LEARNING CONTINUOUS FIELDS WITH HIERARCHICAL LATENT MODELING AND SPARSE CONTEXT ENCODING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reconstructing continuous physical fields from sparse sensor observations is a fundamental challenge in scientific machine learning, with applications in climate, atmosphere, and ocean sciences. Implicit neural representations (INRs) provide a powerful coordinate-based framework, but existing sparse-to-dense INR approaches struggle to represent the multiscale spatiotemporal variability of physical fields. We present `HiLS` (*Hierarchical Latent modeling and Sparse context encoding*), an INR architecture built on two key innovations: *(i) Sparse Context Encoding (SCE)* that uses sparse observations together with sparse coordinates with a latent cross-attention mechanism to compress high-dimensional fields into a latent embedding, enabling accurate reconstruction; *(ii) Hierarchical Latent Modeling (HLM)* learns diverse spatial structures through a hierarchy of blocks built on rational basis functions. We evaluated `HiLS` in 64 configurations that span four geophysical datasets, four reconstruction tasks with fixed or variable sensors and locations, three sparsity levels, and temporal extrapolation. `HiLS` achieves the best average rank across all experiments. Ablation studies confirm the contribution of each architectural component.

1 INTRODUCTION

Monitoring the health of planet Earth is crucial for socio-economic well-being. The effects of climate change fall unevenly, hitting those with the least capacity to adapt the hardest. The planet is a highly complex geophysical system, governed by multiscale processes. Fortunately, many nations and international consortia have invested in remote sensing and in-situ Earth observation infrastructure. However, due to the vast spatial and temporal extent of the Earth system, even these extensive efforts still yield only sparse spatio-temporal datasets. Traditionally, researchers combine physics-based models with classical data assimilation to generate an analysis field, that is, an estimate of the Earth system that leverages both physical laws and observations (Evensen, 1994). With progress in neural modeling and the growing availability of geophysical data, recent work has reframed geophysical field analysis as a sparse-to-dense reconstruction task (e.g., Reichstein et al., 2019; Luo et al., 2024). The central challenge is to design methods that can consistently learn appropriate representations from sparse measurements alone and then generate accurate, dense fields.

Mathematical Problem Statement. Let $v(k) \in \mathbb{R}^{d_c}$ denote the sparse measurement of a physical field observed at the sensor location $k \in \mathbb{R}^{d_x}$ (where d_x is the dimension of the spatial location and d_c is the dimension of spatial content). Let $(\mathcal{K}, v(\mathcal{K})) = \{k_i, v(k_i)\}_{i=1}^{N}$, represent the set of N sparse observations for the sparse-to-dense reconstruction task. The objective of physical field reconstruction is to learn a function that, given the sparse context set $(\mathcal{K}, v(\mathcal{K}))$, can predict the field values at any dense set of target locations $\mathcal{X}$, yielding a continuous estimate $\hat{v}(\mathcal{X})$ of the true physical field $v(\mathcal{X})$.

Implicit neural representations (INRs) are a promising candidate, capable of learning continuous functions directly from coordinates, naturally handling sparse and irregular data through basis functions that encode spatial patterns (Luo et al., 2024). Critically, INRs require only discrete observations for training, eliminating the need for dense ground truth fields for training. This makes them uniquely suited for real-world scenarios. More information on related work is provided in

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

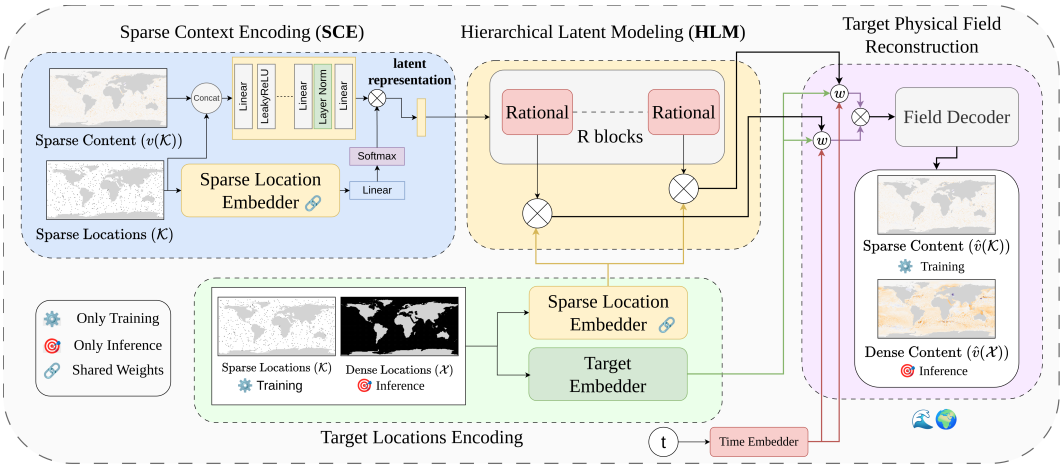

Figure 1: **HiLS** architecture. *(i) Sparse Context Encoding (SCE)* which embeds sparse observations $v(\mathcal{K})$ together with their spatial locations $\mathcal{K}$ into latent representations; *(ii) Hierarchical Latent Modeling (HLM)*, implemented using Rational Function blocks, which models multiscale physical dynamics in the latent space; and *(iii)* the *Target Field Reconstruction Module (TFRM)*, comprising Target Locations Encoding and Target Physical Field Reconstruction, which fuses target embeddings with latent through field decoder to reconstruct the dense field.

the Appendix B. INRs are designed to learn the approximate function $\hat{v}(\mathcal{X})$ from $\hat{v}(\mathcal{K}) = \mathcal{F}_\theta(\mathcal{K})$. Crucially, $\mathcal{F}_\theta$ must be trained using only sparse samples, while at inference time the dense field must be reconstructed from dense target locations $\mathcal{X}$. No explicit physical equations, boundary conditions, or PDE constraints are assumed. As learning is driven solely by the sparse observational data, this approach is broadly applicable across domains where physical models are incomplete or unavailable. Beyond INRs, Encoder-based alternatives such as neural processes (ConvCNP (Gordon et al., 2019)), attention-based sensor encoders (Senseiver Santos et al. (2022)) and operators (FNO (Li et al., 2020), OFormer Li et al. (2022)) were designed for dense supervision or stationary, grid-based priors. When retrained and tuned under our sparse-only protocol, all three yield higher errors than HiLS and trail the strongest INR baselines (Appendix G.1.1), which motivates building on INRs.

Nevertheless, current INRs exhibit four key limitations. *(i)* For physical field reconstruction, existing INRs take as input only the coordinates and ignore the content of the sparse observations, reducing INRs (Luo et al., 2024) to interpolation-based models rather than generalizable systems that exploit contextual information for reconstruction; *(ii)* as physical fields move to higher-dimensional settings, existing methods struggle to exploit global structural correlations while remaining computationally tractable; *(iii)* sequential learning can collapse the representation toward a single dominant direction, limiting the model's ability to preserve the diverse spatial structures required for complex physical field reconstruction; and *(iv)* the use of MLP layers, which in practice often have difficulty modeling periodic functions (Yu et al., 2024; Yang & Wang, 2024) that commonly arise in physical field data.

Contributions and placement within the INR literature. We present HiLS (*Hierarchical Latent modeling* and *Sparse context encoding*), a novel INR architecture based on two key innovations that jointly overcome the four limitations described above, and demonstrate its use across a broad range of experiments.

(1) *Sparse Context Encoding (SCE)*

(i) Crucially, unlike other INRs, the content information is encoded together with the coordinates, to learn a function $\hat{v}(\mathcal{X}) = \mathcal{F}_\theta(\mathcal{X} \mid \mathcal{K}, v(\mathcal{K}))$ with the appropriate representation capability.

(ii) Leverages a latent cross-attention mechanism to compress the original high-dimensional sensor locations into a lower-dimensional embedding space. This enables scalable reconstruction while preserving essential physical relationships across different scales at variable number of locations.

(2) *Hierarchical Latent Modeling (HLM)*

(iii) A new Hierarchical modeling approach is introduced, which helps to learn diverse spatial structures required for complex physical field reconstruction.

(iv) We adopt rational functions as our learnable basis Boullé et al. (2020) that approximate a broader class of functions, including those with singularities or sharp variations, more efficiently and accurately than polynomials (Walsh, 1935; Baker Jr & Gammel, 1961), thus offering greater expressive power for modeling complex physical fields.

(3) We evaluate `HiLS` over 64 configurations: 4 datasets $\times$ 4 tasks $\times$ 3 sparsity levels (48 settings), plus temporal extrapolation on 4 datasets $\times$ 4 tasks (16 settings). To our knowledge, this is the broadest evaluation reported for sparse-to-dense INR reconstruction.

Figure 1 shows our new architecture. Specifically, `HiLS` differs architecturally from existing INR-based models in four components: *Context Input*, *Spatial Context Encoding*, *Hierarchical Modeling*, and use of *Rational* function layers. None of the INR baselines (MMGN, WIRE, MFN, INCODE, FFN-P, CMLP, SIREN, and FINER) incorporates any of these components, and `HiLS` is the only model that integrates all four, underscoring its architectural novelty. Our experimental evaluation is correspondingly broad. We consider four reconstruction tasks that differ in their sensor and location configurations: fixed sensors at fixed locations (T1), variable sensors at fixed locations (T2), fixed sensors at random locations (T3), and variable sensors at random locations (T4). These tasks are evaluated on four geophysical datasets, each under three sparsity levels: Sea Surface Height (SSH; 10%, 25%, 50%), Chlorophyll (CHL; 10%, 25%, 50%), Global Surface Temperature (GST; 5%, 25%, 50%), and Sea Surface Temperature (SST; 0.1%, 0.3%, 0.5%). We further conduct temporal extrapolation experiments that cover all four tasks and all four datasets. Together, these aspects highlight both the distinctive design of `HiLS` and the comprehensive scope of our evaluation.

2 METHODOLOGY

Our architecture (Fig: 1) is composed of three primary components: *(i)* a *Sparse Context Encoding (SCE)* module, which encodes sparse content ($v(\mathcal{K})$) and their locations $\mathcal{K}$ to produce latent representations; *(ii)* a *Hierarchical Latent Modeling (HLM)* module that learns the diverse spatial relationships through hierarchical rational blocks in the learned latent space; and *(iii)* a *Target Field Reconstruction Module (TFRM)* with two submodules: Target Locations Encoding and Target Physical Field Reconstruction that together reconstruct the target content ($\hat{v}(\mathcal{X})$).

2.1 *Sparse Context Encoding (SCE)*

Coordinate-based INRs that learn a map between the coordinates and the target fields exhibit an implicit bias toward smooth reconstruction. Due to aliasing effect, multiple dense reconstructions are possible for the same sparse field. Coordinate-based INRs, even those with structures such as sinusoidal encoding, tend to produce blurry and overly smooth fields in the regions between sensors. This behavior is illustrated in Fig. 2, where existing INR-based baselines fail to recover fine-scale structures and instead generate excessively smoothed reconstructions.

We introduce *Sparse Context Encoding*, where sparse content ($v(\mathcal{K}) \in \mathbb{R}^{N \times d_c}$) and the corresponding sparse locations ($\mathcal{K} \in \mathbb{R}^{N \times d_x}$) are integrated through a latent cross-attention mechanism to produce latent representations, as illustrated in the top-left of Fig. 1. SCE works as follows.

We first encode the sparse spatial locations using a Gabor filter ($\mathcal{G}$) (Appendix A.2) to obtain enhanced coordinate embeddings. The choice of Gabor-based coordinate encoding is motivated by its effectiveness in prior INR-based methods (Saragadam et al., 2023; Fathony et al., 2020; Luo et al., 2024).

However, when the number of sparse locations N increases, a cross-attention mechanism that attends to all N sparse context, with $O(N)$ computations. To reduce the computational burden, we compress the N sparse content values and Gabor-enhanced coordinate embeddings into a fixed-size latent space of dimension L, making the cross-attention operate in the latent space. This interaction enables an effective fusion of spatial and physical information, resulting in richer priors from sparse data and improved computational efficiency.

The new latent cross-attention mechanism is defined as

$$A = \textit{Softmax}\left(\mathcal{G}(\mathcal{K})W_A\right) \in \mathbb{R}^{N\times L},$$
$$V = \phi(h)W_V \in \mathbb{R}^{N\times d_{\text{dim}}}; \qquad h = \textit{concat}\left([\mathcal{K}, v(\mathcal{K})]\right) \in \mathbb{R}^{N\times(d_x+d_c)},$$
$$Z^0 = A^T V \in \mathbb{R}^{L\times d_{\text{dim}}}.$$

Here, d_{dim} denotes the hidden dimension, and $\phi(\cdot)$ is a learnable network composed of linear layers and layer normalization. Since $L \ll N$, the resulting $L \times d_{dim}$ latent representation enables efficient subsequent processing and reduces the compute time for a forward pass (Section 4.1). Additionally, we empirically demonstrate that removing the sparse content leads to a reduction in reconstruction performance (Section 4.1).

2.2 *Hierarchical Latent Modeling (HLM)*

Given the latent representation Z^0, a conventional approach would be to process it through a sequential stack of MLP layers. However, such sequential processing can progressively collapse the learned representation toward a dominant mode (as discussed in Section 4.2), restricting the model's ability to preserve diverse spatial structures. This becomes particularly problematic for physical fields, which often exhibit complex multiscale patterns with distinct spatial characteristics. Furthermore, standard MLPs often struggle to efficiently represent periodic functions, which limits their suitability for capturing the complex structures present in physical field data.

We introduce *Hierarchical Latent Modeling (HLM)* which processes the latent representations through a sequence of R Rational Function blocks ($\mathcal{R}$), where each block operates on the output of the previous one. Going a step ahead, we retain and aggregate intermediate outputs from all blocks to form the final representation. This design induces a hierarchical organization of features, where successive blocks transform the latent representation in a progressive manner. Mathematically,

$$Z^{(r)} = \mathcal{R}_r\left(Z^{(r-1)}\right), \qquad r = 1, \ldots, R, \qquad Z^{(r)} \in \mathbb{R}^{L\times d_{\text{dim}}}.$$

As the model passes through successive blocks, it gradually updates the latent representation. The intermediate representations, denoted by $Z^{(r)}$, therefore capture features at different depths in the hierarchy. For each level r, we combine the information from its L latent tokens by mean pooling, producing a compact global descriptor for that level.

On its own, this global descriptor does not depend on where in space we want to reconstruct the physical field. To make it location-dependent, we combine it with the corresponding sparse location embedding using element-wise modulation. During training, we modulate using the sparse locations $\mathcal{K}$. During inference, we instead use the dense set of target locations $\mathcal{X} \in \mathbb{R}^{M\times d_x}$. This lets the same hierarchical latent representation be conditioned on any spatial coordinates, yielding location-aware features at every hierarchical level. Mathematically,

$$\textit{During Training:} \quad \tilde{Z}^{(r)} = \mathcal{A}\left(Z^{(r)}\right) \odot \mathcal{G}(\mathcal{K}) \in \mathbb{R}^{N\times d_{\text{dim}}},$$
$$\textit{During Inference:} \quad \tilde{Z}^{(r)} = \mathcal{A}\left(Z^{(r)}\right) \odot \mathcal{G}(\mathcal{X}) \in \mathbb{R}^{M\times d_{\text{dim}}},$$

where $\mathcal{A}(\cdot)$ denotes latent aggregation through mean pooling over the L latent tokens. The aggregated latent descriptor is broadcast across the corresponding spatial locations before element-wise modulation. We demonstrate the superiority of hierarchical and rational in Section 4.2.

2.3 *Target Field Reconstruction Module (TFRM)*

TFRM has two modules: *(i)* Target Location Encoding, and *(ii)* Target Physical Field Reconstruction.

Target Location Encoding: To reconstruct the physical field at arbitrary spatial locations, the learned latent representation must be conditioned on the queried coordinates. We therefore use two Gabor-based location encoding pathways with complementary functions. The first reuses the SCE Sparse Location Embedder from *SCE*, providing a consistent spatial representation that conditions each hierarchical latent representation within *HLM*. In parallel, a separate Target Embedder directly encodes the query locations for the Target Physical Field Reconstruction module, preserving

Table 1: Unobserved-location MSE ($\times 10^{-3}$) across three sampling ratios for each dataset. The error is computed exclusively over locations not included as sparse training observations (sensor locations). The final column reports the mean rank of each model across the 16 evaluation settings within the block, where a lower rank indicates better performance.

Data	SSH				CHL				GST				SST				Mean
Task	T1	T2	T3	T4	T1	T2	T3	T4	T1	T2	T3	T4	T1	T2	T3	T4	rank
Model	*Sampling ratio = 10%*				*Sampling ratio = 10%*				*Sampling ratio = 5%*				*Sampling ratio = 0.1%*				
HiLS	**4.47**	**4.41**	**0.72**	**1.02**	**17.23**	**16.93**	**6.30**	7.34	**3.65**	**4.66**	**2.23**	**3.40**	**0.83**	**1.12**	0.49	**0.76**	**1.12**
MMGN	19.23	15.08	3.29	4.07	42.07	37.16	8.23	9.54	4.89	5.31	3.11	3.76	1.01	1.27	**0.47**	1.01	5.12
INCODE	6.05	5.96	2.13	2.11	20.22	20.15	8.28	8.31	16.21	16.96	14.83	14.75	1.37	1.56	0.82	0.94	4.31
SIREN	6.45	6.96	2.97	3.00	19.64	19.92	9.52	9.61	18.40	18.60	18.03	17.99	1.92	2.03	1.40	1.50	6.06
FINER	4.87	4.97	4.62	4.66	18.90	19.29	17.72	18.19	17.10	17.57	15.85	15.82	2.44	2.70	1.91	1.98	6.06
WIRE	5.51	5.66	5.15	5.20	24.13	26.36	24.44	24.27	21.68	22.00	22.28	22.29	5.72	5.55	5.34	5.39	8.69
FFN-P	5.98	5.77	1.21	1.23	23.59	24.30	7.15	**7.16**	9.99	10.60	8.18	8.16	2.79	3.17	0.71	0.82	3.94
FFN-G	6.76	6.49	2.06	2.20	22.95	23.58	7.94	8.00	18.34	20.26	14.15	14.15	8.80	9.99	1.10	1.59	5.81
MFN	7.63	7.36	3.08	3.13	20.88	22.02	10.57	10.53	20.00	21.56	16.10	16.11	16.32	12.58	1.55	1.73	7.62
CMLP	5.90	5.90	3.19	3.19	18.81	19.65	11.07	10.64	21.62	22.34	17.02	16.86	2.01	2.23	1.28	1.31	6.25
	Sampling ratio = 25%				*Sampling ratio = 25%*				*Sampling ratio = 25%*				*Sampling ratio = 0.3%*				
HiLS	**3.55**	**3.27**	**0.26**	**0.48**	13.79	**12.61**	**3.29**	**5.26**	**2.16**	**2.56**	**1.98**	**2.36**	**0.34**	**0.51**	**0.22**	0.38	**1.31**
MMGN	8.39	7.47	2.08	2.74	20.80	18.00	6.87	7.66	3.11	3.33	2.81	3.00	0.40	0.54	0.26	0.44	4.62
INCODE	3.81	3.97	1.74	1.77	12.22	13.31	7.53	7.61	13.97	14.20	13.70	13.89	0.75	0.89	0.59	0.69	3.94
SIREN	3.80	4.31	3.11	3.07	**12.19**	13.29	8.99	9.03	17.66	17.68	17.56	17.60	1.28	1.43	1.21	1.36	6.50
FINER	4.18	4.34	3.98	4.04	14.92	15.12	13.97	13.48	15.44	15.54	15.09	15.32	1.33	2.69	1.17	1.56	7.38
WIRE	5.24	5.11	5.06	5.19	24.64	25.81	23.17	23.49	21.52	21.26	21.79	21.00	6.12	5.32	4.80	5.62	9.88
FFN-P	4.68	4.55	0.84	0.88	17.59	17.70	5.78	5.87	7.50	7.46	7.14	7.03	0.75	0.98	0.34	**0.37**	3.94
FFN-G	4.13	4.20	1.37	1.45	14.51	14.97	6.70	6.60	13.50	13.76	13.02	13.40	1.11	2.39	0.50	0.88	4.31
MFN	4.07	4.47	2.92	2.87	13.02	14.13	10.05	10.14	15.73	16.08	15.24	15.43	2.87	3.82	1.08	1.26	6.56
CMLP	4.33	4.53	2.78	3.06	13.19	14.47	8.69	11.93	15.75	16.92	16.43	16.55	1.12	1.21	1.02	1.11	6.56
	Sampling ratio = 50%				*Sampling ratio = 50%*				*Sampling ratio = 50%*				*Sampling ratio = 0.5%*				
HiLS	2.49	**1.91**	**0.16**	**0.29**	10.93	**8.79**	**2.15**	**3.43**	**1.99**	**2.31**	**1.93**	**2.22**	**0.25**	**0.35**	**0.18**	**0.28**	**1.38**
MMGN	3.11	3.62	1.75	2.16	10.35	10.82	6.48	6.98	2.90	3.07	2.79	2.93	0.30	0.41	0.24	0.35	3.44
INCODE	2.35	2.70	1.65	1.51	**8.89**	10.84	7.25	7.26	13.74	13.81	13.87	14.01	0.60	0.74	0.54	0.63	4.12
SIREN	3.06	3.69	2.69	3.09	9.87	10.68	8.57	8.82	17.43	17.73	17.50	17.50	1.18	1.37	1.17	1.33	7.12
FINER	3.75	3.64	3.57	3.34	12.43	15.05	12.01	13.96	15.01	15.37	15.07	15.28	1.15	2.22	1.06	1.58	7.94
WIRE	5.11	5.01	5.06	5.18	23.90	23.91	24.12	24.07	21.52	21.38	21.58	21.27	4.64	5.03	4.54	5.64	10.00
FFN-P	2.79	2.95	0.73	0.76	10.97	11.51	5.32	5.41	7.10	6.98	7.01	6.83	0.47	0.58	0.29	0.32	3.25
FFN-G	**2.32**	2.52	1.18	1.10	9.22	10.39	6.31	6.17	13.10	13.13	12.70	13.14	0.67	0.87	0.54	0.59	3.38
MFN	3.21	3.56	2.74	2.84	10.96	11.71	9.95	9.84	15.41	15.54	15.31	15.31	1.57	1.91	1.14	1.09	7.19
CMLP	3.35	3.76	2.65	2.92	11.41	12.08	9.72	11.28	16.18	16.72	14.61	16.27	1.06	1.08	0.95	0.86	7.19

location-specific information for the final fusion. During training, the sparse observation locations $\mathcal{K}$ are used as targets, while at inference they are replaced by the dense query locations $\mathcal{X}$. This enables reconstruction at arbitrary spatial locations without altering the learned latent representation.

Target Physical Field Reconstruction: After obtaining location-conditioned representations from *HLM*, the final step is to reconstruct the physical field at the target locations. Rather than directly decoding the hierarchical representations, we progressively integrate spatial, latent, and temporal information through two stages: *(i)* Block-wise fusion, *(ii)* Field decoding.

Block-wise Fusion: Each level in the hierarchy represents a different step in the hidden (latent) computation, so it must be linked to the spatial and time context of the target field. For each hierarchical block, we combine three types of information: (1) an embedding of the target location, (2) the location-conditioned hierarchical representation $\tilde{Z}^{(r)}$, and (3) a time embedding t. The target-location embedding tells the model where in the field it is reconstructing values. The representation $\tilde{Z}^{(r)}$ contains sparse content information passed through the r-th level of the hierarchy. The time embedding adjusts these representations based on the state of the physical system at the requested time. We project all three components into the same representation space, add them together, and include a learnable bias specific to each block. During training and inference, this operation is defined as follows:

$$V_{\text{train}}^{(r)} = \mathcal{P}_x(\mathcal{G}(\mathcal{K})) + \mathcal{P}_z^{(r)}\left(\tilde{Z}^{(r)}\right) + \mathcal{P}_t^{(r)}(t) + b_r \in \mathbb{R}^{N \times d_{\text{dim}}},$$

$$V_{\text{infer}}^{(r)} = \mathcal{P}_x(\mathcal{G}(\mathcal{X})) + \mathcal{P}_z^{(r)}\left(\tilde{Z}^{(r)}\right) + \mathcal{P}_t^{(r)}(t) + b_r \in \mathbb{R}^{M \times d_{\text{dim}}},$$

where $\mathcal{P}_x$, $\mathcal{P}_z^{(r)}$, and $\mathcal{P}_t^{(r)}$ denote learnable projections for the target location, hierarchical latent, and temporal representations, respectively. This block-wise formulation allows each hierarchical level to independently integrate the spatial, content, and temporal information required for reconstruction.

After obtaining complementary information within each hierarchical level, the resulting representations must also be combined across the hierarchy. Instead of aggregating them through simple addition, we employ element-wise multiplicative fusion, $V_{\text{fused}} = \bigodot_{r=1}^{R} V^{(r)} \in \mathbb{R}^{Q \times d_{\text{dim}}}$, where

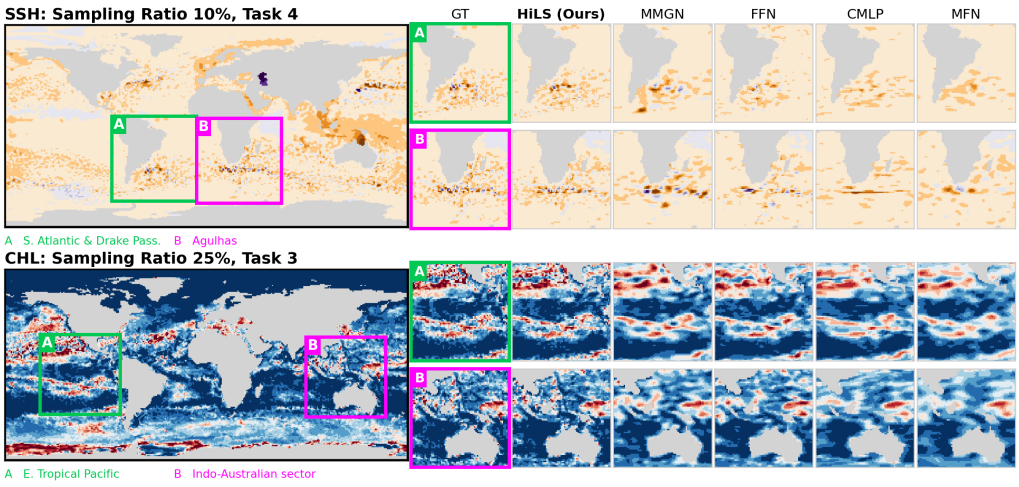

Figure 2: Qualitative comparison of reconstructed fields for SSH at a 10% sampling ratio (Task 4, top) and CHL at a 25% sampling ratio (Task 3, bottom). Ground-truth (GT) fields are compared with HiLS (Ours), MMGN, FFN, CMLP, and MFN. The highlighted regions A and B correspond to the South Atlantic & Drake Passage and Agulhas for SSH, and the Eastern Tropical Pacific and Indo-Australian sector for CHL, respectively. The zoomed regions highlight that HiLS more effectively captures and preserves fine-scale spatial details and local structures, particularly in regions with complex dynamics, while producing reconstructions that are visually closer to the ground truth.

$Q = N$ during training and $Q = M$ during inference. This multiplicative interaction allows the response of one hierarchical level to modulate those of the others, thereby introducing higher-order interactions among features obtained at different stages of latent processing.

Field Decoder: Finally, the hierarchically fused representation is decoded into the physical field through a lightweight Field Decoder $\mathcal{M}(\cdot)$,

$$\hat{v} = \mathcal{M}(V_{\text{fused}}), \qquad \hat{v} \in \begin{cases} \mathbb{R}^{N \times d_c}, & \text{during training,} \\ \mathbb{R}^{M \times d_c}, & \text{during inference.} \end{cases}$$

The Field Decoder is implemented as a shallow MLP with LeakyReLU activation that maps the fused d_{dim} dimensional representation to the d_c physical field variables at each target location.

3 Experiments

3.1 Datasets

To evaluate the performance of the model, we considered four challenging meteorological and ocean datasets derived from simulation and remote sensing. Sea Surface Height (SSH) (Copernicus Marine Service, 2024b) is sourced from altimeter SLA measurements and spans 365 frames on a 180×360 latitude–longitude grid. Chlorophyll (CHL) (Copernicus Marine Service, 2024a) combines NEMO simulation output with satellite observations, providing 579 frames on a 170×360 grid. Global Surface Temperature (GST) (Danabasoglu et al., 2020) is derived from CESM2 and consists of 1024 frames on a 192×288 grid. Sea Surface Temperature (SST) (Martin et al., 2012) draws on multi-source data and offers 360 frames at a finer 901×1001 grid. Appendix D has details.

3.2 Experimental Setup

We conduct extensive experiments to evaluate the reconstruction capability of HiLS under varying degrees of sensor randomness and observation sparsity. We compare against a broad set of established and recent INR-based methods, including SIREN (Sitzmann et al., 2020), FFN-G and FFN-P (Tancik et al., 2020), MMGN (Luo et al., 2024), CMLP (Ramasinghe & Lucey, 2022), MFN

(Fathony et al., 2020), WIRE (Saragadam et al., 2023), INCODE (Kazerouni et al., 2024), and FINER (Liu et al., 2024a). Further details of the baseline models are provided in Appendix F.2.

To systematically assess robustness to sparse observations, we consider the four reconstruction tasks introduced in Section 1, each representing a different level of sensor randomness. An illustrative overview of these tasks is provided in Appendix F.1. For each task, we further vary the observation sparsity to evaluate how reconstruction quality changes as the amount of available information decreases. We consider observation ratios of 5%, 25%, and 50% for GST; 0.1%, 0.3%, and 0.5% for SST; and 10%, 25%, and 50% for SSH and CHL.

For each setting, sparse observations are sampled from the full physical field according to the corresponding task configuration and sparsity level. Importantly, only these sparse observations are provided to the model during training and validation; the unobserved field values are not used for supervision. For each setting, the available sparse observations are divided into 90% for training and 10% for validation across timesteps. Given N observed locations $\mathcal{K} = \{k_i\}_{i=1}^{N}$ and their corresponding measurements $\{v(k_i)\}_{i=1}^{N}$, the model is optimized to reconstruct the field values at the observed locations. The training objective is the mean squared reconstruction error, $\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} \|\hat{v}(k_i) - v(k_i)\|_2^2$.

Evaluation Protocol. We evaluate and report reconstruction errors on the unobserved locations (i.e., dense locations excluding those used for training), thereby assessing the model's ability to reconstruct the physical field beyond the observed samples. Note that this protocol is different from that of baselines such as MMGN where the error is reported on the complete set of locations including those of the sensor.

Hyperparameter Selection and Implementation. For every combination of physical variable, reconstruction task, and sparsity level, `HiLS` is trained for up to 200 epochs using the AdamW optimizer with a learning rate of 10^{-3} and a weight decay of 10^{-5}. The batch sizes are set to 16, 2, 1 and 1 for GST, SST, SSH and CHL, respectively. All experiments are implemented in PyTorch Lightning and run on a single NVIDIA RTX 5000 Ada Generation GPU with 32 GB of VRAM. Complete hyperparameter configurations are provided in Appendix F. To ensure consistent comparisons, all baseline methods are tuned and evaluated under the same experimental protocol as `HiLS`.

Temporal Extrapolation. Beyond spatial reconstruction from sparse observations, we evaluate whether the learned representation can reconstruct physical fields in future time steps that are not observed during training. For all datasets, the temporal sequence is divided chronologically: the first 70% of frames are used for training, the subsequent 10% for validation, and the final 20% for testing. Consequently, the test frames correspond exclusively to future time steps outside the training interval. Models are evaluated by densely reconstructing the physical field at these unseen time steps, and the resulting dense reconstruction errors are reported.

3.3 Results

Quantitative Results. Tasks 1 through 4 are assessed at three sampling ratios on the GST, SST, SSH, and CHL datasets, offering a comprehensive evaluation of `HiLS` in comparison with other leading methods. Table 1 reports the quantitative results in terms of MSE at unobserved locations. `HiLS` consistently achieves strong reconstruction performance across datasets and sparsity levels, resulting in the best overall mean rank among the evaluated methods. Additional evaluations using MSE over the complete dense field, as well as the Pearson correlation coefficient and SSIM, exhibit similar performance trends and are provided in Appendix G.1. The robustness of the results to variations in random seeds is further assessed using error bars and reported in Appendix E.4.

Qualitative Results. Figure 2 compares reconstructions for SSH at 10% sampling (Task 4) and CHL at 25% sampling (Task 3). Across the highlighted regions, `HiLS` more faithfully recovers the ground-truth spatial structure, preserving fine-scale structures and sharper boundaries that are smoothed or distorted by the baselines. This is particularly evident in the Agulhas region for SSH and the Eastern Tropical Pacific for CHL, where `HiLS` retains localized structures that are less distinctly resolved by other methods. Additional qualitative comparisons across tasks and sampling ratios are provided in Appendix G.2.

Table 2: Temporal Extrapolation performance comparison of HiLS with major INR models on all four datasets and tasks for least sampling ratio from each dataset, using MSE(10^{-2}).

Data	SSH				CHL				GST				SST				Mean
Task	T1	T2	T3	T4	T1	T2	T3	T4	T1	T2	T3	T4	T1	T2	T3	T4	Rank
Model	*Sampling ratio = 10%*				*Sampling ratio = 10%*				*Sampling ratio = 5%*				*Sampling ratio = 0.1%*				
HiLS	**1.23**	**1.21**	**1.24**	**1.17**	**4.76**	**4.62**	**4.35**	5.01	**2.90**	**2.92**	**2.83**	**2.87**	**2.10**	**2.36**	**2.37**	**2.09**	**1.09**
MMGN	2.78	2.20	1.41	1.53	9.57	8.79	7.07	7.44	3.63	3.64	3.50	3.60	14.41	14.50	14.54	14.41	5.09
CMLP	1.29	1.32	1.35	1.36	8.46	6.80	6.77	7.14	5.36	4.05	3.42	3.37	12.54	12.58	13.99	13.54	3.81
MFN	2.02	1.97	**1.24**	1.20	10.15	9.21	7.57	9.30	16.98	17.04	13.43	12.85	22.74	19.46	26.22	19.21	6.28
WIRE	2.56	2.87	2.31	2.42	5.01	4.98	5.01	**4.99**	3.43	4.56	3.68	3.65	9.85	9.84	9.72	9.91	4.00
FFN-P	1.33	1.46	1.57	1.79	5.75	5.63	5.33	5.40	7.99	9.99	7.67	8.18	6.21	6.87	6.45	6.57	4.25
FFN-G	1.41	1.40	1.37	1.31	7.94	7.43	7.07	6.89	4.83	4.97	4.52	4.56	6.04	6.28	4.07	9.98	3.84
SIREN	5.19	4.83	1.93	3.83	10.11	13.23	15.35	10.09	7.50	6.67	41.93	31.57	32.73	29.38	45.26	48.10	7.62

Note on Non-INR methods. For the sake of completeness, we also evaluated Non-INR baselines, such as DeepONet (Lu et al., 2021), UNet (Ronneberger et al., 2015), and the FNO (Li et al., 2020). Since these architectures are not inherently suited to sparse-to-dense problems, they require non-trivial modifications to be trainable in our setting and consume more compute than the INR baselines. We give a comparison between HiLS and these methods in the Appendix G.1.1.

Temporal Extrapolation. Table 2 summarizes the results of dense reconstruction at future time frames not included in training. We report performance at the lowest sampling ratio (from main analysis), representing the most sparse and difficult reconstruction scenario. In this challenging task, HiLS consistently achieved strong performance with lowest mean rank. No other model had robust performance across datasets, highlighting the strength of our architecture. As context, previous published papers in this field do not report this important task.

Table 3: Component ablation of HiLS using unobserved-location MSE ($\times 10^{-3}$). Relative increases in error with respect to the full model are shown in parentheses; lower is better.

Data	Task (Ratio)	HiLS	- SCE	- Content	-Hierarchical	-Rational
GST	T3 (5%)	**2.23**	2.56 (+14.8%)	2.42 (+8.52%)	3.06 (+37.4%)	3.66 (+64.1%)
SST	T3 (0.50%)	**0.18**	0.22 (+20.1%)	0.24 (+34.1%)	0.46 (+153.3%)	0.34 (+88.9%)
GST	T2 (25%)	**2.56**	2.68 (+4.6%)	2.95 (+15.2%)	3.76 (+46.8%)	3.76 (+46.9%)
SST	T2 (0.10%)	**1.12**	1.20 (+7.2%)	1.19 (+6.3%)	1.16 (+3.8%)	1.24 (+10.7%)

4 Model Analysis

4.1 Effect of *Sparse Context Encoding (SCE)*

The *SCE* module is designed to address two limitations of coordinate-only INRs: the absence of sparse content information and the increasing computational cost of processing large numbers of spatial observations. To examine these two roles, we perform ablation studies by removing the complete *SCE* module (*-SCE*) and by removing only the sparse content $v(\mathcal{K})$ (*-Content*). We further compare the training cost with and without *SCE* across different sampling ratios.

Table 3 shows that removing either component consistently degrades reconstruction accuracy. Eliminating the complete *SCE* module increases the reconstruction error by 4.6%–20.1%, while removing only the sparse content results in an increase of 6.3%–34.1% across the evaluated settings. This demonstrates that using sparse content $v(\mathcal{K})$, in addition to the sparse coordinates $\mathcal{K}$, provides important contextual information for recovering the underlying physical field. Additionally, Fig. 2 shows that HiLS better preserves fine-scale structures and sharp boundaries, whereas coordinate-only baselines tend to produce smoother reconstructions without access to sparse content.

Beyond reconstruction accuracy, *SCE* also improves computational efficiency. As shown in Fig. 3, the training time of the architecture without *SCE* increases substantially as the sampling ratio grows. In contrast, *SCE* maintains considerably lower and more stable training cost across sampling ratios, providing approximately $2.0\times$–$5.3\times$ faster training in the evaluated settings.

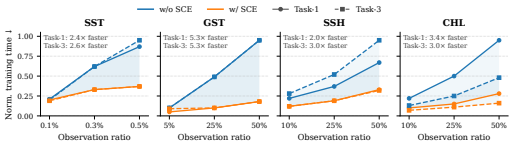

Figure 3: Normalized training time with and without SCE across observation ratios. Latent compression through SCE provides 2.0×–5.3× faster training and reduces sensitivity to increasing observation density.

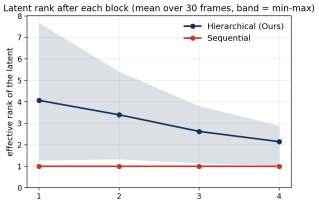

Figure 4: Effective rank across Rational Function blocks for SST Task 3 0.5% sampling ratio. Hierarchical modeling preserves multiple latent directions, while sequential processing collapses toward a dominant direction.

4.2 Effect of *Hierarchical Latent Modeling (HLM)*

The *HLM* module is designed to address two limitations of conventional sequential latent processing: the progressive collapse of latent representations toward a dominant direction and the limited expressive power of MLPs for physical field data. To examine these effects, we analyze the latent representation across successive blocks and perform ablations by removing the hierarchical fusion (*-Hierarchical*) and replacing the Rational Function blocks (*-Rational*).

We first investigate whether hierarchical modeling preserves diversity in the learned latent space. Figure 4 reports the effective rank based on entropy of the latent representation after each Rational Function block, computed from the normalized singular-value spectrum as $r_{\text{eff}}(Z) = \exp\left(-\sum_{i=1}^{q} p_i \log p_i\right), p_i = \frac{\sigma_i}{\sum_{j=1}^{q} \sigma_j}; q = \min(L, d_{\text{dim}})$. In conventional sequential processing, the effective rank remains close to 1 across all blocks, indicating that the representation becomes dominated by a single direction. In contrast, hierarchical modeling maintains an effective rank of approximately 2.2–4.1 across the successive blocks. Although the rank gradually decreases with depth, retaining and jointly utilizing intermediate representations prevents the information learned at earlier stages from being discarded. This provides the reconstruction module with multiple complementary latent representations rather than relying solely on the final sequential state. In addition, this behavior is also visualized using qualitative analysis in the Appendix E.1.

The quantitative ablation study in Table 3 further reinforce this observation. The replacement of hierarchical design with conventional sequential processing increases the error by 3.8%-153.3%, highlighting the benefit of preserving information from multiple stages of latent processing. Similarly, replacing the Rational Function with standard MLPs leads to a consistent performance degradation of 10.7%-88.9% which is also shown qualitatively in Appendix E.2.

Additional Appendix Analysis. We also evaluate the computational efficiency of the model (App. E.3). HiLS achieves the best reconstruction performance while maintaining low inference time and a moderate parameter count, demonstrating an effective efficiency-performance trade-off. Extensive sensitivity studies have been performed by varying the latent dimension and hierarchical blocks (App E.5).

5 Conclusion

`HiLS` introduces a new architecture for reconstructing physical fields from sparse data by combining *Sparse Context Encoding (SCE)* and *Hierarchical Latent Modeling (HLM)*. This architecture addresses key problems that have limited the use of existing INR models in scientific applications. Because *SCE* encodes the sparse observations jointly with the sparse coordinates, and its latent cross-attention keeps reconstruction scalable for high-dimensional fields. *HLM*, through its hierarchy of rational-function blocks, captures diverse structures with high representation capability. Across 64 experimental configurations, `HiLS` achieves the best average rank across all evaluation metrics. Qualitatively, its reconstructions remain spatially coherent while avoiding the artifacts, noise, and over-smoothing observed in competing INRs. Qualitative and quantitative ablation studies further provide consistent evidence that each proposed component contributes to the observed performance gains.

AI USE STATEMENT

In this work, we used generative AI tools solely for language editing and polishing the manuscript, including improving grammar, clarity, readability, and sentence structure. Generative AI tools were not used to generate research ideas, develop the methodology, conduct experiments, analyze results, produce scientific claims, or create figures. All AI-assisted edits were carefully reviewed by the authors to ensure that they preserved the intended technical meaning and accurately reflected the work performed. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The main paper provides the model formulation, training objective, evaluation protocol, and experimental setup. Detailed descriptions of the datasets and preprocessing are provided in Appendix D, while the reconstruction tasks, sampling procedures, baseline implementations, model configurations, and hyperparameter settings are documented in Appendix F. Additional quantitative results, statistical analyses across multiple runs, sensitivity studies, and qualitative comparisons are provided in Appendices E and G. We also provide the complete source code required to reproduce the experiments as part of the supplementary material.

REFERENCES

Alireza Afzal Aghaei. rkan: Rational kolmogorov-arnold networks. *arXiv preprint arXiv:2406.14495*, 2024.

Rico Angell and Daniel R Sheldon. Inferring latent velocities from weather radar data using gaussian processes. *Advances in Neural Information Processing Systems*, 31, 2018.

George A Baker Jr and John L Gammel. The padé approximant. *Journal of Mathematical Analysis and Applications*, 2(1):21–30, 1961.

Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3):24, 2009.

Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pp. 417–424, 2000.

Marc Bocquet. Ensemble kalman filtering without the intrinsic need for inflation. *Nonlinear Processes in Geophysics*, 18(5):735–750, 2011.

Nicolas Boullé, Yuji Nakatsukasa, and Alex Townsend. Rational neural networks. *Advances in neural information processing systems*, 33:14243–14253, 2020.

Lucy Chai, Michael Gharbi, Eli Shechtman, Phillip Isola, and Richard Zhang. Any-resolution training for high-resolution image synthesis. In *European conference on computer vision*, pp. 170–188. Springer, 2022.

Panqi Chen, Yifan Sun, Lei Cheng, Yang Yang, Weichang Li, Yang Liu, Weiqing Liu, Jiang Bian, and Shikai Fang. Generating full-field evolution of physical dynamics from irregular sparse observations. *arXiv preprint arXiv:2505.09284*, 2025.

Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8628–8638, 2021.

Hyungjin Chung, Jeongsol Kim, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. *arXiv preprint arXiv:2209.14687*, 2022.

Copernicus Marine Service. Global ocean biogeochemistry analysis and forecast (product id global_analysisforecast_bgc_001_028). Copernicus Marine Environment Monitoring Service (CMEMS), 2024a. URL https://doi.org/10.48670/moi-00015.

Copernicus Marine Service. Global ocean gridded l4 sea surface heights and derived variables reprocessed 1993–ongoing (product id sealevel_glo_phy_l4_my_008_047). Copernicus Marine Environment Monitoring Service (CMEMS), 2024b. URL https://doi.org/10.48670/moi-00148.

Gokhan Danabasoglu, J-F Lamarque, J Bacmeister, DA Bailey, AK DuVivier, Jim Edwards, LK Emmons, John Fasullo, R Garcia, Andrew Gettelman, et al. The community earth system model version 2 (cesm2). *Journal of Advances in Modeling Earth Systems*, 12(2):e2019MS001916, 2020.

Pan Du, Meet Hemant Parikh, Xiantao Fan, Xin-Yang Liu, and Jian-Xun Wang. Conditional neural field latent diffusion model for generating spatiotemporal turbulence. *Nature Communications*, 15(1):10416, 2024.

Geir Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics. *Journal of Geophysical Research: Oceans*, 99(C5): 10143–10162, 1994.

Rizal Fathony, Anit Kumar Sahu, Devin Willmott, and J Zico Kolter. Multiplicative filter networks. In *International conference on learning representations*, 2020.

Marco Gherlone, Priscilla Cerracchio, Massimiliano Mattone, Marco Di Sciuva, and Alexander Tessler. Shape sensing of 3d frame structures using an inverse finite element method. *International Journal of Solids and Structures*, 49(22):3100–3112, 2012.

Jonathan Gordon, Wessel P Bruinsma, Andrew YK Foong, James Requeima, Yann Dubois, and Richard E Turner. Convolutional conditional neural processes. *arXiv preprint arXiv:1910.13556*, 2019.

Yan Gu, Lei Wang, Wen Chen, Chuanzeng Zhang, and Xiaoqiao He. Application of the meshless generalized finite difference method to inverse heat source problems. *International Journal of Heat and Mass Transfer*, 108:721–729, 2017.

Robert Hecht-Nielsen. Kolmogorov's mapping neural network existence theorem. In *Proceedings of the international conference on Neural Networks*, volume 3, pp. 11–14. IEEE press New York, NY, USA, 1987.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Daniel Zhengyu Huang, Tapio Schneider, and Andrew M Stuart. Iterated kalman methodology for inverse problems. *Journal of Computational Physics*, 463:111262, 2022.

Marco A Iglesias, Kody JH Law, and Andrew M Stuart. Ensemble kalman methods for inverse problems. *Inverse Problems*, 29(4):045001, 2013.

Andrew H Jazwinski. *Stochastic processes and filtering theory*. Courier Corporation, 2007.

Amirhossein Kazerouni, Reza Azad, Alireza Hosseini, Dorit Merhof, and Ulas Bagci. Incode: Implicit neural conditioning with prior knowledge embeddings. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1298–1307, 2024.

Julien Lamouroux, Coralie Perruche, Alexandre Mignot, Julien Paul, Camille Szczypta, and Elodie Gutknecht. Quality information document for global_analysisforecast_bgc_001_028. Technical report, Copernicus Marine Service, September 2024. URL https://documentation.marine.copernicus.eu/QUID/CMEMS-GLO-QUID-001-028.pdf. Accessed: 2025-09-25.

Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5549–5558, 2020.

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.

Zijie Li, Kazem Meidani, and Amir Barati Farimani. Transformer for partial differential equations' operator learning. *arXiv preprint arXiv:2205.13671*, 2022.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

Zhen Liu, Hao Zhu, Qi Zhang, Jingde Fu, Weibing Deng, Zhan Ma, Yanwen Guo, and Xun Cao. Finer: Flexible spectral-bias tuning in implicit neural representation by variable-periodic activation functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2713–2722, 2024a.

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y Hou, and Max Tegmark. Kan: Kolmogorov-arnold networks. *arXiv preprint arXiv:2404.19756*, 2024b.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature machine intelligence*, 3(3):218–229, 2021.

Xihaier Luo, Xiaoning Qian, Nathan Urban, and Byung-Jun Yoon. Reinstating continuous climate patterns from small and discretized data. In *1st Workshop on the Synergy of Scientific and Machine Learning Modeling@ ICML2023*, 2023.

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

Matthew Martin, Prasanjit Dash, Alexander Ignatov, Viva Banzon, Helen Beggs, Bruce Brasnett, Jean-Francois Cayula, James Cummings, Craig Donlon, Chelle Gentemann, et al. Group for high resolution sea surface temperature (ghrsst) analysis fields inter-comparisons. part 1: A ghrsst multi-product ensemble (gmpe). *Deep Sea Research Part II: Topical Studies in Oceanography*, 77:21–30, 2012.

Alejandro Molina, Patrick Schramowski, and Kristian Kersting. Pad\'e activation units: End-to-end learning of flexible activation functions in deep networks. *arXiv preprint arXiv:1907.06732*, 2019.

Evangelos Ntavelis, Mohamad Shahbazi, Iason Kastanis, Radu Timofte, Martin Danelljan, and Luc Van Gool. Arbitrary-scale image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11533–11542, 2022.

Marie-Isabelle Pujol. Copernicus marine product user manual for sealevel_glo_phy_climate_l4_my_008_057. Technical report, Copernicus Marine Service, 2 avenue de l'Aérodrome de Montaudran, 31400 Toulouse, France, November 2024. URL `https://documentation.marine.copernicus.eu/PUM/CMEMS-SL-PUM-008-057.pdf`. Public release.

Sameera Ramasinghe and Simon Lucey. Beyond periodicity: Towards a unifying framework for activations in coordinate-mlps. In *European Conference on Computer Vision*, pp. 142–158. Springer, 2022.

Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer school on machine learning*, pp. 63–71. Springer, 2003.

Markus Reichstein, Gustau Camps-Valls, Bjorn Stevens, Martin Jung, Joachim Denzler, Nuno Carvalhais, and F Prabhat. Deep learning and process understanding for data-driven earth system science. *Nature*, 566(7743):195–204, 2019.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Javier E Santos, Zachary Fox, Arvind Mohan, Hari Viswanathan, and Nicholas Lubbers. The senseiver: attention-based global field reconstruction from sparse observations, 2022.

Vishwanath Saragadam, Daniel LeJeune, Jasper Tan, Guha Balakrishnan, Ashok Veeraraghavan, and Richard G Baraniuk. Wire: Wavelet implicit neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18507–18516, 2023.

Donald Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference*, pp. 517–524, 1968.

Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33:7462–7473, 2020.

Bowen Song, Liyue Shen, and Lei Xing. Piner: Prior-informed implicit neural representation learning for test-time adaptation in sparse-view ct reconstruction. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 1928–1938, 2023a.

Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *International Conference on Learning Representations*, 2023b.

Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in neural information processing systems*, 33:7537–7547, 2020.

Joseph Leonard Walsh. *Interpolation and approximation by rational functions in the complex domain*, volume 20. American Mathematical Soc., 1935.

Mohit Yadav, Daniel Sheldon, and Cameron Musco. Faster kernel interpolation for gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pp. 2971–2979. PMLR, 2021.

Xingyi Yang and Xinchao Wang. Kolmogorov-arnold transformer. *arXiv preprint arXiv:2409.10594*, 2024.

Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5505–5514, 2018.

Runpeng Yu, Weihao Yu, and Xinchao Wang. Kan or mlp: A fairer comparison. *arXiv preprint arXiv:2407.16674*, 2024.

A PRELIMINARIES

A.1 RATIONAL FUNCTION NETWORK

The representation theorem (Hecht-Nielsen, 1987) states that any multivariate continuous function $f : [0, 1]^n \to \mathbb{R}$ can be expressed as:

$$f(x_1, \ldots, x_n) = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^{n} \phi_{q,p}(x_p) \right),$$

where $\phi_{q,p} : [0, 1] \to \mathbb{R}$ and $\Phi_q : \mathbb{R} \to \mathbb{R}$ are continuous univariate functions. These functions generalize this by substituting each weight parameter with a learnable univariate function (Liu et al., 2024b). A general Rational function network is a composition of L layers is $\text{KAN}(x) = (\Phi_{L-1} \circ \Phi_{L-2} \circ \cdots \circ \Phi_0)(x)$, where the j-th feature in layer l transforms as:

$$\Phi(x) = w_a \cdot \textit{SiLU}(x) + w_b \cdot \textit{Spline}(x),$$

where $\textit{Spline}(x) = \sum_i c_i B_i(x)$ is a linear combination of ***B-spline*** functions with learnable coefficients. Unlike MLPs with fixed activations, Rational functions provide adaptive nonlinearities at

each connection, improving multiscale representations, critical for learning dynamics from sparse observations of physical fields.

In our method, we use the rational function (Boullé et al., 2020; Aghaei, 2024) as the base function. Specifically, we parameterize the function $\phi(x)$ on each edge as a rational function over polynomials $P(x)$ and $Q(x)$ of order m and n:

$$\phi(x) = wF(x) = w\frac{P(x)}{Q(x)} = w\frac{a_0 + a_1 x + \cdots + a_m x^m}{b_0 + b_1 x + \cdots + b_n x^n}$$

Here, a_i and b_j are coefficients of the rational function, and w is a scaling factor. This function is said to have degree m/n. We aim to learn the coefficients $\{a_i\}$, $\{b_j\}$, and w through end-to-end backpropagation.

To avoid instability caused by poles, where $Q(x) \to 0$ and $\phi(x) \to \pm\infty$, we employ a Safe Padé Activation Unit (PAU) (Molina et al., 2019) as our basis, which is a modified form of the standard rational function:

$$F(x) = \frac{a_0 + a_1 x + \cdots + a_m x^m}{1 + |b_1 x + \cdots + b_n x^n|}$$

A.2 Gabor-based Coordinate Encoding

Traditional positional encodings, such as Fourier features, employ global basis functions that struggle to capture spatially varying frequency content. We leverage localized Gabor filters combining Gaussian spatial localization with sinusoidal frequency selectivity:

$$\textit{Gabor}(x) = \exp\left(-\frac{\gamma}{2}\|x - \boldsymbol{\mu}\|_2^2\right) \times \sin(\frac{3\delta}{2} \times \textit{Linear}(x)),$$

We initialize $\boldsymbol{\mu} \sim \mathcal{N}(2, 1) \in \mathbb{R}^{N \times d_x}$, $\gamma \sim Gamma(\alpha, \beta) \in \mathbb{R}^{d_{dim}}$ and $\delta \in \mathbb{R}^{d_{dim}}$ is a learnable parameter.

B Related work

Continuous field reconstruction methods can be broadly categorized into two approaches: *(i)* classical and *(ii)* deep learning methods.

One of the first and most widely used classical methods is the linear model. To address non-linearity, spatial interpolation is performed using inverse distance weighting with a power parameter (Shepard, 1968). Later, alternative approaches are based on the Gaussian process and its variants (Rasmussen, 2003), such as the ensemble Kalman filter (Evensen, 1994; Bocquet, 2011; Iglesias et al., 2013) and extended Kalman filter (Jazwinski, 2007; Huang et al., 2022), in which fields are approximated using Gaussian kernels. However, these methods are hindered by computational complexity, which is characterized by time complexity and renders it infeasible for extremely large datasets(Angell & Sheldon, 2018; Yadav et al., 2021). There was also the introduction of model reduction techniques to manage high-dimensional fields and perform field reconstruction (Angell & Sheldon, 2018; Gu et al., 2017; Gherlone et al., 2012).

In deep learning, several techniques can be used for reconstruction, including super-resolution, neural inpainting, and implicit neural representation. Super-resolution techniques are primarily concerned with the enhancement of image quality from a lower to a higher resolution. However, within the domain of continuous field reconstruction of scientific datasets, the paired data for such processes are absent. Our principal objective lies in the construction of a continuous representation u derived from a discretized data set. Recent advances in deep learning-driven super resolution facilitate continuous magnification through a variety of methodologies, including scale-consistent positional encodings (Ntavelis et al., 2022) and training with variable sizes (Chai et al., 2022). Furthermore, these methods encompass local conditioning strategies that take advantage of the deep features of the surrounding area (Chen et al., 2021) or employ neighborhood-based interpolation (Luo et al., 2023). Image inpainting techniques can be broadly classified into traditional and learning-based approaches. Traditional methods utilize low-level features through diffusion(Bertalmio et al., 2000) or patch-based (Barnes et al., 2009) techniques to propagate known information into missing regions. In contrast, learning-based approaches, including GANs(Yu et al., 2018; Lee et al., 2020)

and diffusion models(Song et al., 2023b; Chung et al., 2022), produce more semantically coherent reconstructions but are often computationally intensive and may require additional post-processing.

Implicit Neural Representation (INR): INRs are a type of neural network that implicitly represents data (like images, 3D shapes, or other signals); that is, instead of storing data as arrays of values (like pixels in an image), INRs learn a continuous function that maps coordinates (like x,y,z) to the value at that point (like pixel brightness or density). Two primary purposes of INRs are (1) Parameterizing the sensor domain, i.e., INRs map sensor coordinates (like where a measurement was taken) to predicted sensor values; this is useful when real sensor data are noisy or incomplete, and the INR fills in or improves those data. e.g., MRI (Song et al., 2023a) sensors measure data in the Fourier domain. (2) Parameterizing the density domain, that is, mapping spatial coordinates (like a point in 2D or 3D space) to underlying physical density values, is a more direct representation of the actual object or field that is measured. e.g, Measurement of temperature from locations.

Gabor filter in computer vision : The application of Gabor filters for encoding spatial coordinates has proven to be a significant advancement for Implicit Neural Representations (INRs). For example, WIRE (Saragadam et al., 2023) uses a complex Gabor wavelet as activations to get richer location-dependent features. Multiplicative Filter Networks (MFN) (Fathony et al., 2020) iteratively multiply together the linear functions of Gabor wavelet functions applied to the input to produce richer feature representations from input coordinates; the authors show this construction can be viewed as a linear function approximator operating over a large collection of Fourier/Gabor basis functions. The Multiplicative and Modulated Gabor Network (MMGN) (Luo et al., 2024) adopts an encoder-decoder architecture for location representations, and emphasizes that Gabor-based, localized frequency representations have an advantage over global Fourier bases when modeling complex, non-stationary patterns.

Diffusion based Methods: Recent generative approaches have advanced sparse-to-full reconstruction of physical dynamics. CoNFiLD (Du et al., 2024) uses conditional neural-field latent diffusion to generate high-fidelity turbulent flows and support zero-shot reconstruction and restoration. SDIFT (Chen et al., 2025) employs diffusion in Functional Tucker space with a message-passing posterior to recover full-field dynamics from irregular sparse measurements. S3GM (Li et al., 2024) leverages an unconditional score-based generative prior and conditions it on sparse sensors, achieving robust and accurate reconstruction even under extreme sparsity and noise.In the training stage of these above methods, full-field data is necessary, which is often not feasible in many real-world scenarios. On the other hand, our goal is to learn the physical dynamics using only sparse data.

C HILS MATHEMATICAL PROBLEM FORMULATION

Given N sparse context

$$\mathcal{D}_{\mathrm{s}} = \{(k_i, v(k_i))\}_{i=1}^{N}, \qquad k_i \in \mathbb{R}^{d_x}, \quad v_i = v(k_i) \in \mathbb{R}^{d_c},$$

the model learns a mapping

$$\mathcal{F}_\theta : (\mathcal{K}, v(\mathcal{K})) \mapsto \hat{v}(\cdot).$$

Here, $\mathcal{K} = \{k_i\}_{i=1}^{N} \in \mathbb{R}^{N \times d_x}$ denotes the set of N sparse locations and $v(\mathcal{K}) = \{v(k_i)\}_{i=1}^{N} \in \mathbb{R}^{N \times d_c}$ denotes the set of sparse content, with $d_x = 2$ and $d_c = 1$ in our setting, corresponding respectively to the spatial coordinates (latitude and longitude) and the physical field value at that location (e.g. SST, GST).

During **training**, the model predicts the content at the sparse locations,

$$\hat{v}(\mathcal{K}) = \{\hat{v}(k_i)\}_{i=1}^{N} \in \mathbb{R}^{N \times d_c}.$$

At **inference**, given M dense query locations and N sparse context ($\mathcal{D}_{\mathrm{s}}$)

$$\mathcal{X} = \{k_j\}_{j=1}^{M}, \qquad k_j \in \mathbb{R}^{d_x},$$

the model produces

$$\hat{v}(\mathcal{X}) = \{\hat{v}(k_j)\}_{j=1}^{M} \in \mathbb{R}^{M \times d_c}.$$

D DATASET DETAILS

To test and validate the proposed methodology, we use four datasets that exhibit distinct characteristics. The first data set consists of the merged satellite product derived from a space-borne altimeter. The second data set consists of a data-assimilative forecast of a biological variable from a numerical ocean model. The third dataset is derived from a simulation of a prominent climate model. The fourth dataset consists of remotely sensed observations. All considered datasets are from the metocean domain and represent important aspects of the Earth with different spatiotemporal variability, distributional characteristics, and range. This selection allows us to assess the effectiveness of the proposed approach in reconstructing continuous fields from sparse observations using implicit neural representations (INRs) across various conditions and showcases its potential.

D.1 SEA SURFACE HEIGHT

The first dataset we use comes from the Copernicus Marine Environment Monitoring Service (CMEMS) global gridded Level 4 Sea Surface Height product, processed by the DUACS (Data Unification and Altimeter Combination System) multimission altimeter data processing system. The product provides altimeter satellite gridded Sea Level Anomalies (SLA) computed with respect to a twenty-year [1993, 2012] mean, estimated using Optimal Interpolation that merges L3 along-track measurements from different altimeter missions available (Pujol, 2024).

The DUACS system processes data from all altimeter missions available, including Sentinel-6A, Jason-3, Sentinel-3A, Sentinel-3B, Saral/AltiKa, Cryosat-2, and HY-2B. All missions are homogenized with respect to a reference mission, and the processing is fitted to the global ocean. The system has been designed to ensure quality and stability of the sea level record, with calibration and validation activities funded by space agencies being essential to maintain the quality of these products.

The dataset represents a critical component for monitoring sea level rise and understanding climate evolution, with the system maintained and operated by CMEMS since May 2015, having previously been integrated into the CNES multi-mission ground segment SSALTO.

D.2 CHLOROPHYLL CONCENTRATION

The second dataset we use comes from the Copernicus Marine Service (CMEMS) operational global biogeochemical analysis and forecast system, produced at Mercator-Ocean (Lamouroux et al., 2024). The system provides global chlorophyll mass concentration analysis and forecast products. We obtain and use the surface Chlorophyll-A concentration in a global grid of 1 degree spatial resolution.

The system is based on the NEMO (Nucleus for European Modeling of the Ocean) platform coupled with the PISCES (Pelagic Interaction Scheme of Carbon and Ecosystem Studies) biogeochemical model. PISCES is a biogeochemical model that simulates marine biological productivity and describes the biogeochemical cycles of carbon, oxygen, and the main nutrients (P, N, Si, Fe), representing the marine biogeochemistry component of ocean modeling platforms and Earth System models. The coupled NEMO-PISCES system with data assimilation has been operational since April 2018, and in September 2019, Mercator Ocean International successfully updated its global biogeochemical analysis and forecasting system with Ocean Color data assimilation capability.

Satellite-derived surface chlorophyll data are assimilated weekly into the three-dimensional ensemble configuration of the coupled NEMO-PISCES model using a reduced-order Kalman filter, improving surface analysis and forecast chlorophyll representation across major parts of the model domain. The biogeochemical model is offline coupled with the dynamical ocean from the CMEMS global physical analysis and forecasting operational system, at a daily frequency, and benefits from the assimilation of satellite (SSH-SST-SIC) and in situ physical data. The time series is aggregated in time to reach a two full year's time series sliding window, with output mean fields interpolated on a standard regular grid in NetCDF format.

D.3 Global Surface Temperature

The third dataset we use comes from the pre-industrial control run of the Community Earth System Model version 2 (CESM2). This dataset provides monthly averages of global surface temperature, representing a key atmospheric variable. The CESM2 dataset captures the complex atmospheric patterns, making it an ideal resource for evaluating our method.

CESM2 simulates interactions among major components of the Earth system, including the atmosphere, oceans, land surface, and sea ice. The pre-industrial control run specifically models a stable climate without human influence, serving as a reference point for understanding how the climate may change under future scenarios.

Because the data is simulated and exhibits rich spatial and temporal variation, it offers a robust environment for testing the performance of our proposed Implicit Neural Representation (INR) method. This dataset is part of the Coupled Model Intercomparison Project (CMIP) under the World Climate Research Program (WCRP), which supports standardized climate modeling experiments across research groups. The U.S. Department of Energy's Program for Climate Model Diagnosis and Intercomparison (PCMDI), in collaboration with the Global Organization for Earth System Science Portals, maintains and distributes this data through pcmdi.llnl.gov.

Specifically, we used monthly averaged global surface temperature fields, representing an atmospheric variable, for model evaluation. To increase the complexity of the task, the seasonal cycle has been removed from the data. The resulting data set has dimensions of 1024 time steps, 192 latitudinal points, and 288 longitudinal points.

D.4 Sea Surface Temperature

The fourth dataset we use comes from the Group for High Resolution Sea Surface Temperature (GHRSST) and is known as the Level 4 Multiscale Ultrahigh Resolution (MUR) Global Sea Surface Temperature (SST) dataset, which is available through NASA's Earthdata platform. It includes both retrospective data (available with a four-day delay) and near-real-time data (with a one-day delay), created using an optimal interpolation technique based on wavelet functions, mapped globally at a fine spatial resolution of 0.01 degrees. The version 4 MUR L4 analysis is based on nighttime GHRSST Level 2P skin and subskin SST observations from several instruments. The dataset incorporates observations from multiple sources, including the NASA AMSR-E, JAXA AMSR2 on GCOM-W1, MODIS on Aqua and Terra, U.S. Navy WindSat, NOAA AVHRR, and in situ SST from NOAA iQuam. Ice data are from the OSI SAF High Latitude Processing Center.

For this study, we focus on a subset of the dataset covering the western North Atlantic, specifically a region bounded by coordinates (34°N,-69°E) in the southwest and (44°N,-59°E) in the northeast. Unlike atmospheric data, oceanic fields exhibit unique spatial and temporal patterns. These range from small-scale features like eddies and thermal fronts to large-scale ocean currents and gyres. The GHRSST Level 4 MUR Global SST dataset captures this complexity through its ultra-high resolution, satellite-derived measurements.

This richness and level of detail make the dataset a strong benchmark for evaluating how well our method performs with real-world, high-resolution ocean data. We use one year of daily SST observations—from August 20, 2022, to August 20, 2023—focused on the Gulf Stream region. NASA provides this dataset freely as part of its Earth Science Data Systems (ESDS) Program, accessible through earthdata.nasa.gov.

E Additional Model Analysis

E.1 Qualitative visualizations for Hierarchical vs Sequential

To further understand how hierarchical modeling utilizes successive blocks, we visualize both the cumulative reconstruction and the individual block-wise representations in Figure 5 and Figure 6.

Figure 5 shows individual hierarchical blocks exhibit markedly different spatial patterns, indicating that different stages preserve complementary structures rather than repeatedly encoding the same representation. Conversely, the sequential block outputs remain visually similar across depth, sug-

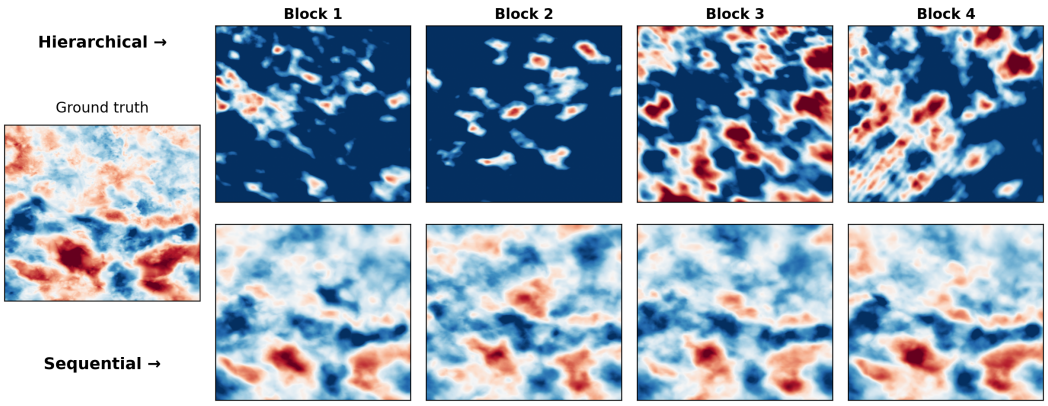

Figure 5: Block-wise representations for hierarchical and sequential modeling for SST - Task 3, 0.5% sampling ratio. Hierarchical blocks capture distinct spatial structures, whereas sequential block outputs remain comparatively similar across depth, indicating greater representation redundancy.

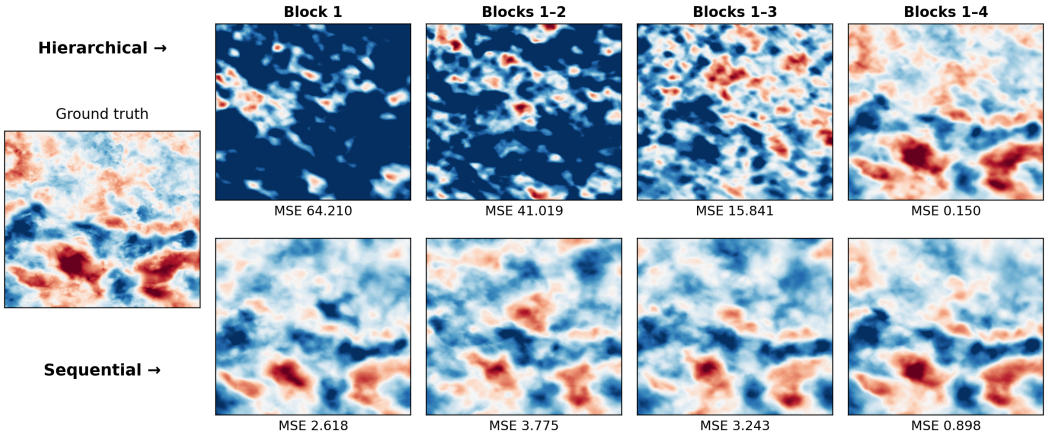

Figure 6: Progressive reconstruction across blocks for hierarchical and sequential modeling for SST - Task 3, 0.5% sampling ratio (with per frame MSE (x 10^{-3})). Hierarchical fusion progressively integrates complementary block representations, reducing the reconstruction error as additional blocks are incorporated, while sequential processing shows less consistent improvement across depth.

gesting greater redundancy between successive representations. Figure 6 provides further insight into this behavior, that the hierarchical formulation progressively improves the reconstruction as additional block representations are fused, reducing the Per frame MSE from 64.21 using the first block to 0.15 after all four blocks. This indicates that the intermediate blocks provide complementary information that is progressively integrated toward the final field. In contrast, the sequential model exhibits less consistent improvement across depth and retains a higher final error of 0.898.

Together with the effective-rank analysis in Fig. 4, these observations show that hierarchical fusion preserves diverse information across blocks and progressively combines it for accurate physical field reconstruction.

E.2 Qualitative visualizations for Rational vs MLP

To further examine the representation capability of rational functions for physical field reconstruction, we qualitatively compare rational-based models with their corresponding MLP-based counterparts. Although MLPs are universal function approximators in theory (Hornik et al., 1989), in practice they can struggle to efficiently represent periodic and complex functional patterns that fre-

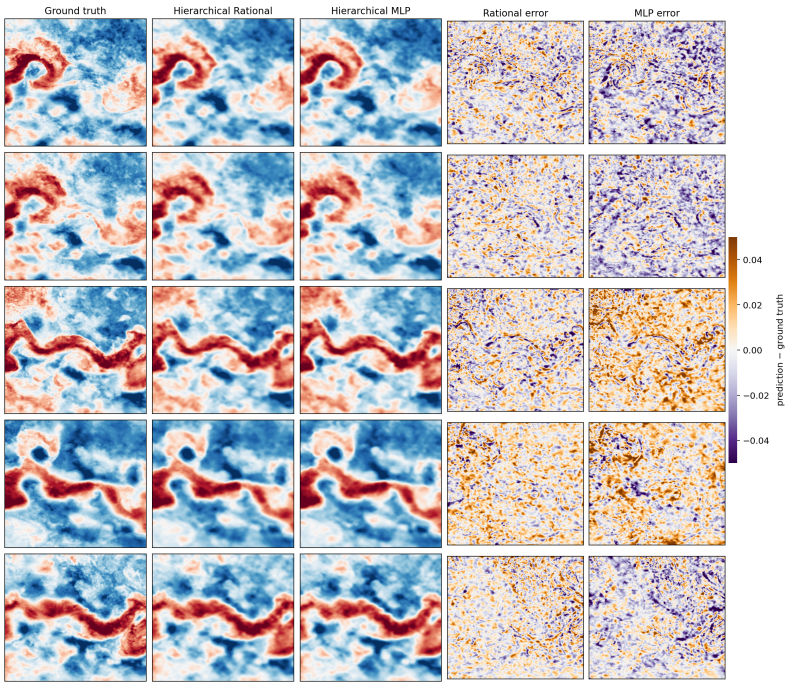

Figure 7: Qualitative comparison of rational and MLP transformations within the hierarchical architecture. From left to right, the columns show the ground-truth field, hierarchical rational reconstruction, hierarchical MLP reconstruction, and the corresponding signed error maps for the rational and MLP variants. Both models recover the overall spatial organization of the physical field, while the rational formulation more faithfully represents the underlying field variations. The corresponding error maps show weaker and less spatially structured residuals for the rational model, highlighting its greater representation capability compared with the MLP counterpart.

quently arise in physical field data (Yu et al., 2024; Yang & Wang, 2024). In contrast, learnable rational functions provide a richer class of basis functions and can approximate a broader family of functions more efficiently (Boullé et al., 2020; Walsh, 1935; Baker Jr & Gammel, 1961). The following comparisons illustrate how this difference in representation capability translates to the reconstructed physical fields.

Figure 7 compares rational and MLP transformations within the hierarchical architecture. Both formulations recover the dominant spatial organization of the ground-truth fields. However, the rational model more faithfully reproduces the spatial variations present across the field, while the MLP counterpart exhibits larger deviations from the ground truth in several regions. This distinction is more clearly visible in the corresponding signed error maps, where the rational formulation generally produces weaker and less spatially structured residuals. The result indicates that the rational basis provides a more expressive representation of the underlying physical field within the same hierarchical framework.

A similar behavior is observed in the sequential architecture, as shown in Figure 8. While the MLP variant captures the overall field configuration, its error maps exhibit stronger and more spatially coherent discrepancies across the domain. In comparison, the rational formulation follows the ground-truth field more closely and produces residuals that are generally more localized around zero. Since the same trend appears under both hierarchical and sequential processing, the improvement is not specific to a particular latent organization, but rather reflects the greater representation capability introduced by the rational basis functions.

Overall, these qualitative comparisons support the use of learnable rational functions as a more expressive alternative to conventional MLP transformations for modeling complex physical fields. Their broader approximation capability enables the model to represent the underlying field dynamics

more faithfully, which is consistently reflected in both the reconstructed fields and their corresponding error distributions.

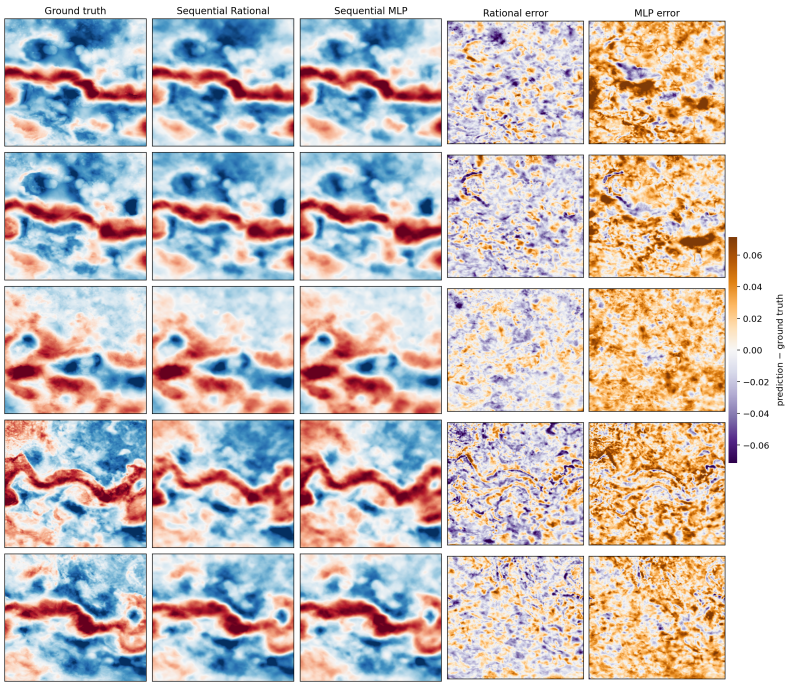

Figure 8: Qualitative comparison of rational and MLP transformations within the sequential architecture. From left to right, the columns show the ground-truth field, sequential rational reconstruction, sequential MLP reconstruction, and their corresponding signed error maps. Although both models capture the overall field configuration, the rational formulation more closely follows the ground-truth physical field and produces weaker residual discrepancies. The comparison further demonstrates the greater expressive capability of rational basis functions over conventional MLP transformations for physical field representation.

E.3 Computational Analysis

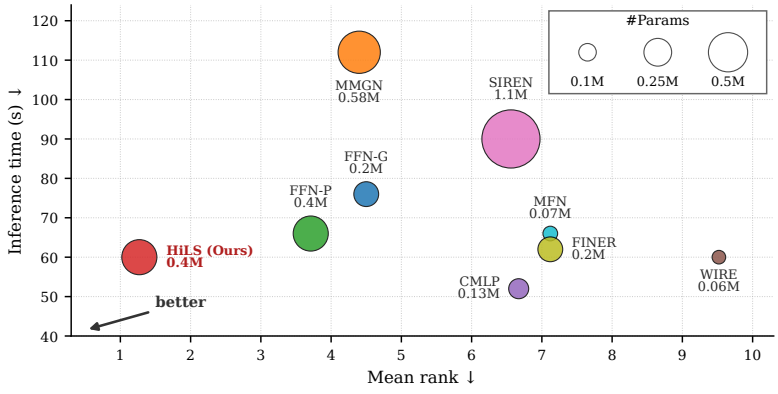

Figure 9: Inference time vs. model ranking plot for various INR models, with circle size indicating parameter count. HiLS achieves the best ranking with the comparable inference time and similar model size, demonstrating superior efficiency and performance compared to other baselines for SST dataset Task 1 with 0.1% sampling ratio.

Figure 9 illustrates the trade-off between inference time and model ranking for various implicit neural representation (INR) models, where the circle size corresponds to the number of parameters. Models such as SIREN and MMGN, despite having larger parameter counts, exhibit significantly higher inference times, making them less efficient. On the other hand, lightweight models like WIRE and MFN achieve faster inference but at the cost of lower ranking. `HiLS` (ours), with a moderate parameter count of 0.4M, achieves the best overall ranking while maintaining a low inference time comparable to smaller models. This demonstrates `HiLS's` superior balance of efficiency and performance, outperforming existing baselines by offering both speed and representational power.

E.4 Statistical Analysis

Table 4: Quantitative Metrics of `HiLS` obtained on five runs.

	SSH		CHL	
	MSE(10^{-3})	SSIM	MSE(10^{-3})	SSIM
0	0.724	0.9589	7.258	0.8253
1	0.711	0.9578	7.322	0.8293
2	0.734	0.9592	7.270	0.8278
3	0.723	0.9587	7.458	0.8302
4	0.715	0.9580	7.370	0.8295
mean	0.721	0.9585	7.336	0.8284
std deviation	0.0079	0.0006	0.0073	0.0019

Table 4 summarizes the quantitative performance of `HiLS` over five independent runs for SSH (10 % Task 3), and CHL (10 %, Task 4),. The results show that across five independent runs, HiLS exhibits very low run-to-run variability, indicating strong stability and reproducibility. The standard deviations for all metrics particularly SSIM (on the order of 10^{-4}–10^{-3}) are negligible, implying that performance fluctuations due to random initialization or training nondeterminism are minimal. The mean MSE values closely match those reported in Tables 1, further corroborating the model's robustness. Overall, these results show that `HiLS` delivers reliable predictions across trials.

E.5 Sensitivity Studies

We performed HiLS's sensitivity studies to two core architectural hyperparameters on GST (task 3, 5% sampling) and SST (task 2, 0.10% sampling), with results summarized in Tables 5 and 6. Adding hierarchical blocks yields monotonic gains on both datasets (MSE: $2.378 \rightarrow 2.187$ on GST, $1.366 \rightarrow 1.098$ on SST), validating the hierarchical-scale design. The latent dimension, by contrast, proves to be a largely inert hyperparameter: sweeping it from 128 to 512 shifts GST MSE by under 1% and SSIM by less than 0.001, while on SST the smallest latent (64) is marginally best, with larger values trading a slight MSE increase for negligible SSIM change. Taken together, these results indicate that HiLS is robust to architectural choices.

Table 5: Sensitivity of HiLS to the number of hierarchical blocks. Unobserved locations MSE ($\times 10^{-3}$, lower is better) and SSIM on the reconstructed field (higher is better). Best per column in bold.

	GST task3, 5%			SST task2, 0.10%	
Blocks	MSE	SSIM	Blocks	MSE	SSIM
2	2.378	0.7675	1	1.366	0.8953
3	2.228	0.7821	2	1.116	0.9022
4	**2.187**	**0.7863**	4	**1.098**	**0.9068**

Table 6: Sensitivity of HiLS to the Latent dim. Unobserved locations MSE ($\times 10^{-3}$, lower is better) and SSIM on the reconstructed field (higher is better). Best per column in bold.

	GST task3, 5%			SST task2, 0.10%	
Latent dim	MSE	SSIM	Latent dim	MSE	SSIM
128	2.248	**0.7826**	64	**1.116**	0.9022
256	**2.228**	0.7821	128	1.136	0.9007
512	2.233	0.7823	256	1.185	**0.9063**

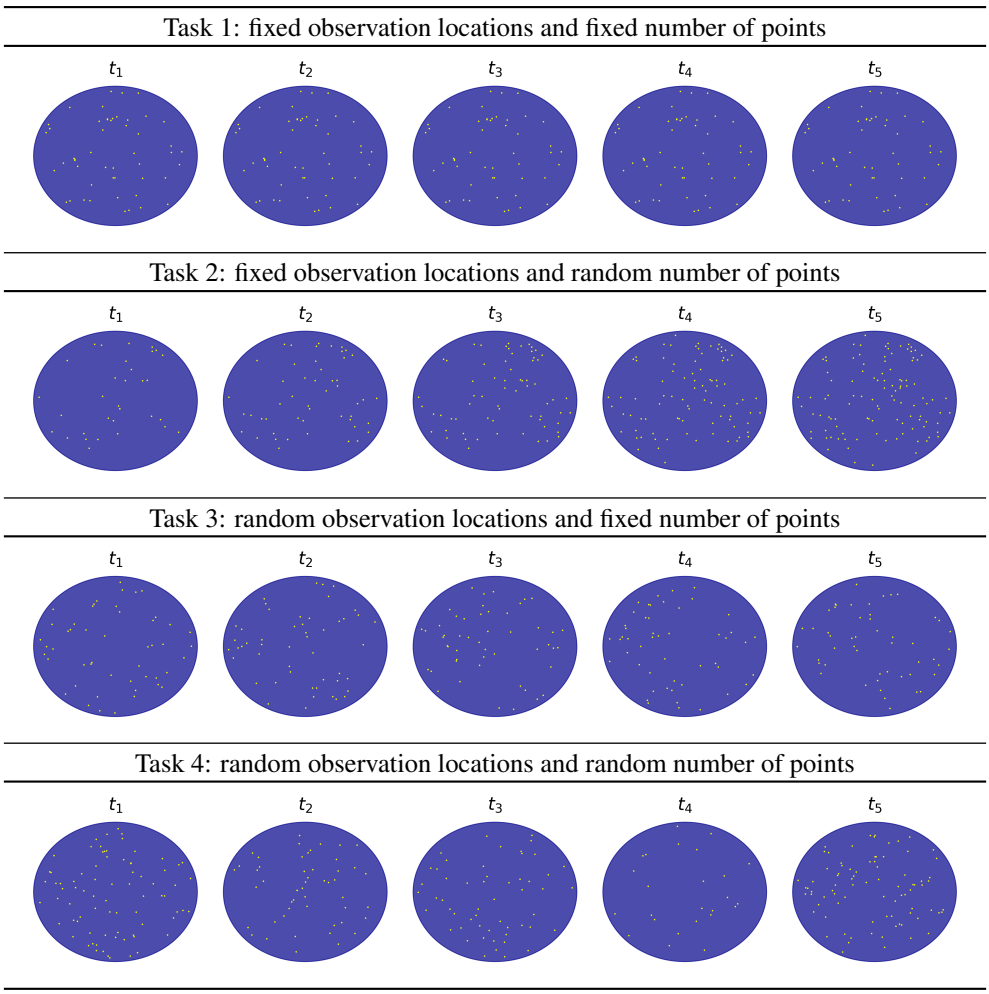

Figure 10: Diagram illustrating different field reconstruction tasks arranged in order of increasing complexity for model evaluation. Each task differs based on the degree of randomness in the number and spatial distribution of data points. These tasks allow the model evaluation over varying sampling conditions.

F Additional Implementation Details

F.1 Tasks

To test how well our model handles different conditions, we designed a series of experiments that gradually increase in complexity. These include: *Task 1*, where both the number and location of input points are fixed; *Task 2*, which uses a random number of points but keeps their locations fixed; *Task 3*, which fixes the number of points but places them at random locations; and *Task 4*, where both the number and locations of points are randomized as shown in Figure 10.

F.2 BASELINE IMPLEMENTATION

SIREN (Sitzmann et al., 2020) uses periodic activation functions with neural networks to represent signals and is ideally suited for representing complex natural signals and their derivatives.

FFN (Tancik et al., 2020) introduced Fourier feature mappings to enable multilayer perceptrons (MLPs) to learn high-frequency functions in low dimensions, this has two variants with positional encoding (P) and Gaussian encoding (G).

MMGN (Luo et al., 2024) factorizes spatio-temporal variability into spatial and temporal components using the separation of variables technique and learns relevant basis functions from sparsely sampled irregular data points to represent represent continuous data. See Section 2.1 of the Supplementary Material for more details of the baselines. In the official implementation, however, the observed values enter only through the training loss. The network's inputs are the query coordinates and a per-snapshot latent code optimized by auto-decoding, so observation content is never passed as an input to the model. HiLS instead encodes the sparse values v(K) jointly with their locations K as an explicit input. We use the released MMGN code for all MMGN results.

WIRE (Saragadam et al., 2023) Inspired by Harmonic analysis, Wavelet Implicit Neural Representation (WIRE) uses complex gabor wavelet as its activation function that is well known to be optimally concentrated in space-frequency and effective biases for representing RGB images.

MFN (Fathony et al., 2020) The authors introduced multiplicative filter networks, namely GaborNet and FourierNet, as alternative functional representations of deep neural networks. These networks are constructed by sequentially applying sinusoidal or Gabor filter-based linear multiplicative functions. The authors demonstrate that, in certain scenarios, these proposed networks can surpass the performance of contemporary deep networks utilizing ReLu activations.

CMLP (Ramasinghe & Lucey, 2022) shifts the focus in coordinate-MLPs away from periodic activations and positional embeddings towards a larger class of robust activations, enabling simpler, more powerful neural architectures for signal representation.

INCODE (Kazerouni et al., 2024) enhances implicit neural representations by dynamically controlling sinusoidal activations through a harmonizer–composer architecture that adapts to task-specific priors. This enables robust, high-fidelity reconstruction across audio, images, 3D shapes, NeRFs, and inverse problems such as denoising, super-resolution, and CT reconstruction.

FINER (Liu et al., 2024a) introduces variable-periodic activation functions that flexibly tune an INR's supported frequency set by adjusting network biases, enabling better representation of complex multi-frequency signals. This leads to improved performance across 2D image fitting, 3D SDFs, and 5D NeRFs compared to existing INR methods.

In addition, We have also compared with Standard methods like DeepONet (Lu et al., 2021), UNet (Ronneberger et al., 2015) and Fourier Neural Operators (Li et al., 2020).

All the models were trained on L2 loss for 200 epochs, using the learning rate of 0.001 with the weight decay of 0.00001 and AdamW optimizer. All experiments were conducted using PyTorch Lightning on a single NVIDIA RTX 5000 Ada Generation GPU with 32 GB of VRAM. The implementation of Rational functions is inspired by (Yang & Wang, 2024).

Table 7: `HiLS` hyperparameters of the reported models.

Data	Sampling ratio	Num Blocks	Latent dim	Hidden dim	Num Layers
SSH	10%, 25%, 50%	2	256	384	3
CHL	10%, 25%, 50%	3	256	384	3
GST	5%	3	256	384	3
	25%, 50%	2	256	256	3
SST	0.10%	2	64	128	4
	0.30%, 0.50%	4	256	256	3

Table 8: Comparison of HiLS with other Non-INR baselines on GST and SST datasets at different sampling ratios for different tasks using Full MSE(10^{-3}) metrics. Best results are in **bold**.

Model	GST		SST	
	Task 1	Task 3	Task 1	Task 3
Sampling ratio = 5% (GST), 0.1% (SST)				
HiLS	**3.49**	**2.14**	**0.83**	**0.49**
UNet	78.2	83.5	65.9	67.14
FNO	20.45	19.65	18.73	19.60
DeepONet	21.27	21.19	20.80	21.09
Senseiver	42.50	9.09	83.12	79.23
ConvCNP	20.92	21.40	8.59	8.62
OFormer	9.08	6.90	1.827	1.54
Sampling ratio = 25% (GST), 0.3% (SST)				
HiLS	**2.05**	**1.93**	**0.34**	**0.22**
UNet	23.32	20.35	20.25	19.93
FNO	2.84	2.86	2.34	2.18
DeepONet	8.22	7.86	8.08	8.43
Senseiver	33.04	7.09	20.74	6.74
ConvCNP	20.19	21.14	8.09	7.91
OFormer	6.27	6.17	1.46	1.47

F.3 Sampling

Sampling for different tasks is applied differently for Task 1 (fixed observation locations and fixed number of points) and Task 3 (random observation locations and fixed number of points). Since the number of points is fixed, the sampling ratio is taken as specified. However for Task 2 (fixed observation locations and random number of points) and Task 4 (random observation locations and random number of points) since the number of data points varies, in that case we give a range of sampling values as indicated in Table 9 to ensure fair evaluation of the model performance.

Table 9: Minimum and maximum sampling ratios used during training for all the datasets. These sampled input locations are used to train the model for continuous field reconstruction from sparse observations.

Data	GST			SST			SSH			CHL		
Ratios	5%	25%	50%	0.10%	0.30%	0.50%	10%	25%	50%	10%	25%	50%
Min	1	1	1	0.01	0.01	0.01	5	5	5	5	5	5
Max	5	25	50	0.10	0.30	0.50	10	25	50	10	25	50

F.4 Model Architecture and Hyperparameters

Model architecture design and hyperparameter optimization play a vital role in model performance, especially in the deep learning domain. To ensure a fair comparison between the architectures, each model was tuned on various ranges of hyperparameters. Table 10 contains the list of model configurations used to train the models. SIREN, FFN+P/G, MMGN, HiLS. Our best hyperparameters for different datasets are listed in Table 7. We initialize Gabor parameters as discussed in Section A.2 as $\boldsymbol{\mu} \sim \mathcal{N}(2, 1)$ in the spatial domain, $\gamma \sim Gamma(\alpha, \beta)$, $W_g \sim \mathcal{N}(0, \beta^2)$, and $\mathbf{b}_g \sim U[-\pi, \pi]$ following WIRE (Saragadam et al., 2023) and MFN (Fathony et al., 2020).

Table 10: Hyperparameter configurations used for training different model architectures. Column 1 lists the various models used in this project. Column 2 specifies the hyperparameters and architectural variants associated with each model, and Column 3 depicts the range of values considered for each hyperparameter during training. These settings were used for model evaluation and optimization across the GST and SST datasets.

Model	Hyperparameters	Values
SIREN	Width	{128, 192, 256, 384, 512}
	Depth	{3, 4, 5, 6, 7}
	Weight scale w_0	{1, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 100}
FFN+P	Width	{128, 192, 256, 384, 512}
	Depth	{3, 4, 5, 6, 7}
	Frequency constant	{30, 40, 50, 60, 70, 80, 90, 100, 110, 120}
	Frequency number	{150, 160, 170, 180, 190, 200, 210, 220, 230, 240, 250}
	Non-linear activation	{ReLU, Sigmoid, Tanh, SELU, GELU, Swish}
FFN+G	Width	{128, 192, 256, 384, 512}
	Depth	{3, 4, 5, 6, 7}
	Gaussian σ	{1, 3, 5, 7, 10, 20, 30, 40, 50}
	Encode size	{64, 128, 256, 512}
	Non-linear activation	{ReLU, Sigmoid, Tanh, SELU, GELU, Swish}
MMGN	Width	{128, 192, 256, 384, 512}
	Depth	{3, 4, 5, 6, 7}
	Input scale	{128, 256, 512}
	Latent size	{1, 2, 4, 8, 16, 32, 64, 128, 256, 512}
	Latent initialization	{Uniform, Gaussian, Ones, Zeros, Orthogonal}
WIRE	Width	{128, 256, 512}
	Depth	{1, 2, 3}
FINER	Width	{128, 256, 512}
	Depth	{2, 3, 4, 5}
MFN	Width	{64, 128, 256, 512}
	Depth	{2, 3, 4}
	Input Scale	{64, 128, 256, 512}
DeepONet	Input dimension	{128, 256, 512}
	Branch layers	{2, 3, 4, 5}
	Trunk layers	{2, 3, 4, 5}
INCODE	Width	{128, 192, 256, 384, 512}
	Depth	{3, 4, 5, 6, 7}
`HiLS`	Latent dimension	{64, 128, 256}
	Blocks	{2, 3, 4}
	MLP Layers	{2, 3, 4}
	Hidden dim	{64, 128, 256, 384}

G Additional Results

G.1 Quantitative Results

G.1.1 Non-INR Baselines

We compare `HiLS` against six non-INR architectures on GST and SST (Tasks 1 and 3, two sampling ratios each). They fall into two groups.

Grid-based models. UNet (Ronneberger et al., 2015), FNO (Li et al., 2020) and DeepONet (Lu et al., 2021) operate on fixed discretizations and cannot ingest a variable set of scattered observations directly. We therefore provide the full spatial grid as input, with unobserved locations as

Table 11: Dense MSE ($\times 10^{-3}$) across three sampling ratios for each dataset. The error is computed exclusively over all locations. The final column reports the mean rank of each model across the 16 evaluation settings within the block, where a lower rank indicates better performance.

Data	SSH				CHL				GST				SST				Mean
Task	T1	T2	T3	T4	T1	T2	T3	T4	T1	T2	T3	T4	T1	T2	T3	T4	rank
Model	*Sampling ratio = 10%*				*Sampling ratio = 10%*				*Sampling ratio = 5%*				*Sampling ratio = 0.1%*				
HiLS	**4.03**	**4.08**	**0.66**	**0.94**	**15.55**	**15.69**	**5.70**	**6.81**	**3.49**	**4.53**	**2.14**	**3.31**	**0.83**	**1.12**	0.49	**0.76**	**1.06**
MMGN	17.30	13.93	3.07	3.84	38.00	34.46	7.95	9.22	4.77	5.22	3.08	3.71	1.01	1.26	**0.47**	1.01	5.06
INCODE	5.68	5.63	2.08	2.06	19.12	19.13	8.15	8.18	16.11	16.85	14.75	14.67	1.37	1.56	0.82	0.94	4.19
SIREN	6.11	6.62	2.95	2.98	18.70	19.02	9.43	9.53	18.38	18.57	18.02	17.98	1.92	2.03	1.40	1.50	5.94
FINER	4.81	4.95	4.62	4.66	18.71	19.09	17.71	18.19	17.02	17.48	15.79	15.77	2.44	2.70	1.91	1.98	6.25
WIRE	5.50	5.65	5.15	5.20	24.08	26.29	24.44	24.26	21.66	21.98	22.28	22.29	5.72	5.55	5.34	5.39	8.88
FFN-P	5.61	5.44	1.15	1.16	22.15	22.97	6.87	6.87	9.88	10.49	8.12	8.10	2.78	3.17	0.71	0.82	3.94
FFN-G	6.34	6.13	1.95	2.08	21.57	22.30	7.63	7.69	18.17	20.06	14.06	14.06	8.79	9.99	1.10	1.59	5.81
MFN	7.22	7.00	3.03	3.08	19.90	21.04	10.48	10.44	19.83	21.37	16.05	16.06	16.31	12.58	1.55	1.73	7.62
CMLP	5.61	5.65	3.12	3.12	18.04	18.84	10.97	10.53	21.36	22.14	16.96	16.78	2.01	2.22	1.28	1.31	6.25
	Sampling ratio = 25%				*Sampling ratio = 25%*				*Sampling ratio = 25%*				*Sampling ratio = 0.3%*				
HiLS	**2.67**	**2.78**	**0.22**	**0.42**	**10.43**	**10.76**	**2.69**	**4.58**	**2.05**	**2.44**	**1.93**	**2.28**	**0.34**	**0.51**	**0.22**	0.38	**1.06**
MMGN	6.46	6.43	1.92	2.53	16.43	15.78	6.64	7.39	2.99	3.23	2.78	2.96	0.40	0.54	0.26	0.44	4.56
INCODE	3.33	3.56	1.68	1.72	11.03	12.22	7.39	7.48	13.89	14.11	13.60	13.80	0.75	0.89	0.59	0.69	3.88
SIREN	3.51	4.04	3.09	3.05	11.33	12.45	8.90	8.94	17.64	17.65	17.55	17.59	1.28	1.43	1.21	1.36	6.81
FINER	4.06	4.21	3.97	4.04	14.60	14.72	13.94	13.44	15.38	15.47	15.03	15.26	1.33	2.69	1.17	1.56	7.62
WIRE	5.26	5.05	5.06	5.18	24.58	25.72	23.17	23.50	21.51	21.25	21.79	21.00	6.12	5.32	4.80	5.62	9.88
FFN-P	3.98	4.00	0.79	0.83	15.19	15.75	5.52	5.61	7.39	7.35	7.09	6.99	0.75	0.98	0.34	**0.37**	3.62
FFN-G	3.53	3.71	1.28	1.36	12.65	13.42	6.40	6.29	13.41	13.67	12.93	13.31	1.11	2.39	0.50	0.88	4.19
MFN	3.75	4.16	2.88	2.82	12.25	13.34	9.96	10.04	15.65	15.99	15.19	15.38	2.87	3.81	1.08	1.26	6.75
CMLP	3.97	4.24	2.70	3.02	12.28	13.65	8.53	11.84	15.65	16.78	16.37	16.49	1.12	1.21	1.01	1.11	6.62
	Sampling ratio = 50%				*Sampling ratio = 50%*				*Sampling ratio = 50%*				*Sampling ratio = 0.5%*				
HiLS	**1.27**	**1.40**	**0.13**	**0.24**	**5.82**	**6.55**	**1.65**	**2.79**	**1.90**	**2.17**	**1.88**	**2.12**	**0.25**	**0.35**	**0.18**	**0.28**	**1.00**
MMGN	2.18	2.92	1.63	1.98	7.65	9.15	6.27	6.74	2.81	2.97	2.76	2.87	0.30	0.41	0.24	0.35	3.12
INCODE	1.95	2.29	1.59	1.46	7.97	9.77	7.10	7.13	13.66	13.72	13.76	13.91	0.60	0.74	0.53	0.63	4.38
SIREN	2.85	3.46	2.67	3.07	9.22	9.96	8.47	8.73	17.42	17.72	17.50	17.50	1.18	1.37	1.17	1.33	7.62
FINER	3.63	3.43	3.56	3.32	12.05	14.62	11.96	13.92	14.96	15.31	15.03	15.21	1.15	2.22	1.06	1.58	8.00
WIRE	5.08	4.98	5.05	5.17	23.83	23.86	24.11	24.07	21.52	21.36	21.58	21.27	4.64	5.03	4.54	5.64	10.00
FFN-P	2.04	2.33	0.69	0.71	8.65	9.57	5.07	5.16	7.01	6.89	6.97	6.79	0.47	0.58	0.29	0.32	3.00
FFN-G	1.83	2.06	1.09	1.03	7.75	8.89	6.01	5.87	13.03	13.05	12.62	13.06	0.67	0.87	0.54	0.59	3.50
MFN	2.99	3.27	2.70	2.79	10.37	11.02	9.85	9.75	15.34	15.47	15.26	15.26	1.56	1.90	1.14	1.09	7.38
CMLP	2.95	3.43	2.58	2.88	10.75	11.35	9.59	11.17	16.08	16.61	14.51	16.22	1.06	1.08	0.95	0.86	7.00

masking, and compute the loss only at the sparse locations where observations are available. Inference produces the full grid directly.

Observation-conditioned models. ConvCNP (Gordon et al., 2019), Senseiver (Santos et al., 2022) and OFormer (Li et al., 2022) encode the sparse observation values together with their locations, as HiLS does. Senseiver was originally trained with dense fields, and OFormer is typically supervised on the full output mesh. Here, all three are retrained under our sparse-only protocol: the loss is computed only at observed locations, and no dense field is used at any stage.

Results. Table 8 shows that HiLS achieves the lowest MSE in every setting. OFormer is the strongest non-INR baseline, yet its error remains 2.6–4.3$\times$ higher than that of HiLS. ConvCNP is 6–36$\times$ worse; we attribute this to its gridded, translation-equivariant prior, which conflicts with the strongly location-dependent statistics of geophysical fields. Senseiver performs markedly better on Task 3 than on Task 1 (e.g., 9.09 vs. 42.50 on GST at 5%). This is consistent with its reliance on dense supervision: with fixed sensors (Task 1), its decoder is only ever supervised at the same locations, whereas randomly placed sensors (Task 3) expose it to the more different points over the course of training. HiLS performs well in both regimes.

G.1.2 Extended Results

To evaluate the performance of our proposed model quantitatively, we conducted a comprehensive quantitative analysis using MSE over the complete dense field, as well as the Pearson correlation coefficient and the Structural Similarity Index Measure (SSIM) as our primary metrics. These metrics are standard benchmarks for assessing image reconstruction quality, where higher values for both SSIM indicate a higher fidelity and perceptual quality of the generated result compared to the ground truth. The detailed results of our comparison are presented in Tables 11, 12, 13, and 14. These tables document the performance of our model, HiLS, against several leading baseline models (SIREN, FFN-P, FFN-G, MMGN, CMLP, MFN, FINER, INCODE and WIRE) across four distinct datasets, multiple tasks, and a wide range of data sampling ratios.

Table 12: SSIM of the reconstructed field. The last column is the mean rank of that model over the 16 cells of the block.

Data	SSH				CHL				GST				SST				Mean
Task	T1	T2	T3	T4	T1	T2	T3	T4	T1	T2	T3	T4	T1	T2	T3	T4	rank
Model	*Sampling ratio = 10%*				*Sampling ratio = 10%*				*Sampling ratio = 5%*				*Sampling ratio = 0.1%*				
HiLS	**0.812**	**0.811**	**0.959**	**0.944**	**0.632**	**0.629**	**0.857**	**0.829**	**0.714**	0.674	**0.803**	0.751	**0.912**	**0.905**	0.924	**0.917**	**1.19**
MMGN	0.740	0.720	0.871	0.854	0.610	0.613	0.760	0.734	0.687	**0.683**	0.775	**0.763**	0.908	0.905	**0.933**	0.908	3.38
INCODE	0.769	0.768	0.876	0.879	0.599	0.621	0.744	0.743	0.458	0.423	0.537	0.538	0.902	0.899	0.920	0.915	3.69
SIREN	0.774	0.759	0.839	0.838	0.619	0.612	0.707	0.706	0.475	0.472	0.484	0.484	0.904	0.901	0.911	0.909	4.50
FINER	0.754	0.746	0.758	0.755	0.482	0.479	0.492	0.478	0.433	0.412	0.496	0.498	0.868	0.746	0.882	0.868	7.38
WIRE	0.720	0.708	0.737	0.732	0.409	0.383	0.396	0.393	0.412	0.402	0.416	0.417	0.857	0.862	0.851	0.862	9.12
FFN-P	0.742	0.744	0.921	0.921	0.597	0.586	0.800	0.801	0.477	0.449	0.602	0.602	0.657	0.644	0.853	0.835	5.25
FFN-G	0.735	0.738	0.878	0.869	0.609	0.593	0.752	0.748	0.382	0.342	0.517	0.517	0.753	0.758	0.893	0.864	6.06
MFN	0.738	0.715	0.829	0.825	0.565	0.554	0.645	0.646	0.369	0.357	0.489	0.489	0.763	0.763	0.884	0.875	7.75
CMLP	0.768	0.759	0.837	0.839	0.580	0.570	0.676	0.684	0.243	0.245	0.480	0.474	0.883	0.880	0.907	0.905	6.69
	Sampling ratio = 25%				*Sampling ratio = 25%*				*Sampling ratio = 25%*				*Sampling ratio = 0.3%*				
HiLS	**0.869**	**0.864**	**0.983**	**0.972**	0.756	**0.747**	**0.918**	**0.879**	**0.784**	**0.764**	**0.793**	**0.781**	**0.931**	0.926	0.943	0.935	**1.25**
MMGN	0.810	0.800	0.901	0.882	**0.774**	0.735	0.792	0.770	0.766	0.753	0.784	0.779	0.929	**0.927**	**0.945**	**0.937**	2.81
INCODE	0.843	0.834	0.894	0.894	0.717	0.697	0.764	0.763	0.567	0.553	0.581	0.576	0.919	0.916	0.928	0.924	3.62
SIREN	0.829	0.812	0.835	0.839	0.691	0.670	0.721	0.720	0.493	0.488	0.492	0.490	0.913	0.909	0.914	0.912	6.25
FINER	0.783	0.775	0.786	0.784	0.552	0.554	0.562	0.573	0.533	0.511	0.548	0.531	0.868	0.762	0.899	0.877	8.25
WIRE	0.725	0.733	0.735	0.731	0.399	0.387	0.410	0.408	0.413	0.414	0.416	0.418	0.857	0.866	0.865	0.860	9.69
FFN-P	0.817	0.809	0.938	0.936	0.709	0.696	0.827	0.825	0.621	0.591	0.637	0.645	0.837	0.803	0.924	0.917	4.12
FFN-G	0.837	0.819	0.906	0.898	0.703	0.683	0.777	0.781	0.552	0.533	0.557	0.558	0.903	0.838	0.928	0.902	4.56
MFN	0.807	0.786	0.836	0.838	0.624	0.609	0.660	0.656	0.501	0.481	0.525	0.519	0.877	0.848	0.905	0.896	7.62
CMLP	0.815	0.793	0.855	0.839	0.662	0.635	0.736	0.645	0.515	0.454	0.510	0.509	0.907	0.900	0.912	0.909	6.81
	Sampling ratio = 50%				*Sampling ratio = 50%*				*Sampling ratio = 50%*				*Sampling ratio = 0.5%*				
HiLS	**0.932**	**0.926**	**0.987**	**0.980**	**0.857**	**0.838**	**0.937**	**0.914**	**0.795**	**0.780**	**0.795**	**0.786**	**0.939**	**0.933**	0.945	0.939	**1.12**
MMGN	0.892	0.865	0.913	0.898	0.814	0.786	0.803	0.788	0.778	0.770	0.786	0.782	0.937	0.933	**0.947**	**0.941**	2.56
INCODE	0.890	0.875	0.897	0.904	0.758	0.697	0.766	0.770	0.582	0.573	0.589	0.575	0.926	0.922	0.930	0.927	4.25
SIREN	0.846	0.825	0.851	0.837	0.720	0.705	0.733	0.723	0.497	0.492	0.495	0.491	0.914	0.910	0.915	0.912	6.81
FINER	0.802	0.813	0.806	0.817	0.613	0.541	0.613	0.553	0.548	0.522	0.551	0.521	0.884	0.791	0.907	0.872	8.31
WIRE	0.734	0.739	0.735	0.733	0.408	0.398	0.401	0.401	0.415	0.416	0.416	0.416	0.865	0.861	0.867	0.851	9.94
FFN-P	0.899	0.880	0.943	0.942	0.786	0.767	0.839	0.833	0.647	0.624	0.642	0.639	0.886	0.867	0.933	0.929	3.25
FFN-G	0.887	0.877	0.912	0.917	0.760	0.742	0.785	0.791	0.567	0.562	0.570	0.561	0.916	0.907	0.924	0.925	4.38
MFN	0.832	0.820	0.843	0.840	0.654	0.643	0.661	0.666	0.523	0.509	0.524	0.522	0.895	0.884	0.902	0.902	7.56
CMLP	0.849	0.827	0.859	0.845	0.681	0.669	0.705	0.661	0.513	0.490	0.536	0.519	0.909	0.908	0.914	0.913	6.81

Table 13: Pearson correlation on unobserved locations. The last column is the mean rank of that model over the 16 cells of the block.

Data	SSH				CHL				GST				SST				Mean
Task	T1	T2	T3	T4	T1	T2	T3	T4	T1	T2	T3	T4	T1	T2	T3	T4	rank
Model	*Sampling ratio = 10%*				*Sampling ratio = 10%*				*Sampling ratio = 5%*				*Sampling ratio = 0.1%*				
HiLS	**0.671**	**0.677**	**0.954**	**0.934**	**0.815**	**0.819**	**0.937**	0.925	**0.920**	**0.898**	**0.951**	**0.926**	**0.949**	**0.934**	0.969	**0.954**	**1.12**
MMGN	0.379	0.422	0.785	0.742	0.670	0.695	0.916	0.903	0.890	0.881	0.930	0.916	0.945	0.928	**0.972**	0.942	5.00
INCODE	0.618	0.630	0.856	0.858	0.800	0.801	0.915	0.915	0.579	0.557	0.621	0.623	0.925	0.914	0.954	0.948	3.69
SIREN	0.605	0.590	0.792	0.789	0.804	0.803	0.902	0.900	0.491	0.483	0.505	0.507	0.896	0.890	0.925	0.920	5.62
FINER	0.629	0.619	0.649	0.645	0.794	0.788	0.807	0.802	0.544	0.529	0.586	0.586	0.869	0.852	0.898	0.894	6.44
WIRE	0.556	0.540	0.594	0.587	0.725	0.696	0.721	0.725	0.337	0.317	0.297	0.297	0.682	0.691	0.697	0.708	9.50
FFN-P	0.586	0.599	0.921	0.920	0.760	0.752	0.928	**0.927**	0.762	0.747	0.807	0.807	0.846	0.828	0.958	0.952	4.00
FFN-G	0.572	0.596	0.864	0.854	0.776	0.772	0.919	0.918	0.532	0.493	0.643	0.642	0.625	0.615	0.939	0.913	5.56
MFN	0.560	0.558	0.784	0.780	0.790	0.780	0.890	0.890	0.477	0.446	0.577	0.577	0.541	0.570	0.913	0.904	7.56
CMLP	0.595	0.590	0.775	0.775	0.805	0.798	0.884	0.889	0.430	0.394	0.545	0.551	0.894	0.883	0.931	0.928	6.50
	Sampling ratio = 25%				*Sampling ratio = 25%*				*Sampling ratio = 25%*				*Sampling ratio = 0.3%*				
HiLS	**0.755**	**0.774**	**0.984**	**0.969**	0.858	**0.870**	**0.967**	**0.948**	**0.952**	**0.943**	**0.956**	**0.948**	**0.978**	**0.968**	**0.987**	**0.977**	**1.25**
MMGN	0.601	0.626	0.860	0.819	0.809	0.828	0.930	0.922	0.930	0.925	0.937	0.933	0.975	0.967	0.984	0.973	4.44
INCODE	0.751	0.743	0.884	0.881	**0.875**	0.864	0.923	0.922	0.647	0.640	0.656	0.650	0.957	0.950	0.966	0.960	3.69
SIREN	0.735	0.701	0.782	0.784	0.874	0.863	0.907	0.907	0.520	0.518	0.523	0.522	0.930	0.923	0.935	0.927	6.81
FINER	0.692	0.682	0.708	0.702	0.840	0.838	0.851	0.857	0.599	0.596	0.611	0.603	0.927	0.854	0.937	0.917	7.56
WIRE	0.582	0.602	0.604	0.590	0.719	0.704	0.738	0.735	0.343	0.354	0.331	0.372	0.662	0.699	0.740	0.702	10.00
FFN-P	0.702	0.705	0.946	0.944	0.824	0.823	0.942	0.941	0.823	0.824	0.832	0.835	0.955	0.942	0.979	0.977	3.75
FFN-G	0.741	0.731	0.910	0.905	0.856	0.850	0.932	0.933	0.662	0.654	0.676	0.665	0.936	0.872	0.971	0.950	4.19
MFN	0.716	0.693	0.795	0.800	0.864	0.852	0.895	0.894	0.590	0.578	0.606	0.600	0.844	0.805	0.940	0.930	6.62
CMLP	0.689	0.673	0.808	0.785	0.863	0.849	0.910	0.875	0.588	0.548	0.565	0.561	0.939	0.934	0.945	0.940	6.69
	Sampling ratio = 50%				*Sampling ratio = 50%*				*Sampling ratio = 50%*				*Sampling ratio = 0.5%*				
HiLS	0.833	**0.870**	**0.990**	**0.982**	0.889	**0.909**	**0.979**	**0.966**	**0.956**	**0.950**	**0.957**	**0.952**	**0.985**	**0.977**	**0.988**	**0.982**	**1.38**
MMGN	0.798	0.773	0.883	0.857	0.895	0.890	0.934	0.929	0.935	0.932	0.938	0.935	0.981	0.974	0.985	0.978	3.25
INCODE	0.843	0.818	0.890	0.900	**0.909**	0.888	0.926	0.926	0.655	0.652	0.651	0.646	0.965	0.958	0.969	0.963	4.12
SIREN	0.786	0.739	0.813	0.781	0.898	0.889	0.912	0.909	0.527	0.517	0.525	0.525	0.936	0.927	0.937	0.929	7.25
FINER	0.727	0.740	0.743	0.762	0.869	0.838	0.874	0.851	0.614	0.601	0.612	0.605	0.937	0.880	0.943	0.914	7.94
WIRE	0.599	0.612	0.604	0.592	0.728	0.729	0.726	0.727	0.344	0.353	0.338	0.357	0.746	0.727	0.754	0.693	10.00
FFN-P	0.822	0.810	0.953	0.951	0.889	0.882	0.946	0.945	0.833	0.835	0.836	0.839	0.971	0.965	0.981	0.980	3.19
FFN-G	**0.846**	0.835	0.923	0.928	0.906	0.894	0.936	0.937	0.673	0.673	0.686	0.672	0.961	0.950	0.968	0.966	3.38
MFN	0.775	0.750	0.809	0.802	0.885	0.878	0.897	0.898	0.601	0.596	0.604	0.604	0.911	0.890	0.938	0.940	7.31
CMLP	0.764	0.732	0.818	0.797	0.880	0.874	0.899	0.882	0.573	0.554	0.626	0.570	0.943	0.940	0.948	0.953	7.19

Table 11 reports dense reconstruction error (MSE, computed over all locations) for ten implicit neural representation models on four geophysical datasets (SSH, CHL, GST, and SST), each evaluated on four tasks (T1–T4) at three increasing sampling ratios, with models summarized by their mean rank across the 16 settings in each block. `HiLS` is clearly the strongest model, reaching a mean rank

Table 14: Pearson correlation over all dense points. The last column is the mean rank of that model over the 16 cells of the block.

Data	SSH				CHL				GST				SST				Mean
Task	T1	T2	T3	T4	T1	T2	T3	T4	T1	T2	T3	T4	T1	T2	T3	T4	rank
Model	*Sampling ratio = 10%*				*Sampling ratio = 10%*				*Sampling ratio = 5%*				*Sampling ratio = 0.1%*				
HiLS	**0.709**	**0.705**	**0.958**	**0.939**	**0.835**	**0.833**	**0.943**	**0.931**	**0.923**	**0.901**	**0.953**	**0.928**	**0.949**	**0.934**	0.969	**0.954**	**1.06**
MMGN	0.415	0.450	0.799	0.756	0.696	0.713	0.919	0.906	0.893	0.883	0.931	0.917	0.945	0.928	**0.972**	0.942	4.88
INCODE	0.641	0.650	0.860	0.861	0.811	0.811	0.916	0.916	0.582	0.560	0.623	0.625	0.925	0.914	0.954	0.948	3.69
SIREN	0.623	0.608	0.793	0.791	0.813	0.811	0.902	0.901	0.491	0.484	0.505	0.507	0.896	0.890	0.925	0.920	5.69
FINER	0.633	0.623	0.649	0.645	0.796	0.791	0.808	0.802	0.547	0.532	0.587	0.588	0.869	0.852	0.898	0.894	6.62
WIRE	0.557	0.542	0.595	0.587	0.726	0.697	0.721	0.725	0.338	0.317	0.297	0.297	0.682	0.691	0.697	0.708	9.56
FFN-P	0.614	0.624	0.926	0.925	0.775	0.766	0.931	0.930	0.764	0.749	0.808	0.808	0.847	0.828	0.958	0.952	4.00
FFN-G	0.597	0.618	0.872	0.862	0.789	0.784	0.922	0.921	0.536	0.497	0.645	0.645	0.625	0.615	0.940	0.914	5.50
MFN	0.581	0.577	0.788	0.784	0.800	0.789	0.891	0.891	0.481	0.451	0.579	0.579	0.541	0.570	0.913	0.904	7.50
CMLP	0.615	0.607	0.780	0.780	0.813	0.806	0.885	0.890	0.437	0.400	0.547	0.553	0.894	0.883	0.931	0.928	6.50
	Sampling ratio = 25%				*Sampling ratio = 25%*				*Sampling ratio = 25%*				*Sampling ratio = 0.3%*				
HiLS	**0.819**	**0.810**	**0.986**	**0.973**	**0.894**	**0.890**	**0.973**	**0.954**	**0.954**	**0.946**	**0.957**	**0.949**	**0.978**	**0.968**	**0.987**	**0.977**	**1.00**
MMGN	0.670	0.667	0.871	0.832	0.845	0.848	0.932	0.924	0.933	0.927	0.938	0.934	0.975	0.967	0.984	0.973	4.25
INCODE	0.782	0.770	0.888	0.885	0.887	0.875	0.924	0.924	0.650	0.643	0.659	0.653	0.957	0.950	0.966	0.960	3.81
SIREN	0.756	0.721	0.783	0.786	0.883	0.872	0.908	0.908	0.520	0.519	0.523	0.522	0.930	0.923	0.935	0.927	6.88
FINER	0.702	0.692	0.708	0.702	0.844	0.843	0.852	0.857	0.601	0.598	0.613	0.605	0.927	0.854	0.937	0.917	7.88
WIRE	0.584	0.607	0.604	0.591	0.720	0.705	0.738	0.735	0.343	0.355	0.331	0.372	0.662	0.699	0.740	0.702	10.00
FFN-P	0.748	0.741	0.949	0.947	0.847	0.842	0.944	0.943	0.826	0.826	0.833	0.836	0.955	0.943	0.979	0.977	3.62
FFN-G	0.779	0.761	0.917	0.911	0.874	0.866	0.935	0.936	0.664	0.656	0.678	0.667	0.936	0.872	0.971	0.950	4.00
MFN	0.738	0.716	0.799	0.803	0.872	0.860	0.896	0.895	0.592	0.581	0.608	0.602	0.844	0.806	0.940	0.930	6.94
CMLP	0.717	0.697	0.814	0.789	0.872	0.858	0.912	0.876	0.592	0.552	0.567	0.563	0.939	0.934	0.945	0.940	6.62
	Sampling ratio = 50%				*Sampling ratio = 50%*				*Sampling ratio = 50%*				*Sampling ratio = 0.5%*				
HiLS	**0.918**	**0.909**	**0.992**	**0.985**	**0.942**	**0.934**	**0.984**	**0.972**	**0.958**	**0.952**	**0.958**	**0.953**	**0.985**	**0.977**	**0.988**	**0.982**	**1.00**
MMGN	0.856	0.813	0.892	0.868	0.922	0.907	0.936	0.931	0.937	0.933	0.938	0.936	0.981	0.974	0.985	0.978	3.12
INCODE	0.870	0.849	0.894	0.903	0.918	0.899	0.927	0.927	0.657	0.655	0.654	0.649	0.965	0.958	0.969	0.963	4.44
SIREN	0.801	0.757	0.815	0.783	0.905	0.897	0.913	0.910	0.528	0.517	0.525	0.525	0.936	0.927	0.937	0.929	7.56
FINER	0.737	0.756	0.744	0.764	0.873	0.843	0.874	0.852	0.615	0.604	0.613	0.607	0.937	0.880	0.943	0.914	8.06
WIRE	0.602	0.613	0.604	0.592	0.729	0.729	0.726	0.727	0.344	0.353	0.338	0.357	0.746	0.727	0.754	0.693	10.00
FFN-P	0.869	0.850	0.956	0.954	0.912	0.902	0.949	0.948	0.835	0.838	0.837	0.841	0.971	0.965	0.981	0.980	2.94
FFN-G	0.879	0.865	0.929	0.933	0.921	0.909	0.939	0.940	0.676	0.675	0.688	0.675	0.961	0.950	0.968	0.966	3.50
MFN	0.791	0.771	0.813	0.806	0.892	0.885	0.898	0.899	0.603	0.598	0.606	0.606	0.911	0.890	0.938	0.940	7.38
CMLP	0.796	0.758	0.823	0.800	0.888	0.882	0.900	0.883	0.577	0.558	0.630	0.572	0.943	0.940	0.948	0.953	7.00

Table 15: Average rank over the 48 evaluation schemes (4 datasets $\times$ 4 tasks $\times$ 3 sampling ratios).

Model	Dense MSE	Unobserved MSE	SSIM	Dense Pearson	Unobserved Pearson	Mean
HiLS	**1.04**	**1.27**	**1.19**	**1.02**	**1.25**	**1.15**
FFN-P	3.52	3.71	4.21	3.52	3.65	3.72
MMGN	4.25	4.40	2.92	4.08	4.23	3.98
INCODE	4.15	4.12	3.85	3.98	3.83	3.99
FFN-G	4.50	4.50	5.00	4.33	4.38	4.54
SIREN	6.79	6.56	5.85	6.71	6.56	6.50
CMLP	6.62	6.67	6.77	6.71	6.79	6.71
MFN	7.25	7.12	7.65	7.27	7.17	7.29
FINER	7.29	7.12	7.98	7.52	7.31	7.45
WIRE	9.58	9.52	9.58	9.85	9.83	9.68

of 1.06 in all three blocks, which means it ranks first in 15 of the 16 settings per block and second in the remaining one. Its advantage holds across datasets with very different spatial characteristics and across both moderate (10–50%) and extremely sparse (0.1–0.5% for SST) sampling regimes. As expected, errors decrease for nearly all models as the sampling ratio grows; for example, HiLS's SSH T1 error falls from 4.03 to 2.67 to 1.27 across the three blocks. HiLS's consistent lead at every ratio, however, shows that its benefit is not limited to data-rich conditions and remains robust under severe observational sparsity.

Table 12 evaluates the same ten models with the structural similarity index (SSIM, where higher is better), which measures how well each reconstruction preserves the spatial structure of the true field rather than just its pointwise error. HiLS again achieves the best mean rank in every block (1.19, 1.25, and 1.12 at low, medium, and high sampling ratios), confirming that its accuracy advantage in dense MSE comes with more faithful spatial patterns. SSIM rises steadily with sampling density for nearly every model and dataset; for instance, HiLS's SSH T1 score increases from 0.812 to 0.869 to 0.932 across the three blocks. The metric also shows a clear difference in difficulty across datasets: SSH and SST reach high structural fidelity (often above 0.9), whereas CHL and especially GST at the sparsest ratio stay considerably lower, which suggests that fields with sharper, more

heterogeneous features remain challenging for all methods. Even in these harder settings, HiLS preserves its top overall ranking.

Tables 13 and 14 report Pearson correlation computed on unobserved locations and on all dense points, respectively. HiLS achieves the best mean rank in every block of both tables. On unobserved locations its ranks are 1.12, 1.25, and 1.38, and over dense points they are 1.06, 1.00, and 1.00, meaning it ranks first in every setting at the medium and high sampling ratios. Correlations on unobserved locations are consistently lower than over the full field; for example, HiLS's SSH T1 correlation is 0.671 versus 0.709 at the 10% ratio. Both metrics improve with sampling density. These results show that HiLS generalizes well to regions it never observed.

Table 15 summarizes performance across all 48 evaluation cells (4 datasets × 4 tasks × 3 sampling ratios) and five evaluations. HiLS ranks first on every metric, with an overall mean rank of 1.15. Its advantage is strongest on the dense metrics (1.04 for MSE and 1.02 for Pearson) and remains clear on unobserved locations (1.27 and 1.25) and SSIM (1.19). The remaining models trail by a wide margin. FFN-P (3.72), MMGN (3.98), and INCODE (3.99) form a closely matched second tier, with MMGN strongest on SSIM (2.92). FFN-G follows, while SIREN, CMLP, MFN, and FINER occupy the lower ranks and WIRE is consistently last (9.68). The results show that HiLS's superiority is consistent across datasets, tasks, sampling regimes, and evaluation criteria, rather than confined to any single setting.

G.2 Qualitative Results

To complement the quantitative evaluation, Figures 11-12-13-14 show reconstructions from HiLS and four representative baselines (MMGN, FFN, CMLP, and MFN) for all four tasks. Each dataset is shown at its lowest sampling ratio, the most challenging regime, where the model must infer most of the field from very sparse observations. Across all tasks and datasets, HiLS produces the reconstructions that most closely match the reference. It captures the dominant large-scale structures, such as the warm and cold anomaly patterns in GST and the mesoscale eddy-like features in SST, while also retaining localized details such as the SSH anomalies along the Antarctic Circumpolar Current and western boundary currents, and the filamentary chlorophyll patterns in CHL. The baselines fail in distinct ways. MMGN recovers the broad structure of GST and SST reasonably well but introduces spurious high-amplitude artifacts in SSH and blob-like noise in CHL, particularly in Tasks 1 and 2. FFN and MFN produce noisy, speckled reconstructions that lose large-scale coherence, most visibly on GST and SST, where the true spatial patterns are largely washed out. CMLP produces overly smooth and oversaturated fields: it exaggerates warm anomalies in SST and adds streak-like striations in GST and in the Southern Ocean region of CHL. The baselines improve somewhat in Tasks 3 and 4, where more large-scale structure is recovered, but HiLS remains visually closest to the reference in every case. At the finest scales, HiLS is slightly smoother than the reference, which is expected under such extreme sparsity, but it avoids the artifacts and noise that dominate the competing methods. These visual results are consistent with the quantitative rankings in Tables 11-12-13-14 and show that HiLS's gains in MSE, SSIM, and correlation come with physically plausible and spatially coherent reconstructions.

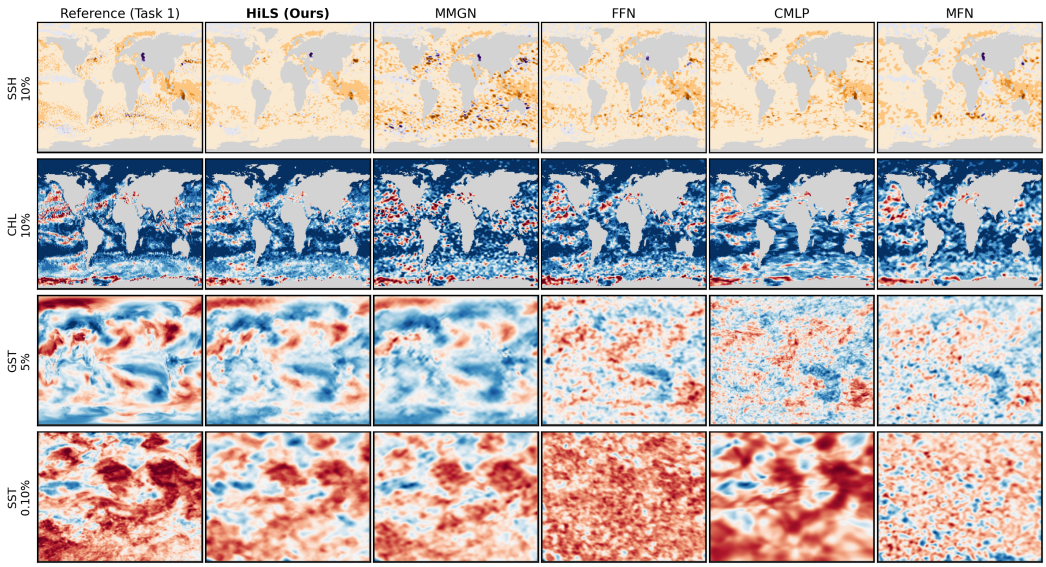

Figure 11: Qualitative comparison of reconstructions for Task 1 at the lowest sampling ratio of each dataset (SSH 10%, CHL 10%, GST 5%, SST 0.1%). The first column shows the reference field and the remaining columns show reconstructions from `HiLS` (ours), MMGN, FFN, CMLP, and MFN. `HiLS` recovers both the large-scale patterns and the localized features of the reference, whereas the baselines show spurious artifacts, high-frequency noise, or excessive smoothing.

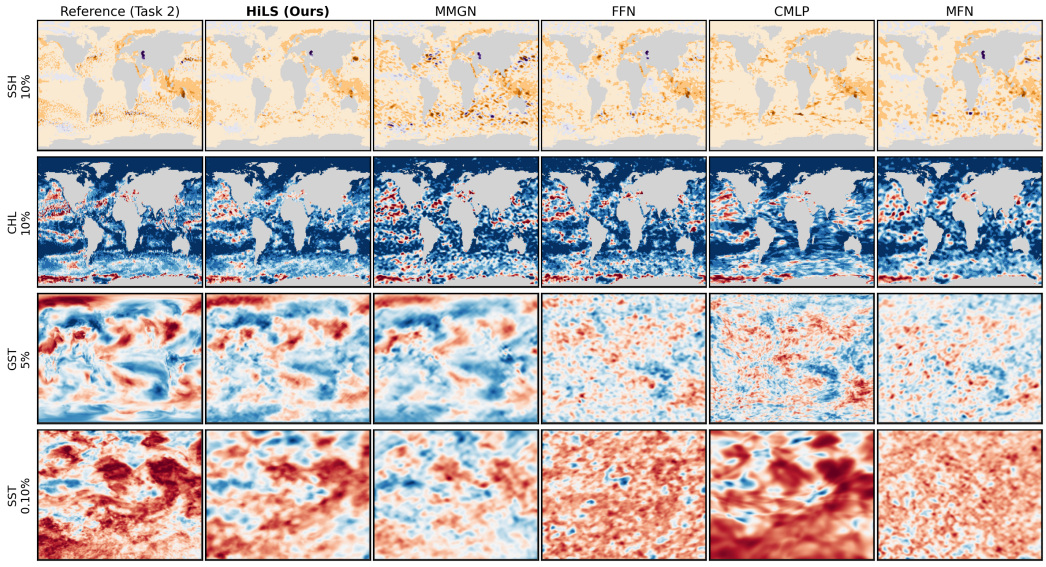

Figure 12: Qualitative comparison of reconstructions for Task 2 at the lowest sampling ratio of each dataset (SSH 10%, CHL 10%, GST 5%, SST 0.1%). `HiLS` stays closest to the reference across all four datasets, while FFN and MFN lose large-scale coherence on GST and SST, and CMLP overestimates warm anomalies on SST.

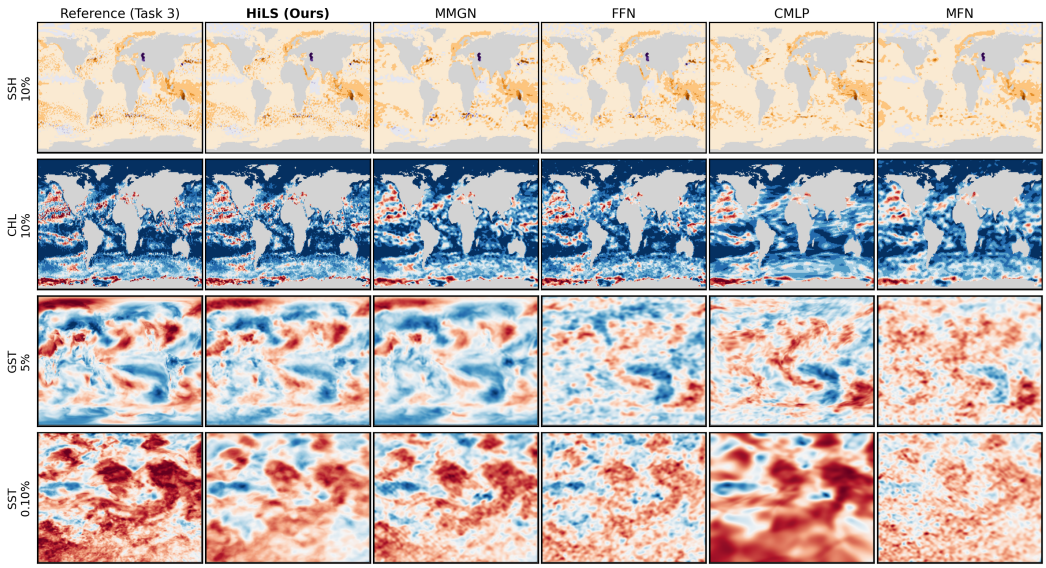

Figure 13: Qualitative comparison of reconstructions for Task 3 at the lowest sampling ratio of each dataset (SSH 10%, CHL 10%, GST 5%, SST 0.1%). The baselines recover more of the large-scale structure than in Tasks 1–2, but `HiLS` still reproduces the fine-scale variability most faithfully, including the SSH features along the Southern Ocean and the filamentary chlorophyll patterns.

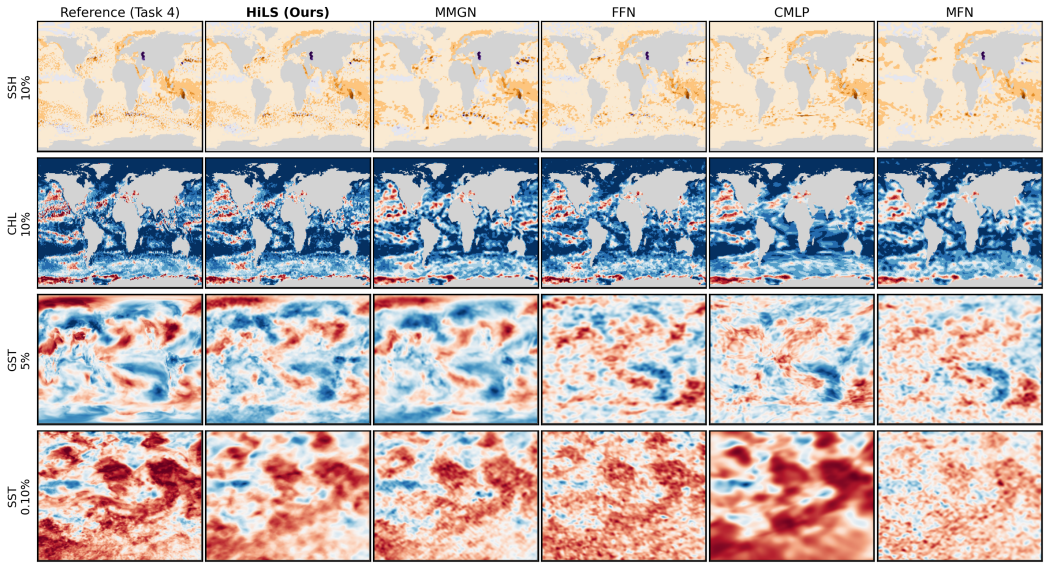

Figure 14: Qualitative comparison of reconstructions for Task 4 at the lowest sampling ratio of each dataset (SSH 10%, CHL 10%, GST 5%, SST 0.1%). `HiLS` produces the most coherent reconstructions across datasets, while CMLP shows streaking artifacts and oversaturation and MFN produces grainy, speckled fields.