matrix after centering it over its locations, and the redundancy, the mean cosine similarity between distinct locations of the uncentered features, each averaged over the slices and, unless a cascade is named, over the eight cascades. The full grid keeps every bottleneck location, and the ground truth region keeps the 20×20 locations whose center falls inside the ground truth. Figure 3 and Table 7 use the ground truth region and Tables 8, 9 and 10 the full grid, all on raw

channels. The orderings below also hold after standardizing each channel and on a second set of 64 slices.

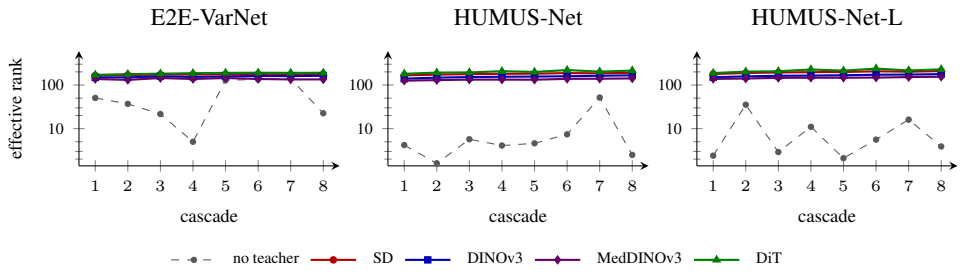

Figure 3: Effective rank of the bottleneck at every cascade after 300k iterations, locations inside the ground truth, log scale, first seed of each student, without a teacher (dashed) and with each of the four teachers of Table 1.

Without a teacher the bottleneck collapses, most strongly on the transformers. Inside the ground truth their controls stay at an effective rank of 9 to 18 through 300k iterations, with 69 to 88% of the energy in one direction (Table 7). On the full grid HUMUS-Net stays between 9.0 and 15.3 and HUMUS-Net-L between 8.8 and 17.2 at every checkpoint to 300k iterations, at most 3% of their width (Table 9).

The collapse reaches almost every cascade. After 300k iterations seven of the eight HUMUS-Net control cascades lie between 2.4 and 6.4 and only the seventh reaches 49.0, and five HUMUS-Net-L cascades lie between 2.6 and 4.6; no unregularized transformer cascade exceeds 50 at any budget of Table 8.

Only the convolutional student partly recovers. The rank of its control rises from 27.85 and 36.83 at 30k iterations to 80.96 and 82.56 at 300k on its two seeds, and on its first seed the rise is concentrated in the fifth to seventh cascades, the highest at 30k iterations, which reach 154 to 186 while the fourth stays below 4 (Table 8).

Alignment keeps every cascade at high rank. Every aligned run uses 18 to 42% of its width, and with DINOv3 no cascade of any student falls below 103 at 30k iterations or below 152 at 300k. The ratio of aligned to control rank falls on E2E-VarNet from 3.9 to 6.4 at 30k iterations to 1.8 to 2.3 at 300k, while on the transformers it stays at least 8.9.

Brain data show the same pattern. The stalled HUMUS-Net control stays below 7, while every aligned brain run lies between 120 and 238 (Table 10).

These are associations across students and training time; the measurements do not show that the rank causes the gain.

Table 7: Bottleneck statistics inside the ground truth on the 64 analysis slices, mean over the eight cascades, one entry per seed. Rank is the effective rank, top-1 the share of energy in the leading direction and redundancy the mean cosine similarity between locations, all on raw channels. The runs are those of Table 1 at the same three budgets, each teacher at its own weight.

Student, iterations	teacher	rank	top-1	redundancy
E2E-VarNet, 30k	none	26.3 / 30.3	0.62 / 0.61	0.50 / 0.43
	SD	156.3 / 155.1	0.13 / 0.13	0.12 / 0.12
	DINOv3	137.8 / 137.8	0.13 / 0.13	0.11 / 0.12
	MedDINOv3	125.5 / 126.4	0.16 / 0.15	0.12 / 0.12
	DiT	171.6 / 170.3	0.13 / 0.13	0.08 / 0.08
E2E-VarNet, 180k	none	60.6 / 56.3	0.45 / 0.52	0.47 / 0.46
	SD	171.3 / 170.1	0.12 / 0.12	0.13 / 0.12
	DINOv3	153.4 / 154.1	0.13 / 0.13	0.12 / 0.12
	MedDINOv3	136.4 / 136.4	0.15 / 0.15	0.13 / 0.13
	DiT	183.8 / 183.9	0.15 / 0.15	0.11 / 0.10
E2E-VarNet, 300k	none	67.4 / 64.3	0.44 / 0.49	0.47 / 0.46
	SD	173.8 / 172.2	0.13 / 0.13	0.13 / 0.13
	DINOv3	156.6 / 156.6	0.13 / 0.13	0.12 / 0.12
	MedDINOv3	137.7 / 138.2	0.15 / 0.15	0.13 / 0.13
	DiT	184.9 / 184.5	0.16 / 0.16	0.12 / 0.12
HUMUS-Net, 30k	none	10.7 / 11.9	0.81 / 0.78	0.66 / 0.61
	SD	136.0 / 135.9	0.16 / 0.16	0.16 / 0.18
	DINOv3	107.4 / 110.3	0.17 / 0.17	0.19 / 0.19
	MedDINOv3	93.6 / 105.0	0.20 / 0.19	0.18 / 0.17
	DiT	160.0 / 160.7	0.14 / 0.15	0.03 / 0.04
HUMUS-Net, 180k	none	9.3 / 17.4	0.84 / 0.70	0.89 / 0.78
	SD	172.1 / 176.9	0.13 / 0.13	0.16 / 0.15
	DINOv3	146.9 / 149.0	0.15 / 0.15	0.16 / 0.16
	MedDINOv3	127.5 / 131.5	0.17 / 0.16	0.15 / 0.15
	DiT	192.1 / 191.5	0.12 / 0.13	0.04 / 0.04
HUMUS-Net, 300k	none	10.2 / 17.7	0.82 / 0.71	0.88 / 0.78
	SD	180.7 / 184.1	0.13 / 0.13	0.15 / 0.14
	DINOv3	155.0 / 156.9	0.14 / 0.14	0.16 / 0.15
	MedDINOv3	134.3 / 138.5	0.16 / 0.15	0.14 / 0.14
	DiT	201.6 / 202.6	0.12 / 0.12	0.03 / 0.03
HUMUS-Net-L, 30k	none	14.4 / 15.4	0.74 / 0.70	0.69 / 0.73
	SD	148.9 / 147.2	0.15 / 0.16	0.16 / 0.17
	DINOv3	117.2 / 119.0	0.17 / 0.17	0.19 / 0.18
	MedDINOv3	113.1 / 111.5	0.19 / 0.19	0.16 / 0.17
	DiT	169.7 / 172.1	0.15 / 0.15	0.03 / 0.03
HUMUS-Net-L, 180k	none	13.6 / 14.3	0.82 / 0.74	0.79 / 0.68
	SD	188.4 / 189.6	0.13 / 0.14	0.15 / 0.15
	DINOv3	157.1 / 159.4	0.15 / 0.15	0.16 / 0.16
	MedDINOv3	139.9 / 140.0	0.16 / 0.16	0.15 / 0.15
	DiT	203.1 / 206.7	0.12 / 0.12	0.04 / 0.04
HUMUS-Net-L, 300k	none	9.9 / 15.3	0.88 / 0.69	0.89 / 0.77
	SD	196.9 / 198.9	0.12 / 0.13	0.15 / 0.14
	DINOv3	166.0 / 168.1	0.14 / 0.14	0.16 / 0.15
	MedDINOv3	146.8 / 148.0	0.15 / 0.15	0.14 / 0.14
	DiT	215.2 / 218.8	0.11 / 0.11	0.03 / 0.03

Table 8: Effective rank of the bottleneck at each cascade on the full grid, raw channels, single-coil knee at 8×, first seed of each student, without a teacher and with DINOv3, with the mean over the eight cascades in the last column. The runs are those of Table 1.

Student	iterations	teacher	1	2	3	4	5	6	7	8	mean
E2E-VarNet	30k	none	9.8	14.2	11.1	3.6	42.4	46.5	78.3	16.9	27.85
		DINOv3	129.0	149.7	155.1	146.1	157.2	149.3	152.1	144.4	147.87
	180k	none	48.9	35.5	16.7	2.9	132.4	137.4	169.8	21.8	70.68
		DINOv3	147.7	162.9	175.5	159.7	173.2	172.2	167.1	164.9	165.41
	300k	none	60.3	42.2	18.1	2.8	153.5	159.0	185.8	25.9	80.96
		DINOv3	152.8	164.9	178.8	159.3	176.3	177.5	170.0	169.1	168.58
HUMUS-Net	30k	none	2.8	3.6	4.8	6.2	5.3	3.2	42.7	24.9	11.70
		DINOv3	110.2	103.2	104.0	121.3	131.7	137.5	134.6	146.4	123.61
	180k	none	5.1	2.4	4.2	3.5	5.1	4.5	45.5	3.6	9.24
		DINOv3	161.4	154.1	152.4	167.1	184.3	187.0	188.2	195.1	173.69
	300k	none	4.0	2.4	6.4	3.7	4.5	3.1	49.0	3.7	9.59
		DINOv3	171.6	159.7	163.0	177.7	197.7	196.9	197.8	206.0	183.80
HUMUS-Net-L	30k	none	3.9	21.3	3.8	11.7	3.9	13.3	35.2	18.3	13.93
		DINOv3	107.1	142.4	146.8	143.3	136.4	160.3	150.7	163.7	143.85
	180k	none	3.9	20.2	3.1	13.7	3.4	3.4	49.1	4.5	12.67
		DINOv3	158.8	190.1	211.7	201.3	210.2	222.2	217.6	221.1	204.12
	300k	none	2.6	28.6	2.9	9.8	3.0	3.2	16.4	4.6	8.88
		DINOv3	177.5	201.9	223.9	208.2	223.5	227.1	232.0	231.3	215.69

Table 9: Effective rank of the bottleneck on the full grid, raw channels, mean over the eight cascades, for the runs of Table 1. Two entries are two seeds, first / second. On the first seed, the the HUMUS-Net control destabilizes briefly around iteration 84k and recovers.

Student	teacher	30k iter.	180k iter.	300k iter.
E2E-VarNet	none	27.85 / 36.83	70.68 / 72.74	80.96 / 82.56
	rank regularizer	465.98 / 458.88	479.12 / 470.48	484.83 / 474.84
	deep supervision	40.51 / 34.88	67.89 / 59.09	74.81 / 66.87
	SD	167.14 / 168.01	187.93 / 189.13	188.87 / 188.04
	DINOv3	147.87 / 158.13	165.41 / 174.49	168.58 / 175.80
	MedDINOv3	143.55 / 143.58	156.21 / 152.04	156.10 / 152.21
	DiT	177.80 / 181.71	190.14 / 194.57	187.65 / 191.69
HUMUS-Net	none	11.70 / 12.61	9.24 / 13.45	9.59 / 13.62
	SD	152.14 / 161.93	189.61 / 211.65	207.68 / 221.46
	DINOv3	123.61 / 121.10	173.69 / 162.50	183.80 / 171.10
	MedDINOv3	111.75 / 122.43	154.07 / 154.71	163.17 / 161.98
	DiT	162.52 / 158.80	203.39 / 205.67	215.12 / 219.02
HUMUS-Net-L	none	13.93 / 15.84	12.67 / 15.88	8.88 / 17.20
	SD	179.31 / 189.71	235.80 / 256.08	251.97 / 272.84
	DINOv3	143.85 / 141.62	204.12 / 191.19	215.69 / 205.08
	MedDINOv3	142.78 / 143.74	181.69 / 179.99	192.19 / 192.90
	DiT	196.44 / 195.20	252.61 / 254.78	269.90 / 274.30

Table 10: Effective rank of the bottleneck on the full grid, raw channels, mean over the eight cascades, for the multi-coil brain runs of Table 3, on 100 test slices.

Student	teacher	30k iter.	150k iter.	420k iter.
HUMUS-Net	none	4.48	6.99	5.26
	SD	187.15	218.05	224.13
	DINOv3	119.95	154.15	161.82
	DiT	197.72	231.57	237.76
E2E-VarNet	none	25.23	34.82	40.93
	SD	137.94	132.01	125.48
	DINOv3	138.48	138.32	133.15
	DiT	163.52	163.72	157.80

B.2 ALIGNED BOTTLENECK VISUALIZATIONS

Figures 4 and 5 show, for one validation slice and at cascades one, four and eight, three quantities: the raw student bottleneck, the same bottleneck after the cascade's projection head and the frozen teacher's features. Both figures show the first seed of each student after 300k iterations, without a teacher and with each of the four teachers. Each map shows the first three principal components of its features as color, and panels with the same border color share one basis, so equal colors denote equal directions within one basis. The raw bottlenecks of all runs (gray) share a basis fitted on their pooled locations. Each teacher, DINOv3 (blue), MedDINOv3 (purple), SD (red) and DiT (green), has its own basis and color range fitted on its features, and the projected bottleneck of the matching run is shown in that basis, which is the space in which the loss compares the two. For HUMUS-Net, the bottleneck channels of each run are standardized over its locations before the shared basis is fitted. The ground truth is shown at the top of the teacher column.

Without a teacher the bottleneck looks different at every cascade and generally devoid of any input image features, especialy for a transformer. With a teacher the projected bottlenecks closely resemble the teacher's features.

Companion figures. Figures 6 and 7 show principal components four to six of the same bases, which carry the structure the leading three leave out. Figures 8 and 10 show the multi-coil brain cells of Table 3, and Figures 9 and 11 their components four to six.

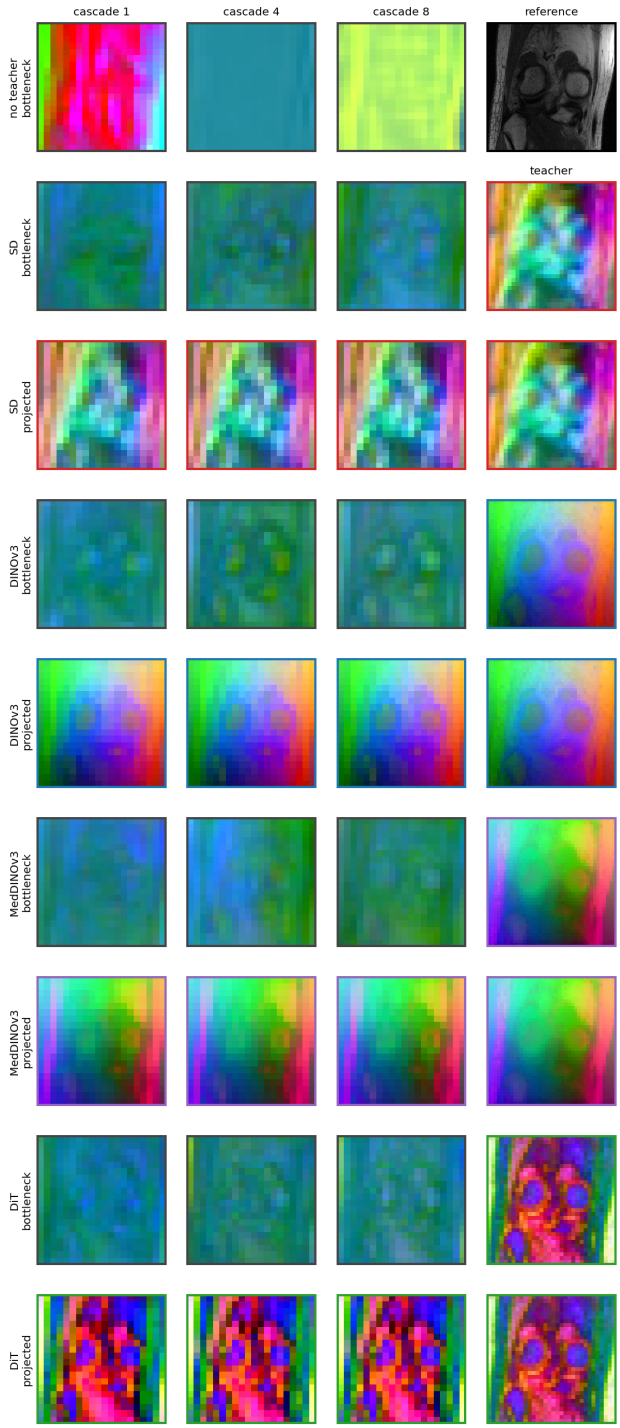

Figure 4: Principal components one to three of the bottleneck features at cascades one, four and eight for one validation slice, E2E-VarNet on its first seed after 300k iterations, without a teacher and with each of the four teachers. Panels with the same border color share one basis.

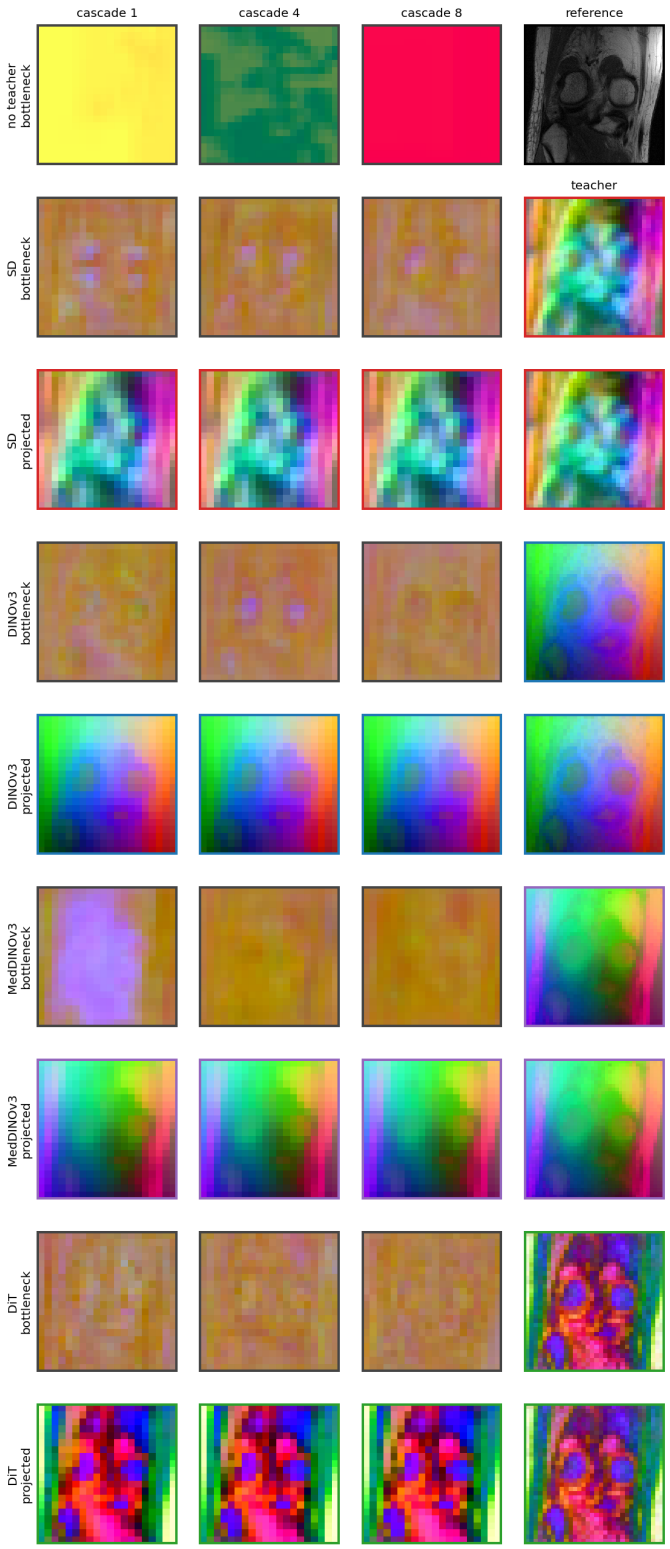

Figure 5: The quantities of Figure 4 for HUMUS-Net on its first seed after 300k iterations.

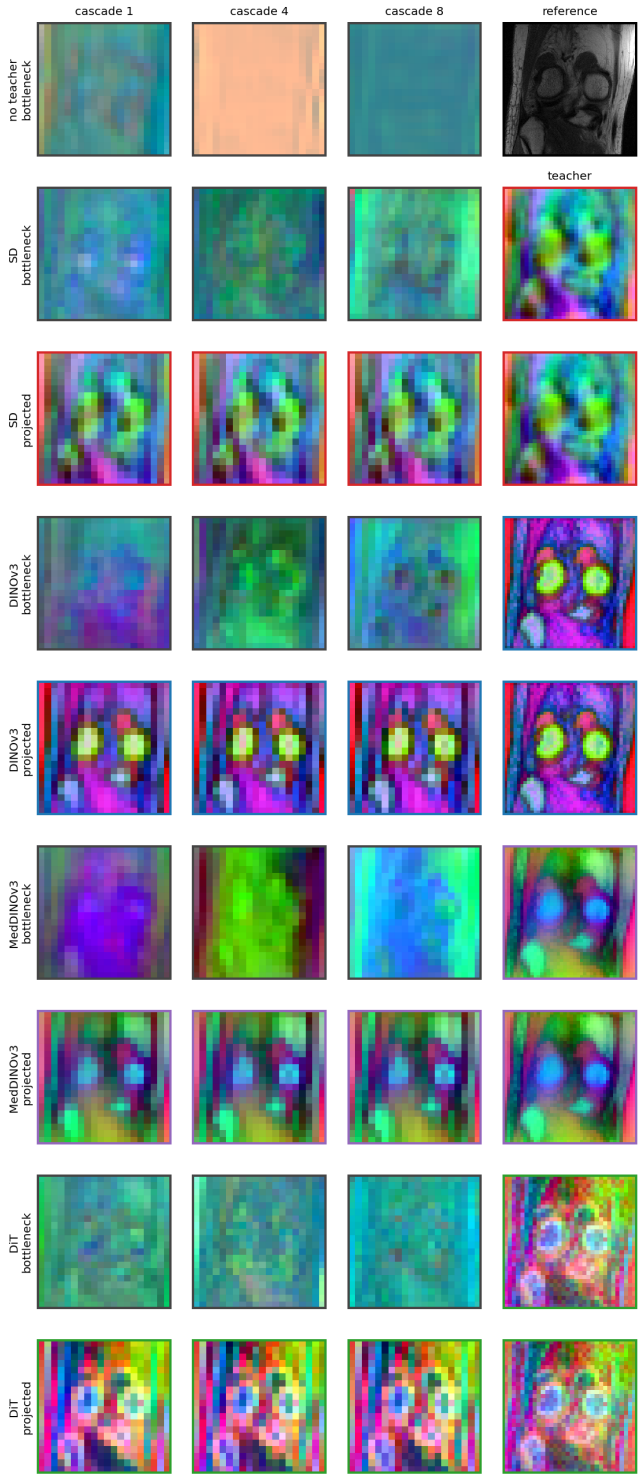

Figure 6: Components four to six of the basis of Figure 4, E2E-VarNet.

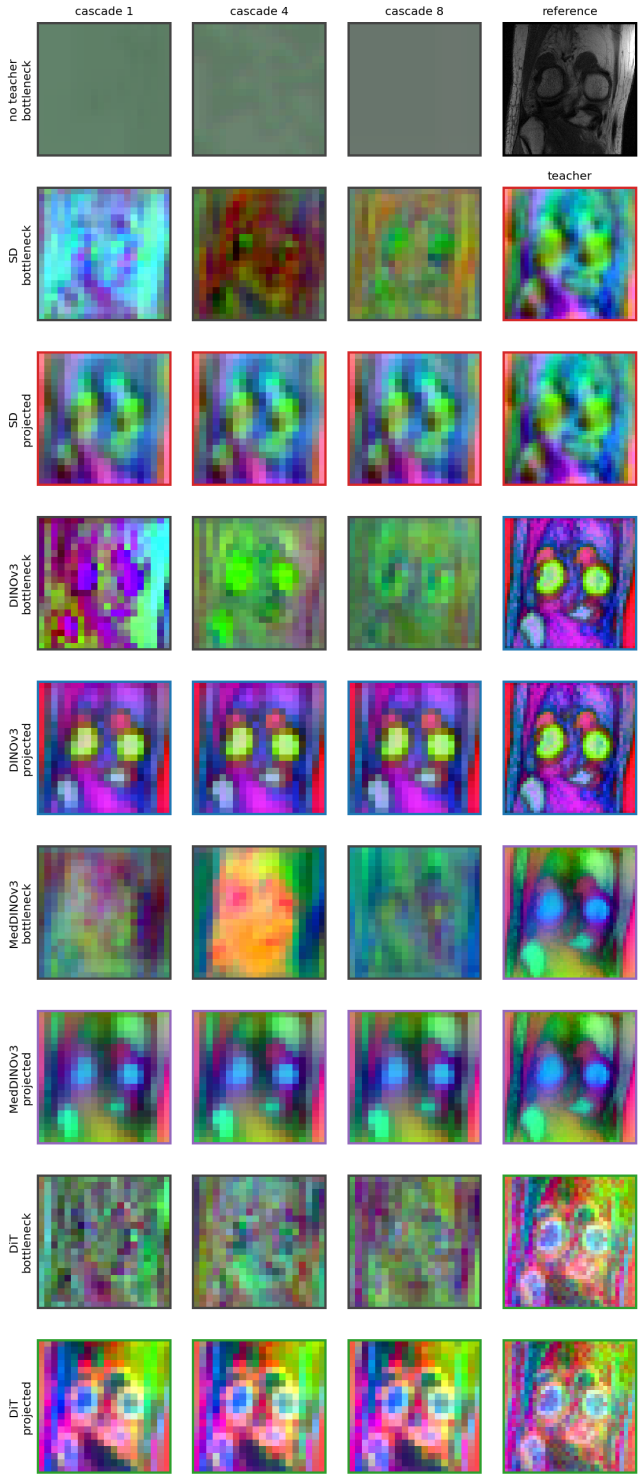

Figure 7: Components four to six of the basis of Figure 5, HUMUS-Net.

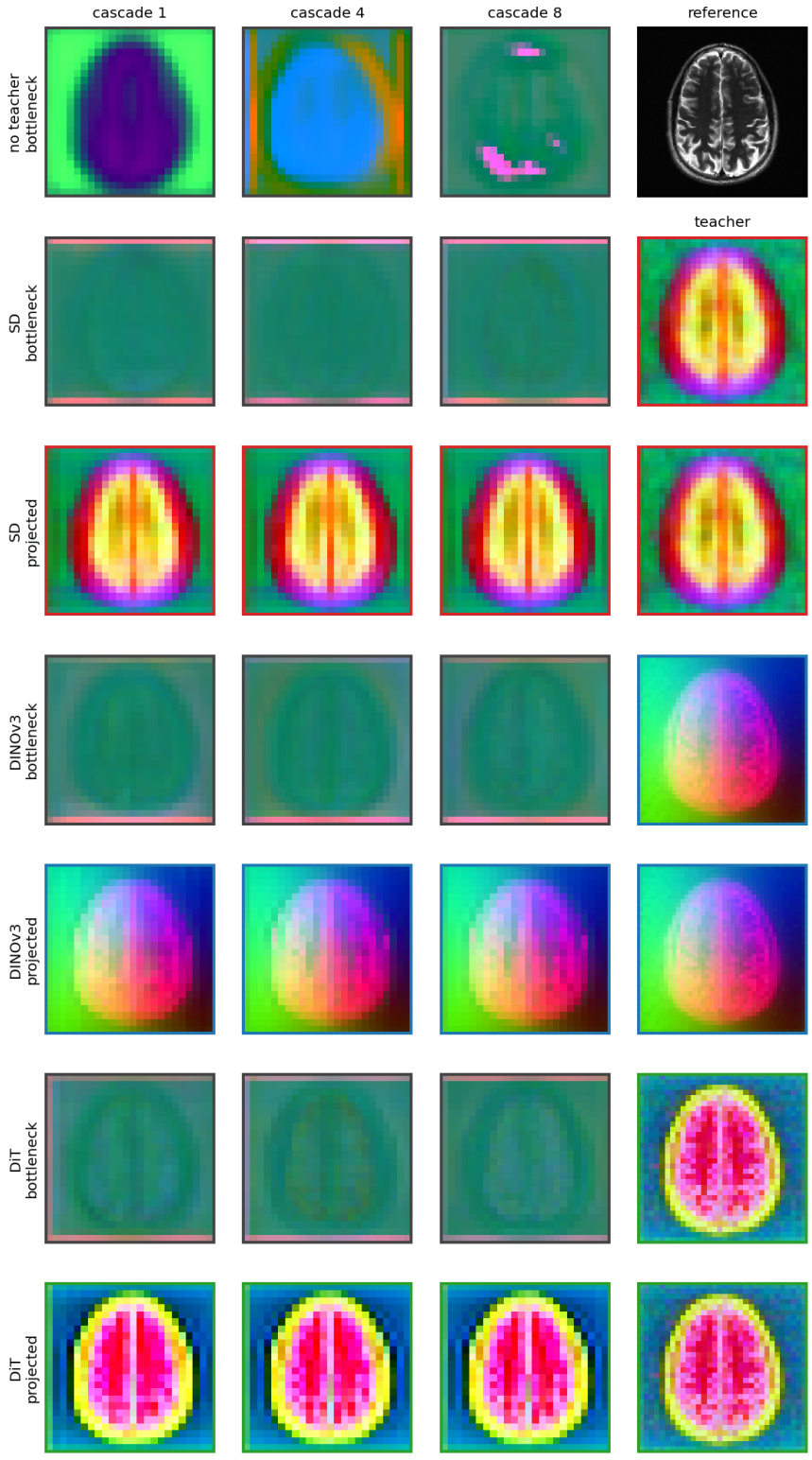

Figure 8: The quantities of Figure 4 for the multi-coil brain runs of Table 3, E2E-VarNet after 300k iterations, an intermediate checkpoint of the 420k runs.

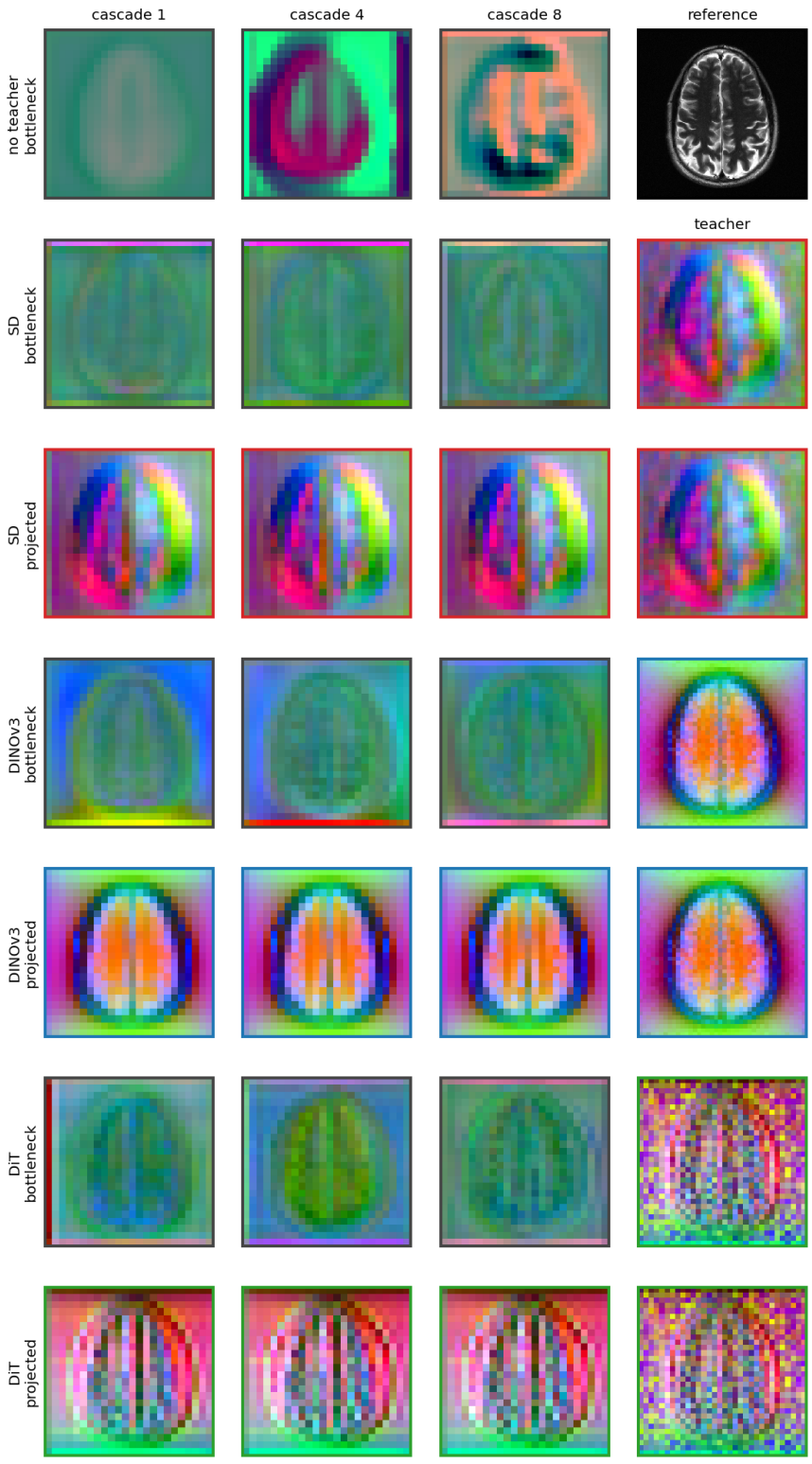

Figure 9: Components four to six of the basis of Figure 8, E2E-VarNet on brain data.

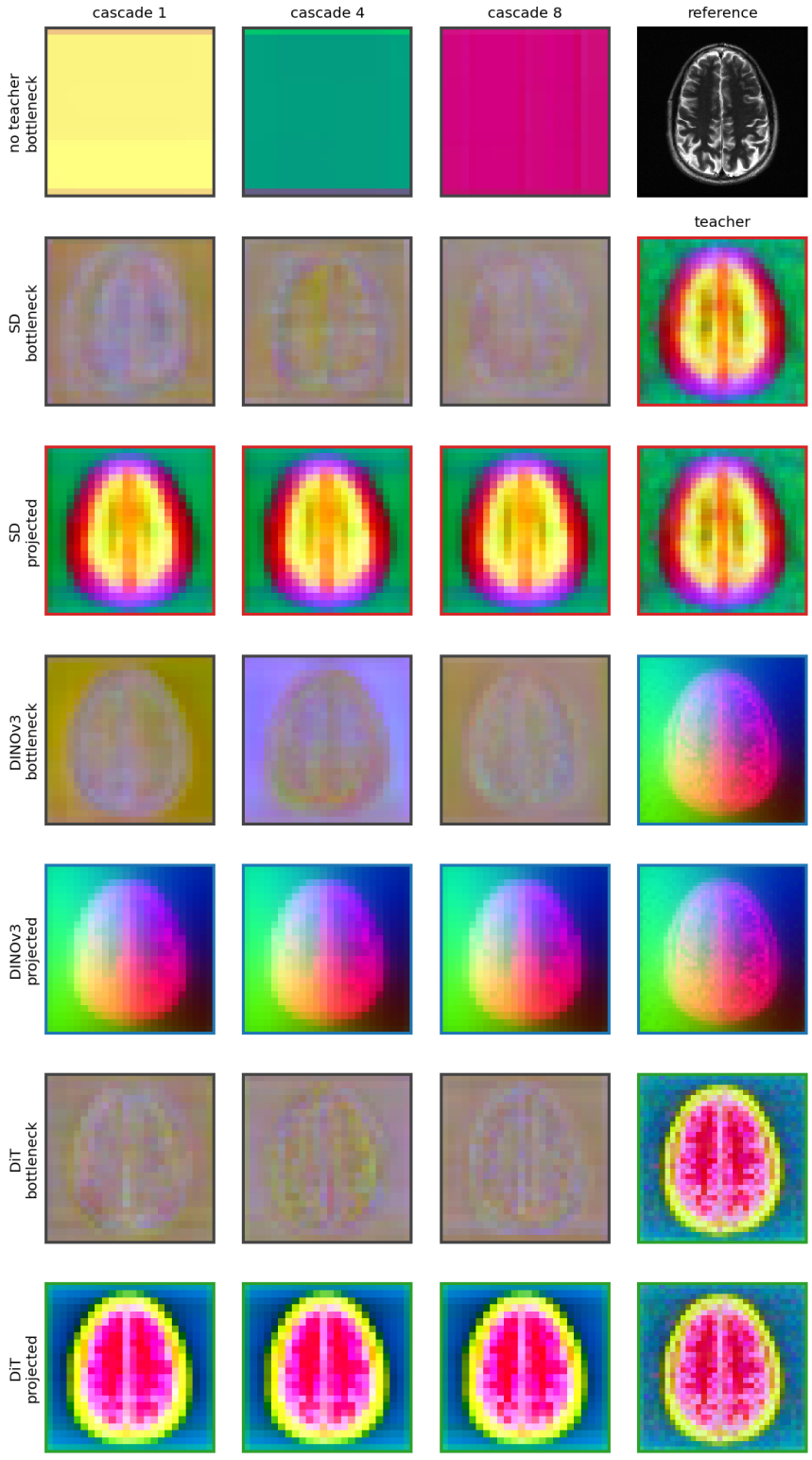

Figure 10: The quantities of Figure 4 for HUMUS-Net on brain data after 300k iterations, an intermediate checkpoint of the 420k runs.

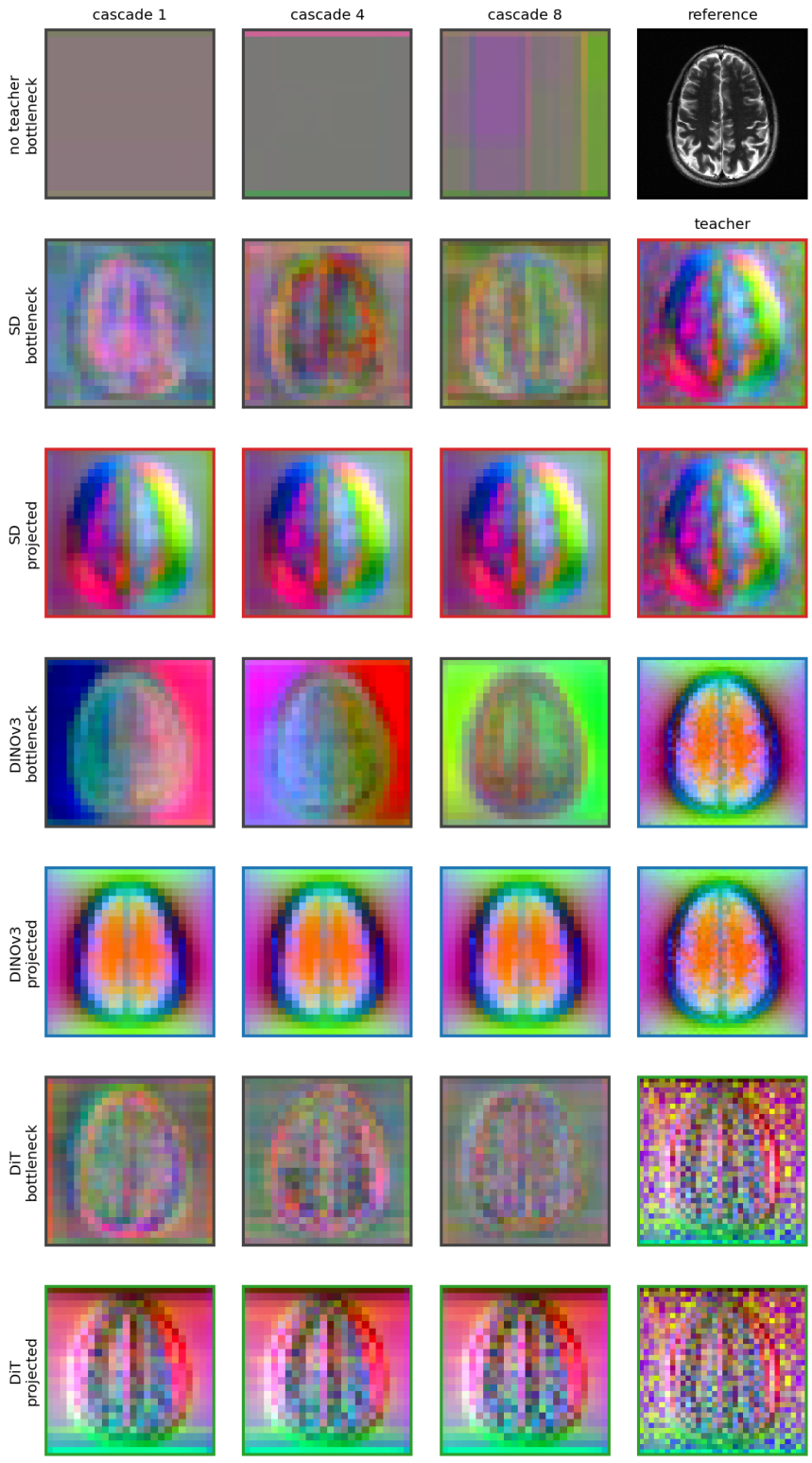

Figure 11: Components four to six of the basis of Figure 10, HUMUS-Net on brain data.

C FURTHER RESULTS

These sections extend the main tables: their volume-wise metrics, the brain protocol with its $16\times$ block and the teacher comparison on the convolutional student repeated at lower teacher resolution.

C.1 VOLUME-WISE RESULTS

Tables 11 to 14 repeat Tables 1, 2 and 3 with the volume metrics of fastMRI: PSNR from the summed error of a volume and its maximum, SSIM with the volume maximum as data range and NMSE from the summed error and reference energy, each averaged over volumes. On knee data the metrics cover every slice of the 199 validation volumes except those used to choose a weight, and on brain data every slice of the 150 test volumes of Table 3.

Table 11: Table 1 with volume-wise PSNR in dB, gain over the same-seed no-teacher run as a subscript.

Student	teacher	30k iter.	180k iter.	300k iter., drop at 240k
E2E-VarNet	none	29.245 / 29.380	29.948 / 29.999	30.115 / 30.192
	rank regularizer	$29.425_{+0.179}$ / $29.346_{-0.034}$	$30.019_{+0.071}$ / $29.958_{-0.041}$	$30.148_{+0.033}$ / $30.134_{-0.058}$
	deep supervision	$29.298_{+0.052}$ / $29.239_{-0.141}$	$29.925_{-0.023}$ / $29.968_{-0.031}$	$30.159_{+0.044}$ / $30.165_{-0.026}$
	SD	$29.744_{+0.499}$ / $29.683_{+0.303}$	$30.143_{+0.195}$ / $30.131_{+0.131}$	$30.260_{+0.145}$ / $30.298_{+0.106}$
	DINOv3	$29.620_{+0.375}$ / $29.565_{+0.185}$	$30.037_{+0.089}$ / $30.177_{+0.178}$	$30.242_{+0.127}$ / $30.315_{+0.124}$
	MedDINOv3	$29.672_{+0.427}$ / $29.610_{+0.230}$	$30.025_{+0.077}$ / $30.136_{+0.136}$	$30.232_{+0.117}$ / $30.272_{+0.080}$
	DiT	$29.794_{+0.549}$ / $29.738_{+0.358}$	$30.088_{+0.140}$ / $30.333_{+0.334}$	$30.280_{+0.165}$ / $30.302_{+0.111}$
HUMUS-Net	none	29.040 / 29.406	29.928 / 29.802	29.979 / 30.101
	SD	$29.789_{+0.749}$ / $29.714_{+0.308}$	$30.338_{+0.410}$ / $30.182_{+0.380}$	$30.371_{+0.392}$ / $30.399_{+0.299}$
	DINOv3	$29.665_{+0.625}$ / $29.727_{+0.320}$	$30.080_{+0.152}$ / $30.144_{+0.342}$	$30.332_{+0.353}$ / $30.399_{+0.298}$
	MedDINOv3	$29.590_{+0.550}$ / $29.592_{+0.186}$	$30.244_{+0.316}$ / $30.149_{+0.347}$	$30.380_{+0.401}$ / $30.330_{+0.229}$
	DiT	$29.783_{+0.743}$ / $29.646_{+0.239}$	$30.180_{+0.252}$ / $30.108_{+0.305}$	$30.377_{+0.398}$ / $30.369_{+0.269}$
HUMUS-Net-L	none	29.012 / 29.109	29.711 / 29.788	27.669 / 30.089
	SD	$29.697_{+0.685}$ / $29.693_{+0.585}$	$30.332_{+0.621}$ / $30.255_{+0.468}$	$30.394_{+2.725}$ / $30.410_{+0.321}$
	DINOv3	$29.804_{+0.792}$ / $29.628_{+0.520}$	$30.119_{+0.408}$ / $30.238_{+0.450}$	$30.390_{+2.721}$ / $30.408_{+0.319}$
	MedDINOv3	$29.710_{+0.698}$ / $29.537_{+0.428}$	$30.199_{+0.488}$ / $30.153_{+0.365}$	$30.377_{+2.708}$ / $30.363_{+0.274}$
	DiT	$29.878_{+0.865}$ / $29.707_{+0.598}$	$30.170_{+0.458}$ / $30.261_{+0.473}$	$30.444_{+2.774}$ / $30.404_{+0.314}$

Table 12: Table 1 with volume-wise SSIM, gain over the same-seed no-teacher run as a subscript.

Student	teacher	30k iter.	180k iter.	300k iter., drop at 240k
E2E-VarNet	none	0.6642 / 0.6671	0.6776 / 0.6792	0.6813 / 0.6828
	rank regularizer	$0.6688_{+0.0045}$ / $0.6667_{-0.0004}$	$0.6793_{+0.0017}$ / $0.6793_{+0.0001}$	$0.6826_{+0.0013}$ / $0.6823_{-0.0005}$
	deep supervision	$0.6640_{-0.0003}$ / $0.6642_{-0.0029}$	$0.6773_{-0.0003}$ / $0.6783_{-0.0009}$	$0.6821_{+0.0008}$ / $0.6820_{-0.0007}$
	SD	$0.6722_{+0.0080}$ / $0.6721_{+0.0050}$	$0.6817_{+0.0041}$ / $0.6823_{+0.0031}$	$0.6846_{+0.0033}$ / $0.6850_{+0.0022}$
	DINOv3	$0.6707_{+0.0065}$ / $0.6707_{+0.0036}$	$0.6807_{+0.0031}$ / $0.6827_{+0.0035}$	$0.6845_{+0.0032}$ / $0.6853_{+0.0025}$
	MedDINOv3	$0.6705_{+0.0063}$ / $0.6715_{+0.0044}$	$0.6798_{+0.0022}$ / $0.6815_{+0.0023}$	$0.6836_{+0.0022}$ / $0.6846_{+0.0018}$
	DiT	$0.6733_{+0.0090}$ / $0.6738_{+0.0067}$	$0.6817_{+0.0041}$ / $0.6835_{+0.0043}$	$0.6850_{+0.0037}$ / $0.6854_{+0.0026}$
HUMUS-Net	none	0.6535 / 0.6611	0.6739 / 0.6749	0.6784 / 0.6802
	SD	$0.6708_{+0.0173}$ / $0.6699_{+0.0087}$	$0.6825_{+0.0086}$ / $0.6827_{+0.0078}$	$0.6857_{+0.0072}$ / $0.6860_{+0.0058}$
	DINOv3	$0.6703_{+0.0168}$ / $0.6703_{+0.0091}$	$0.6830_{+0.0091}$ / $0.6817_{+0.0068}$	$0.6854_{+0.0070}$ / $0.6856_{+0.0054}$
	MedDINOv3	$0.6684_{+0.0149}$ / $0.6690_{+0.0079}$	$0.6825_{+0.0086}$ / $0.6817_{+0.0069}$	$0.6849_{+0.0065}$ / $0.6847_{+0.0045}$
	DiT	$0.6734_{+0.0199}$ / $0.6708_{+0.0097}$	$0.6836_{+0.0097}$ / $0.6824_{+0.0076}$	$0.6857_{+0.0073}$ / $0.6855_{+0.0053}$
HUMUS-Net-L	none	0.6532 / 0.6556	0.6700 / 0.6751	0.6291 / 0.6793
	SD	$0.6728_{+0.0196}$ / $0.6721_{+0.0165}$	$0.6841_{+0.0142}$ / $0.6835_{+0.0084}$	$0.6865_{+0.0574}$ / $0.6862_{+0.0069}$
	DINOv3	$0.6726_{+0.0194}$ / $0.6703_{+0.0148}$	$0.6834_{+0.0134}$ / $0.6832_{+0.0082}$	$0.6865_{+0.0574}$ / $0.6862_{+0.0069}$
	MedDINOv3	$0.6704_{+0.0172}$ / $0.6693_{+0.0137}$	$0.6823_{+0.0124}$ / $0.6823_{+0.0072}$	$0.6860_{+0.0568}$ / $0.6853_{+0.0060}$
	DiT	$0.6740_{+0.0208}$ / $0.6721_{+0.0165}$	$0.6836_{+0.0136}$ / $0.6835_{+0.0084}$	$0.6868_{+0.0576}$ / $0.6860_{+0.0067}$

C.2 MULTI-COIL BRAIN TRAINING PROTOCOL AND HIGHER BRAIN ACCELERATION REGIME

The brain runs use $8\times$ with a 4% center, and compute the SSIM loss over the whole ground truth. They follow the knee training protocol, except that the learning rate drops tenfold at iteration 96k and HUMUS-Net trains on two GPUs with distributed data parallelism, so each optimization step processes two slices, counts as two iterations and averages their gradients. The brain HUMUS-Net runs therefore take half as many optimization steps as E2E-VarNet at the same iteration count, and the learning rate drop at iteration 96k falls at step 48k. Tables 3 and 15 score all 2382 slices of

Table 13: Table 2 with volume-wise PSNR in dB and SSIM, gain over the same student's no-teacher run as a subscript.

		4×		8×		16×	
Student	teacher	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
E2E-VarNet, 30k iter.	none	31.681	0.7403	29.297	0.6652	26.240	0.5891
	SD	$31.768_{+0.088}$	$0.7420_{+0.0017}$	$29.509_{+0.211}$	$0.6702_{+0.0050}$	$26.936_{+0.696}$	$0.6057_{+0.0166}$
	DINOv3	$31.864_{+0.183}$	$0.7423_{+0.0020}$	$29.633_{+0.336}$	$0.6697_{+0.0046}$	$26.846_{+0.606}$	$0.6058_{+0.0167}$
E2E-VarNet, 240k iter.	none	31.948	0.7444	30.035	0.6801	27.591	0.6203
	SD	$31.943_{-0.005}$	$0.7451_{+0.0007}$	$30.129_{+0.094}$	$0.6825_{+0.0025}$	$27.864_{+0.272}$	$0.6273_{+0.0070}$
	DINOv3	$31.960_{+0.012}$	$0.7454_{+0.0010}$	$30.085_{+0.051}$	$0.6822_{+0.0022}$	$27.819_{+0.228}$	$0.6272_{+0.0068}$
HUMUS-Net, 30k iter.	none	31.647	0.7368	29.040	0.6535	25.528	0.5729
	SD	$31.685_{+0.038}$	$0.7382_{+0.0014}$	$29.789_{+0.749}$	$0.6708_{+0.0173}$	$27.234_{+1.705}$	$0.6076_{+0.0347}$
	DINOv3	$31.706_{+0.059}$	$0.7381_{+0.0013}$	$29.665_{+0.625}$	$0.6703_{+0.0168}$	$27.234_{+1.706}$	$0.6079_{+0.0350}$
HUMUS-Net, 240k iter.	none	31.933	0.7431	29.880	0.6758	26.404	0.5974
	SD	$32.047_{+0.113}$	$0.7453_{+0.0022}$	$30.049_{+0.169}$	$0.6832_{+0.0075}$	$28.016_{+1.611}$	$0.6299_{+0.0325}$
	DINOv3	$32.039_{+0.106}$	$0.7446_{+0.0015}$	$30.286_{+0.406}$	$0.6839_{+0.0081}$	$28.158_{+1.754}$	$0.6316_{+0.0342}$

Table 14: Table 3 with volume-wise PSNR in dB, SSIM and NMSE, gain over the no-teacher run of the same student and budget as a subscript.

		30k iter.			150k iter.			420k iter.		
Student	teacher	PSNR	SSIM	NMSE	PSNR	SSIM	NMSE	PSNR	SSIM	NMSE
HUMUS-Net	none	29.99	0.8527	0.0465	30.24	0.8674	0.0441	31.96	0.8989	0.0297
	SD	$33.83_{+3.85}$	$0.9175_{+0.0647}$	$0.0193_{-0.0272}$	$35.62_{+5.39}$	$0.9321_{+0.0647}$	$0.0129_{-0.0312}$	$35.83_{+3.86}$	$0.9340_{+0.0351}$	$0.0123_{-0.0174}$
	DINOv3	$33.51_{+3.52}$	$0.9176_{+0.0649}$	$0.0209_{-0.0256}$	$35.89_{+5.65}$	$0.9340_{+0.0665}$	$0.0122_{-0.0319}$	$36.13_{+4.17}$	$0.9359_{+0.0370}$	$0.0116_{-0.0182}$
	DiT	$33.67_{+3.68}$	$0.9126_{+0.0598}$	$0.0201_{-0.0264}$	$35.93_{+5.69}$	$0.9342_{+0.0668}$	$0.0121_{-0.0320}$	$36.18_{+4.21}$	$0.9361_{+0.0372}$	$0.0114_{-0.0183}$
E2E-VarNet	none	33.93	0.9200	0.0189	35.62	0.9325	0.0129	35.89	0.9343	0.0122
	SD	$34.35_{+0.42}$	$0.9214_{+0.0014}$	$0.0172_{-0.0017}$	$35.93_{+0.32}$	$0.9345_{+0.0020}$	$0.0120_{-0.0009}$	$36.15_{+0.26}$	$0.9361_{+0.0018}$	$0.0115_{-0.0007}$
	DINOv3	$34.23_{+0.29}$	$0.9197_{-0.0003}$	$0.0177_{-0.0012}$	$35.93_{+0.31}$	$0.9346_{+0.0022}$	$0.0121_{-0.0009}$	$36.23_{+0.33}$	$0.9365_{+0.0022}$	$0.0113_{-0.0009}$
	DiT	$34.14_{+0.21}$	$0.9226_{+0.0025}$	$0.0179_{-0.0010}$	$35.99_{+0.37}$	$0.9349_{+0.0024}$	$0.0119_{-0.0011}$	$36.22_{+0.32}$	$0.9366_{+0.0022}$	$0.0113_{-0.0009}$

150 test volumes drawn in proportion to the five contrasts, and Figure 12 uses 100 fixed test slices. The first-seed HUMUS-Net control stalls; on a second seed after 30k iterations the control reaches 30.01 dB and SD, DINOv3 and DiT reach 31.44, 31.48 and 31.65 dB. Table 15 repeats the setting at 16× with a 2% fully sampled center.

Table 15: Multi-coil brain at 16× with a 2% center, protocol as in Table 3, one seed, SSIM loss over the whole ground truth, scored on all 2382 slices of 150 test volumes. Subscripts give the PSNR gain over the no-teacher run of the same student and budget, as in Table 3.

		30k iter.			150k iter.		
Student	teacher	PSNR	SSIM	NMSE	PSNR	SSIM	NMSE
HUMUS-Net	none	24.78	0.7547	0.0975	26.01	0.7899	0.0726
	DINOv3	$27.69_{+2.92}$	0.8210	0.0493	$29.95_{+3.94}$	0.8533	0.0291
E2E-VarNet	none	27.31	0.8220	0.0529	29.21	0.8471	0.0341
	DINOv3	$28.09_{+0.78}$	0.8297	0.0453	$29.61_{+0.40}$	0.8512	0.0314

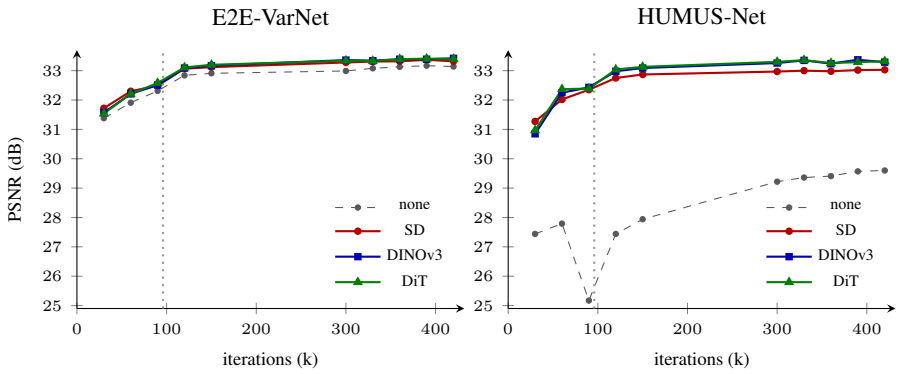

Figure 12: Training curves of the runs of Table 3, to 420k iterations for both students, PSNR on 100 fixed brain test slices against training iterations, no teacher dashed and each teacher solid. The dotted line marks the learning rate drop at iteration 96k.

D QUALITATIVE RECONSTRUCTIONS

Figure 13 shows three knee slices with a zoomed-in anatomical detail window. Figures 14, 15 and 16 show the same comparison for the two transformer students and for multi-coil brain.

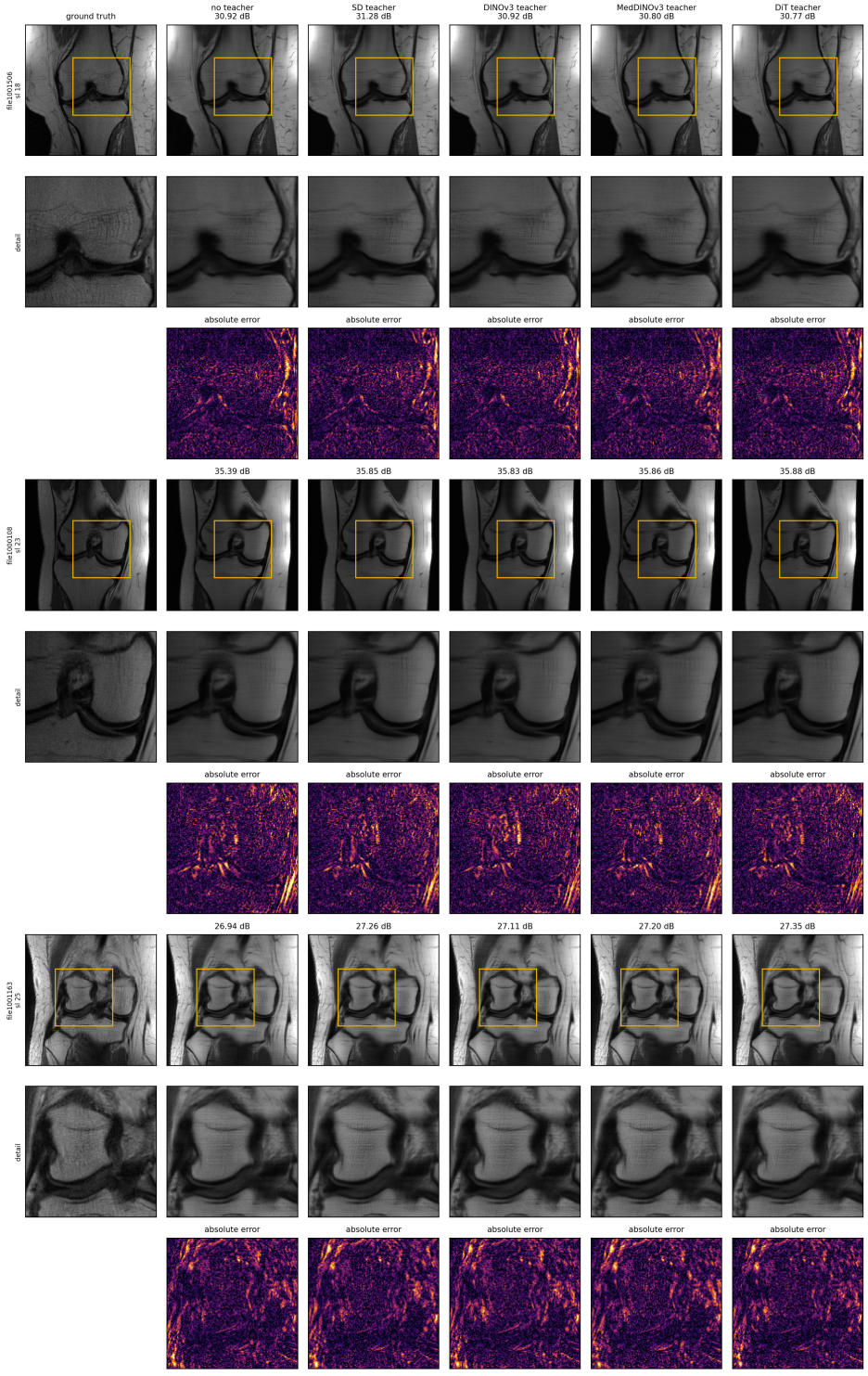

Figure 13: Single-coil knee at $8\times$, three evaluation slices, E2E-VarNet on one seed without a teacher and with each of the four teachers after 300k iterations. Each slice shows the full image with the detail window, the window enlarged and the absolute error in the window, on one error scale per slice.

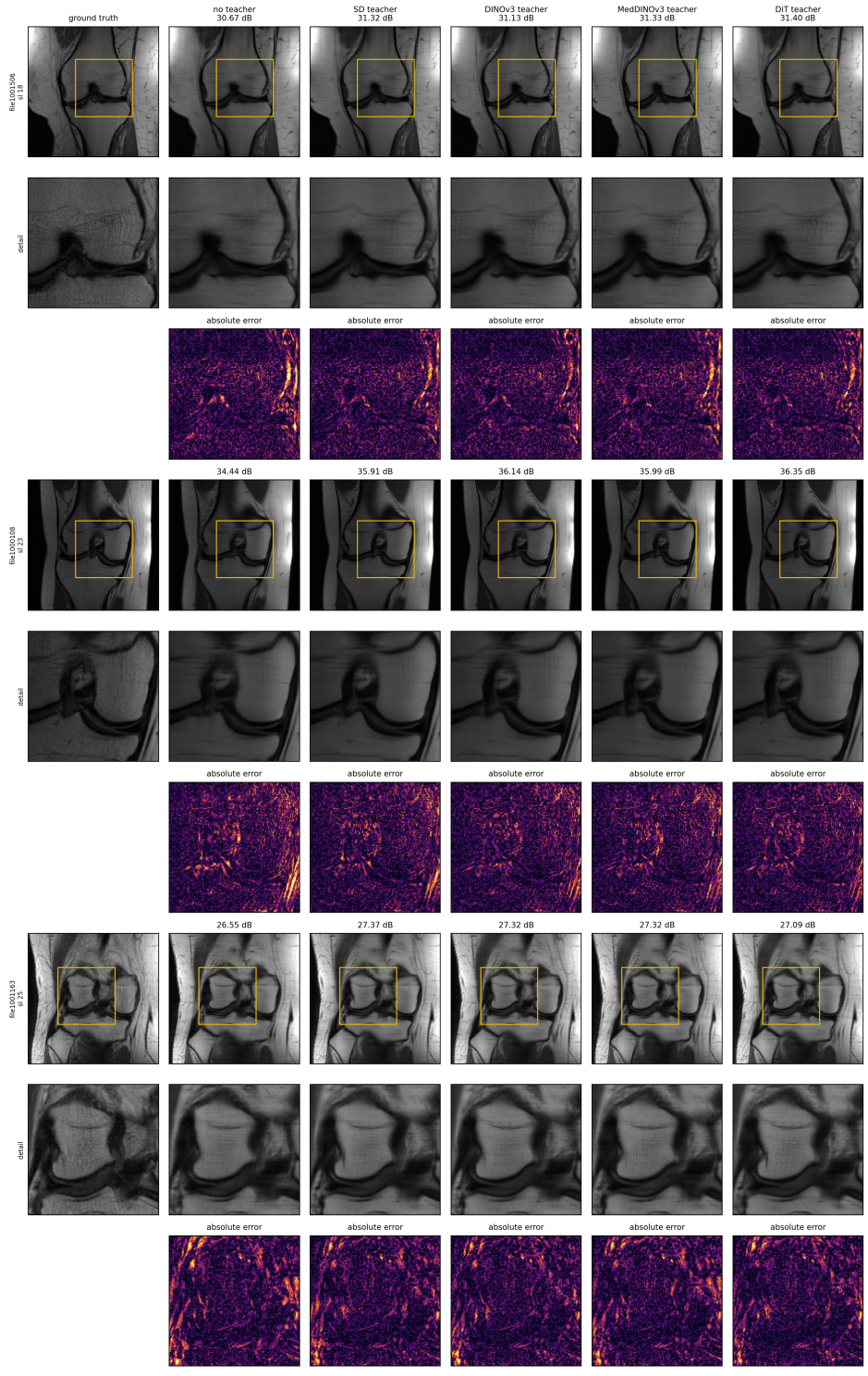

Figure 14: The comparison of Figure 13 for HUMUS-Net on one seed without a teacher and with each of the four teachers after 300k iterations, on the same three slices.

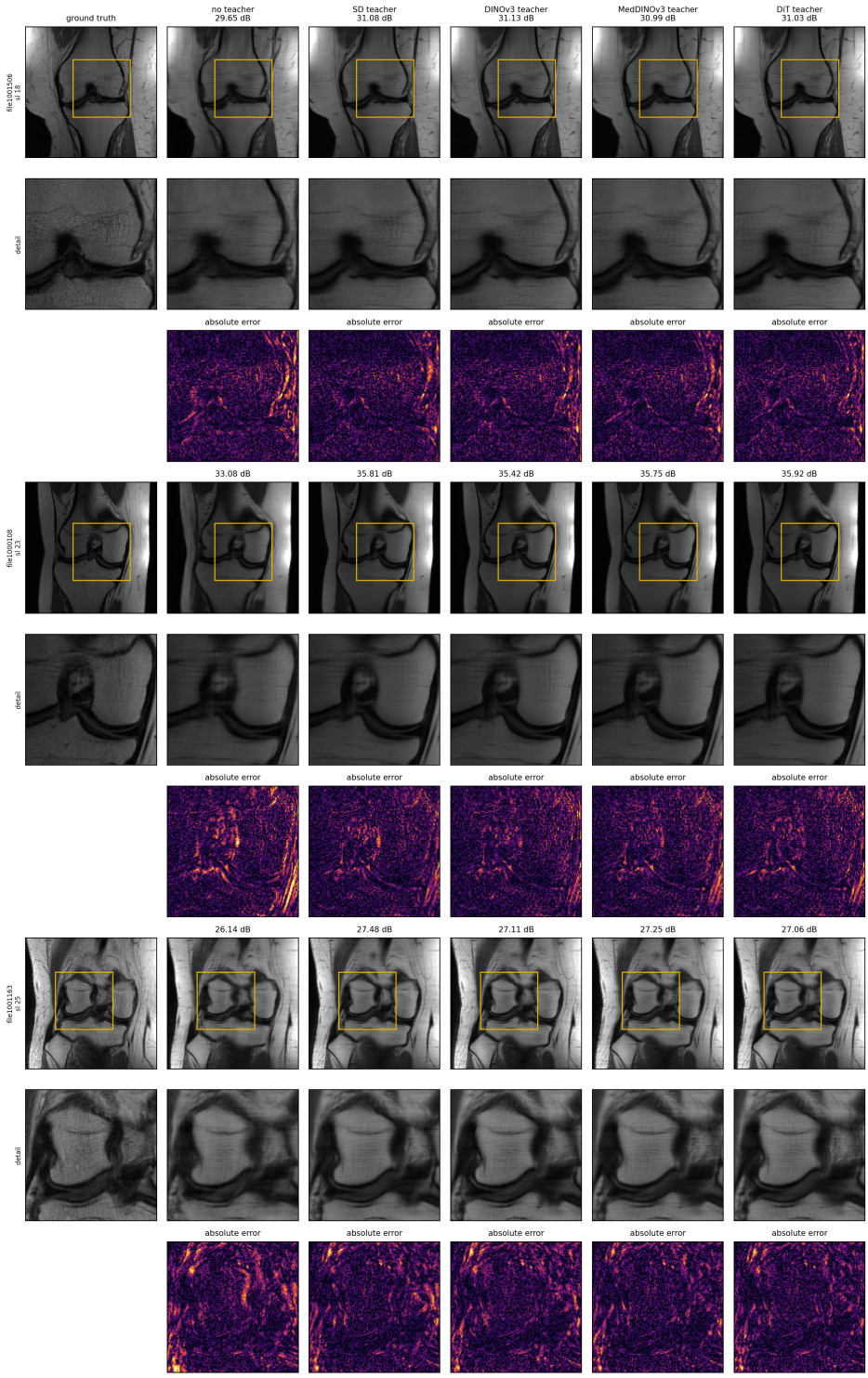

Figure 15: The comparison of Figure 13 for HUMUS-Net-L on the first seed without a teacher and with each of the four teachers after 180k iterations, the last budget before its control collapses, on the same three slices.

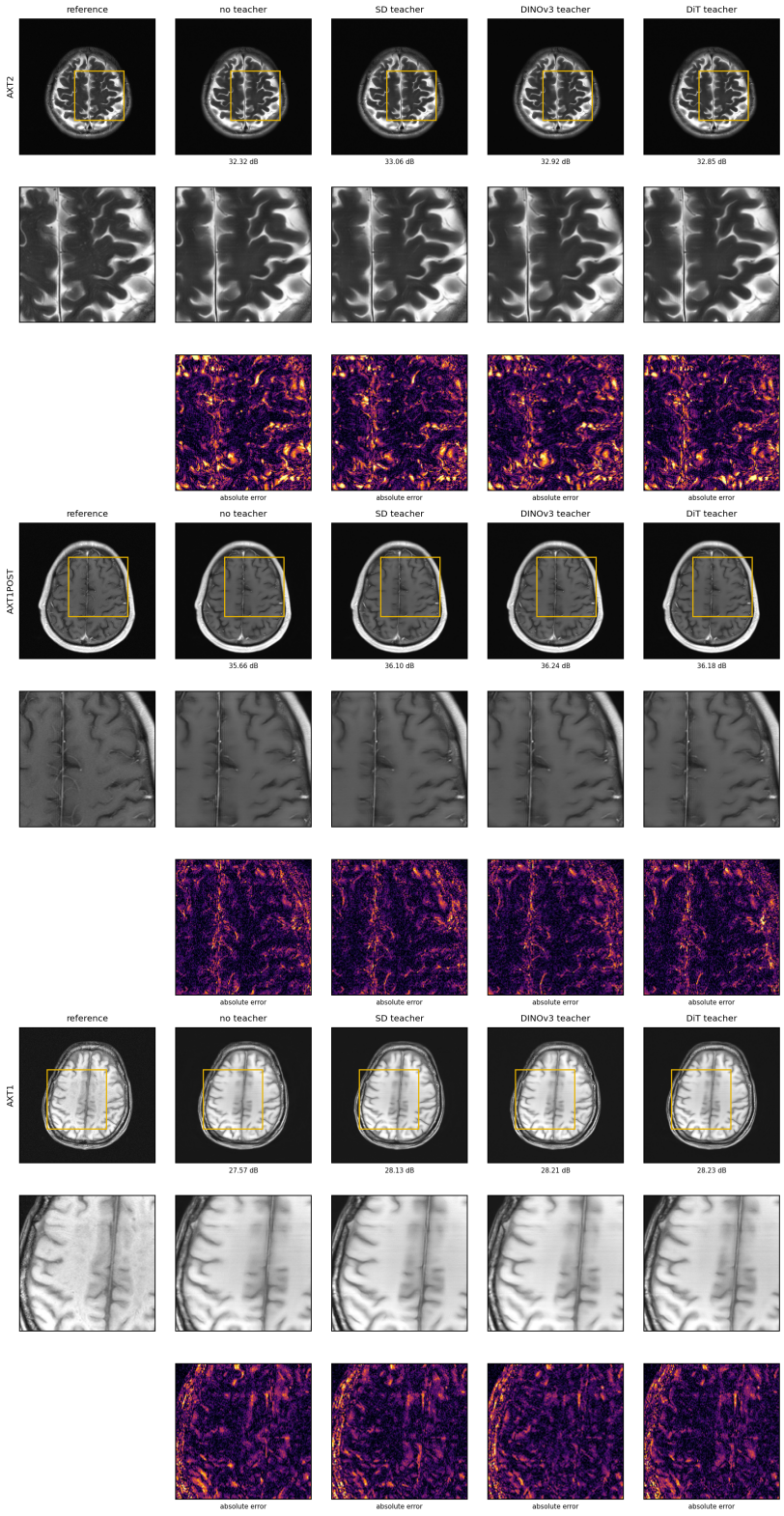

Figure 16: Multi-coil brain at $8\times$, E2E-VarNet without a teacher and with SD, DINOv3 and DiT after 300k iterations, one validation slice for each of three contrasts, so the PSNR values differ from those of Table 3.