A Unified Model for Compressed Sensing MRI Across Undersampling Patterns

Supplementary Material

In the supplementary, we first present more details of the proposed U-shaped DISCO Neural Operator (UDNO, in Section A), a main building block of $\mathrm{NO}_\mathbf{i}$ and $\mathrm{NO}_\mathbf{k}$ of our framework. We then provide more details of the machine learning framework implementation (Section B) as well as additional numerical results of the multi-pattern and multi-rate undersampling experiments (Section C). In Section D we include additional ablation and analysis, on comparing CNN and NO kernels and their performance under the same parameter size, followed by details about DISCO and the justification of its basis choice in Section E.

A. UDNO Architecture

The motivation behind the U-shaped architecture is to capture multi-scale features by integrating high-level contextual information with low-level details. Its encoder-decoder structure, enhanced by skip connections, enables precise localization of features even with limited annotated data. In our approach, we extend this idea through UDNO, which is applied to both the physical and frequency domains for MRI reconstruction—unlike methods such as FNO [24] used for PDE data that incorporate a frequency cut. This difference arises because PDE data typically comes from smooth functions, where low frequencies are dominant and high frequencies mainly represent noise. In contrast, imaging data benefits from retaining both low-frequency information and high-frequency details (e.g., edges).

We provide additional details of the proposed UDNO (U-Shaped DISCO Neural Operator) architecture. Fig. 7 depicts the overall architecture, which mimics the U-Net [37]. We use the updated implementation of the U-Net in [40]. Our network architecture has two differences. First, all traditional convolutions are replaced with their DISCO counterparts. Second, transpose convolutions are replaced by an interpolation upsampling step, followed by a DISCO2d layer, InstanceNorm layer, and LeakyReLU activation. DISCO2d layers function as drop-in replacements for traditional 2d convolution layers. They do not change the spatial dimension of the input. The UDNO is an end-to-end *neural operator*.

As in the traditional U-Net [37], each encoder block halves the spatial dimensions and doubles the feature channels. Each decoder step (upsampling + decoder) doubles the spatial dimensions and halves the feature channels. Skip connections are included, as in the original architecture. All components of the UDNO operate in the function space and are not tied to a specific discretization, thus making the model an *end-to-end* neural operator.

B. Additional Implementation Details

B.1. Undersampling Configurations

We summarize the configurations of different CS-MRI undersampling rates in Table 5 and undersampling patterns in Fig. 8.

B.2. Learning Sensitivity Maps for Multi-Coil MRI

In MRI reconstruction, the sensitivity map S_i for the i^{th} coil is needed for coil reductions and expansions. Inspired by [40], we use a UDNO with 4 encoder/decoder steps, 8 hidden channels, 0.02 DISCO radius (assuming the domain is $[-1, 1]^2$), and the kernel basis from [26] with 1 isotropic basis and 5 anisotropic basis rings, each containing 7 basis functions. We use this UDNO to predict the sensitivity map S_i from the input coil measurement $\mathbf{k}_i$. We then follow [40] to combine multiple coils weighted by the corresponding learned sensitivity maps.

B.3. UDNO and DISCO Implementation Details

Both $\mathrm{NO}_\mathbf{k}$ and $\mathrm{NO}_\mathbf{i}$ use DISCO layers using the *linear-piecewise* kernel basis from [26] with 1 isotropic basis and 5 anisotropic basis rings, each containing 7 basis functions. The $\mathrm{NO}_\mathbf{k}$ (measurement space neural operator) is implemented as a UDNO with 2 input and output channels, 16 hidden channels, and 4 depth (encoder/decoder steps). $\mathrm{NO}_\mathbf{k}$ DISCO $\mathrm{NO}_\mathbf{i}$ have a radius cutoff of 0.02. The $\mathrm{NO}_\mathbf{i}$ (image-space neural operator) is implemented as a UDNO with 2 input and output channels, 18 hidden channels, and 4 encoder/decoder steps. $\mathrm{NO}_\mathbf{i}$ DISCO kernels have a radius cutoff of 0.02 with the same internal basis shape. We train both our model and the baseline with SSIM loss, and 0.0003 learning rate.

To compare the choice of basis function (piecewise linear, Zernike, and Morlet), we train our neural operator with a single cascade on a 30% subset of the fastMRI knee dataset for 15 epochs. We find that empirically, the piecewise linear basis outperforms both the Zernike and Morlet bases by at least 3 PSNR. All kernels have a similar number of parameters. Results are provided in Table 4.

B.4. Baseline Hyperparameter Search Details

For the diffusion baseline CSGM, we tuned `step_lr` and `mse` parameters in their official github repo) using Bayesian optimization. The search algorithm was run on 6 representative images outside of the test set for around 50 iterations with the search space defined in Table 9. For E2E-VN baselines, we tune the number of layers in each cascade, learning rate and schedule.

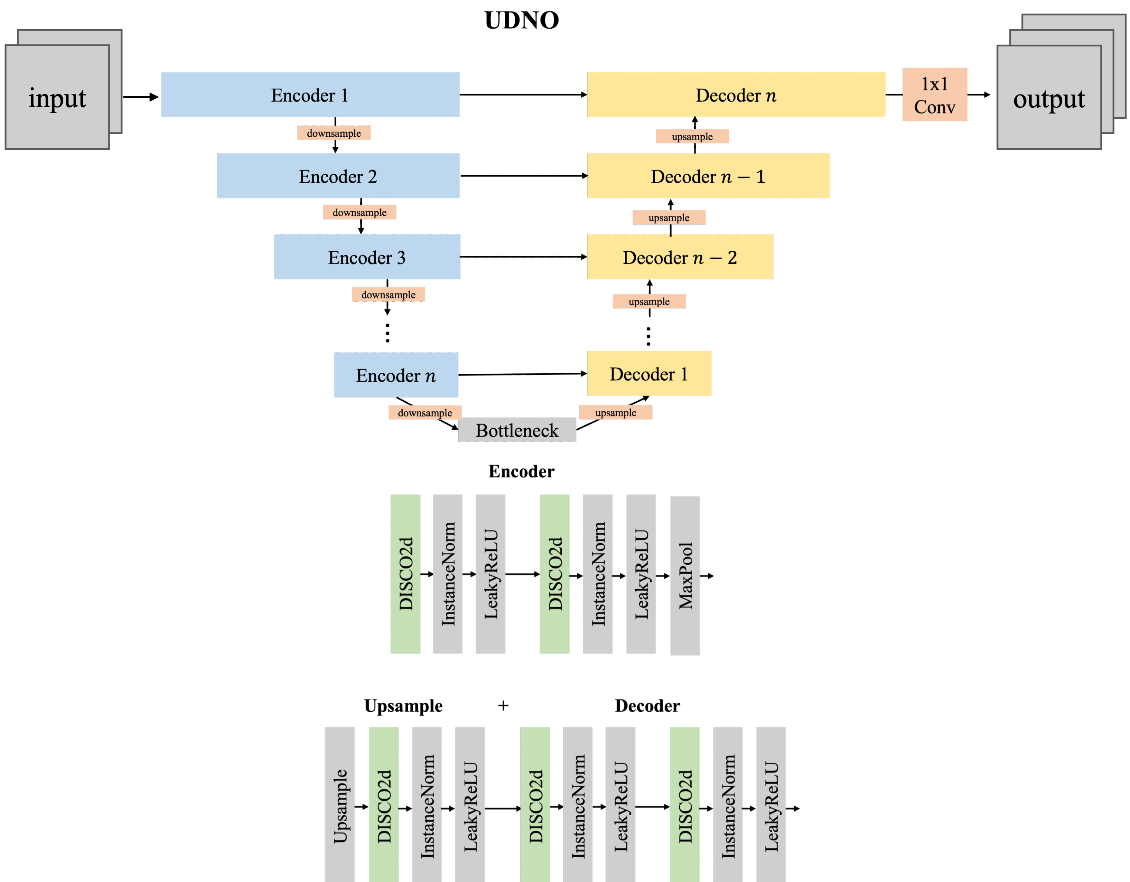

Figure 7. **UDNO architecture**. We propose a U-shaped neural operator (UDNO) to capture multi-scale features of the input. The UDNO uses discrete-continuous convolutions (DISCOs) [31] as the local integral operator. The final 1x1 convolution allows the module to flexibly project to the desired number of output channels and is resolution invariant by virtue of being a pointwise operation. The UDNO is an end-to-end *neural operator*.

Kernel Basis	PSNR ↑	SSIM ↑	NMSE ↓
Piecewise Linear	**36.125**	**0.884**	**0.009**
Zernike	32.983	0.838	0.017
Morlet	32.755	0.835	0.018

Table 4. **Kernel basis experiment results.** We train our neural operator model with the piecewise-linear, Zernike, and Morlet bases, comparing empirical reconstruction results. The Piecewise Linear basis outperforms both the Zernike and Morlet by at least 3 PSNR.

Alias	Acceleration rate	Center fraction rate
16×	16	0.02
8×	8	0.04
6×	6	0.06
4×	4	0.08

Table 5. **k space undersampling configurations** (acceleration and center fraction parameters) used for MRI experiments. We follow the [40] and [40]

C. Additional Results Across Undersampling Patterns and Rates

We summarize the numerical results of the performance of the proposed neural operator (NO) and the End-to-End VarNet baseline [40] across different undersampling patterns and rates on the fastMRI [44] knee and brain dataset.

fastMRI Knee. Results for multiple patterns are in Table 2 of the paper and those for multiple rates are in Table 6.

fastMRI Brain. Results for multiple patterns are in Table 7 and those for multiple rates are in Table 8.

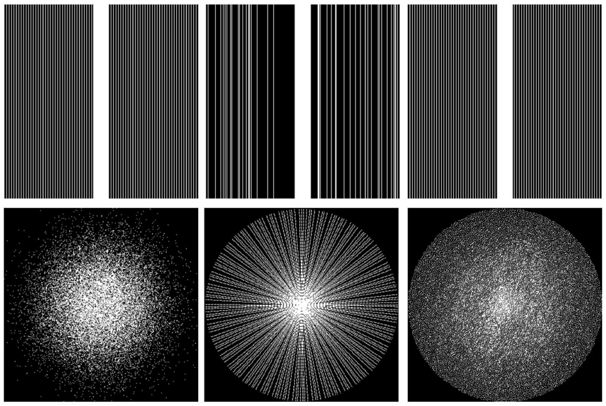

Figure 8. **Undersampling mask patterns**. The visualized patterns are all for the 4× acceleration rate. **Top:** Rectilinear patterns: Equispaced, Random, Magic. **Bottom:** Irregular patterns: Gaussian, Radial, Poisson.

	NO (ours)		E2E-VN [40]	
Rate	**PSNR ↑**	**SSIM ↑**	**PSNR ↑**	**SSIM ↑**
4×	37.215 ± 2.466	0.897 ± 0.071	$\mathbf{38.329 \pm 3.062}$	$\mathbf{0.905 \pm 0.073}$
6×	$\mathbf{35.452 \pm 2.150}$	$\mathbf{0.872 \pm 0.073}$	32.770 ± 2.064	0.851 ± 0.069
8×	$\mathbf{33.598 \pm 1.892}$	$\mathbf{0.848 \pm 0.071}$	28.346 ± 2.407	0.780 ± 0.062
16×	$\mathbf{29.241 \pm 2.402}$	$\mathbf{0.779 \pm 0.070}$	23.181 ± 3.558	0.629 ± 0.090

Table 6. **fastMRI Knee performance across different undersampling rates.** We compare our NO model's knee reconstruction performance to the E2E-VN [40], assessing for robustness against different undersampling rates. Both models are trained on equispaced 4× knee samples, and evaluated across 4×, 6×, 8×, and 16× equispaced validation samples. Notice that over the irregular patterns, our model shows an increase of 3.22 dB PSNR and 5.8% SSIM.

	NO (ours)		E2E-VN [40]	
Pattern	**PSNR ↑**	**SSIM ↑**	**PSNR ↑**	**SSIM ↑**
Equispaced	37.106 ± 1.646	0.952 ± 0.010	38.063 ± 2.701	0.962 ± 0.011
Random	36.051 ± 1.665	0.945 ± 0.015	37.025 ± 2.187	0.957 ± 0.010
Magic	38.270 ± 1.985	0.960 ± 0.011	38.463 ± 2.967	0.965 ± 0.011
Radial	36.498 ± 1.792	0.948 ± 0.015	25.225 ± 2.126	0.722 ± 0.063
Poisson	33.936 ± 2.047	0.924 ± 0.020	22.117 ± 1.487	0.670 ± 0.046
Gaussian	32.725 ± 2.004	0.910 ± 0.018	25.283 ± 3.336	0.730 ± 0.098

Table 7. **fastMRI Brain performance across different undersampling patterns.** We compare our NO model's brain reconstruction performance to the E2E-VN [40], assessing for robustness against different undersampling patterns. Both models are trained on equispaced 4× brain samples, and evaluated across multiple patterns. Notice that over the irregular patterns, our model shows a significant 10 dB PSNR and 22% SSIM improvement on average. Our NO model is robust to different patterns, while the E2E-VN overfits to the rectilinear patterns (equispaced, random, magic).

D. Additional Ablation Studies and Analysis

Rescaling CNN Kernel Size for Consistent Ratio. As illustrated in Fig. 1b, CNNs have inconsistent kernel size to image size ratio when image resolution changes. We want to compare NO kernels, parameterized in the function space, with CNN kernels by eliminating the factor of kernel size ratio with CNN kernel interpolation. In this ablation study, we manually resize the CNN kernel in [40] with bilinear interpolation to make its relative kernel size consistent for different resolutions and call it E2E-VN-INTERP. We compare its performance with the NO. Specifically, in a super-resolution experiment as follows, we train both our NO and the E2E-VN-INTERP on 320 × 320 equispaced 4× knee samples, with a similar setting as in Section 4.4 (NO_i MRI higher-resolution experiment). Then, we perform zero-shot inference on higher resolution 640 × 640 samples in image space.

Our NO model leverages DISCO convolutions, which

	NO (ours)		E2E-VN [40]	
Rate	**PSNR ↑**	**SSIM ↑**	**PSNR ↑**	**SSIM ↑**
4×	36.851 ± 2.334	0.952 ± 0.027	38.294 ± 3.030	0.959 ± 0.027
6×	34.575 ± 2.404	0.934 ± 0.030	33.418 ± 2.675	0.925 ± 0.030
8×	32.246 ± 2.306	0.912 ± 0.031	28.827 ± 3.197	0.856 ± 0.047
16×	27.561 ± 2.620	0.836 ± 0.044	22.694 ± 3.097	0.594 ± 0.104

Table 8. **fastMRI Brain performance across different undersampling rates.** Comparisons of the reconstruction quality of our NO model with the E2E-VN [40] across various undersampling rates demonstrate that our model maintains robustness at higher undersampling rates and the E2E-VN shows significant degradation in both metrics, particularly at extreme undersampling (e.g., 16×).

Parameter	Lower bound	Upper bound	Distribution
`step_lr`	10^{-5}	10^{-4}	Log-uniform
`mse`	0	5	Uniform

Table 9. Hyperparameter search space for the diffusion baseline.

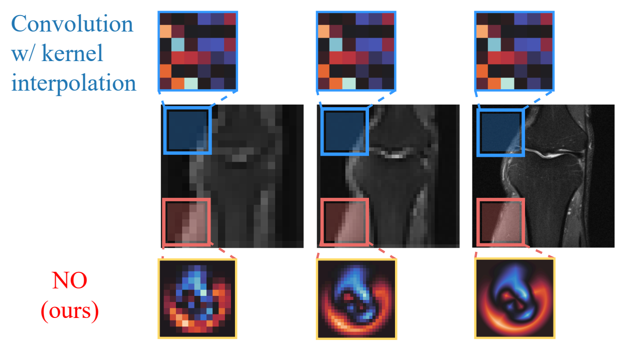

Figure 9. **Ablation study: consistent kernel size to image size ratio for both CNNs and NOs**. As illustrated in Fig. 1b, CNNs have inconsistent relative kernel size when image resolution changes. In this ablation study, we manually resize the CNN kernel with bilinear interpolation to make its relative kernel size consistent for different resolutions and compare the performance with the NO.

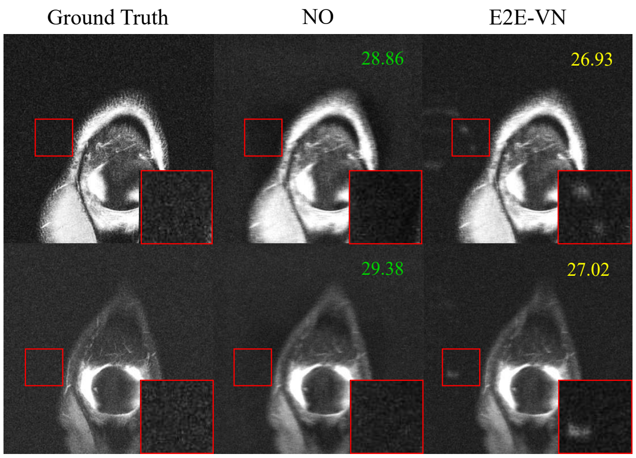

Figure 10. **Zero-shot inference on higher-resolution samples (NO vs. E2E-VN with interpolated kernels).** While both models are able to recover overall structure, notably, the E2E-VN suffers from hallucinations and noise artifacts in the area surrounding the subject's knee.

enable zero-shot inference on arbitrary resolutions, making them inherently resolution-agnostic (Fig. 1a). In contrast, traditional CNN kernels are designed for fixed resolutions. For instance, the original 3×3 kernels of the E2E-VN model, backed up by CNNs, cannot directly scale to the larger 640×640 inference resolution. One approach to address this is by resizing the learned kernels to 6×6 with bilinear interpolation while preserving their norms, as we follow [26] and use quadrature weights to perform the integration. We adopt this method, comparing our NO model with the kernel-scaled E2E-VN-INTERP model. Side-by-side visualization results are presented in Fig. 10, where we observe a slightly worse reconstruction performance in the background region of E2E-VN-INTERP compared to VN. Also, E2E-VN-INTERP outperforms the E2E-VN with inconsistent kernel size, validating the need to keep a consistent relative kernel size.

Performance Under Same Parameter Size. Additionally, we conduct an experiment comparing the NO and E2E-VN models, ensuring both have an identical number of trainable parameters (21.7M). Both models are pretrained on $4\times$ equispaced fastMRI brain samples for 10 epochs. Then, both are trained for an additional epoch, in which they see samples from all patterns together. We plot cross-pattern performance in Fig. 11.

Functions of $\mathbf{NO_k}$. We perform an ablation study of our $NO_{\mathbf{k}}$ module, training both models on a small subset of the full $4\times$ equispaced training set and plot zero-shot SSIM scores across all patterns. The $NO_{\mathbf{k}}$ increases zero-shot SSIM by 5.3% across irregular patterns (Fig. 12).

FLOPs of models. We measure the number of forward passes and GFLOPs required in a single inference in Table 10. Notice that diffusion requires multiple forward passes for a single inference, which is why the computational cost is several order of magnitudes greater.

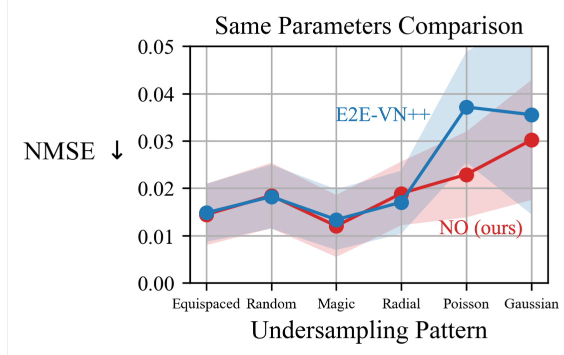

Figure 11. **Comparison between same parameter (21.7M) NO and E2E-VN++ with NMSE (↓).** While performance is similar on rectilinear patterns, on irregular patterns our NO model achieves lower NMSE than the E2E-VN of same size. On the Poisson undersampling pattern, we achieve 45% lowering NMSE. On the Gaussian undersampling pattern, we achieve 15% lower NMSE. We also notice that our NO model exhibits lower variance in its prediction performance.

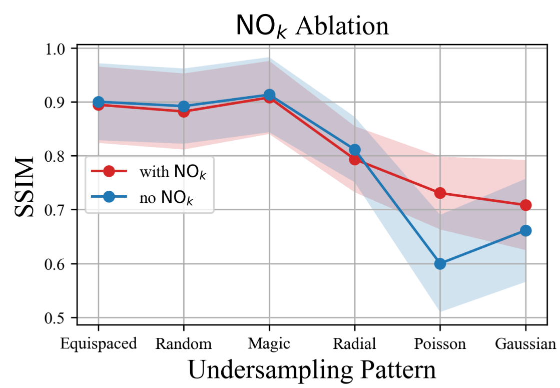

Figure 12. The ablation study of $NO_{\mathbf{k}}$.

Metric	NO (ours)	CSGM	ScoreMRI	PnP-DM
# forward passes	1	3465	4000	2651
GFLOPs	171	823K	950K	630K

Table 10. Comparison of GFLOPs and forward passes required per method.

E. DISCO: Discrete-Continuous Convolutions

E.1. Definition

Discrete-continuous (DISCO) convolutions [31] generalize the standard (continuous) convolution to Lie groups and quotient spaces. The approach is inspired by conventional convolutional layers, which efficiently implement local operations in neural networks but—upon grid refinement—converge to pointwise linear operators.

Definition E.1 (Group Convolution)**.** Let $\kappa, v : G \to \mathbb{R}$ be functions on a group G. Their convolution is defined as

$$(\kappa \star v)(g) = \int_G \kappa(g^{-1}x)\, v(x)\, \mathrm{d}\mu(x), \tag{9}$$

with $g, x \in G$ and $\mathrm{d}\mu(x)$ the invariant Haar measure.

Definition E.2 (DISCO Convolutions)**.** Given a quadrature rule with points $x_j \in G$ and weights q_j, the convolution (9) is approximated by

$$(\kappa \star v)(g) \approx \sum_{j=1}^{m} \kappa(g^{-1}x_j)\, v(x_j)\, q_j. \tag{10}$$

Here, the group action is applied analytically to κ, while the integral is discretized.

For a discrete set of output locations $\{g_i\}$, this becomes a matrix-vector product:

$$\sum_{j=1}^{m} \kappa(g_i^{-1}x_j)\, v(x_j)\, q_j = \sum_{j=1}^{m} K_{ij}\, v(x_j)\, q_j, \tag{11}$$

with $K_{ij} = \kappa(g_i^{-1}x_j)$. When κ is compactly supported, K_{ij} is sparse, with sparsity determined by the grid resolution and kernel support. A learnable filter is obtained by parameterizing κ as a linear combination of a chosen set of basis functions.

For comparison, consider a standard convolutional layer with stride 1, n input channels, a single output channel, and kernel $K = (K_i)_{i=1}^{S} \subset \mathbb{R}^n$ (with odd size S). On a regular grid $D_h = \{x_j\}_{j=1}^{m} \subset \mathbb{R}$ with spacing h, the output at $y \in D_h$ is given by

$$\mathrm{Conv}_K[v](y) = \sum_{i=1}^{S} K_i \cdot v\Big(y + z_i\Big), \tag{12}$$

with $z_i = h\Big(i - 1 - \frac{S-1}{2}\Big)$, and zero-padding.We see that $h \to 0$, $\lim_{h\to 0} \mathrm{Conv}_K[v](y) = \bar{K} \cdot v(y)$ with $\bar{K} = \sum_{i=1}^{S} K_i$, this means the convolutional layer is converging to a pointwise linear operator as the receptive field with respect to the underlying domain D is shrinking to a point. DISCO, however, does not converge to the pointwise operator.

E.2. Kernel Basis

In our DISCO framework, the kernel κ is parameterized using a basis for $L^2(\mathbb{D})$. The piecewise-linear, Zernike, and Morlet kernels are all parameterized by bases for $L^2(\mathbb{D})$. We show a specific case, using the (complex) Zernike polynomials, defined by

$$V_n^l(x, y) = R_n^l(\rho) e^{il\varphi}, \quad x = \rho\cos\varphi,\ y = \rho\sin\varphi, \tag{13}$$

where n is the total degree, $|l| \le n$, and $n - |l|$ is even. The radial polynomials $R_n^l(\rho)$ satisfy

$$\int_0^1 R_n^l(\rho) R_m^l(\rho)\, \rho\, d\rho = c_n^l\, \delta_{n,m}, \tag{14}$$

for some nonzero constant c_n^l. There are exactly $\frac{(n+1)(n+2)}{2}$ linearly independent polynomials of degree $\le n$, so every monomial $x^i y^j$ is a finite linear combination of Zernike polynomials. By the Weierstrass theorem, they form a complete basis for $L^2(\mathbb{D})$. (More details are in Appendix VII of [2].)