SS-REPA: SPECTRAL-SPATIAL REPRESENTATION ALIGNMENT FOR ONBOARD SPARSE PHYSICAL FIELD RECONSTRUCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Onboard sparse physical field reconstruction aims to recover dense physical states from a limited number of sensor measurements under tight memory and latency constraints. Compact convolutional encoder–decoders are attractive for deployment, but their capacity-constrained bottlenecks may struggle to capture the field-wide context required to infer unobserved regions. We propose Spectral-Spatial Representation Alignment (SS-REPA), a framework for improving bottleneck representations in compact convolutional encoder–decoders through alignment with a pretrained spectral neural operator. Specifically, SS-REPA introduces a dual-branch bottleneck with a mean-shift spatial branch and a spectral-context branch, providing separate pathways for local features reconstruction and global spectral interaction. Only the spectral-context branch features are mapped to the spectral neural operator feature space, where a Gaussian-weighted frequency-domain objective aligns their real and imaginary components while emphasizing global low-frequency structure. Extensive experiments on three sparse-sensor reconstruction benchmarks demonstrate that our method achieves competitive reconstruction accuracy with approximately 94% fewer parameters and 83.3–85.5% lower inference latency than the standard baseline on an edge computing device.

1 INTRODUCTION

Reconstructing dense physical fields from sparse sensor measurements is a fundamental problem in scientific computing and engineering, with applications ranging from structural health monitoring and digital twins to real-time state estimation and control (Wang & Zaki, 2021; Wang et al., 2025). Recent studies have demonstrated the potential of deep learning for reconstruction from sparse sensors (Fukami et al., 2021; Santos et al., 2023; Zhao et al., 2024; Ma et al., 2025). In an onboard setting, however, reconstruction accuracy is only one of several considerations, as deployment is simultaneously constrained by memory footprint and computational throughput (Manoj et al., 2023), making the capacity of the inference model a first-order design constraint.

Convolutional encoder-decoder networks (Ronneberger et al., 2015; Gong et al., 2023) provide a particularly attractive backbone under these constraints. Their hierarchical convolutions, resolution changes, and regular dense-tensor operations enable efficient extraction and reconstruction of spatial information. Recent theoretical analyses have further characterized encoder-decoder networks as multi-resolution mappings in which the encoder progressively transforms the input into a condensed latent representation and the decoder reconstructs the output from this representation (Williams et al., 2023). Under a strict deployment budget, however, reducing network width and depth inevitably restricts the amount of information that can be represented at this bottleneck interface (Silva et al., 2025). This issue is especially consequential for sparse physical field reconstruction: the bottleneck must aggregate measurements that may be widely separated in space and encode sufficient global context to infer large unobserved regions. This motivates the central question of this work: **how can we improve the bottleneck representation of a compact convolutional encoder-decoder without substantially enlarging the model deployed onboard?**

Spectral neural operators offer a promising source of global context representation (Li et al., 2021; Kovachki et al., 2023). By parameterizing interactions in a spectral basis, these operators can cap-

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.

ture long dependencies across the physical domain (Du et al., 2024; Xiao et al., 2024). Recent works such as U-FNO (Wen et al., 2022), U-FFNO (Liu et al., 2025a), PhySF-UNO (Chen et al., 2026), IRNO (Liu et al., 2026), and DRIFT-Net (Li & Salim, 2026) illustrate the complementary roles of the spectral and spatial representation. Nevertheless, directly incorporating a complete spectral neural operator into the deployed reconstruction architecture increases parameter storage and inference computation, which runs counter to the requirements of onboard reconstruction. This inspires an alternative use of spectral neural operators: rather than deploying them together with the compact reconstruction model, we can exploit their internal representations to serve as training-time alignment targets. However, achieving such alignment efficiently requires bridging a fundamental representational gap: the convolutional network is dominated by spatial convolutional features, whereas the spectral neural operator organizes information through spectral interactions. Directly forcing the entire bottleneck to match a spectral representation may impose conflicting requirements on local-detail preservation and global-context incorporation. Moreover, a conventional spatial-domain loss does not distinguish the spectral components that should be transferred from those already well modeled by spatial convolutions.

To address these challenges, we propose Spectral-Spatial Representation Alignment (SS-REPA), a regularization paradigm for capacity-constrained convolutional encoder-decoder networks. SS-REPA develops a decoupled dual-branch bottleneck in which a spatial-detail branch preserves local variations while a spectral-context branch performs frequency-domain channel interaction. Instead of aligning the complete bottleneck representation, only the spectral-context branch is mapped to the feature space of a pretrained spectral neural operator. A Gaussian-weighted frequency-domain alignment objective is then introduced to emphasize globally informative low-frequency components while progressively reduces the constraint on higher-frequency components. The outputs of the two branches are fused before decoding. Our main contributions are as follows:

- We propose SS-REPA, a regularization framework that aligns heterogeneous spectral and spatial features to enhance the bottleneck representation of compact convolutional encoder-decoder networks.
- We develop a decoupled dual-branch bottleneck with selective frequency-domain alignment, enabling the compact convolutional encoder-decoder network to acquire global spectral structure without sacrificing the local detail modeling afforded by spatial convolutions.
- We evaluate SS-REPA on three sparse physical-field reconstruction benchmarks, conducted on an NVIDIA Orin Nano edge platform, demonstrating a favorable accuracy–efficiency trade-off; relative to the standard convolutional baseline, the resulting model reduces model size by approximately 94% and inference latency by approximately 83.3–85.5% while maintaining competitive reconstruction fidelity.

2 Related Work

2.1 Deep Learning for Sparse Physical Field Reconstruction

Practical sensor deployment constraints often yield highly sparse observations (Liu et al., 2025b), rendering dense field recovery inherently ill-posed (Luo et al., 2024; Li et al., 2026). To address this challenge, a variety of deep learning approaches have been proposed, including convolutional networks (Voronoi-CNN (Fukami et al., 2021), UNet (Zhu et al., 2025b)), spectral neural operators (RecFNO (Zhao et al., 2024)), Transformer-based models (Senseiver (Santos et al., 2023)), and generative models (S3GM (Li et al., 2024), PhySense (Ma et al., 2025)). Despite their effectiveness, many of these methods rely on deeply stacked architectures, complex attention mechanisms, or intricate generative sampling, resulting in large model footprints and limited computational efficiency. Deploying them onboard typically requires additional compression or acceleration techniques. Among these approaches, convolutional networks offer a clear computational advantage (Khunarsa & Doungpaisan, 2026). In particular, Zhou et al. (2027) proposed progressive single-stage channel pruning for UNet-based sparse field reconstruction on resource-constrained edge platforms, achieving a favorable trade-off between parameter efficiency and reconstruction accuracy. In this work, we instantiate the SS-REPA framework with a compact UNet as the deployment encoder-decoder network.

2.2 Efficient Deployment and Knowledge Distillation

Recent efforts toward efficient deployment have explored a variety of methods, including manually engineered lightweight architectures (Qin et al., 2024), model pruning (Zhu et al., 2025a), quantization (Rokh et al., 2023), low-rank decomposition (Ou et al., 2024), and knowledge distillation (Mansourian et al., 2025). Unlike other approaches that directly modify the architecture or parameters of the model itself, knowledge distillation employs a teacher–student learning paradigm that leverages auxiliary information from an external teacher to guide the training of a compact student network, thereby enabling faster convergence or superior performance. Early knowledge distillation methods primarily transferred knowledge between models with homogeneous architectures; subsequent work has progressively extended to cross-architecture distillation, which has demonstrated efficacy (Liu et al., 2022; Zhang et al., 2025; Nguyen & Nguyen, 2025). In this work, we propose SS-REPA, which draws upon the idea of knowledge distillation to preserve the compactness of the student while effectively injecting a global spectral prior during training.

3 Methodology

Problem Setup: Consider a sparse physical field reconstruction task, let $\mathbf{x} \in \mathbb{R}^{n\times d}$ denote the locations of n sensors, and $\mathbf{s} \in \mathbb{R}^n$ denote the corresponding measurements from these sensors. The ground-truth physical field is denoted as $\mathbf{Y} \in \mathbb{R}^m$, representing the physical values at m discrete spatial locations ($n \ll m$). The objective is to learn a model parameterized by Θ, which realizes a mapping $G(\cdot;\Theta) : \mathbb{R}^{n\times d} \times \mathbb{R}^n \to \mathbb{R}^m$, such that the predicted field $\hat{\mathbf{Y}} = G(\mathbf{x}, \mathbf{s}; \Theta)$ closely approximates the ground-truth $\mathbf{Y}$.

3.1 Overall Framework

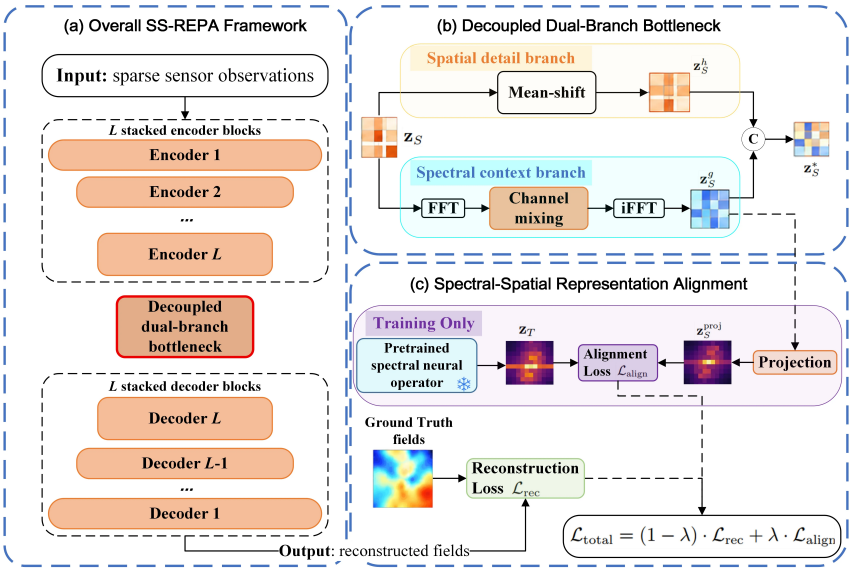


Figure 1: Schematic diagram of the overall framework of SS-REPA.

The overall framework of our proposed SS-REPA is illustrated in Fig. 1. Our approach is formulated as a training regularization paradigm, where a pretrained neural operator serves as a global spectral prior provider and a convolutional encoder-decoder network acts as the compact deployment model. For a sparse physical field reconstruction task, we freeze the pretrained spectral neural operator and use its intermediate features $\mathbf{z}_T$ as alignment targets. Suppose the compact model $G(\cdot;\Theta_S)$ consist of L encoder blocks $\{\mathcal{E}_i\}_{i=1}^L$ and L decoder blocks $\{\mathcal{D}_i\}_{i=1}^L$. The overall architecture can be expressed as a nested composition:

$$G(\cdot;\Theta_S) = \mathcal{D}_1 \circ \mathcal{D}_2 \circ \cdots \circ \mathcal{D}_L \circ \mathcal{E}_L \circ \cdots \circ \mathcal{E}_2 \circ \mathcal{E}_1, \tag{1}$$

where each encoder block $\mathcal{E}_i$ performs downsampling and feature extraction, and each decoder block $\mathcal{D}_i$ performs upsampling and feature reconstruction. The bottleneck feature is obtained as $\mathbf{z}_S = \mathcal{E}_L \circ \cdots \circ \mathcal{E}_1(\mathbf{x}, \mathbf{s})$, which is then fed into the decoupled dual branches: one applies a mean-shift method to preserve local details, yielding $\mathbf{z}_S^h$, while the other performs frequency-domain channel mixing, producing $\mathbf{z}_S^g$. The outputs of the two branches are fused to obtain $\mathbf{z}_S^*$, which is subsequently processed by the decoder. The decoder reconstructs the output recursively: starting from $\mathbf{d}_L = \mathbf{z}_S^*$, for $i = L, L-1, \ldots, 1$, the i-th decoder block computes

$$\mathbf{d}_{i-1} = \mathcal{D}_i(\mathbf{d}_i, \mathbf{h}_{i-1}), \qquad \hat{\mathbf{Y}} = \mathbf{d}_0, \tag{2}$$

where $\mathbf{h}_{i-1}$ is the skip feature from the corresponding encoder level and can be omitted for architectures without skip connections, and $\hat{\mathbf{Y}}$ is the final prediction of the convolutional encoder-decoder. In parallel, $\mathbf{z}_S^g$ is mapped to match the pretrained operator feature space, yielding $\mathbf{z}_S^{\text{proj}}$. A frequency-domain regularization loss $\mathcal{L}_{\text{align}}$ is then applied to enforce alignment between $\mathbf{z}_S^{\text{proj}}$ and $\mathbf{z}_T$, guiding the convolutional encoder-decoder to incorporate global structural patterns. At inference time, only the compact model is deployed, the operator and projection module are removed.

3.2 Decoupled Dual-Branch Bottleneck

The bottleneck is the central interface of an encoder–decoder network: it aggregates the encoder's multiscale features into a compact representation that conditions all subsequent decoding. Its quality therefore directly determines how much global context can be propagated to unobserved regions, making it a critical representational pressure point under a strict capacity budget. When a pretrained neural operator is introduced to supply global spectral priors, the bottleneck is expected to serve two roles simultaneously: preserving the local details that spatial convolutions excel at, and assimilating the global frequency information provided by the operator. These two roles, however, place conflicting demands on the learned representation. A single branch is less effective at representing both types of information within a shared feature space (see 4.3 for a detailed ablation study). This incompatibility motivates our decoupled dual-branch bottleneck: the spatial detail branch shifts the mean values to emphasize local intensity variations; the spectral context branch performs channel mixing in the frequency domain via separate convolutions on the real and imaginary components, with its outputs regularized by the proposed alignment loss to ensure low-frequency fidelity. Given the bottleneck feature $\mathbf{z}_S$, the two branches operate as follows:

$$\mathbf{z}_S^h = \mathbf{z}_S - \mathrm{GAP}(\mathbf{z}_S), \tag{3}$$

$$\mathbf{z}_S^g = \mathcal{F}^{-1}\left(\mathbf{W}_r * \mathrm{Re}(\mathcal{F}(\mathbf{z}_S)) + i \cdot \mathbf{W}_i * \mathrm{Im}(\mathcal{F}(\mathbf{z}_S))\right), \tag{4}$$

where $\mathrm{GAP}(\cdot)$ denotes global average pooling over the spatial dimensions; $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote the Fourier and inverse Fourier transforms, respectively; $\mathrm{Re}(\cdot)$ and $\mathrm{Im}(\cdot)$ extract the real and imaginary parts of a complex tensor; $\mathbf{W}_r$ and $\mathbf{W}_i$ are learnable real-valued convolution kernels applied independently to the real and imaginary components; and $*$ denotes the channel-mixing operator. The outputs of the two branches are then concatenated and fused:

$$\mathbf{z}_S^* = \phi\left(\mathrm{Concat}(\mathbf{z}_S^h, \mathbf{z}_S^g)\right), \tag{5}$$

where $\phi(\cdot)$ denotes channel fusion. The fused representation $\mathbf{z}_S^*$ is subsequently processed by the decoder, enriched with both local spatial and global spectral information.

3.3 Spectral-Spatial Representation Alignment

The core challenge of aligning convolutional encoder-decoders with spectral neural operators lies in their inherent representation mismatch: the encoder-decoder features are spatial-domain activations, whereas the neural operator features arise from spectral convolutions, leading to fundamentally different feature resolution, channel organization, and representation statistics. To this end, we adopt a mapping strategy—MLP first and then upscale—to bridge this gap (Tian et al., 2025). Specifically, the bottleneck feature $\mathbf{z}_S^g$ is transformed via projection followed by upsampling to yield $\mathbf{z}_S^{\text{proj}} \in \mathbb{R}^{C_T \times H_T \times W_T}$:

$$\mathbf{z}_S^{\text{proj}} = \mathrm{Upsample}\left(\mathcal{P}(\mathbf{z}_S^g)\right), \quad \mathcal{P} : \mathbb{R}^{C_S \times H_S \times W_S} \to \mathbb{R}^{C_T \times H_S \times W_S}, \tag{6}$$

where C_T, H_T, W_T denote the channel dimension and spatial resolution of the operator intermediate features, while C_S, H_S, W_S denote those of the bottleneck features.

To better measure feature discrepancy, we propose a frequency-domain regularization loss tailored to the spectral nature of physical fields. Unlike prior representation alignment methods that operate directly in the spatial domain with similarity-based objectives or ℓ_2 losses, our loss performs alignment in the frequency domain (see 4.3 for a detailed ablation study). Physical fields are typically smooth in space, with their energy concentrated in low-frequency modes, while high-frequency components correspond to fine-scale variations that the encoder-decoder's convolutions may already represent. The proposed Gaussian-weighted frequency-domain alignment therefore acts as a soft, frequency-selective transfer operator. It selectively injects the global low-frequency structures that the compact model cannot fully capture under a strict capacity budget, while preserving the local high-frequency details that the model already models well. Unlike a hard cutoff, the Gaussian mask attenuates the transfer gradually, thereby avoiding the spectral leakage artifacts of ideal low-pass filtering. The alignment loss is defined as:

$$\begin{aligned} \mathcal{L}_{\text{align}} &= \mathbb{E}\left[\left\|\mathcal{G}_\sigma \odot \left(\hat{\mathbf{z}}_S^{\text{proj}} - \hat{\mathbf{z}}_T\right)\right\|_F^2\right], \\ \mathcal{G}_\sigma(u,v) &= \exp\left(-\frac{\left(\frac{2u}{H}-1\right)^2 + \left(\frac{2v}{W}-1\right)^2}{2\sigma^2}\right), \end{aligned} \tag{7}$$

where σ controls the bandwidth of the low-frequency passband in normalized frequency coordinates. $u = 0, \ldots, H-1$ and $v = 0, \ldots, W-1$ index the shifted frequency grid. $\hat{\mathbf{z}}_S^{\text{proj}}$ and $\hat{\mathbf{z}}_T$ denote the frequency-domain representations of $\mathbf{z}_S^{\text{proj}}$ and $\mathbf{z}_T$, respectively.

3.4 Training Objective

The overall training objective comprises two terms: a reconstruction loss $\mathcal{L}_{\text{rec}}$ and a frequency-domain alignment loss $\mathcal{L}_{\text{align}}$. For reconstruction, we adopt mean squared error between the convolutional encoder-decoder network prediction and the ground truth:

$$\mathcal{L}_{\text{rec}} = \frac{1}{N}\sum_{i=1}^{N}\left\|\hat{\mathbf{Y}}_i - \mathbf{Y}_i\right\|_2^2, \tag{8}$$

where $\hat{\mathbf{Y}}$ denotes the prediction of the convolutional encoder–decoder network and N is the number of samples. The total loss is then formulated as a weighted combination:

$$\mathcal{L}_{\text{total}} = (1-\lambda)\cdot\mathcal{L}_{\text{rec}} + \lambda\cdot\mathcal{L}_{\text{align}}, \tag{9}$$

where $\lambda \in [0,1]$ balances the two objectives. We adopt a curriculum learning strategy that adaptively adjusts λ during training, see Appendix Eq. 10 for the detailed schedule.

4 Experiments

4.1 Experimental Setup

Benchmarks. We evaluate the proposed SS-REPA method on three high-fidelity benchmarks spanning diverse flow regimes: AirfRANS (steady airfoil Reynolds-Averaged Navier-Stokes) (Mazari et al., 2022), three-cylinder flow (unsteady vortex interaction) (Dou et al., 2025), and turbulent channel flow (wall-bounded turbulence) (Fukagata et al., 2006). We report the test relative ℓ_2 error, defined as Rel-$\ell_2 = \frac{\|\mathbf{Y}-\hat{\mathbf{Y}}\|_2}{\|\mathbf{Y}\|_2}$. For efficiency assessment, we measure parameter count (in millions), per-sample inference time (ms), and FLOPs (G). Detailed benchmark descriptions and data splits are deferred to the Appendix.

Baselines. We compare the proposed SS-REPA method against a comprehensive set of baselines spanning dominant paradigms in sparse field reconstruction. These include standard UNet (Zhou et al., 2027), Senseiver (Santos et al., 2023), RecFNO (Zhao et al., 2024), hybrid architectures (U-FNO (Wen et al., 2022), U-FFNO (Liu et al., 2025a), and IRNO (Liu et al., 2026)), as well as the

Table 1: Accuracy comparison: Rel-ℓ_2 across different sensor settings on three benchmarks. For the three-cylinder benchmark, the sensor setting is reported as sensors per cylinder, with the total number of sensors in parentheses. The last row reports the relative improvement of SS-REPA over the standard UNet, in percentage. Best results are in **bold**, second-best are underlined.

Method	AirfRANS			Three-cylinder flow			Turbulent channel flow		
	4	8	12	4 (12)	6 (18)	8 (24)	30	50	100
U-FNO	0.321	0.292	0.271	0.051	0.049	0.047	0.588	0.409	0.275
Senseiver	0.304	0.266	0.237	0.094	0.083	0.075	0.571	0.430	0.288
RecFNO	0.368	0.308	0.297	0.051	0.051	0.045	0.651	0.477	0.278
U-FFNO	0.345	0.276	0.249	0.048	0.054	0.043	0.636	0.474	0.321
IRNO	0.325	0.279	0.263	0.051	0.048	0.045	0.648	0.449	0.271
Standard UNet	0.301	0.273	0.235	0.079	0.051	0.054	0.593	0.412	0.277
Standard UNet-pruned	0.290	0.267	0.239	0.079	0.054	0.055	0.608	0.471	0.295
MobileNet-UNet	0.330	0.297	0.306	0.072	0.065	0.059	0.728	0.585	0.439
SS-REPA (Ours)	**0.290**	**0.254**	**0.233**	**0.045**	**0.039**	**0.040**	**0.555**	**0.382**	**0.257**
Improvement over UNet (%)	+3.65	+6.96	+0.85	+43.04	+23.53	+25.93	+6.41	+7.28	+7.22

lightweight method (standard UNet-pruned (Zhou et al., 2027) and MobileNet-UNet (Lakra & Dass, 2026)). For fair comparisons, all methods are trained and evaluated under strictly identical sensor configurations, data splits, and training protocols on each benchmark.

Implementation Details. We instantiate the SS-REPA framework with a compact UNet backbone, and train it for 100 epochs using the Adam optimizer. For spectral neural operator, we adopt a pretrained 4-layer RecFNO as a frozen supervisor that provides alignment targets during training. The initial learning rate is set to 1×10^{-3}, with the OneCycleLR scheduler (Smith & Topin, 2019) employed for dynamic adjustment. The bandwidth σ is fixed at 0.5, and the batch size is set to 16. All training procedures are conducted on a single NVIDIA RTX 4090 GPU. To evaluate the practical feasibility of our method for deployment on resource-constrained edge platforms, we additionally benchmark its inference performance on an NVIDIA Orin Nano. Comprehensive implementation details are deferred to the Appendix.

4.2 Main results

Our evaluation is organized around two central questions: **(Q1)** How does the proposed SS-REPA compare against representative sparse-field reconstruction methods? **(Q2)** How does model capacity influence the performance of convolutional encoder-decoder networks?

Quantitative Results. Table 1 compares the proposed SS-REPA with representative baselines across three benchmarks and varying sensor budgets. i) **Accuracy.** Overall, SS-REPA achieves the best or competitive reconstruction accuracy under every sensor configuration, indicating robustness to varying observation densities. We highlight three observations. First, MobileNet-UNet perform poorly across all datasets, which is consistent with the intuition that reducing model capacity can limit the ability to reconstruct high-fidelity physical fields. Second, on the three-cylinder flow benchmark, the standard UNet and its pruned variant still perform poorly despite their larger capacity. This gap cannot be attributed to model capacity alone, suggesting that other factors may also contribute. Third, RecFNO exhibits highly task-dependent behavior—performing strongly on the three-cylinder flow but poorly on AirfRANS—whereas SS-REPA achieves consistently strong accuracy across all three benchmarks. ii) **Efficiency.** Beyond accuracy, SS-REPA also offers substantial efficiency advantages, as illustrated in Table 2 and Appendix Figure 5. Compared with the standard UNet, SS-REPA reduces the parameter count by approximately 94% and inference time by approximately 84.3-85.5%, respectively. In contrast, while SS-REPA has a slightly larger parameter count than MobileNet-UNet, it achieves substantially lower inference time on edge devices, supporting its suitability for onboard sparse physical field reconstruction tasks.

Model Capacity Study. To investigate how the capacity of convolutional encoder–decoder networks affects performance, we vary the number of channels and the number of encoder/decoder blocks in the standard UNet, as illustrated in Figure 2. SS-REPA is denoted by a red star marker. We refer to a variant that shares the same architecture as SS-REPA but omits both the alignment strategy and the decoupled dual-branch bottleneck as **Standard UNet-small**, marked in black in the figure.

Table 2: Efficiency comparison: number of parameters (M), FLOPs (G), and inference time (ms/sample) on three benchmarks. The last row reports the relative improvement of SS-REPA over the standard UNet, in percentage.

Method	Params	AirfRANS		Three-cylinder flow		Turbulent channel flow	
		FLOPs	Infer. Time	FLOPs	Infer. Time	FLOPs	Infer. Time
U-FNO	6.74	8.33	32.74	11.42	69.39	2.02	9.09
Senseiver	10.11	31.38	53.46	43.44	102.40	6.32	12.30
RecFNO	8.41	1.88	20.57	2.60	47.56	0.40	4.79
U-FFNO	8.73	21.59	163.41	29.42	239.80	4.65	35.30
IRNO	10.57	4.03	80.05	5.52	126.73	0.78	36.50
Standard UNet	13.39	30.15	22.64	41.28	32.27	5.81	4.87
Standard UNet-pruned	6.57	26.08	21.47	35.70	31.59	5.00	4.78
MobileNet-UNet	**0.71**	**1.40**	5.56	**1.99**	8.09	**0.27**	1.61
SS-REPA (Ours)	0.86	1.66	**3.47**	2.26	**4.69**	0.31	**0.81**
Improvement over UNet (%)	+93.58	+94.49	+84.68	+94.53	+85.46	+94.66	+83.32

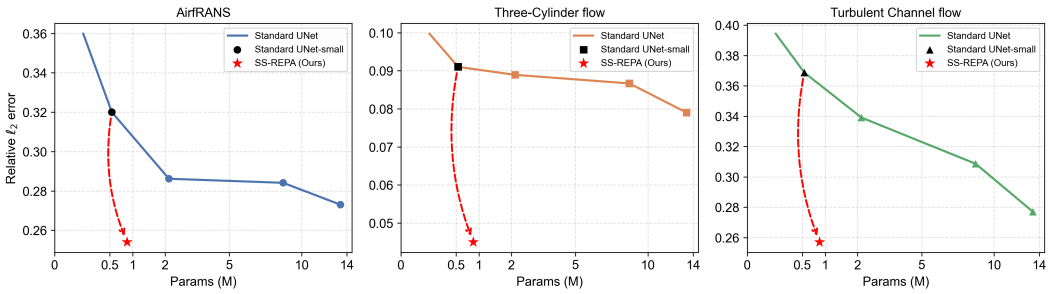


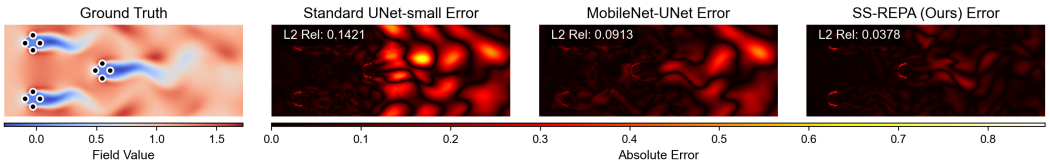
Figure 2: Reconstruction performance of the convolutional encoder-decoder network versus the number of parameters. The sensor configurations are as follows: AirfRANS with 8 sensors, the Three-cylinder flow with 4 sensors/cylinder, and the Turbulent channel flow with 100 sensors.

Figure 2 shows that increasing the number of parameters of the convolutional encoder–decoder network reduces the relative ℓ_2 error on all three benchmarks. Notably, SS-REPA achieves a substantial reduction in relative ℓ_2 error while maintaining a compact model size.

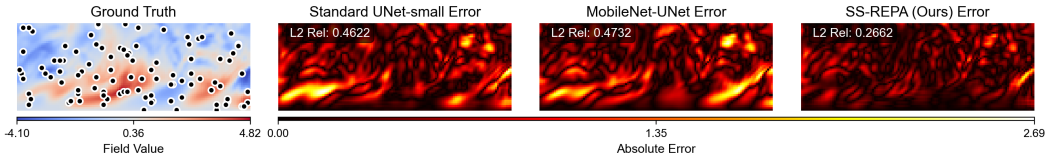
Case Study. Figure 3 visualizes the reconstruction errors for representative test samples from the Three-cylinder and Turbulent channel benchmarks. The corresponding AirfRANS result is deferred to Appendix Figure 6. Across all benchmarks, SS-REPA consistently yields lower errors than the standard UNet-small and MobileNet-UNet, with the most pronounced improvements observed in regions exhibiting complex flow dynamics, such as the wake behind cylinders. Quantitatively, compared to standard UNet-small, SS-REPA reduces the relative ℓ_2 error from 0.1421 to 0.0378 on the three-cylinder flow sample, and from 0.4622 to 0.2662 on the turbulent channel flow sample, demonstrating consistent gains across diverse physical settings.

4.3 Further Discussion

Ablation Study. To systematically investigate the impact of each component in SS-REPA, we design four ablation variants: **Single-branch w/ alignment**, a single-branch UNet with representation alignment; **Dual-branch w/o alignment**, a decoupled dual-branch UNet without alignment; **Dual-branch w/o mean-shift**, the full model without the spatial branch's mean-shift; and **Dual-branch w/o frequency-domain loss**, the full model with the Gaussian-weighted frequency-domain loss replaced by a spatial-domain ℓ_2 loss. We also report results for the **Standard UNet-small**, i.e., standard UNet with neither the dual-branch bottleneck nor alignment. As shown in Table 3, removing any component consistently degrades performance. Notably, Single-branch w/ alignment is even worse than standard UNet-small on AirfRANS and Three-cylinder flow, indicating that spatial convolutions alone cannot effectively incorporate the global spectral priors. Dual-branch w/o alignment and Dual-branch w/o frequency-domain loss also exhibit clear drops across all datasets, confirming the necessity of the selective frequency-domain alignment. Dual-branch w/o mean-shift



(a) Three-cylinder flow (4 sensors/cylinder)



(b) Turbulent channel flow (100 sensors)

Figure 3: Qualitative case study on two benchmarks. The first column shows the ground-truth fields, with black markers indicating sensor locations; the remaining columns present the absolute error fields produced by the standard UNet-small, MobileNet-UNet, and SS-REPA (ours), respectively.

Table 3: Ablation study of SS-REPA components. For each variant, we report the relative ℓ_2 error and the performance change (in parentheses, %) relative to the full model. Negative values indicate a performance degradation relative to the full model.

Variant	AirfRANS (8 sensors)	Three-cylinder flow (4 sensors/cylinder)	Turbulent channel flow (100 sensors)
SS-REPA (Full)	**0.254**	**0.045**	**0.257**
Standard UNet-small	0.320 (-26.0%)	0.091 (-102.2%)	0.369 (-43.6%)
Single-branch w/ alignment	0.332 (-30.8%)	0.093 (-105.8%)	0.326 (-26.5%)
Dual-branch w/o alignment	0.292 (-15.3%)	0.061 (-35.9%)	0.296 (-15.1%)
Dual-branch w/o mean-shift	0.282 (-11.3%)	0.079 (-76.0%)	0.309 (-20.1%)
Dual-branch w/o frequency-domain loss	0.262 (-3.5%)	0.085 (-88.4%)	0.292 (-13.6%)

degrades across all datasets as well, with the largest drop on the Three-cylinder flow, suggesting that mean-shift improves the complementarity of the two branches. Overall, these results support the synergistic effect of all components in SS-REPA.

Bottleneck Representation Improvement. To assess whether SS-REPA enhances bottleneck representations, we evaluate it from an information-theoretic perspective, using three metrics: effective rank, feature entropy, and mutual information. Effective rank measures the number of independent directions the representation occupies via the entropy of its singular value spectrum; feature entropy quantifies the total information volume encoded in the representation via the log-determinant of the empirical covariance; and mutual information estimates how much of the target field is linearly decodable from the bottleneck using a linear CCA estimator. Formal definitions are provided in the Appendix. As shown in Table 4, SS-REPA improves all three metrics consistently across the three benchmarks. The most pronounced gain is observed in feature entropy, while effective rank and mutual information also increase, albeit by smaller margins. Overall, these results indicate that SS-REPA yields a richer and more informative bottleneck representation.

Effective Receptive Field and Distance-Conditioned Jacobian. To examine whether SS-REPA extends the spatial extent over which the model aggregates information, we use two gradient-based diagnostics: the effective receptive field (ERF) and the distance-conditioned Jacobian. The ERF measures how much each input pixel contributes to a central output pixel, reflecting the spatial context captured by the network; the distance-conditioned Jacobian quantifies the expected absolute gradient between output and input pixels as a function of their Euclidean distance, characterizing the decay rate of long-range input-output dependence. Formal definitions are provided in the Appendix. Figure 4 presents the results on the Three-cylinder flow benchmark: the ERF heat-map of SS-REPA is markedly broader and more uniform than that of the standard UNet-small, and the distance-conditioned Jacobian of SS-REPA decays more slowly with distance.

Table 4: Bottleneck representation information metrics on three benchmarks.

Benchmark	Model	Effective Rank ↑	Feature Entropy ↑	Mutual Info. ↑
AirfRANS (8 sensors)	standard UNet-small	40.73	-224.31	18.83
	SS-REPA (Ours)	**43.97**	**-68.85**	**19.14**
Three-Cylinder (4 sensors/cylinder)	standard UNet-small	11.00	247.57	114.86
	SS-REPA (Ours)	**11.05**	**291.22**	**123.04**
Turbulent (100 sensors)	standard UNet-small	451.38	21.21	20.75
	SS-REPA (Ours)	**458.48**	**168.84**	**21.35**

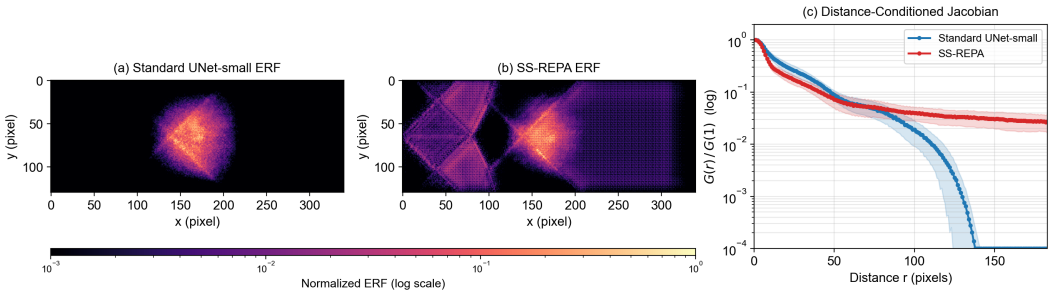


Figure 4: ERF and distance-conditioned Jacobian on the three-cylinder flow benchmark. (a) ERF heat-map of the Standard UNet-small. (b) ERF heat-map of SS-REPA. (c) Distance-conditioned Jacobian on a linear distance axis and a logarithmic vertical axis.

Alignment Target Position Selection. To identify the most effective intermediate layer of spectral neural operators for representation alignment, we conduct a study using a 4-layer RecFNO as an example. Specifically, we extract features from each of its four intermediate layers—denoted as RecFNO-1 through RecFNO-4—as alignment targets, and evaluate the resulting SS-REPA performance. As shown in Appendix Table 5, SS-REPA achieves consistently strong performance across all three datasets regardless of which layer is selected. On the Three-Cylinder, aligning to the last layer (RecFNO-4) attains a relative ℓ_2 error of 0.0451, whereas RecFNO-1/2/3 yield 0.0471/0.0483/0.0477, corresponding to 4.49%–7.12% larger errors. Similar trends hold on the other datasets, with the last layer consistently yielding the best or comparable results.

Generalization to Unseen Operating Regimes. To evaluate the out-of-distribution generalization capability of SS-REPA, we benchmark it against representative baselines on AirfRANS under two additional data splits—AoA and Reynolds—each with an 8 sensor configuration. As shown in Appendix Table 6, across all data splits, SS-REPA consistently attains the lowest relative ℓ_2 error. Specifically, SS-REPA achieves 0.2536, 0.2596, and 0.2700 on the Full, AoA, and Reynolds splits, respectively, compared to 0.2733, 0.2600, and 0.2738 for the strongest baseline standard UNet.

5 Conclusions

We propose SS-REPA, a regularization framework tailored for compact convolutional encoder–decoder networks in the context of onboard physical field reconstruction. SS-REPA enhances the capacity-constrained bottleneck representation by aligning it with that of a pretrained spectral neural operator, thereby capturing global information in physical fields more effectively. Specifically, we develop a decoupled dual-branch bottleneck and map only the features from the spectral context branch to the neural operator feature space, where the proposed Gaussian-weighted frequency-domain alignment loss is used. Extensive experiments on the AirfRANS, Three-cylinder flow, and Turbulent channel flow benchmarks, conducted on an edge computing platform, demonstrate that SS-REPA achieves competitive performance while reducing model size by 94% and inference latency by 83.3–85.5%. Ablation studies validate the effectiveness of each architectural component, and further analyses confirm that SS-REPA enhances the information content of bottleneck representations and broadens effective receptive fields.

AI USE STATEMENT

During the preparation of this work, the authors used AI to assist with language polishing and improving the readability of the manuscript. The tool was used solely for editing purposes, including grammar correction, sentence restructuring, and improving academic expression. The authors have not used generative AI tools for generating research ideas, designing experiments, implementing methods, analyzing results, or creating figures. All AI-assisted content was reviewed and edited by the authors, who take full responsibility for the final content of this work, including all text, claims, and artifacts.

REFERENCES

Yuxuan Chen, Yan Wang, Haotian Cheng, Hongna Zhang, Yuke Li, Wenhua Zhang, Xiaobin Li, and Fengchen Li. A physics-guided spectral-fourier u-net operator: From velocity field to hidden tensor field in viscoelastic turbulence. *Physics of Fluids*, 38(7):073103, 2026. doi: 10.1063/5.0333863.

Yehao Dou, Xun Han, and Pengzhi Lin. Flow field reconstruction and prediction of the two-dimensional cylinder flow using data-driven physics-informed neural network combined with long short-term memory. *Engineering Applications of Artificial Intelligence*, 149:110547, 2025. doi: 10.1016/j.engappai.2025.110547.

Yiheng Du, Nithin Chalapathi, and Aditi S. Krishnapriyan. Neural spectral methods: Self-supervised learning in the spectral domain. In *12th International Conference on Learning Representations, ICLR 2024*, 2024.

Koji Fukagata, Nobuhide Kasagi, and Petros Koumoutsakos. A theoretical prediction of friction drag reduction in turbulent flow by superhydrophobic surfaces. *Physics of Fluids*, 18(5):051703, 2006. doi: 10.1063/1.2205307.

Kai Fukami, Romit Maulik, Nesar Ramachandra, Koji Fukagata, and Kunihiko Taira. Global field reconstruction from sparse sensors with voronoi tessellation-assisted deep learning. *Nature Machine Intelligence*, 3(11):945–951, 2021. doi: 10.1038/s42256-021-00402-2.

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.

Zhiqiang Gong, Weien Zhou, Jun Zhang, Wei Peng, and Wen Yao. Joint deep reversible regression model and physics-informed unsupervised learning for temperature field reconstruction. *Engineering Applications of Artificial Intelligence*, 118:105686, 2023. doi: 10.1016/j.engappai.2022.105686.

Peerapol Khunarsa and Pafan Doungpaisan. A unified big-o-based framework for comparative complexity analysis of cnn and transformer architectures. *PeerJ Computer Science*, 12:e3966, 2026. doi: 10.7717/peerj-cs.3966.

Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research*, 24(89):1–97, 2023.

Wazir Lakra and Rajeshwar Dass. Edge detection using clip relu-based enhanced hybrid network. *Journal of Image and Graphics*, 14(1):84–95, 2026.

Jiayi Li and Flora D. Salim. DRIFT-Net: A spectral-coupled neural operator for PDEs learning. In *International Conference on Learning Representations (ICLR)*, 2026. URL `https://iclr.cc/virtual/2026/poster/10007240`.

Shihang Li, Zhiqiang Gong, Weien Zhou, Yue Gao, and Wen Yao. Learning to reconstruct temperature field from sparse observations with implicit physics priors. *International Communications in Heat and Mass Transfer*, 175:111147, 2026. doi: 10.1016/j.icheatmasstransfer.2026.111147.

Zeyu Li, Wang Han, Yue Zhang, Qingfei Fu, Jingxuan Li, Lizi Qin, Ruoyu Dong, Hao Sun, Yue Deng, and Lijun Yang. Learning spatiotemporal dynamics with a pretrained generative model. *Nature Machine Intelligence*, 6(12):1566–1579, 2024. doi: 10.1038/s42256-024-00938-z.

Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021. URL https://mlanthology.org/iclr/2021/li2021iclr-fourier/.

Hui Liu, Peizhi Zhao, and Tao Song. U-shaped factorized fourier neural operator for solving partial differential equations. *Computers & Mathematics with Applications*, 196:233–245, 2025a. doi: 10.1016/j.camwa.2025.07.013.

Xiaotian Liu, Shuyuan Shang, Xiaopeng Wang, Pu Ren, and Yaoqing Yang. Iterative refinement neural operators are learned fixed-point solvers: A principled approach to spectral bias mitigation. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026.

Xu Liu, Wei Peng, Xiaoya Zhang, Xiaoyu Zhao, Weien Zhou, Wen Yao, and Xiaoqian Chen. Enhancing deep learning-based field reconstruction with a differentiable learning framework. *Nature Machine Intelligence*, 7(7):1129–1140, 2025b. doi: 10.1038/s42256-025-01063-1.

Yufan Liu, Jiajiong Cao, Bing Li, Weiming Hu, Jingting Ding, and Liang Li. Cross-architecture knowledge distillation. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pp. 3396–3411, 2022.

Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2016.

Xihaier Luo, Wei Xu, Balasubramanya T. Nadiga, Yihui Ren, and Shinjae Yoo. Continuous field reconstruction from sparse observations with implicit neural networks. In *The Twelfth International Conference on Learning Representations (ICLR 2024)*, 2024.

Yuezhou Ma, Haixu Wu, Hang Zhou, Huikun Weng, Jianmin Wang, and Mingsheng Long. Physense: Sensor placement optimization for accurate physics sensing. In *Proceedings of the Thirty-ninth Annual Conference on Neural Information Processing Systems (NeurIPS 2025)*, 2025.

S. Manoj, Sannidhi Kasturi, Chandrakala G. Raju, H. N. Suma, and Jayanthi K. Murthy. Overview of on-board computing subsystem. In *Smart Small Satellites: Design, Modelling and Development*. Springer, 2023.

Amir M. Mansourian, Rozhan Ahmadi, Masoud Ghafouri, Amir Mohammad Babaei, Elaheh Badali Golezani, Zeynab Yasamani Ghamchi, Vida Ramezanian, Alireza Taherian, Kimia Dinashi, Amirali Miri, and Shohreh Kasaei. A comprehensive survey on knowledge distillation. *Transactions on Machine Learning Research*, 2025.

Ahmed Jocelyn Mazari, Florent Bonnet, Paola Cinnella, and Patrick Gallinari. Airfrans: High fidelity computational fluid dynamics dataset for approximating reynolds-averaged navier-stokes solutions. *Advances in Neural Information Processing Systems*, 35:23463–23478, 2022.

Khanh Nguyen and Huy Tien Nguyen. Cross-architecture knowledge distillation for speech enhancement: From cmgan to unet. *Neurocomputing*, 649:130798, 2025. doi: 10.1016/j.neucom.2025.130798.

Xinwei Ou, Zhangxin Chen, Ce Zhu, and Yipeng Liu. Low rank optimization for efficient deep learning: Making a balance between compact architecture and fast training. *Journal of Systems Engineering and Electronics*, 35(3):509–531, 2024. doi: 10.23919/JSEE.2024.000036.

Danfeng Qin, Chas Leichner, Manolis Delakis, Marco Fornoni, Shixin Luo, Fan Yang, Weijun Wang, Colby Banbury, Chengxi Ye, Berkin Akin, Vaibhav Aggarwal, Tenghui Zhu, Daniele Moro, and Andrew Howard. Mobilenetv4: Universal models for the mobile ecosystem. In *18th European Conference on Computer Vision, ECCV 2024*. Springer, 2024.

Babak Rokh, Ali Azarpeyvand, and Alireza Khanteymoori. A comprehensive survey on model quantization for deep neural networks in image classification. *ACM Transactions on Intelligent Systems and Technology*, 14(6):97:1–97:50, 2023. doi: 10.1145/3623402.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241. Springer, 2015.

Javier E. Santos, Zachary R. Fox, Arvind Mohan, Daniel O'Malley, Hari Viswanathan, and Nicholas Lubbers. Development of the senseiver for efficient field reconstruction from sparse observations. *Nature Machine Intelligence*, 5(11):1317–1325, 2023. doi: 10.1038/s42256-023-00746-x.

Jorge F. Silva, Victor Faraggi, Camilo Ramirez, Alvaro Egaña, and Eduardo Pavez. Understanding encoder–decoder structures in machine learning using information measures. *Signal Processing*, 234:109983, 2025. doi: 10.1016/j.sigpro.2025.109983.

Leslie N. Smith and Nicholay Topin. Super-convergence: very fast training of neural networks using large learning rates. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, 2019. doi: 10.1117/12.2520589.

Yuchuan Tian, Hanting Chen, Mengyu Zheng, Yuchen Liang, Chao Xu, and Yunhe Wang. U-REPA: Aligning diffusion U-Nets to ViTs. *Advances in Neural Information Processing Systems*, 38:11003–11024, 2025.

Kunpeng Wang, Wenhao Yang, Wenbo Li, Yi Chai, Juan Yao, Xiaofeng Huang, and Tong Wang. Progress in autonomous intelligent maintenance technologies for spacecraft. *Yuhang Xuebao/Journal of Astronautics*, 46(2):215–231, 2025.

Mengze Wang and Tamer A. Zaki. State estimation in turbulent channel flow from limited observations. *Journal of Fluid Mechanics*, 917:A9, 2021. doi: 10.1017/jfm.2021.255.

Gege Wen, Zongyi Li, Kamyar Azizzadenesheli, Anima Anandkumar, and Sally M. Benson. U-fno: An enhanced fourier neural operator-based deep-learning model for multiphase flow. *Advances in Water Resources*, 163:104180, 2022. doi: 10.1016/j.advwatres.2022.104180.

Christopher Williams, Fabian Falck, George Deligiannidis, Chris Holmes, Arnaud Doucet, and Saifuddin Syed. A unified framework for u-net design and analysis. In *37th Conference on Neural Information Processing Systems, NeurIPS 2023*, 2023.

Zipeng Xiao, Siqi Kou, Zhongkai Hao, Bokai Lin, and Zhijie Deng. Amortized fourier neural operators. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*, 2024.

Haonan Zhang, Longjun Liu, Yi Zhang, Xinyu Lei, Fei Hui, and Bihan Wen. Densekd: Dense knowledge distillation by exploiting region and sample importance. *IEEE Transactions on Neural Networks and Learning Systems*, 36(6):11243–11257, 2025. doi: 10.1109/TNNLS.2025.3525737.

Xiaoyu Zhao, Xiaoqian Chen, Zhiqiang Gong, Weien Zhou, Wen Yao, and Yunyang Zhang. Recfno: A resolution-invariant flow and heat field reconstruction method from sparse observations via fourier neural operator. *International Journal of Thermal Sciences*, 195:108619, 2024. doi: 10.1016/j.ijthermalsci.2023.108619.

Tao Zhou, Zhiqiang Gong, Weien Zhou, Xianzong Bai, and Wen Yao. Progressive channel pruning: Lightweight surrogate modeling of physical fields in aerial vehicle digital twins. *Expert Systems with Applications*, 332:133497, 2027. doi: 10.1016/j.eswa.2026.133497.

Zhanghao Zhouyin and Ding Liu. Understanding neural networks with logarithm determinant entropy estimator. *Neurocomputing*, 657:131520, 2025. doi: 10.1016/j.neucom.2025.131520.

Kehan Zhu, Fuyi Hu, Yuanbing Ding, Wei Zhou, and Ruxin Wang. A comprehensive review of network pruning based on pruning granularity and pruning time perspectives. *Neurocomputing*, 626:129382, 2025a. doi: 10.1016/j.neucom.2025.129382.

Tong Zhu, Dehao Liu, and Yanglong Lu. Finite-volume physics-informed u-net for flow field reconstruction with sparse data. *Journal of Computing and Information Science in Engineering*, 25(7): 071004, 2025b. doi: 10.1115/1.4067583.

A Appendix

A.1 Experimental Setup

A.1.1 Data preprocessing and sensor encoding

For AirfRANS, there are 1000 samples in total, with data splits including Full, AOA-shift, and Reynolds-shift. We use the Full split in the main results and the other two splits in the further discussion. Since the data are in point-cloud format, we randomly sample 32,000 points (all airfoil-surface points plus randomly sampled interior points) and convert them into a regular 160×200 grid. In the main experiments, we place 4, 8, and 12 surface-pressure sensors at fixed locations uniformly spaced along the normalized airfoil arc length. For the three-cylinder flow, there are 1000 samples in total. In all experiments, we randomly select 50 samples for training and 950 for testing. The data are already on a regular grid, and we crop a 130×340 region containing the three cylinders, resulting in 44,200 grid points. In the main experiments, we deploy 4, 6, and 8 velocity sensors in the flow field around each cylinder, uniformly distributed along the normalized arc length, capturing the local velocity components needed to compute vorticity. For the turbulent channel flow, there are 10,000 samples in total. The data are on a regular grid with 128×48 grid points, totaling 6,144 points. In the main experiments, 25–300 sensors are randomly selected for training, with their positions uniformly sampled across the entire domain. The model is trained on all available simulation data and evaluated on its ability to reconstruct the corresponding fields with 30, 50, and 100 sensors. Throughout all experiments, we use fixed random seeds and identical sensor placements for every method to guarantee a fair comparison.

A.1.2 SS-REPA Implementation Details

SS-REPA backbone. We instantiate the deployment model as a compact UNet with $L = 3$ encoder blocks and 3 decoder blocks. The base channel width is 16, and the channel width is doubled after each downsampling stage. Each encoder/decoder block contains two 3×3 convolutions, each followed by normalization and a GeLU activation. Downsampling is performed by strided convolution with stride 2. Upsampling is performed by bilinear interpolation followed by a 3×3 convolution. Skip connections are used between the corresponding encoder and decoder blocks. In the decoupled dual-branch bottleneck, $\mathbf{W}_r$ and $\mathbf{W}_i$ are implemented as shared, bias-free 1×1 convolutions over the channel dimension, applied to the real and imaginary parts. The two branch outputs are concatenated and fused by a 3×3 convolution (denoted as $\phi(\cdot)$ in the main text).

Alignment target and projection module. We adopt a pretrained RecFNO as the example of spectral neural operator supervisor, configured with 4 Fourier layers, 16 retained modes per axis, and a hidden width of 64. It is pretrained on the same benchmark training split under the same sensor configuration as the deployment model, and all its parameters are frozen during SS-REPA training. We use the intermediate feature from the last Fourier layer as the alignment target. The projection head $\mathcal{P}$ is a two-layer MLP with GeLU activation and BatchNorm. The upsampling operator is bilinear, mapping the bottleneck spatial resolution to the operator feature resolution.

Training objective and curriculum. We adopt a bounded adaptive curriculum for λ that scales the alignment weight according to the current reconstruction performance, the detailed curriculum schedule is given by:

$$\lambda(t) = \mathrm{clip}\left(\lambda_{\max} \cdot \frac{\mathcal{L}_{\text{rec}}(t-1)}{\mathcal{L}_{\text{rec}}(0) + \epsilon},\ \lambda_{\min},\ \lambda_{\max}\right), \tag{10}$$

where $\mathcal{L}_{\text{rec}}(t-1)$ denotes the average reconstruction loss over the previous epoch, $\mathcal{L}_{\text{rec}}(0)$ is that of the first epoch, and ϵ is a small constant for numerical stability. We set $\lambda_{\max} = 0.8$ and $\lambda_{\min} = 0.2$. The clipping operation ensures that $\lambda(t) \in [\lambda_{\min}, \lambda_{\max}]$, so the reconstruction term in the total objective never receives a negative weight. This schedule allows the model to initially emphasize alignment with the spectral neural operator and then progressively shift its focus toward reconstruction as training proceeds.

Edge deployment and efficiency measurement. Inference latency is measured on an NVIDIA Orin Nano operating at the 15 W power mode. The platform is equipped with a six-core ARM

Cortex-A78AE CPU running at 1.5 GHz and an NVIDIA Ampere GPU with 1024 CUDA cores at a maximum clock of 625 MHz, and provides 6.3 GB of available LPDDR5 memory. We report per-sample latency with batch size 1. Each measurement is averaged over 100 runs after 50 warm-up iterations. FLOPs are computed for a single forward pass. Unless otherwise stated, all models are evaluated at their default precision without additional hardware-specific optimization.

A.1.3 Baseline Architectures and Configurations

Standard UNet. We use the original UNet architecture (Ronneberger et al., 2015) as the full-capacity reference model and as the starting point for pruning. The backbone consists of 4 downsampling and upsampling stages, resulting in 5 spatial resolution levels. Each level employs a double-convolution block of two 3×3 convolutions with batch normalization and ReLU activation, and the decoder restores resolution through bilinear upsampling followed by skip concatenation. The encoder and decoder channel widths are $\{64, 64, 128, 128, 256, 256, 512, 512, 512, 512, 1024\}$ and $\{256, 256, 512, 128, 128, 256, 64, 64, 128, 64, 64\}$, respectively.

RecFNO. RecFNO (Zhao et al., 2024) learns the reconstruction mapping in infinite-dimensional function space, enabling resolution-invariant field reconstruction. Sparse observations are modeled via one of three embedding strategies—MLP, mask, or Voronoi—and the embedded input is then processed by stacked Fourier layers that reconstruct the physical field in Fourier space through Fourier mode superposition. Following the original implementation, we use 4 Fourier layers with 16 retained modes per axis and a hidden width of 64.

Senceiver. Senseiver (Santos et al., 2023) is a Transformer-based model that encodes sparse sensor observations into a compact latent representation via cross-attention and reconstructs the physical field by querying the latent space at arbitrary spatial coordinates. We follow the official configuration with a Fourier positional encoding of the sensor coordinates using 6 frequency bands. The encoder employs a preprocessing channel width of 256, 16 latent tokens with a latent dimension of 256, 8 cross-attention layers, 4 cross-attention heads, and 4 self-attention heads with 2 self-attention layers per block. The decoder uses a preprocessing channel width of 256, a latent dimension of 256, and 4 cross-attention heads.

U-FNO. U-FNO (Wen et al., 2022) couples a Fourier branch with a mini UNet branch in a parallel manner. Following the original setup, U-FNO stacks 6 Fourier blocks with 16 modes per axis and a hidden width of 32; in the first three blocks each Fourier branch is combined with a 1×1 convolution, while in the last three blocks it is further combined with a mini UNet branch via element-wise addition and a ReLU activation.

U-FFNO. U-FFNO (Liu et al., 2025a) introduces a U-shaped architecture prior to the factorized Fourier operator layers, allowing high-frequency information to be captured before spectral truncation discards it. We follow the official configuration with 6 factorized Fourier layers, 12 modes per axis, and a hidden width of 32. A lifting layer maps the input field, augmented with normalized grid coordinates, to a hidden representation, which is first processed by a U-shaped architecture and then fed into a stack of factorized Fourier operator blocks.

IRNO. IRNO (Liu et al., 2026) is an implicit residual neural operator that couples a frozen base operator with an iterative refinement network. The base operator is a pre-trained FNO with 4 Fourier layers, 16 modes per axis, and a hidden width of 64; its parameters are frozen during training and provide an initial field estimate h_0. Starting from h_0, a UNet-style refinement operator Φ_θ iteratively corrects the estimate through

$$h_{k+1} = h_k + \alpha\, \Phi_\theta\big([x, h_k]\big), \quad k = 0, \ldots, K-1, \tag{11}$$

where x is the input field, $\alpha = 0.2$ is the step size, and $K = 6$ refinement steps are used. The refinement operator takes the concatenated $[x, h_k]$ as input, uses 4 encoder/decoder levels with a base width of 16, circular padding for periodic domains. Training combines a spatial MSE loss with a progressive spectral loss and a fixed-point regularization term.

Table 5: Relative ℓ_2 error of SS-REPA with different RecFNO layers as alignment targets. Values in parentheses indicate the relative performance change compared to RecFNO-4; negative values indicate a larger error than RecFNO-4, i.e., worse performance. Best results per dataset are in **bold**.

Layer	AirfRANS (8 sensors)	Three-Cylinder (4 sensors)	Turbulent Channel (100 sensors)
RecFNO-1	0.265 (-4.62%)	0.0471 (-4.49%)	0.280 (-8.57%)
RecFNO-2	0.256 (-0.96%)	0.0483 (-7.12%)	0.261 (-1.22%)
RecFNO-3	0.257 (-1.39%)	0.0477 (-5.76%)	0.257 (-0.07%)
RecFNO-4	**0.254**	**0.0451**	**0.257**

Table 6: Relative ℓ_2 error on AirfRANS under different data splits (8 sensors). Best results are in **bold**, second-best are underlined.

Model	Full	AOA	Reynolds
RecFNO	0.3077	0.3165	0.3241
standard UNet	<u>0.2733</u>	<u>0.2600</u>	<u>0.2738</u>
U-FNO	0.2921	0.3211	0.3211
U-FFNO	0.2756	0.3164	0.3164
SS-REPA (Ours)	**0.2536**	**0.2596**	**0.2700**

Standard UNet-Pruned. We apply progressive channel pruning (Zhou et al., 2027) to the standard UNet. Channel importance is estimated by combining a first-order Taylor score, a mutual-information term, and a FLOPs regularization. The pruning schedule and training budget follow the original recipe.

MobileNet-UNet. MobileNet-UNet (Lakra & Dass, 2026) adopts a MobileNetV2 backbone as the encoder for sparse-field reconstruction. The single-channel input field is concatenated with normalized x- and y-coordinate channels to form a three-channel tensor. Multi-scale features from three stages (at $1/4$, $1/8$, and $1/16$ resolution with 24, 32, and 96 channels, respectively) are progressively fused in a lightweight decoder composed of bilinear upsampling, skip concatenation, and double-convolution blocks.

A.2 Additional Experimental Results

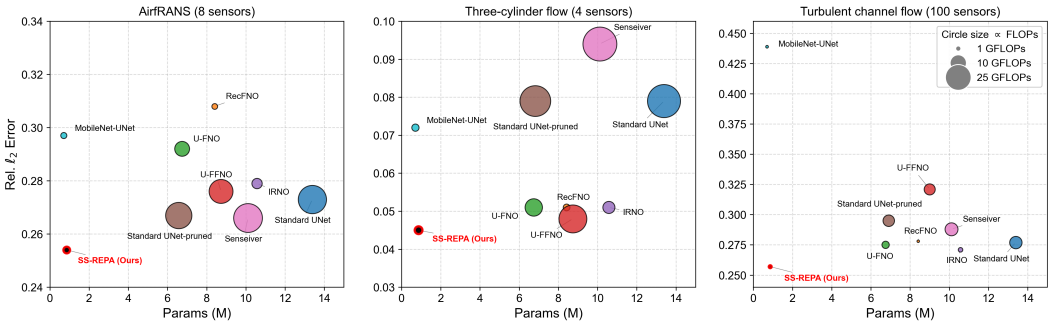


Figure 5: Accuracy–efficiency trade-off of SS-REPA and baselines on three sparse-field reconstruction benchmarks. Each point denotes a method; the circle size is proportional to FLOPs.

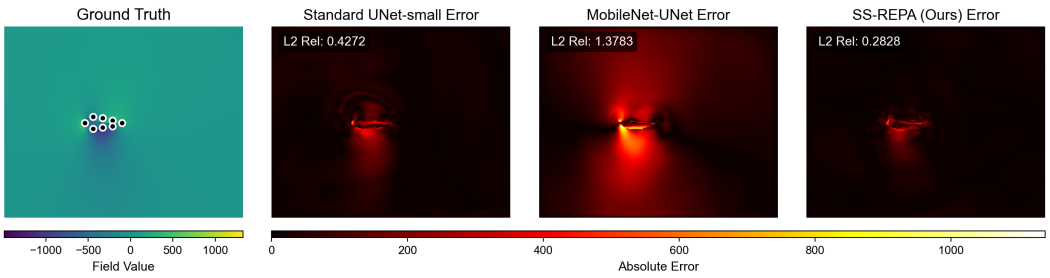


Figure 6: Qualitative case study on AirfRANS (8 sensors). The first column shows the ground-truth field, with black markers indicating sensor locations; the remaining columns present the absolute error fields produced by the standard UNet-small, MobileNet-UNet, and SS-REPA (ours), respectively, for a representative test sample.

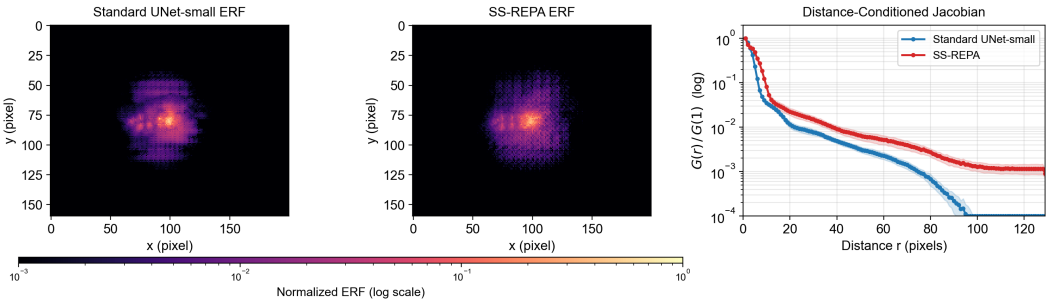


Figure 7: ERF and distance-conditioned Jacobian on the AirfRANS flow benchmark. (a) ERF heat-map of the Standard UNet-small. (b) ERF heat-map of SS-REPA. (c) Distance-conditioned Jacobian on a linear distance axis and a logarithmic vertical axis; shaded regions denote ± 1 standard deviation across 20 samples.

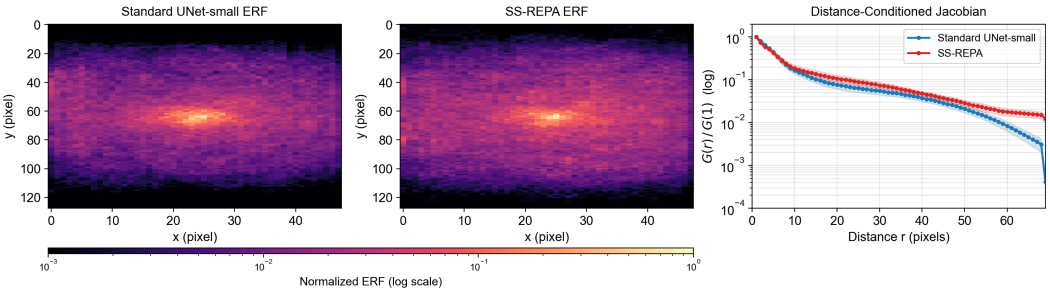


Figure 8: ERF and distance-conditioned Jacobian on the Turbulent channel flow benchmark. (a) ERF heat-map of the Standard UNet-small. (b) ERF heat-map of SS-REPA. (c) Distance-conditioned Jacobian on a linear distance axis and a logarithmic vertical axis; shaded regions denote ± 1 standard deviation across 20 samples.

A.3 Information-Theoretic Analysis of Bottleneck Representations

To quantitatively characterize how the dual-branch bottleneck in SS-REPA differs from the single-branch bottleneck in the standard UNet-small, we extract the bottleneck feature map $\mathbf{z}_S^* \in \mathbb{R}^{C_S \times H_S \times W_S}$ from SS-REPA and $\mathbf{z}_S \in \mathbb{R}^{C_S \times H_S \times W_S}$ from the standard UNet-small, and compute three representation-theoretic quantities. For all metrics, we flatten the spatial dimensions to obtain $\tilde{\mathbf{z}}_S^* \in \mathbb{R}^{N \times D}$ and $\tilde{\mathbf{z}}_S \in \mathbb{R}^{N \times D}$, where $D = C_S \cdot H_S \cdot W_S$ and N is the number of test samples (we use the entire test set). Unless otherwise stated, the feature tensors are centered column-wise before each computation.

Effective Rank. Following the RankMe criterion of Garrido et al. (2023), we measure the effective dimensionality of the bottleneck representation using the entropy of its singular value spectrum. This metric captures *how many independent directions the representation occupies*. Concretely, we compute the singular values $\{\sigma_i\}$ of the centered features via SVD, discard those below a numerical threshold $\epsilon = 10^{-12}$, and normalize them into a probability distribution $p_i = \sigma_i / \sum_j \sigma_j$. The effective rank is defined as the exponential of the Shannon entropy of this distribution:

$$\mathrm{erank}(\tilde{\mathbf{z}}) = \exp\left(-\sum_i p_i \log p_i \right). \tag{12}$$

A larger value indicates that the representation spreads its variance over more independent directions.

Feature Entropy. We quantify the information volume of the bottleneck through the LogDet entropy estimator of Zhouyin & Liu (2025), which approximates Shannon differential entropy from the log-determinant of the empirical covariance. This metric captures *the total information volume encoded in the representation*. Since D can be large relative to N, we first reduce the centered features to a k-dimensional subspace via PCA with $k = \min(k_{\max}, N-1, D)$ (we set $k_{\max} = 100$). Let $\mathbf{\Sigma}$ be the empirical covariance of the projected features. To ensure numerical stability we apply shrinkage regularization,

$$\hat{\mathbf{\Sigma}} = (1-\lambda)\,\mathbf{\Sigma} + \lambda\,\frac{\mathrm{tr}(\mathbf{\Sigma})}{k}\,I_k, \qquad \lambda = 0.1, \tag{13}$$

and compute the feature entropy as

$$\mathcal{H}(\tilde{\mathbf{z}}) = \log\det\hat{\mathbf{\Sigma}} = 2\sum_j \log L_{jj}, \tag{14}$$

where L is the Cholesky factor $\hat{\mathbf{\Sigma}} = LL^\top$; if the factorization fails, we fall back to the eigenvalue decomposition and sum $\log\lambda_j$ over the eigenvalues of $\hat{\mathbf{\Sigma}}$.

Mutual Information. We estimate the information that the bottleneck retains about the target field Y using the linear CCA estimator, which is a standard lower-bound estimator of mutual information for Gaussian variables. This metric captures *how much of the target field is linearly decodable from the bottleneck*. We first reduce both $\tilde{\mathbf{z}}$ and the flattened target $\tilde{\mathbf{y}} \in \mathbb{R}^{N \times D_y}$ to their leading $k = 32$ principal components, yielding $\mathbf{z}' \in \mathbb{R}^{N \times k}$ and $\mathbf{y}' \in \mathbb{R}^{N \times k}$. Let C_{zz}, C_{yy}, C_{zy} be the corresponding empirical covariance and cross-covariance matrices, and define the whitened cross-covariance

$$M = C_{zz}^{-1/2}\, C_{zy}\, C_{yy}^{-1/2}. \tag{15}$$

Its singular values $\rho_1, \ldots, \rho_k \in [0,1]$ are the canonical correlations between $\mathbf{z}'$ and $\mathbf{y}'$. Under the standard Gaussian assumption, the mutual information admits the closed form

$$\hat{I}(\mathbf{z}; Y) = -\tfrac{1}{2}\sum_{i=1}^{k} \log\bigl(1-\rho_i^2\bigr). \tag{16}$$

A.4 Effective Receptive Field and Distance-Conditioned Jacobian

We quantify the spatial extent of the bottleneck representation using the Effective Receptive Field (ERF) Luo et al. (2016) and the distance-conditioned Jacobian. The ERF measures how much

each input pixel contributes to a central output pixel, reflecting the spatial context captured by the network. For each of the $N = 20$ randomly selected test samples, we back-propagate from the center pixel of the output feature map and compute

$$\mathrm{ERF}(u, v) = \frac{1}{C} \sum_{c=1}^{C} \left| \frac{\partial f(\mathbf{X})(h_c, w_c)}{\partial \mathbf{X}_c(u, v)} \right|, \tag{17}$$

where $(h_c, w_c) = (\lfloor H_S/2 \rfloor, \lfloor W_S/2 \rfloor)$ is the center of the output feature map. To analyze long-range dependency, we define the distance-conditioned Jacobian $G(r)$ as the expected absolute gradient between an output pixel p and an input pixel q at Euclidean distance $r = \|p - q\|$:

$$G(r) = \mathbb{E}_{p,\, q\,:\, \|p-q\|=r} \left[\left| \frac{\partial \mathbf{Y}(p)}{\partial \mathbf{X}(q)} \right| \right]. \tag{18}$$

We approximate both quantities by back-propagating from sampled output locations and averaging the absolute input gradients. For ERF, each per-sample map is normalized by its maximum before cross-sample averaging. For $G(r)$, each per-sample curve is normalized so that $G(1) = 1$, isolating the relative decay rate of information flow. A slower decay indicates stronger long-range dependency.

A.5 Future Work

While the present work focuses on 2D sparse reconstruction, extending SS-REPA to 3D scenarios is a natural and promising direction for future work. This would involve generalizing the decoupled dual-branch bottleneck and the frequency-domain alignment loss to volumetric data.