LMO: Linear Mamba Operator for MRI Reconstruction

Supplementary Material

A. Supplementary Materials

A.1. Proof by Formulas

The derivation proof from Eq. (8) to Eq. (9) is intuitive, given as follows.

$$\begin{aligned} h(x) =& e^{Ax}\int_{-\infty}^{x} B(e^{-Ay})v(y)\mathrm{d}y, \\ h^{'}(x) =& Ae^{Ax}\int_{-\infty}^{x} B(e^{-Ay})v(y)\mathrm{d}y + e^{Ax}Be^{-Ax}v(x), \\ =& Ae^{Ax}\int_{-\infty}^{x} B(e^{-Ay})v(y)\mathrm{d}y + Bv(x), \\ =& Ah(x) + Bv(x). \end{aligned} \tag{15}$$

Below, we provide the derivation proof from Eq. (10) to Eq. (12). It can be deduced from Eq. (8) that

$$\begin{aligned} F(x) = e^{-Ax}h(x) = \int_{-\infty}^{x} & B(e^{-Ay})v(y)\mathrm{d}y \\ =& F(0) + \int_{0}^{x} B(e^{-Ay})v(y)\mathrm{d}y \\ =& h(0) + \int_{0}^{x} B(e^{-Ay})v(y)\mathrm{d}y. \end{aligned} \tag{16}$$

Continuing the derivation, we can obtain

$$\begin{aligned} e^{-Ax}h(x) =& h(0) + \int_{0}^{x} B(e^{-Ay})v(y)\mathrm{d}y, \\ h(x) =& e^{Ax}h(0) + e^{Ax}\int_{0}^{x} B(e^{-Ay})v(y)\mathrm{d}y. \end{aligned} \tag{17}$$

Assume x_k and x_{k+1} are two adjacent sampling points after discretization, we will show below how to go from $h(x_k)$ to $h(x_{k+1})$. Firstly, we provide their definitions below:

$$\begin{aligned} & h(x_k) = e^{Ax_k}h(0) + e^{Ax_k}\int_{0}^{x_k} B(e^{-Ay})v(y)\mathrm{d}y, \\ & h(x_{k+1}) = e^{Ax_{k+1}}h(0) + e^{Ax_{k+1}}\int_{0}^{x_{k+1}} B(e^{-Ay})v(y)\mathrm{d}y, \\ & = e^{A(x_k+(x_{k+1}-x_k))}h(0) \\ & + e^{A(x_k+(x_{k+1}-x_k))}\int_{0}^{x_{k+1}} B(e^{-Ay})v(y)\mathrm{d}y, \\ & = e^{A(x_{k+1}-x_k)}[e^{Ax_k}h(0) + e^{Ax_k}\int_{0}^{x_k} B(e^{-Ay})v(y)\mathrm{d}y] \\ & + e^{Ax_{k+1}}\int_{x_k}^{x_{k+1}} B(e^{-Ay})v(y)\mathrm{d}y, \\ & = e^{A(x_{k+1}-x_k)}h(x_k) + \int_{x_k}^{x_{k+1}} B(e^{A(x_{k+1}-y)})v(y)\mathrm{d}y. \end{aligned} \tag{18}$$

Assume $\Delta = x_{k+1} - x_k$ as the step length parameter for the time interval, then

$$h(x_{k+1}) = e^{A\Delta}h(x_k) + \int_{x_k}^{x_{k+1}} B(e^{A(x_{k+1}-y)})v(y)\mathrm{d}y. \tag{19}$$

According to the concept of zero-order hold, when Δ approaches 0, we can regard the value of function v in the interval $[x_k, x_{k+1}]$ as a constant $v(x_{k+1})$, then Eq. (19) can be written as:

$$\begin{aligned} h(x_{k+1}) \approx& e^{A\Delta}h(x_k) + \int_{x_k}^{x_{k+1}} e^{A(x_{k+1}-y)}\mathrm{d}yBv(x_{k+1}), \\ \approx& e^{A\Delta}h(x_k) + Bv(x_{k+1})e^{Ax_{k+1}}\int_{x_k}^{x_{k+1}} (e^{-Ay})\mathrm{d}y, \\ \approx& e^{A\Delta}h(x_k) + Bv(x_{k+1})\frac{1}{A}(e^{A(x_{k+1}-x_k)} - 1), \\ \approx& e^{A\Delta}h(x_k) + \Delta Bv(x_{k+1})\frac{1}{\Delta A}(e^{\Delta A} - 1). \end{aligned} \tag{20}$$

Let $\bar{\mathbf{A}} = e^{\Delta A}$, $\bar{\mathbf{B}} = (\Delta A)^{-1}(e^{\Delta A} - \mathbf{I}) \cdot \Delta B$, then:

$$h(x_{k+1}) = \bar{\mathbf{A}}h(x_k) + \bar{\mathbf{B}}v(x_{k+1}). \tag{21}$$

Lastly, with Eq. (8) combined, we can obtain:

$$\begin{aligned} h(x_{k+1}) =& \bar{\mathbf{A}}h(x_k) + \bar{\mathbf{B}}v(x_{k+1}), \\ \mathcal{K}(v)(x_{k+1}) =& Ch(x_{k+1}). \end{aligned} \tag{22}$$

This is easy from Eq. (22) to Eq. (12).

A.2. Experimental Setup

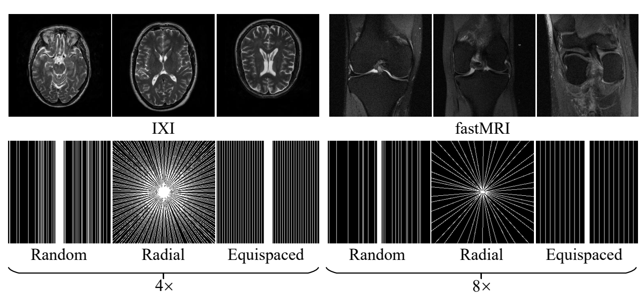

Figure 6. Representative MRI images used in our experiments. The first row showcases some IXI and fastMRI samples, while the second row exhibits images at $\times 4$ and $\times 8$ acceleration levels as well as three distinct mask types.

Datasets: For single-channel MRI evaluation, two MRI datasets, IXI [†] and knee fastMRI[57] are adopted to evaluate

[†]https://brain-development.org/ixi-dataset/

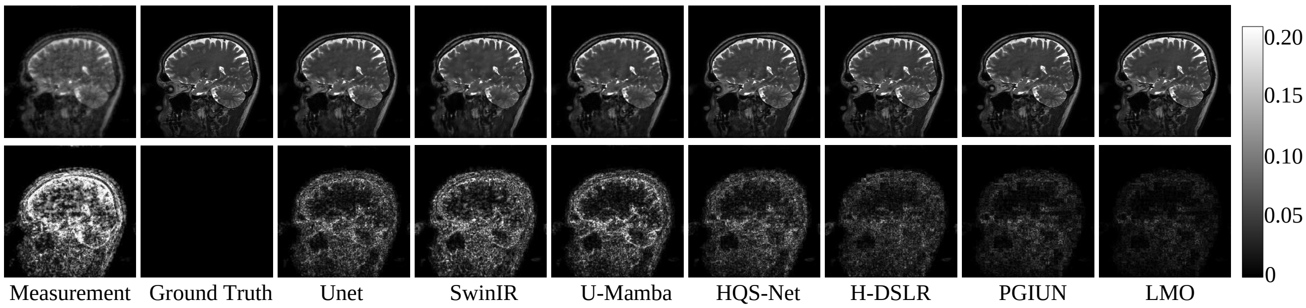

Figure 7. Visual results of multi-coil recovery with $\times 6$ AR of random mask.

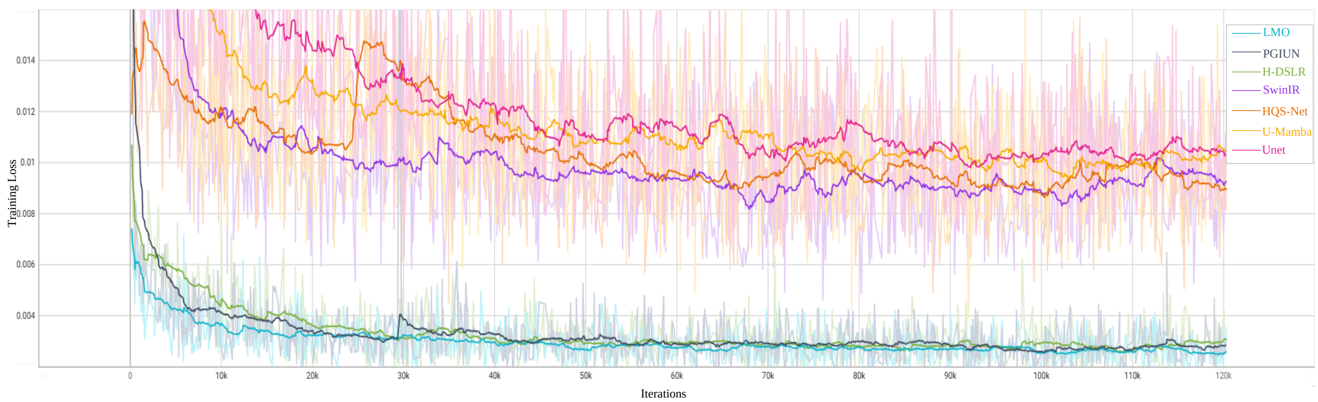

Figure 8. Loss curves during different model training on IXI dataset with $\times 4$ AR of radial mask.

the clinical efficacy of the proposed method. IXI comprises of 578 registered T2 images, each with a size of 256×256. For fastMRI dataset, 588 volumes of fat-suppressed proton density (FSPD) weighted images are selected, each with a size of 320×320. We split the dataset into training, validation, and testing sets with a ratio of $7 : 1 : 2$. In addition, we employ three different k-space undersampling masks, i.e., 1D Cartesian with random fraction, 1D Cartesian with equispaced fraction, and 2D radial, with acceleration rates (ARs) of $4\times$ and $8\times$ in our experiments.

For multi-channel MRI evaluation, a brain dataset [1] is employed, which includes fully sampled multi-coil images from five volunteers. Data from the first four participants are used for training, while those from the fifth participant are split into validation and test sets. Specifically, the dimensions of training, validation, and testing data are $12 \times 360 \times 256 \times 232$, $12 \times 100 \times 256 \times 232$, and $12 \times 64 \times 256 \times 232$, respectively. Note that the dimensions are sequentially denoted as coils, slices, width, and height. Coil sensitivity maps are estimated using ESPIRiT [45] from the central k-space region of each slice. Several representative samples and masks used in the main texts are showcased in Fig. 6.

Implementation details: All experiments are coded using the Pytorch framework on an NVIDIA GeForce RTX 3090 GPU. We employ Charbonnier loss and Adam optimizer with parameters ($\beta_1 = 0.9, \beta_2 = 0.999$) for model update. Besides, the initial learning rate, the batch size, and the epochs are set as 10^{-4}, 1, and 100, respectively.

Competing methods: Six previously reported state-of-the-art methods are selected, comprising of three deep unfolding methods including HQS-Net [53], H-DSLR [37], and PGIUN [17], as well as three purely deep learning-based methods including Unet [57], SwinIR [29] and U-Mamba [40]. For all competitors, the parameter configurations suggested by the original authors are employed for a fair comparison.

A.3. More visual results on multi-coil recovery

Fig. 7 provides the most complete comparisons, with other competing methods introduced. Evidently, the error maps from the additionally added competitors all suffer from much more scattered points and dramatic changes, demonstrating the inferior recovery performance compared to H-DSLR, PGIUN, and LMO.

A.4. The comparisons on running efficiency

In the main texts, we have claimed that our LMO enjoys only $O(n)$ time complexity. Here, we further provide the decreasing loss curves, as illustrated in Fig. 8, showing

Metrics	Unet	SwinIR	U-Mamba	HQS-Net	H-DSLR	PGIUN	LMO
GPU memory	1374M	12896M	2194M	2228M	4950M	9492M	4705M
Epochs	100	100	100	100	100	100	100
Training time	1.71h	25.23h	9.39h	6.16h	7.29h	8.06h	8.11h

Table 6. The comparisons on training efficiency, including memory, epochs, and ellapsed time (IXI-Radial-$\times$4).

Dataset	Batchsize	Epochs	Loss	η	γ	ω	Inputs
IXI	1	100	Charbonnier	0.0001	0.98	1e-06	Sampled Signal+Mask
fastMRI	1	100	Charbonnier	0.0001	0.98	1e-10	Sampled Signal+Mask
Multi-Coil Recovery	1	100	Charbonnier	0.0001	0.98	1e-06	Sampled Signal+Mask+Coil Sensitivity Maps

Table 7. Hyperparameter setting for model training. η, γ, and ω represent the learning rate, weight decay, and scheduler gamma, respectively. 'Loss' is short for loss function during training stage.

the actual training process of all competing models. Our first observation is that H-DSLR, PGIUN, and LMO share a similar convergence level, not only presenting faster dropping trends, but also enjoying fewer error fluctuations. With a deeper inspection, we can further find that LMO achieves lower errors than H-DSLR and PGIUN, confirming closely to the numerical results shown in Table 1.

Additionally, some other efficiency metrics, including GPU memory, used epochs, and training time, are also ablated on the IXI dataset, as shown in Table 6. As seen, Unet and SwinIR occupy the two extremes among all competing methods. While the former enjoys the most efficient training stage, the latter suffers from the heaviest computational burden. Our LMO ranks at the middle place in these regards. Compared to U-Mamaba that also adopts the state space model, our proposal is limited by more parameters due to the combination of both global and local integrals. However, in terms of training time, LMO is about one hour ahead of U-Mamba. In contrast, compared to PGIUN that performs the second best in most experiments, our proposed model enjoys nearly half the parameters, but shares a similar level of training time. Note that all results are achieved with hyperparameters set as in Table 7. Most of the values are constant in different experiments, with partial of them fine-tuned in specific cases. For other competing methods, we use the optimal parameters mentioned in the original papers to ensure best practice.

A.5. Statistical analysis

Table 8 showcases the numerical results that have been already presented in Table 1, yet with the p-values added. It can be seen that the differences in the experimental results are statistically significant, indicating that these results are not due to random fluctuations. However, such a table is too redundant, so we have omitted the statistical comparison in the main text of the paper. Note again that each experimental metric is obtained from 10 repeated runs.

A.6. More ablation experiments

Effect of different kernel integrals: The effects from differing amounts of our elaborated kernel integrals were scrutinized during the creation of LMO, with the results given in Table 9. As evidenced, the more kernel integrals evidently promote the higher model performance, yet resulting in more parameters and computational burden. Among the options of $\{4, 5, 6, 7\}$, the final selection of six kernel integrals is considered with a better balance between efficiency and efficacy, hence selected as the default value.

Moreover, to better demonstrate the data consistency of our scanning plus convolution integration, we have compared the performance of LMO with the classical neural operators, including FNO [25] and CNO [39], for MRI reconstruction. The results are given in Table 10. For fairness, the number of integration layers has been regulated to ensure that all three models share the same level of parameters. It can be seen that FNO and CNO lag significantly behind our LMO. This is because they are not specifically designed for MRI tasks. Directly transferring these models may overlook critical information in MRI signal reconstruction, resulting in a lack of data consistency.

Effect of the unfolding directions: In our Scan Unfolding (SU) and Scan Merging (SM) module, we use four different directional unfoldings for the function. To evaluate the specific impact of different direction selections on model performance, the ablation results are shown in Table 11, in which the variable $direction$ is used to represent the number of unfolding directions in the experiments. Evidently, it is observed that although with more parameters and FLOPs required, the value of $direction = 4$ leads the efficacy by a large margin, hence suggested as our default selection.

Effect of the latent space dimension: To explore the

AR	Methods	IXI									fastMRI								
		Random			Radial			Equispaced			Random			Radial			Equispaced		
		PSNR	SSIM	P-Value	PSNR	SSIM	P-Value	PSNR	SSIM	P-Value	PSNR	SSIM	P-Value	PSNR	SSIM	P-Value	PSNR	SSIM	P-Value
×4	Unet	31.28	0.954	<0.001/<0.001	34.03	0.935	<0.001/<0.001	30.22	0.946	<0.001/<0.001	27.96	0.811	<0.001/<0.001	28.69	0.830	<0.001/<0.001	27.26	0.780	<0.001/<0.001
	SwinIR	32.51	0.962	<0.001/<0.001	35.57	0.940	<0.001/<0.001	31.30	0.951	<0.001/<0.001	28.45	0.822	<0.001/<0.001	29.50	0.840	<0.001/<0.001	28.12	0.794	<0.001/<0.001
	U-Mamba	32.10	0.958	<0.001/<0.001	34.03	0.930	<0.001/<0.001	30.92	0.945	<0.001/<0.001	28.17	0.813	<0.001/<0.001	28.93	0.833	<0.001/<0.001	27.57	0.782	<0.001/<0.001
	HQS-Net	32.49	0.948	<0.001/<0.001	35.14	0.969	<0.001/<0.001	30.34	0.942	=0.001/<0.001	28.57	0.819	<0.001/<0.001	29.32	0.839	<0.001/<0.001	27.82	0.787	<0.001/<0.001
	H-DSLR	36.01	0.982	<0.001/<0.001	45.31	0.994	<0.001/<0.001	33.65	0.968	<0.001/<0.001	29.04	0.834	<0.001/<0.001	30.23	0.866	<0.001/<0.001	28.25	0.799	<0.001/<0.001
	PGIUN	37.98	0.985	<0.001/<0.001	47.09	0.994	<0.001/<0.001	35.51	0.978	<0.001/<0.001	30.02	0.850	<0.001/<0.001	30.98	0.876	<0.001/<0.001	28.55	0.809	<0.001/<0.001
	LMO	39.14	0.986	-	48.16	0.996	-	35.70	0.980	-	30.17	0.853	-	31.11	0.883	-	28.65	0.832	-
×8	Unet	29.06	0.932	<0.001/<0.001	29.86	0.890	<0.001/<0.001	27.91	0.922	<0.001/<0.001	26.38	0.754	<0.001/<0.001	26.42	0.723	<0.001/<0.001	26.21	0.745	<0.001/<0.001
	SwinIR	30.13	0.947	<0.001/<0.001	29.79	0.891	<0.001/<0.001	28.54	0.925	<0.001/<0.001	27.41	0.768	<0.001/<0.001	27.88	0.742	<0.001/<0.001	27.26	0.752	<0.001/<0.001
	U-Mamba	29.88	0.935	<0.001/<0.001	29.31	0.879	<0.001/<0.001	28.06	0.923	<0.001/<0.001	26.89	0.758	<0.001/<0.001	27.75	0.765	<0.001/<0.001	26.89	0.748	<0.001/<0.001
	HQS-Net	28.70	0.923	<0.001/<0.001	29.02	0.921	<0.001/<0.001	27.40	0.909	<0.001/<0.001	26.64	0.755	=0.001/<0.001	27.82	0.764	<0.001/<0.001	26.42	0.747	<0.001/<0.001
	H-DSLR	31.96	0.955	=0.001/<0.001	34.50	0.971	<0.001/<0.001	29.80	0.938	<0.001/<0.001	27.04	0.762	<0.001/<0.001	27.28	0.741	<0.001/=0.001	26.84	0.752	<0.001/<0.001
	PGIUN	34.15	0.970	<0.001/<0.001	36.21	0.980	<0.001/<0.001	31.74	0.957	<0.001/<0.001	27.83	0.783	<0.001/<0.001	28.01	0.791	<0.001/<0.001	27.41	0.768	<0.001/<0.001
	LMO	34.26	0.971	-	36.53	0.983	-	31.80	0.960	-	27.88	0.808	-	28.29	0.837	-	27.51	0.787	-

Table 8. The numerical results under three masks as well as $\times$ 4 and $\times$ 8 ARs. P $<$ 0.001 denotes a statistically significant level.

KIs	PSNR	SSIM	Params	FLOPs
7	48.40	0.997	2.07M	135.84G
6	48.16	0.996	1.78M	116.44G
5	47.52	0.995	1.48M	97.04G
4	46.57	0.993	1.18M	77.65G

Table 9. The results from different kernel integral (KI) numbers. Our choices are marked in green (IXI-Radial-×4).

Methods	PSNR	SSIM	Params	FLOPs
FNO	27.37	0.723	1.77M	87.54G
CNO	27.96	0.773	1.78M	93.94G
LMO	48.16	0.996	1.78M	116.44G

Table 10. Numerical results of FNO, CNO, and LMO for MRI reconstruction (IXI-Radial-×4).

SU&SM Directions	PSNR	SSIM	Params	FLOPs
$direction = 1$	37.48	0.934	1.43M	87.33G
$direction = 2$	43.56	0.981	1.55M	97.80G
$direction = 4$	48.16	0.996	1.78M	116.44G

Table 11. The results by using different directions in SU and SM. Our choices are marked in green (IXI-Radial-×4).

Dimensions	PSNR	SSIM	Params	FLOPs
32	36.44	0.982	0.45M	29.23G
48	42.58	0.990	1.00M	65.59G
64	48.16	0.996	1.78M	116.44G
96	48.26	0.996	3.99M	261.64G
128	48.32	0.997	7.09M	464.81G

Table 12. The ablation results within different latent dimensions. The final choice is colored green (IXI-Radial-×4).

practical efficacy of the dimension in which kernel integration is performed, we have provided the ablation results as shown in Table 12. Note "Dimension" represents the value that original function is lifted. Through kernel integration learnt in this higher dimensional space, a concerned projection operation is then conducted to reconstruct the target signal. The lifting and projection modules are composed of convolution layers. As seen, a higher dimension enforces the model learning more enriched features, further leading to a higher performance. However, when the dimension is beyond 64, the performance gain is limited. Taking efficiency into account, we finally choose 64 as the default value.

A.7. The numerical results with difference degree

Due to space limitations, the changes in numerical increments corresponding to Tables 1 and 4 were originally not provided in the main texts, which are now added in Tables 13, 14, and 15. It can be observed that our LMO method wins the others in all cells. Furthermore, in generalization experiments with regard to Table 15, LMO again achieves the best results. When transferring the model trained at ×8 to other scales, LMO was the only one that shows an increase in performance, while all others experienced a sharp decline in performance to varying degrees.

AR	Methods	IXI					
		Random		Radial		Equispaced	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
×4	Unet	31.28▼20.1%	0.954▼3.2%	34.03▼29.3%	0.935▼6.1%	30.22▼15.3%	0.946▼3.5%
	SwinIR	32.51▼16.9%	0.962▼2.4%	35.57▼26.1%	0.940▼5.6%	31.30▼12.3%	0.951▼3.0%
	U-Mamba	32.10▼18.0%	0.958▼2.8%	34.03▼29.3%	0.930▼6.6%	30.92▼13.4%	0.945▼3.6%
	HQS-Net	32.49▼17.0%	0.948▼3.9%	35.14▼27.0%	0.969▼2.7%	30.34▼15.0%	0.942▼3.9%
	H-DSLR	36.01▼8.0%	0.982▼0.4%	45.31▼5.9%	0.994▼0.2%	33.65▼5.7%	0.968▼1.2%
	PGIUN	37.98▼3.0%	0.985▼0.1%	47.09▼2.2%	0.994▼0.2%	35.51▼0.5%	0.978▼0.2%
	LMO (Ours)	39.14	0.986	48.16	0.996	35.70	0.980
×8	Unet	29.06▼15.2%	0.932▼4.0%	29.86▼18.3%	0.890▼9.5%	27.91▼12.2%	0.922▼4.0%
	SwinIR	30.13▼12.0%	0.947▼2.5%	29.79▼18.4%	0.891▼9.3%	28.54▼10.3%	0.925▼3.6%
	U-Mamba	29.88▼12.8%	0.935▼3.7%	29.31▼19.8%	0.879▼10.6%	28.06▼11.8%	0.923▼3.9%
	HQS-Net	28.70▼16.2%	0.923▼4.9%	29.02▼20.5%	0.921▼6.3%	27.40▼13.8%	0.909▼5.3%
	H-DSLR	31.96▼6.7%	0.955▼1.6%	34.50▼5.5%	0.971▼1.2%	29.80▼6.3%	0.938▼2.3%
	PGIUN	34.15▼0.3%	0.970▼0.1%	36.21▼0.9%	0.980▼0.3%	31.74▼0.2%	0.957▼0.3%
	LMO (Ours)	34.26	0.971	36.53	0.983	31.80	0.960

Table 13. Numerical comparisons of IXI dataset under three masks as well as × 4 and × 8 ARs.

AR	Methods	fastMRI					
		Random		Radial		Equispaced	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
×4	Unet	27.96▼7.3%	0.811▼4.9%	28.69▼7.8%	0.830▼6.0%	27.26▼4.9%	0.780▼6.2%
	SwinIR	28.45▼5.7%	0.822▼3.6%	29.50▼5.2%	0.840▼4.9%	28.12▼1.9%	0.794▼4.6%
	U-Mamba	28.17▼6.6%	0.813▼4.7%	28.93▼7.0%	0.833▼5.7%	27.57▼3.8%	0.782▼6.0%
	HQS-Net	28.57▼5.3%	0.819▼4.0%	29.32▼5.7%	0.839▼5.0%	27.82▼2.9%	0.787▼5.4%
	H-DSLR	29.04▼3.8%	0.834▼2.2%	30.23▼2.8%	0.866▼1.9%	28.25▼1.4%	0.799▼4.0%
	PGIUN	30.02▼0.5%	0.850▼0.4%	30.98▼0.4%	0.876▼0.8%	28.55▼0.4%	0.809▼2.8%
	LMO (Ours)	30.17	0.853	31.11	0.883	28.65	0.832
×8	Unet	26.38▼5.4%	0.754▼6.7%	26.42▼6.6%	0.723▼13.6%	26.21▼4.7%	0.745▼5.3%
	SwinIR	27.41▼1.7%	0.768▼5.0%	27.88▼1.4%	0.742▼11.4%	27.26▼0.9%	0.752▼4.4%
	U-Mamba	26.89▼3.5%	0.758▼6.2%	27.75▼1.9%	0.765▼8.6%	26.89▼2.3%	0.748▼5.0%
	HQS-Net	26.64▼4.4%	0.755▼6.6%	27.82▼1.7%	0.764▼8.7%	26.42▼4.0%	0.747▼5.1%
	H-DSLR	27.04▼3.0%	0.762▼5.7%	27.28▼3.6%	0.741▼11.5%	26.84▼2.4%	0.752▼4.4%
	PGIUN	27.83▼0.2%	0.783▼3.1%	28.01▼1.0%	0.791▼5.5%	27.41▼0.7%	0.768▼2.4%
	LMO (Ours)	27.88	0.808	28.29	0.837	27.51	0.787

Table 14. Numerical comparisons of fastMRI dataset under three masks as well as × 4 and × 8 ARs.

SG	LMO		PGIUN		H-DSLR	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
×4 to ×4	48.16	0.996	47.09	0.994	45.31	0.994
×4 to ×6	43.76▼9.1%	0.989▼0.7%	41.06▼12.8%	0.982▼1.3%	29.80▼34.2%	0.835▼16.0%
×4 to ×8	30.25▼37.2%	0.860▼13.7%	28.31▼39.9%	0.826▼17.0%	24.10▼46.8%	0.746▼24.9%
×8 to ×4	**43.50**▲18.8%	0.988▲0.5%	*29.80*▼17.7%	0.941▼4.0%	*29.20*▼14.4%	0.754▼22.1%
×8 to ×6	**41.98**▲14.6%	0.985▲0.2%	*32.58*▼10.0%	0.963▼1.7%	*31.50*▼7.7%	0.850▼12.2%
×8 to ×8	**36.53**	0.983	*36.21*	0.980	*34.50*	0.971

Table 15. The generalization comparisons across different scales. Bold and italic denote different performance trends.