# DESSNO: ENHANCING NEURAL OPERATORS FOR MULTISCALE PDE SYSTEMS WITH DCT-BASED EXPLICIT SCALE SEPARATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Neural operators have emerged as efficient surrogates for PDE simulations, yet accurately resolving multiscale spatial dynamics remains challenging. Existing methods often represent different scales within a shared latent space, which may entangle smooth large-scale structures with localized fine-scale dynamics and make the latter difficult to capture. To address this limitation, we propose **DESSNO**, which explicitly separates latent fields into macroscopic and microscopic components using a discrete cosine transform (DCT) and processes them through asymmetric spatial pathways. Specifically, the macroscopic pathway captures broad spatial dependencies, whereas the microscopic pathway models localized structures with a dedicated transient evolution module. The two components are then recoupled through a content-adaptive spatial gate before global temporal propagation. This *decompose–evolve–couple* design turns explicit scale separation into structured dynamical routing, enabling scale-dependent evolution while preserving cross-scale interactions. Experiments on six challenging PDE benchmarks demonstrate that DESSNO consistently improves prediction accuracy and parameter efficiency, better preserves fine-scale structures, and remains stable over long-horizon autoregressive rollouts.

## 1 INTRODUCTION

Partial differential equations (PDEs) form the mathematical foundation of many scientific and engineering problems and are traditionally solved using numerical methods, often at a high computational cost. In recent years, neural operators have emerged as efficient approaches for solving partial differential equations (PDEs) in real-world problems, and have achieved remarkable success across a wide range of scientific computing tasks Lu et al. (2021); Li et al. (2021); Cao (2021). However, many real-world systems exhibit pronounced spatial multiscale structures, in which globally smooth background fields often coexist with localized regions of sharp variation. For instance, wind-speed fields are typically dominated by smooth large-scale patterns while containing spatially confined cyclone-associated regions with abrupt intensity changes. Such multiscale characteristics pose a fundamental challenge to neural operators.

Existing neural operators address multiscale challenges through adaptive grids Pfaff et al. (2021); Brandstetter et al. (2022), multi-resolution architectures Rahman et al. (2023); Herde et al. (2024); Guo & Li (2024), and local feature enhancement strategies Liu et al. (2025); Hao et al. (2023). For instance, POSEIDON Herde et al. (2024) incorporates Transformer blocks into a U-Net-style hierarchical encoding framework, enabling information aggregation across multiple resolutions to balance global trends and local details. However, these approaches still learn multiscale features within a shared implicit representation space. In practice, smooth global patterns, corresponding to low-frequency components, typically dominate the PDE systems, whereas localized regions with sharp variations correspond to high-frequency components that occupy limited spatial extent but exhibit complex dynamics. This imbalance drives conventional deep learning models to prioritize fitting the smooth background, often at the expense of accurately capturing localized high-frequency structures. More critically, learning cross-scale interactions between global smooth patterns and localized sharp structures remains extremely challenging.



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.

This limitation raises a natural question: can separating macroscopic and microscopic information alleviate the representational interference introduced by a shared latent space? Such explicit scale separation has long been practiced in traditional numerical simulation. In computational fluid dynamics, Large Eddy Simulation (LES) SMAGORINSKY (1963) separates a flow field into resolved large-scale motions and unresolved small-scale effects. The large eddies are directly simulated, while the effects of the unresolved motions are represented through a subgrid-scale model. Inspired by this principle, we formulate explicit scale separation as a structured dynamical routing mechanism for neural operators: features at different scales are identified explicitly, evolved through scale-specific pathways, and subsequently recoupled to retain cross-scale interactions.

Based on this idea, we propose **DESSNO**, an explicit *decompose–evolve–couple* neural operator for multiscale PDE dynamics. Within each encoder block, DESSNO employs a lightweight two-dimensional discrete cosine transform (DCT) to decompose latent features into complementary low-frequency macroscopic and high-frequency microscopic components. Rather than processing them with a shared operator, DESSNO routes the two components through asymmetric spatial and temporal updates. The macroscopic pathway uses larger attention windows to capture broad spatial dependencies, whereas the microscopic pathway uses smaller windows to resolve localized fine-scale structures and incorporates a dedicated Transient Evolution Module (TEM) to model rapid temporal variations. The resulting features are then recoupled through a content-adaptive spatial gate, allowing the model to dynamically balance information from different scales.

To efficiently capture long-range temporal dependencies, DESSNO further adopts factorized spatiotemporal attention, applying temporal attention after cross-scale coupling while avoiding the prohibitive cost of joint spatiotemporal attention. In summary, our main contributions are as follows:

- We propose **DESSNO**, an explicit *decompose–evolve–couple* neural operator that uses scale separation as a structured dynamical routing mechanism, rather than merely as a multiresolution representation.
- We develop scale-conditioned evolution pathways that assign asymmetric spatial updates and dedicated transient modeling to macroscopic and microscopic components, followed by content-adaptive cross-scale recoupling and efficient global temporal propagation.
- Extensive experiments on six multiscale PDE benchmarks demonstrate superior accuracy, parameter efficiency, and long-horizon rollout stability.

## 2 Related Works

### 2.1 Neural Operators for PDEs

Neural operators learn mappings between infinite-dimensional function spaces, serving as efficient surrogate models for solving partial differential equations (PDEs). The Fourier Neural Operator (FNO) Li et al. (2021) is a representative method that parameterizes the integral kernel in the frequency domain to capture global dependencies and has inspired numerous variants such as Geo-FNO Li et al. (2023b), NUNO Liu et al. (2023), and GINO Li et al. (2023c). Concurrently, Transformer-based architectures have further advanced neural operators, including the Galerkin Transformer Cao (2021), AFNO Guibas et al. (2021), OFormer Li et al. (2023a), GNOT Hao et al. (2023), and CViT Wang et al. (2025). Moreover, models like MPP McCabe et al. (2024), DPOT Hao et al. (2024), and POSEIDON Herde et al. (2024) employ pretraining strategies to enhance transferability and generalization across diverse PDEs.

### 2.2 Multiscale Modeling in Neural Operators

Given that multiscale phenomena are ubiquitous in engineering and scientific domains such as fluid dynamics and meteorology, multiscale modeling has become a critical direction for neural operators. Existing approaches primarily mitigate scale-related challenges through adaptive meshes, multi-resolution hierarchical structures, and local enhancement strategies. Specifically, MeshGraphNet Pfaff et al. (2021) and MP-PDE Brandstetter et al. (2022) leverage graph neural networks on adaptive meshes to efficiently capture multiscale features in regions with sharp local variations. In terms of multiresolution hierarchical structures, POSEIDON Herde et al. (2024) and U-NO Rahman

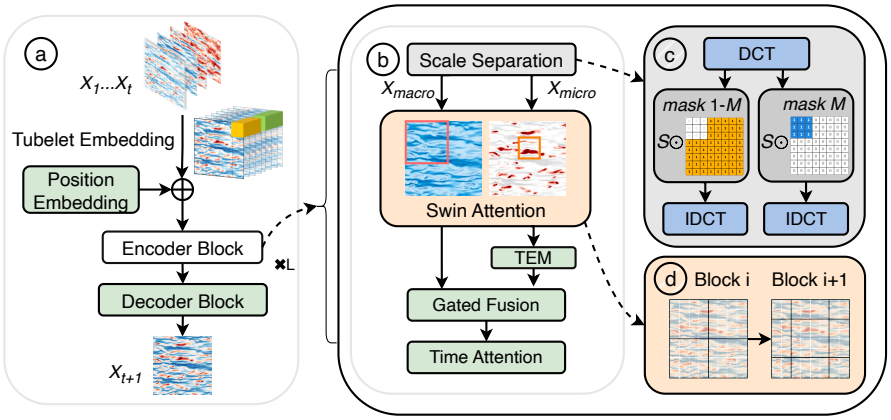


Figure 1: **Overview of the DESSNO framework.** (a) Overall model architecture. (b) Structure of the spatiotemporal factorized encoder. (c) DCT-based scale separation module. (d) Window-based attention mechanism.

et al. (2023) employ U-Net-style hierarchical encoder-decoders to fuse information across multiscale feature spaces, integrating Transformer and FNO-style operators, respectively, to balance global trends with local details. Regarding local enhancement, Conv-FNO Liu et al. (2025) improves local representation by combining convolutional modeling with frequency-domain global fusion, while GNOT Hao et al. (2023) employs a mixture-of-experts (MoE) gating mechanism for region-adaptive scale modeling. Since these methods process multiscale information within a shared implicit feature space, macroscopic trends and microscopic details tend to become entangled, making it difficult for models to capture both in a balanced manner.

Beyond methods that learn multiscale features implicitly, neural operator research has also explored explicit multiscale representations. MWT Gupta et al. (2021), for example, employs multiwavelet decomposition to obtain a sparse and resolution-independent parameterization of operator kernels. Related methods, including WNO Tripura & Chakraborty (2022), MAWNO Su et al. (2024), and WLNO Abid et al. (2026), similarly leverage wavelet transforms for operator parameterization or multiscale feature enhancement.

Unlike these wavelet-based methods, which primarily use multiscale transforms to parameterize or enhance operator representations, DESSNO employs DCT-based macro–micro separation as a dynamical routing mechanism. This design enables scale-dependent feature evolution and adaptive cross-scale coupling, making the decomposition part of the dynamics rather than merely a transform-domain representation.

## 3 DESSNO Design

### 3.1 Problem Formulation

We model the evolution of a physical system as an operator learning problem, aiming to predict future physical field states based on historical observational data. Consider a dynamic system defined on the spatiotemporal domain $\mathcal{D} \times [0, \mathcal{T}]$. The evolution of its state variable $u(x, t)$ is governed by a general partial differential equation (PDE):

$$\frac{\partial u}{\partial t} = \mathcal{F}(u, \nabla u, \ldots; \lambda), \quad x \in \mathcal{D}, t \in [0, \mathcal{T}] \tag{1}$$

where $\mathcal{F}$ denotes a nonlinear operator, and $\lambda$ represents physical parameters.

Our goal is to learn a neural operator $\mathcal{G}_\theta$ that maps a sequence of historical frames to a sequence of future frames. Given an observation sequence $\mathbf{X} = \{u(x, t_i)\}_{i=1}^{T_{in}}$, the model predicts the future states $\hat{\mathbf{Y}} = \{\hat{u}(x, t_j)\}_{j=T_{in}+1}^{T_{in}+T_{out}}$.

### 3.2 Training and Autoregressive Inference

To support stable long-horizon forecasting, we standardize each physical channel, train with randomly sampled temporal windows, and generate future trajectories through autoregressive rollout.

**Data Normalization.** Physical quantities in PDE problems (e.g., pressure, velocity, density) often possess distinct units and vastly different magnitudes. To ensure training stability, we apply channel-wise standardization. The input variable $x$ for each channel of the training set is normalized as:

$$x_{norm} = \frac{x - \mu_c}{\sigma_c} \tag{2}$$

where $\mu_c$ and $\sigma_c$ represent the mean and standard deviation of the $c$-th physical channel calculated over the entire training set, respectively. This normalization prevents variables with larger numerical scales from dominating the optimization.

**Autoregressive Rolling Prediction.** During inference, we adopt an autoregressive rolling prediction mode to generate multi-step future trajectories. The model first predicts the next state $\hat{u}_{T_{in}}$ based on the initial observation window $\mathbf{X}_{0:T_{in}}$. Subsequently, this prediction is treated as a genuine physical observation and fed back to the end of the input sequence, while the earliest time step is discarded, forming a new input window:

$$\mathbf{X}_{1:T_{in}+1} = \text{Concat}(\mathbf{X}_{1:T_{in}}, \hat{u}_{T_{in}}) \tag{3}$$

By recursively executing $\hat{u}_{t+1} = \mathcal{G}_\theta(\mathbf{X}_{t-T_{in}+1:t})$, the model generates continuous future trajectories of the physical fields. This mechanism demands that the model maintains not only high single-step accuracy but also physical conservation and structural stability over multi-step iterations.

**Random Temporal Window Sampling.** To robustly support the autoregressive rolling prediction described above, we implement a random temporal window sampling strategy during training. Since autoregressive inference requires the model to iteratively predict future states based on continuously shifting input windows, the model must possess the capability to generalize across different temporal phases rather than overfitting to a fixed initial time $t = 0$. Specifically, for each training iteration, we randomly select a starting frame index $s$ within the interval $[0, T_{out} - 1]$. We then extract the subsequence $\mathbf{X} = \{u_t\}_{t=s}^{s+T_{in}-1}$ as input and $\mathbf{Y} = \{u_t\}_{t=s+T_{in}}$ as the supervision target. This strategy ensures that the model learns a time-independent evolution operator capable of handling arbitrary temporal contexts encountered during rolling inference.

### 3.3 Overall Architecture

DESSNO comprises four key components: Tubelet Embedding, Positional Encoding, Spatiotemporal Factorized Encoder, and Reconstruction Decoder. The overall architecture is illustrated in Figure 1(a).

**Tubelet Embeddings.** To capture local spatiotemporal correlations, we adopt the 3D tubelet embedding strategy Arnab et al. (2021) instead of standard 2D patch embedding. Given an input sequence $\mathbf{X} \in \mathbb{R}^{T_{\text{in}} \times C_{\text{in}} \times H_0 \times W_0}$, where $C_{\text{in}}$ is the number of physical variables and $(H_0, W_0)$ is the input spatial resolution, we extract non-overlapping tubelets of size $(p_t \times p_s \times p_s)$, where $p_t$ and $p_s$ denote the temporal and spatial tubelet sizes, respectively. Each tubelet is flattened and linearly projected to an embedding dimension $d_{\text{emb}}$. This produces a latent token grid with dimensions $T = T_{\text{in}}/p_t$, $H = H_0/p_s$, and $W = W_0/p_s$, and feature dimension $C = d_{\text{emb}}$, assuming divisibility. The resulting tokens encode short-term temporal dynamics and local spatial structures.

**Positional Encoding.** Since the subsequent attention mechanisms are permutation-invariant, explicit positional information is required to preserve the spatiotemporal grid structure. We employ a hybrid positional encoding scheme:

1. Spatial Position Embedding: We add learnable parameters to the tokens to represent spatial grid positions, allowing the model to adaptively capture domain-specific boundary conditions and topology.

2. Temporal Position Embedding: To preserve the causality and time step order of PDE evolution, we add fixed sinusoidal embeddings to the temporal dimension. Unlike learnable embeddings, sinusoidal functions generalize naturally to continuous time coordinates, facilitating adaptation to unseen time steps during inference.

**Spatiotemporal Factorized Encoder.** To mitigate the high computational cost of spatiotemporal global attention and capture multiscale physical phenomena, we propose the **Dual-Channel Scale-Separated Spatiotemporal Factorized Encoder**. This module explicitly decouples features into macroscopic and microscopic components via frequency-domain transformation. It subsequently processes these components through parallel, scale-specific spatial attention mechanisms, followed by temporal attention to capture evolution dynamics. The detailed design is in Section 3.4.

**Decoder and Prediction.** The decoder maps the latent representation back to the physical state space to predict the next frame. It first applies Layer Normalization (LayerNorm) and a linear projection, followed by spatial rearrangement to the original grid. A shallow convolutional reconstruction head, consisting of 2D convolutional layers and ReLU activations, is then used to suppress blocking artifacts introduced by patch-based processing. Finally, the output layer projects the features to the $C$ target channels, producing the one-step prediction $\hat{\mathbf{u}}_{t+1}$. Multi-step forecasts are generated through the autoregressive procedure.

### 3.4 Encoder Design

The encoder is the core component of DESSNO, responsible for learning complex spatiotemporal evolution operators in the latent space. To efficiently process spatiotemporal data and capture multiscale features, we design the **Dual-Channel Scale-Separated Spatiotemporal Factorized Encoder**. As shown in Figure 1(b), each encoder layer comprises five submodules: (1) DCT-based scale separation module, (2) parallel multiscale spatial attention module, (3) transient evolution module, (4) gated fusion module, and (5) temporal attention module. Assuming the input of the $l$-th layer is $\mathbf{Z}^{l-1} \in \mathbb{R}^{B\times T\times H\times W\times C}$, the processing flow is as follows.

**DCT-based Scale Separation.** We employ a two-dimensional DCT-II for explicit macro–micro scale decomposition, as illustrated in Figure 1(c). DCT transforms signals from the spatial domain to the frequency domain, where low-frequency components correspond to the macroscopic structure of the physical field, while high-frequency components represent microscopic details.

Specifically, let $\mathbf{S} = \mathrm{DCT}(\mathbf{Z})$ denote the frequency spectrum of the input feature $\mathbf{Z}$. We construct a binary low-pass mask $\mathbf{M} \in \{0,1\}^{H\times W}$, where $\mathbf{M}_{u,v} = 1$ if $u < K \cdot H$ and $v < K \cdot W$ (retaining the top-left low-frequency corner), and 0 otherwise. The hyperparameter $K \in (0, 1]$ controls the bandwidth ratio. The decoupled macro and micro spatial features are then reconstructed via Inverse DCT (IDCT):

$$\mathbf{Z}_{macro} = \mathrm{IDCT}(\mathbf{S} \odot \mathbf{M}), \quad \mathbf{Z}_{micro} = \mathrm{IDCT}(\mathbf{S} \odot (\mathbf{1} - \mathbf{M})) \tag{4}$$

where $\odot$ denotes the Hadamard product.

**Parallel Multi-scale Spatial Attention.** After obtaining separated macro and micro features, we adopt a dual-path parallel processing strategy. Using the window-based self-attention (W-MSA) mechanism of the Swin Transformer (Figure 1(d)), we configure different window sizes to match physical feature scales. The Swin block includes standard W-MSA and shifted-window attention (SW-MSA), along with LayerNorm and residual connections. This design not only reduces the complexity of global attention from $O((HW)^2)$ to $O(M^2 \cdot HW)$, where $M$ is the window side length, but also enables the model to use the most suitable receptive field for features of different frequencies. In this paper, we employ a larger window size of $W_{\mathrm{large}} = 8 \times 8$ for the macro branch to capture long-range dependencies and a smaller window size of $W_{\mathrm{small}} = 4 \times 4$ for the micro branch to model local details.

**Transient Evolution Module (TEM).** High-frequency features exhibit rapid temporal evolution characteristics. To explicitly model this high-frequency temporal dynamics, we introduce a dedicated temporal convolution module in the micro branch, while the macro branch (representing slowly varying trends) does not undergo this operation. Specifically, we reshape the micro features $\hat{\mathbf{Z}}_{micro} \in \mathbb{R}^{(B\cdot T)\times H\times W\times C}$ to $\mathbb{R}^{(B\cdot HW)\times C\times T}$ and apply 1D temporal convolution:

$$\mathbf{Z}_{micro}^{temp} = \text{Conv1D}(\hat{\mathbf{Z}}_{micro}; k = 3) + \hat{\mathbf{Z}}_{micro} \tag{5}$$

Here, the convolution kernel size $k = 3$ is used to capture short-term dependencies between adjacent frames while keeping the time length unchanged. This design enables the model to better predict the propagation and decay processes of high-frequency fluctuations.

**Gated Fusion.** After separate spatial processing, macro and micro features are adaptively fused via a convolution-based context gating mechanism, allowing the model to dynamically prioritize macro structures or micro details based on local flow states.

We concatenate the outputs of the two branches along the channel dimension and generate spatially adaptive gating weights through a convolutional network:

$$\mathbf{G} = \sigma\left(\text{Conv}_{1\times1}\left(\text{GELU}\left(\text{GN}\left(\text{Conv}_{3\times3}([\hat{\mathbf{Z}}_{macro}, \mathbf{Z}_{micro}^{temp}])\right)\right)\right)\right) \tag{6}$$

where $[\cdot,\cdot]$ denotes channel concatenation, $\text{Conv}_{3\times3}$ captures local context, GN represents GroupNorm, $\text{Conv}_{1\times1}$ implements channel mixing, and $\sigma$ is the Sigmoid activation function. The elements of the gating map $\mathbf{G} \in \mathbb{R}^{B\times T\times H\times W\times C}$ range in $[0, 1]$; if a position favors macro information, its value approaches 1, otherwise it approaches 0.

The fused spatial feature is:

$$\mathbf{Z}_{spatial} = \text{Proj}\left(\mathbf{G} \odot \hat{\mathbf{Z}}_{macro} + (\mathbf{1} - \mathbf{G}) \odot \mathbf{Z}_{micro}^{temp}\right) + \mathbf{Z}^{l-1} \tag{7}$$

where Proj is a linear projection layer, and the residual connection promotes gradient propagation.

**Temporal Attention.** To model temporal evolution, we treat spatial positions as the batch dimension and apply full attention along the time axis. After reshaping the features to $\mathbb{R}^{(B\cdot HW)\times T\times C}$, we compute

$$\mathbf{Z}_{\text{temporal}} = \text{MSA}(\text{LN}(\mathbf{Z}_{\text{spatial}})) + \mathbf{Z}_{\text{spatial}}. \tag{8}$$

This allows each spatial position to interact across all time steps. With window side lengths $M_{\text{macro}}$ and $M_{\text{micro}}$, the factorized attention reduces complexity from $\mathcal{O}((THW)^2)$ to $\mathcal{O}\big(THW(M_{\text{macro}}^2 + M_{\text{micro}}^2) + HWT^2\big)$, enabling efficient modeling of high-resolution PDE dynamics.

Finally, the output of the $l$-th layer $\mathbf{Z}^l$ is obtained by reshaping $\mathbf{Z}_{temporal}$ back to the original dimensions, serving as input for the next layer.

The complete encoder is constructed by stacking $L$ identical layers, each consisting of five sequential submodules: *DCT separation*, *multiscale spatial attention*, *TEM*, *gated fusion*, and *temporal attention*. By stacking these layers, DESSNO progressively extracts and integrates deep multiscale spatiotemporal features, enabling a high-fidelity approximation of the PDE evolution operator.

# 4 Experiments

## 4.1 Experimental Setup

**Datasets.** We evaluate DESSNO on six benchmarks spanning four two-dimensional numerical PDE systems, real-world atmospheric dynamics, and three-dimensional magnetohydrodynamic turbulence. The four 2D benchmarks comprise SWE, a shallow-water benchmark from PDEArena Gupta & Brandstetter (2023); CE-KH and CE-CRP, two compressible Euler benchmarks from POSEIDON Herde et al. (2024); and NSE, a turbulent incompressible Navier–Stokes benchmark from Spectral-Refiner Cao et al. (2025). For real-world forecasting, we use hourly 10-m wind fields from ERA5 Hersbach et al. (2020), a global atmospheric reanalysis produced by ECMWF. For 3D evaluation, we use the MHD benchmark from The Well Burkhart et al. (2020), which models isothermal compressible magnetohydrodynamic turbulence. Further dataset details are provided in Appendix E.

**Baselines.** We compare DESSNO with representative neural operators spanning spectral methods (FNO Li et al. (2021)), Transformer-based operators (Galerkin Transformer, OFormer, GNOT, and

Table 1: **Quantitative comparison on six PDE benchmarks.** Relative $\ell_2$ errors (%) and parameter counts are reported; boldface and underlining indicate the best and second-best results, respectively.

| Model | Params | SWE | CE-KH | CE-CRP | NSE | ERA5 | MHD |
|---|---|---|---|---|---|---|---|
| FNO | 52M | 2.59 | 10.09 | 30.94 | 8.79 | 25.60 | <u>39.7</u> |
| MWT | 107M | 4.52 | 6.29 | 23.37 | 9.59 | 37.64 | 57.83 |
| U-NO | 11M | 4.62 | 10.66 | 38.30 | 26.96 | 43.82 | 53.11 |
| MgFNO | 2M | 5.50 | 7.79 | 28.19 | 12.37 | 29.43 | — |
| GNOT | 2M | 2.48 | 15.30 | 38.55 | 7.90 | 46.27 | — |
| MPP | 115M | 4.59 | 6.17 | 17.22 | 9.32 | 36.89 | — |
| OFormer | 16M | 4.71 | 8.21 | 34.27 | 15.95 | 43.84 | 52.38 |
| CViT | 71M | 8.66 | 14.37 | 26.82 | 12.78 | 49.65 | — |
| Poseidon(Finetuned) | 157M | 1.60 | 66.93 | 17.11 | 6.09 | 31.48 | — |
| Poseidon(Scratch) | 157M | 1.26 | 75.79 | 19.43 | 8.12 | 31.64 | — |
| DPOT(Finetuned) | 112M | 1.46 | <u>4.69</u> | <u>14.35</u> | <u>3.32</u> | <u>21.46</u> | — |
| DPOT(Scratch) | 112M | <u>0.98</u> | 4.92 | 14.66 | 3.40 | 22.02 | 40.62 |
| **DESSNO** | **55M** | **0.93** | **4.18** | **11.19** | **1.92** | **19.41** | **29.21** |

CViT Cao (2021); Li et al. (2023a); Hao et al. (2023); Wang et al. (2025)), pretrained PDE models (MPP, DPOT, and POSEIDON McCabe et al. (2024); Hao et al. (2024); Herde et al. (2024)), and multiscale architectures (MWT, U-NO, and MgFNO Gupta et al. (2021); Rahman et al. (2023); Guo & Li (2024)).

**Evaluation Metric.** We report the mean relative $\ell_2$ error:

$$\mathcal{E}_{\text{rel}} = \frac{1}{BC} \sum_{b=1}^{B} \sum_{c=1}^{C} \frac{\|\hat{\mathbf{y}}_{b,c} - \mathbf{y}_{b,c}\|_2}{\|\mathbf{y}_{b,c}\|_2 + \varepsilon}, \tag{9}$$

where $B$ is the number of test samples, $C$ is the number of physical variables, and $\hat{\mathbf{y}}_{b,c}$ and $\mathbf{y}_{b,c}$ denote the predicted and ground-truth trajectories, respectively. The norm is computed after flattening the temporal and spatial dimensions, and $\varepsilon = 10^{-8}$ ensures numerical stability.

**Implementation Details.** Unless otherwise specified, DESSNO uses four encoder layers, an embedding dimension of 336, 14 attention heads, and a DCT frequency ratio of $K = 0.1$. The macro/micro window side lengths are $8/4$ for 2D experiments and $4/2$ for the 3D experiment. All models take ten historical frames as input and are trained for one-step prediction. Models trained from scratch use the Adam optimizer with a learning rate of $10^{-4}$ for 350 epochs, whereas models initialized from pretrained weights are fine-tuned for 200 epochs. Evaluation is performed autoregressively for the specified number of steps. All experiments use a single NVIDIA H100 GPU. Baselines follow their recommended architecture-specific hyperparameters while using the same data splits, preprocessing, rollout protocol, evaluation metrics, and hardware as DESSNO.

### 4.2 Main Results

To evaluate DESSNO, we compare it with representative state-of-the-art neural operators on six benchmark datasets: SWE, CE-KH, CE-CRP, NSE, ERA5, and MHD. For MHD, we evaluate only models that support three-dimensional inputs. Table 1 reports the test-set relative $\ell_2$ errors and parameter counts. All models perform autoregressive rollouts for four time steps. DESSNO consistently achieves the best performance across all six benchmarks, while using only 55M parameters—substantially fewer than large pretrained Transformer-based models such as DPOT and POSEIDON.

Two observations merit discussion. First, on SWE, both DPOT and POSEIDON perform better when trained from scratch than when fine-tuned, possibly because SWE's non-square $96 \times 192$ resolution creates a mismatch with the square-grid data used during pretraining. Second, POSEIDON performs poorly on CE-KH, consistent with the results originally reported for this benchmark (Fig. 59) Herde et al. (2024); this may reflect limitations of its multiscale Swin Transformer in modeling complex anisotropic structures.

We further evaluate long-horizon forecasting on the five datasets with sufficiently long trajectories, using 10- or 20-step autoregressive rollouts according to the available sequence length. MHD is ex-

Table 2: **Relative $\ell_2$ errors (%) for long-horizon forecasting.** Numbers in parentheses indicate the number of autoregressive rollout steps.

| Model | SWE (20) | CE-KH (10) | CE-CRP (10) | NSE (20) | ERA5 (10) |
|---|---|---|---|---|---|
| strongest baseline | 5.13 | 7.42 | 21.33 | 7.31 | 34.28 |
| DESSNO | **4.76** | **7.07** | **17.44** | **6.58** | **33.60** |

Table 3: **Frequency-wise prediction errors.** Relative $\ell_2$ errors (%) of the low- and high-frequency components. For a consistent comparison, the predictions of both DESSNO and the strongest baseline are decomposed using DCT with $K = 0.1$.

| Model | SWE | | CE-KH | | CE-CRP | | NSE | | ERA5 | | MHD | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Low | High | Low | High | Low | High | Low | High | Low | High | Low | High |
| strongest baseline | **0.73** | 1.36 | 3.20 | 34.03 | 3.79 | 40.83 | 0.67 | 10.27 | 14.98 | 47.35 | 16.85 | 58.37 |
| DESSNO | **0.73** | **1.12** | **3.12** | **27.09** | **3.68** | **33.78** | **0.56** | **5.64** | **13.18** | **38.18** | **16.29** | **37.61** |

cluded because no additional frames are available beyond the standard evaluation horizon. Throughout the remainder of this paper, **strongest baseline** refers, for each dataset, to the best-performing baseline indicated by the underlined entry in Table 1. As shown in Table 2, DESSNO consistently outperforms these baselines, demonstrating greater long-horizon rollout stability. Parameter-matched comparisons and per-sample $\ell_2$ error statistics are provided in Appendix D.

### 4.3 Ablation Studies

**Effectiveness of Explicit Scale Separation.** For visualization, we apply DCT-based scale separation directly to the raw ERA5 input fields before learned embedding. As shown in Fig. 2, the resulting low- and high-frequency components capture large-scale spatial patterns and fine-scale details, respectively, consistent with the intended decomposition. To assess whether this separation improves fine-scale modeling, Table 3 compares the frequency-wise prediction errors of DESSNO and the strongest baseline. Although the two methods achieve comparable low-frequency errors, DESSNO yields substantially lower high-frequency errors, indicating a stronger ability to capture fine-scale dynamics. Appendix C discusses the rationale for frequency-based macro–micro decomposition and presents comparisons with an implicit multiscale variant and alternative transforms.

**Effectiveness of Spatiotemporal Factorized Attention.** To reduce computational cost, existing neural operators commonly compress or collapse the temporal dimension before spatial modeling. We evaluate the proposed spatiotemporal factorized attention on the SWE dataset by comparing it with three representative temporal compression strategies:

1. **Direct Channel Stacking** (e.g., POSEIDON Herde et al. (2024)), which concatenates multiple input frames along the channel dimension to form a static embedding.
2. **Temporal Aggregation** (e.g., DPOT Hao et al. (2024)), which uses a learnable Fourier-based transformation to aggregate features across time.
3. **Cross-Attention Compression** (e.g., CViT Wang et al. (2025)), which employs learnable latent queries to compress the temporal sequence through cross-attention.

As shown in Table 4, our design achieves the lowest relative $\ell_2$ error of **0.93%**. Its parameter count (55.39M) remains comparable to those of the compression-based variants, with only a marginal increase over direct channel stacking. These results indicate that preserving temporal resolution mitigates information loss and provides a better accuracy–parameter trade-off.

**Effectiveness of the Transient Evolution Module (TEM).** To isolate the contribution of TEM, we conduct an ablation study by removing it from the microscopic branch while keeping the remaining architecture unchanged. The results show that the relative $\ell_2$ error increases from $0.93\%$ to $1.29\%$, despite only a marginal reduction in parameter count from 55M to 52M. Thus, TEM yields a $27\%$

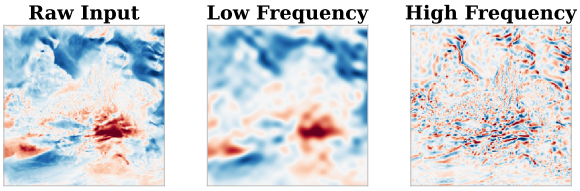


Figure 2: **DCT-based scale separation on ERA5 with $K = 0.1$.**

Table 4: **Impact of different temporal modeling strategies on the SWE dataset**.

| Strategy | Rel. $\ell_2$ | Params |
|---|---|---|
| Channel Stack | 1.13% | 48.19M |
| Time Aggregation | 1.21% | 55.51M |
| Cross Attention | 1.42% | 51.93M |
| **Spatiotemporal Sep.** | **0.93%** | 55.39M |

Table 5: **Sensitivity to the DCT frequency ratio.** Relative $\ell_2$ errors (%) on different datasets.

| $K$ | SWE | CE-KH | CE-CRP | NSE | ERA5 | MHD |
|---|---|---|---|---|---|---|
| 0.05 | 1.00 | 4.46 | 12.43 | 2.02 | 19.64 | 29.45 |
| 0.10 | **0.93** | **4.18** | **11.19** | **1.92** | **19.41** | **29.21** |
| 0.15 | 0.96 | 4.79 | 12.79 | 1.95 | 19.64 | 29.79 |

Table 6: **Sensitivity to model scale.** The ratio $\mathrm{dim}/\mathrm{head}$ is fixed to 24.

| $[\mathrm{dim}, \mathrm{head}]$ | Rel. $\ell_2$ | Params | Time |
|---|---|---|---|
| $[240, 10]$ | 1.13% | 28M | 21 h |
| $[288, 12]$ | 1.11% | 40M | 27 h |
| $[336, 14]$ | **0.93%** | 55M | 33 h |
| $[384, 16]$ | 0.90% | 72M | 40 h |

relative error reduction with modest parameter overhead, demonstrating its effectiveness in modeling rapid fine-scale dynamics.

### 4.4 Sensitivity to Hyperparameters

We examine the sensitivity of DESSNO to the DCT frequency ratio and model capacity; additional results on macro–micro window configurations are provided in Appendix D.

**Sensitivity to the DCT Frequency Ratio $K$.** We examine the effect of $K$ on model performance by reporting the relative $\ell_2$ errors (%) in Table 5. Across all evaluated datasets, $K = 0.1$ yields consistently strong performance. As illustrated in Fig. 2, this setting also produces the intended separation between large-scale structures and fine-scale details.

**Sensitivity to Model Capacity.** We study the effect of the embedding dimension and the number of attention heads on the SWE dataset, while keeping the ratio $\mathrm{dim}/\mathrm{head} = 24$ fixed. As shown in Table 6, increasing the model capacity from smaller configurations, such as $\mathrm{dim} = 240$ and $\mathrm{head} = 10$, leads to noticeable performance improvements, indicating insufficient capacity in low-dimensional settings. However, scaling beyond $\mathrm{dim} = 336$ and $\mathrm{head} = 14$ yields only marginal accuracy gains while substantially increasing the parameter count and training time. Based on this accuracy–efficiency trade-off, we adopt $\mathrm{dim} = 336$ and $\mathrm{head} = 14$ as the default configuration in all experiments.

## 5 Conclusion

In this work, we introduced DESSNO, a decompose–evolve–couple neural operator that explicitly separates latent PDE fields via DCT, evolves macroscopic and microscopic components through scale-specific pathways, and adaptively recouples them for global temporal propagation. DESSNO consistently outperforms the evaluated baselines across six benchmarks while maintaining a favorable parameter–accuracy trade-off and improved long-horizon rollout performance. However, real-world physical systems may exhibit richer and more continuous spectral structures than the binary macro–micro decomposition considered here. Future work will explore multi-band hierarchical partitioning, such as energy-based quadtrees in the DCT domain, to better accommodate such complex spectra. More broadly, we envision explicit, physics-aligned spectral routing as a general foundation for scalable and robust neural simulation of complex multiscale systems.

## 6 AI USE STATEMENT

The authors used ChatGPT to assist with language editing and presentation, including grammatical correction, stylistic refinement, and LaTeX formatting. It was not used to generate research ideas, formulate scientific claims, design the methodology or experiments, implement the proposed method, process or analyze data. All AI-assisted edits were reviewed and verified by the authors, who take full responsibility for the final content of this paper.

## REFERENCES

Muhammad Abid, Arth Sojitra, and Omer San. WLNO: wavelet-laplace neural operator for solving partial differential equations. *CoRR*, abs/2605.24658, 2026. doi: 10.48550/ARXIV.2605.24658. URL https://doi.org/10.48550/arXiv.2605.24658.

Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lucic, and Cordelia Schmid. Vivit: A video vision transformer. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pp. 6816–6826. IEEE, 2021. doi: 10.1109/ICCV48922.2021.00676. URL https://doi.org/10.1109/ICCV48922.2021.00676.

Johannes Brandstetter, Daniel E. Worrall, and Max Welling. Message passing neural PDE solvers. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=vSix3HPYKSU.

B Burkhart, SM Appel, S Bialy, J Cho, AJ Christensen, D Collins, Christoph Federrath, DB Fielding, D Finkbeiner, AS Hill, et al. The catalogue for astrophysical turbulence simulations (cats). *The Astrophysical Journal*, 905(1):14, 2020.

Shuhao Cao. Choose a transformer: Fourier or galerkin. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 24924–24940, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/d0921d442ee91b896ad95059d13df618-Abstract.html.

Shuhao Cao, Francesco Brarda, Ruipeng Li, and Yuanzhe Xi. Spectral-refiner: Accurate fine-tuning of spatiotemporal fourier neural operator for turbulent flows. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=MKP1g8wU0P.

John Guibas, Morteza Mardani, Zongyi Li, Andrew Tao, Anima Anandkumar, and Bryan Catanzaro. Adaptive fourier neural operators: Efficient token mixers for transformers. *CoRR*, abs/2111.13587, 2021. URL https://arxiv.org/abs/2111.13587.

Zi-Hao Guo and Hou-Biao Li. Mgfno: Multi-grid architecture fourier neural operator for parametric partial differential equations. *CoRR*, abs/2407.08615, 2024. doi: 10.48550/ARXIV.2407.08615. URL https://doi.org/10.48550/arXiv.2407.08615.

Gaurav Gupta, Xiongye Xiao, and Paul Bogdan. Multiwavelet-based operator learning for differential equations. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 24048–24062, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/c9e5c2b59d98488fe1070e744041ea0e-Abstract.html.

Jayesh K. Gupta and Johannes Brandstetter. Towards multi-spatiotemporal-scale generalized PDE modeling. *Trans. Mach. Learn. Res.*, 2023, 2023. URL https://openreview.net/forum?id=dPSTDbGtBY.

Zhongkai Hao, Zhengyi Wang, Hang Su, Chengyang Ying, Yinpeng Dong, Songming Liu, Ze Cheng, Jian Song, and Jun Zhu. GNOT: A general neural operator transformer for operator learning. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 12556–12569. PMLR, 2023. URL https://proceedings.mlr.press/v202/hao23c.html.

Zhongkai Hao, Chang Su, Songming Liu, Julius Berner, Chengyang Ying, Hang Su, Anima Anandkumar, Jian Song, and Jun Zhu. DPOT: auto-regressive denoising operator transformer for large-scale PDE pre-training. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=X7UnDevHOM.

Maximilian Herde, Bogdan Raonic, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel de Bézenac, and Siddhartha Mishra. Poseidon: Efficient foundation models for pdes. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/84e1b1ec17bb11c57234e96433022a9a-Abstract-Conference.html.

Hans Hersbach, Bill Bell, Paul Berrisford, Shoji Hirahara, András Horányi, Joaquín Muñoz-Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Dinand Schepers, Adrian Simmons, Cornel Soci, Saleh Abdalla, Xavier Abellan, Gianpaolo Balsamo, Peter Bechtold, Gionata Biavati, Jean Bidlot, Massimo Bonavita, Giovanna De Chiara, Per Dahlgren, Dick Dee, Michail Diamantakis, Rossana Dragani, Johannes Flemming, Richard Forbes, Manuel Fuentes, Alan Geer, Leo Haimberger, Sean Healy, Robin J. Hogan, Elías Hólm, Marta Janisková, Sarah Keeley, Patrick Laloyaux, Philippe Lopez, Cristina Lupu, Gabor Radnoti, Patricia de Rosnay, Iryna Rozum, Freja Vamborg, Sebastien Villaume, and Jean-Noël Thépaut. The era5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730):1999–2049, 2020. doi: 10.1002/qj.3803. URL https://doi.org/10.1002/qj.3803.

Zijie Li, Kazem Meidani, and Amir Barati Farimani. Transformer for partial differential equations' operator learning. *Trans. Mach. Learn. Res.*, 2023, 2023a. URL https://openreview.net/forum?id=EPPqt3uERT.

Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew M. Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=c8P9NQVtmnO.

Zongyi Li, Daniel Zhengyu Huang, Burigede Liu, and Anima Anandkumar. Fourier neural operator with learned deformations for pdes on general geometries. *J. Mach. Learn. Res.*, 24:388:1–388:26, 2023b. URL http://jmlr.org/papers/v24/23-0064.html.

Zongyi Li, Nikola B. Kovachki, Christopher B. Choy, Boyi Li, Jean Kossaifi, Shourya Prakash Otta, Mohammad Amin Nabian, Maximilian Stadler, Christian Hundt, Kamyar Azizzadenesheli, and Animashree Anandkumar. Geometry-informed neural operator for large-scale 3d pdes. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023c. URL http://papers.nips.cc/paper_files/paper/2023/hash/70518ea42831f02afc3a2828993935ad-Abstract-Conference.html.

Chaoyu Liu, Davide Murari, Chris J. Budd, Lihao Liu, and Carola-Bibiane Schönlieb. Enhancing fourier neural operators with local spatial features. *CoRR*, abs/2503.17797, 2025. doi: 10.48550/ARXIV.2503.17797. URL https://doi.org/10.48550/arXiv.2503.17797.

Songming Liu, Zhongkai Hao, Chengyang Ying, Hang Su, Ze Cheng, and Jun Zhu. NUNO: A general framework for learning parametric pdes with non-uniform data. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 21658–21671. PMLR, 2023. URL https://proceedings.mlr.press/v202/liu23o.html.

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. *Nat. Mach. Intell.*, 3(3):218–229, 2021. doi: 10.1038/S42256-021-00302-5. URL https://doi.org/10.1038/s42256-021-00302-5.

Michael McCabe, Bruno Régaldo-Saint Blancard, Liam Holden Parker, Ruben Ohana, Miles D. Cranmer, Alberto Bietti, Michael Eickenberg, Siavash Golkar, Géraud Krawezik, François Lanusse, Mariel Pettee, Tiberiu Tesileanu, Kyunghyun Cho, and Shirley Ho. Multiple physics pretraining for spatiotemporal surrogate models. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/d7cb9db5ade2db7814fbd01ee59f4c7b-Abstract-Conference.html.

Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W. Battaglia. Learning mesh-based simulation with graph networks. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=roNqYL0_XP.

Md Ashiqur Rahman, Zachary E. Ross, and Kamyar Azizzadenesheli. U-NO: u-shaped neural operators. *Trans. Mach. Learn. Res.*, 2023, 2023. URL https://openreview.net/forum?id=j3oQF9coJd.

J. SMAGORINSKY. General circulation experiments with the primitive equations: I. the basic experiment. *Monthly Weather Review*, 91(3):99 – 164, 1963. doi: 10.1175/1520-0493(1963)091⟨0099:GCEWTP⟩2.3.CO;2. URL https://journals.ametsoc.org/view/journals/mwre/91/3/1520-0493_1963_091_0099_gcewtp_2_3_co_2.xml.

Jiayang Su, Junbo Ma, Songyang Tong, Enze Xu, and Minghan Chen. Multiscale attention wavelet neural operator for capturing steep trajectories in biochemical systems. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan (eds.), *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2024, February 20-27, 2024, Vancouver, Canada*, pp. 15100–15107. AAAI Press, 2024. doi: 10.1609/AAAI.V38I13.29432. URL https://doi.org/10.1609/aaai.v38i13.29432.

Tapas Tripura and Souvik Chakraborty. Wavelet neural operator: a neural operator for parametric partial differential equations. *CoRR*, abs/2205.02191, 2022. doi: 10.48550/ARXIV.2205.02191. URL https://doi.org/10.48550/arXiv.2205.02191.

Sifan Wang, Jacob H. Seidman, Shyam Sankaran, Hanwen Wang, George J. Pappas, and Paris Perdikaris. Cvit: Continuous vision transformer for operator learning. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=cRnCcuLvyr.

# A NOTATION

Table 7: Summary of the main notation.

| Notation | Meaning |
|---|---|
| $\mathbf{x} = (x, y)$, $t$ | Spatial coordinate and time |
| $\mathcal{D}$, $\Omega$, $\mathbb{T}^2$ | Spatial domains; $\mathbb{T}^2$ denotes the 2D periodic domain |
| $\mathcal{F}$, $\boldsymbol{\lambda}$ | PDE evolution operator and physical parameters |
| $\mathbf{u}$, $\mathbf{F}(\mathbf{u})$ | Physical state and flux of conserved variables |
| $\mathcal{G}_\theta$ | Neural operator parameterized by $\theta$ |
| $\mathbf{X}_t$, $\mathbf{u}_{t+1}$, $\hat{\mathbf{u}}_{t+1}$ | Input window, next-frame target, and one-step prediction |
| $\hat{\mathbf{Y}}_t$ | Autoregressively generated future trajectory |
| $T_{\text{in}}$, $T_{\text{out}}$ | Input-window length and autoregressive rollout horizon |
| $B$, $T$, $H$, $W$, $C$ | Batch size, temporal length, spatial token-grid dimensions, and feature dimension |
| $d_{\text{emb}}$, $p_t$, $p_s$ | Embedding dimension and temporal/spatial tubelet sizes |
| $\mathbf{Z}^l$, $\mathbf{Z}_{\text{spatial}}$, $\mathbf{Z}_{\text{temporal}}$ | Latent features at layer $l$ and after spatial/temporal attention |
| $S$, $M$, $K$ | DCT spectrum, binary frequency mask, and frequency separation ratio |
| $\mathbf{Z}_{\text{macro}}$, $\mathbf{Z}_{\text{micro}}$ | Low- and high-frequency latent components |
| $\mathbf{G}$ | Content-adaptive cross-scale gating map |
| $M_{\text{macro}}$, $M_{\text{micro}}$ | Window side lengths of the macro and micro branches |
| DCT, IDCT | Discrete Cosine Transform and its inverse |
| $\rho$, $\mathbf{v}$ | Density and velocity field |
| $p$, $E$, $\gamma$ | Pressure, total energy, and adiabatic constant |
| $\mathbf{B}$, $p^*$, $c_s$ | Magnetic field, total pressure, and isothermal sound speed (MHD) |
| $\zeta$, $D$, $\eta$ | Vorticity, divergence, and fluid height (SWE) |
| $f$, $g$ | Coriolis parameter and gravitational acceleration (SWE) |
| $\nu$, $\mathbf{f}$ | Kinematic viscosity and external forcing (NSE) |
| $\sigma_x$, $\sigma_y$ | Spatial deformation functions (CE-CRP) |
| $D_{i,j}$, $n_p$ | Curved subdomain and partition parameter (CE-CRP) |
| $\mathcal{E}_{\text{rel}}$ | Mean relative $\ell_2$ error |
| $\|\cdot\|_2$, $\varepsilon$ | $\ell_2$ norm and numerical stability constant |

# B PHYSICAL MOTIVATION: CONNECTION TO LARGE EDDY SIMULATION

Large Eddy Simulation (LES) SMAGORINSKY (1963) is a classical methodology in computational fluid dynamics for modeling turbulent flows through explicit scale separation. As a traditional numerical simulation approach, LES is built upon first-principles governing equations and has been widely adopted to balance physical fidelity and computational efficiency in multiscale flow problems. The core idea of LES is to decompose a complex flow field into large-scale, energy-dominant motions and small-scale fluctuations via a filtering operation. This design is motivated by the fact that accurately resolving all small-scale eddies requires extremely fine spatial grids and incurs prohibitively high computational cost, while many small-scale turbulent structures exhibit statistically similar behavior and can be effectively modeled rather than explicitly resolved.

Formally, LES applies a spatial filter $G$ to the velocity field $\mathbf{u}$, yielding the resolved large-scale component $\bar{\mathbf{u}}(\mathbf{x}, t) = \int G(\mathbf{x} - \mathbf{x}')\mathbf{u}(\mathbf{x}', t)\, d\mathbf{x}'$. Applying this filtering operation to the Navier–Stokes equations results in the governing equation for the resolved scales:

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial(\bar{u}_i \bar{u}_j)}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \bar{p}}{\partial x_i} + \nu \frac{\partial^2 \bar{u}_i}{\partial x_j \partial x_j} - \frac{\partial \tau_{ij}}{\partial x_j}, \tag{10}$$

where $\tau_{ij} = \overline{u_i u_j} - \bar{u}_i \bar{u}_j$ denotes the *subgrid-scale (SGS) stress tensor*, capturing the net influence of unresolved small-scale fluctuations on the resolved large-scale dynamics.

Despite their strong physical grounding, traditional numerical simulation methods, including LES, remain computationally expensive when applied to large-scale or long-term spatiotemporal simulations. In recent years, neural operators have emerged as data-driven surrogate models that learn

mappings between infinite dimensional function spaces, enabling fast approximation of PDE solution operators and significantly accelerating physical simulation. However, existing neural operators often represent multiscale dynamics implicitly within a shared latent space, which can lead to feature entanglement and degraded performance when macroscopic trends and fine-scale variations coexist.

Motivated by the explicit scale separation principle in LES, we transfer this concept to the neural operator setting and propose DESSNO, a DCT-based Explicit Scale Separation Neural Operator. DESSNO employs the Discrete Cosine Transform (DCT) as a spectral filtering mechanism to explicitly decouple input fields into macroscopic and microscopic components, which are then processed by scale-specific spatiotemporal modeling modules. By explicitly modeling cross-scale interactions rather than entangling them in a single representation, DESSNO provides a structured and efficient approach for learning multiscale physical dynamics.

## C Additional Model Design and Analysis

**Implementation Details.** For the 2D benchmarks, scale separation is implemented using a separable orthonormal DCT-II over the two spatial axes of each latent feature map, independently for every batch element, time step, and channel. For an axis of length $N$, the transform matrix is defined as

$$C_{k,n}^{(N)} = \alpha_k \cos\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)k\right], \qquad \alpha_k = \begin{cases} N^{-1/2}, & k=0, \\ \sqrt{2/N}, & k>0. \end{cases} \tag{11}$$

Accordingly, the 2D spectrum is computed as $S = C^{(H)} Z (C^{(W)})^\top$. Because the transform is orthonormal, its inverse is given by the corresponding transposed transform matrices, enabling exact reconstruction before frequency masking.

The reconstruction decoder applies LayerNorm and a linear projection, rearranges the features to the original spatial grid, and uses a three-layer convolutional head:

$$C_{\text{dec}} \xrightarrow[\text{ReLU}]{\text{Conv}_{3\times3}} 64 \xrightarrow[\text{ReLU}]{\text{Conv}_{3\times3}} 32 \xrightarrow{\text{Conv}_{1\times1}} C. \tag{12}$$

The $3 \times 3$ convolutions use unit stride and padding, and the resulting $C$-channel output represents the next-frame prediction.

In the gated fusion module, we use GroupNorm with one group. For each spatiotemporal slice, its statistics are computed jointly over the channel and spatial dimensions $(C, H, W)$, while retaining channel-wise affine parameters.

**Extension to Three-Dimensional MHD.** For the MHD experiment, we extend all spatial operations in DESSNO to three dimensions. Each input consists of ten frames of the three Cartesian velocity components, $(v_x, v_y, v_z)$, on a $64^3$ grid. Non-overlapping tubelets of size $2 \times 4 \times 4 \times 4$ reduce the input to five temporal tokens on a $16^3$ latent grid, with an embedding dimension of 336.

At each encoder layer, scale separation is performed independently for every temporal token and latent channel using a separable orthonormal 3D DCT-II along the depth, height, and width axes. Given a latent feature $Z \in \mathbb{R}^{C\times D'\times H'\times W'}$, we construct a cuboidal low-frequency mask $M \in \{0,1\}^{D'\times H'\times W'}$ with cutoffs

$$d_c = \max(1, \lfloor KD' \rfloor), \quad h_c = \max(1, \lfloor KH' \rfloor), \quad w_c = \max(1, \lfloor KW' \rfloor), \tag{13}$$

and set $M_{d,h,w} = 1$ for $d < d_c$, $h < h_c$, and $w < w_c$. The macro- and micro-scale components are reconstructed as

$$Z_{\text{macro}} = \text{IDCT}_{3D}(S \odot M), \qquad Z_{\text{micro}} = \text{IDCT}_{3D}(S \odot (1-M)), \tag{14}$$

where $S = \text{DCT}_{3D}(Z)$. Each component is subsequently processed by an independent pointwise MLP with hidden dimension $2C$ and GELU activation.

The two branches employ 3D window attention with cubic window sizes of $4^3$ and $2^3$ for the macro and micro components, respectively. Each branch contains two consecutive blocks, where the second block applies a cyclic shift of half the corresponding window size. Three-dimensional relative

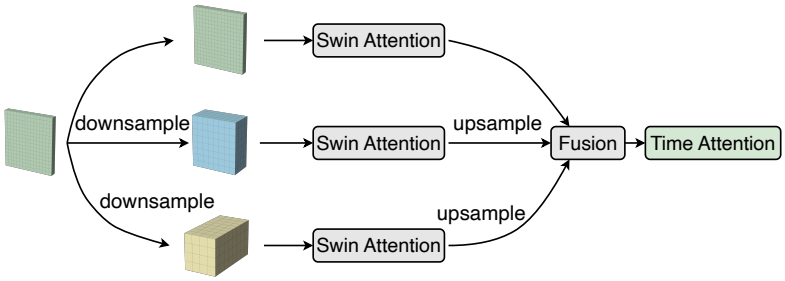


Figure 3: **Schematic of the implicit multiscale baseline.** It uses spatial downsampling instead of DCT to construct multiresolution branches, serving as a control experiment to verify the necessity of explicit scale separation.

Table 8: Ablation study on the effectiveness of the explicit scale separation mechanism.

| Model | Relative $\ell_2$ Error | Params | Inference Time |
|---|---|---|---|
| **DESSNO** | **0.93%** | **55M** | **50.86s** |
| Implicit Baseline | 1.65% | 300M | 60.96s |

position biases are included in the attention computation. The micro branch retains the temporal TEM, implemented using two 1D convolutions with kernel size 3 and a residual connection.

Cross-scale fusion is implemented using $\mathrm{Conv3D}_{3\times3\times3}$, GroupNorm, GELU, and $\mathrm{Conv3D}_{1\times1\times1}$ to generate a channel-wise spatial gate. After temporal attention, the decoder applies LayerNorm and linear unpatchification, followed by a $3 \times 3 \times 3$ convolution from 30 to 64 channels, GELU activation, and a $1 \times 1 \times 1$ convolution from 64 to 3 channels to predict the next MHD state.

**Comparison with Implicit Multiscale Modeling** We validate the necessity of our DCT-based decoupling by comparing DESSNO against an implicit multiscale variant. In this control setting, the frequency separation module is replaced by a standard hierarchical spatial downsampling pyramid (Figure 3). As shown in Table 8, DESSNO significantly outperforms this implicit variant, reducing the relative $\ell_2$ error by **43%** (0.93% vs. 1.65%) while requiring substantially fewer parameters (55M vs. 300M) and lower inference latency. The superior accuracy-efficiency trade-off suggests that explicit spectral decomposition is more effective than spatial downsampling for this task. We hypothesize that this performance gap arises because implicit methods may struggle to fully disentangle macroscopic dynamics from fine-grained details within shared latent spaces. In contrast, DESSNO's explicit separation likely mitigates such feature aliasing, enabling more precise modeling of cross-scale interactions.

**Frequency-Based Macro–Micro Decomposition.** A common observation in physical systems is that large-scale, macroscopic phenomena tend to vary smoothly in space, while small-scale, microscopic phenomena exhibit more rapid and irregular variations. Although exceptions exist, for example, large-scale engineered structures such as bridges may undergo rapid oscillations under extreme wind forcing. For all datasets considered in this paper, macroscopic dynamics such as large-scale flow structures and coherent patterns evolve smoothly in space, whereas fine-scale features, including sharp gradients, vortices, and small-scale fluctuations, are predominantly manifested in higher-frequency components. Therefore, adopting a frequency-based decomposition as a proxy for macro–micro separation is a reasonable and effective modeling choice for the multiscale dynamics studied in this work.

**Choice of Frequency Decomposition Method.** Discrete Cosine Transform (DCT), Fast Fourier Transform (FFT), and Wavelet Transform (WT) are classical frequency decomposition techniques widely used for multiscale representation learning. DCT offers a compact real-valued representation and can therefore be readily integrated into neural networks, whereas FFT naturally accommodates

Table 9: Ablation study on frequency decomposition methods. We report the relative $\ell_2$ error (%).

| Method | SWE | CE-KH | CE-CRP | NSE |
|---|---|---|---|---|
| WT | 1.74 | 4.53 | 15.05 | 2.79 |
| FFT | 1.00 | 5.02 | 14.49 | **1.77** |
| DCT | **0.93** | **4.18** | **11.19** | 1.92 |

Table 10: **Parameter-controlled comparisons.** Relative $\ell_2$ errors (%) are reported.

| Model | Params (M) | SWE | CE-KH | CE-CRP | NSE | ERA5 | MHD |
|---|---|---|---|---|---|---|---|
| *Comparable parameter scale* | | | | | | | |
| U-NO | 56 | 3.43 | 10.35 | 36.49 | 20.73 | 41.80 | 53.10 |
| MgFNO | 51 | 4.77 | 7.30 | 24.68 | 9.52 | 27.40 | — |
| GNOT | 50 | 2.28 | 12.28 | 32.23 | 6.82 | 43.82 | — |
| DESSNO | 55 | **0.93** | **4.18** | **11.19** | **1.92** | **19.41** | **29.21** |
| *Lightweight configurations* | | | | | | | |
| OFormer | 16 | 4.71 | 8.21 | 34.27 | 15.95 | 43.84 | 52.38 |
| DESSNO-Light | 18 | **1.32** | **5.60** | **17.50** | **2.75** | **20.38** | **30.92** |

periodic boundary conditions. In contrast, WT provides spatially localized, hierarchical multiresolution representations.

To examine the effect of the decomposition method, we replace the DCT module with FFT- and WT-based variants while keeping the remaining architecture and training protocol unchanged. For the WT variant, we follow MWT Gupta et al. (2021), adopt the Legendre wavelet basis, and use a two-band low-/high-frequency decomposition to match our macro–micro architecture. As shown in Table 9, FFT achieves the best performance on the NSE dataset, whose periodic boundary conditions align well with the periodic basis functions of the Fourier transform. On the other benchmarks, DCT consistently yields the lowest relative $L_2$ error, suggesting that its real-valued and energy-compact representation provides a robust inductive bias across different physical systems. The WT variant performs less favorably under the current setting. One possible explanation is that restricting WT to only two bands prevents it from fully exploiting its hierarchical multiresolution structure.

Based on its consistently competitive performance and implementation simplicity, we adopt DCT as the default decomposition method throughout this work. Nevertheless, our explicit scale-separation framework is transform-agnostic: DCT can be replaced by a decomposition better suited to the boundary conditions and spectral characteristics of a particular PDE system. Extending the framework to dataset-dependent transformations and multi-band decompositions, where the hierarchical structure of WT may offer additional advantages, is left for future work.

# D Additional Results

## D.1 Parameter-Controlled Comparisons

To disentangle architectural improvements from model capacity, we scale U-NO, MgFNO, and GNOT to approximately match the parameter count of DESSNO (50–56M). As shown in Table 10, DESSNO consistently achieves the lowest error across all evaluated benchmarks, indicating that its gains cannot be attributed solely to increased model capacity.

Scaling OFormer to a comparable parameter count exceeded the memory capacity of an 80GB H100 GPU. We therefore additionally compare its standard 16M-parameter configuration with DESSNO-Light, an 18M-parameter variant of our model. DESSNO-Light substantially outperforms OFormer on all six evaluated datasets. Together with the main results in Table 1, where DESSNO also outperforms substantially larger pretrained models, these comparisons demonstrate a favorable accuracy–parameter trade-off.

Table 11: **Per-sample error statistics.** Mean and standard deviation of the relative $\ell_2$ errors (%) across test samples. Boldface denotes the best result.

| Model | Params (M) | SWE | CE-KH | CE-CRP | NSE |
|---|---|---|---|---|---|
| FNO | 52 | $2.59 \pm 0.37$ | $10.09 \pm 0.72$ | $30.94 \pm 3.98$ | $8.79 \pm 0.74$ |
| U-NO | 56 | $3.43 \pm 0.87$ | $10.35 \pm 0.53$ | $36.49 \pm 3.36$ | $20.73 \pm 1.60$ |
| POSEIDON (fine-tuned) | 157 | $1.60 \pm 0.25$ | $66.93 \pm 0.66$ | $17.11 \pm 3.06$ | $6.09 \pm 0.55$ |
| DPOT (fine-tuned) | 112 | $1.46 \pm 0.26$ | $4.69 \pm 0.32$ | $14.35 \pm 3.09$ | $3.32 \pm 0.38$ |
| DESSNO | 55 | $\mathbf{0.93 \pm 0.23}$ | $\mathbf{4.18 \pm 0.28}$ | $\mathbf{11.19 \pm 2.20}$ | $\mathbf{1.92 \pm 0.21}$ |

### D.2 Error Distribution Across Test Samples

To assess whether the performance gains are consistent across test cases, we report the mean and standard deviation of the per-sample relative $\ell_2$ errors. As shown in Table 11, DESSNO achieves both the lowest mean error and the smallest standard deviation on all four benchmarks. These results indicate that its improvements are not driven by a small subset of favorable samples; rather, DESSNO provides consistently accurate predictions across diverse test trajectories.

### D.3 Multiscale Window Configurations

Finally, we analyze the effect of macro–micro window size configurations by varying the window size of one branch while keeping the other fixed. As shown in Table 12, the asymmetric configuration with $W_{\text{macro}} = 8$ and $W_{\text{micro}} = 4$ consistently achieves the best performance among all tested settings. This result supports our design choice of using larger windows for macroscopic modeling and smaller windows for capturing fine-grained microscopic dynamics.

Table 12: **Sensitivity analysis of macro–micro window configurations.**

| Setting | $[W_{\text{macro}}, W_{\text{micro}}]$ | Relative $L_2$ Error |
|---|---|---|
| *Macro window fixed at 8* | $[8, 2]$ | 0.9419% |
| | $[8, 4]$ | **0.9310%** |
| | $[8, 8]$ | 0.9668% |
| *Micro window fixed at 4* | $[4, 4]$ | 0.9892% |
| | $[8, 4]$ | **0.9310%** |
| | $[12, 4]$ | 0.9885% |

## E Benchmark Details

We evaluate DESSNO on six PDE benchmarks: four numerically simulated 2D datasets (SWE, CE-KH, CE-CRP, and NSE), the real-world ERA5 atmospheric reanalysis, and the 3D MHD turbulence dataset. Together, they span geophysical flows, compressible and incompressible fluid dynamics, atmospheric evolution, and magnetized turbulence.

### E.1 SWE

The SWE dataset is designed to evaluate operator learning models on large-scale geophysical fluid dynamics governed by shallow water equations. It captures nonlinear interactions between vorticity, divergence, and free-surface height under the influence of rotation and gravity, and serves as a standard benchmark for spatiotemporal prediction in climate and weather modeling.

**Governing Equations.** Let $\mathbf{u} = (u, v)$ denote the horizontal velocity field and $\eta$ the interface height, which conceptually represents pressure. We define the vorticity $\zeta = \nabla \times \mathbf{u}$ and the divergence

$D = \nabla \cdot \mathbf{u}$. The vorticity–divergence formulation of the shallow water equations is given by

$$\frac{\partial \zeta}{\partial t} + \nabla \cdot \big(\mathbf{u}(\zeta + f)\big) = 0, \tag{15}$$

$$\frac{\partial D}{\partial t} - \nabla \times \big(\mathbf{u}(\zeta + f)\big) = -\nabla^2 \left(\frac{1}{2}(u^2 + v^2) + g\eta\right), \tag{16}$$

$$\frac{\partial \eta}{\partial t} + \nabla \cdot (\mathbf{u}h) = 0, \tag{17}$$

subject to periodic boundary conditions. Here $f$ denotes the Coriolis parameter, $g = 9.81$ is the gravitational acceleration, and $h$ is the dynamic layer thickness.

**Dataset Information.** The SWE dataset is generated by PDEARENA Gupta & Brandstetter (2023) using the SPEEDYWEATHER.JL solver. The dataset contains 6,000 trajectories, each with spatiotemporal resolution $30 \times 96 \times 192$. We use 5,000 trajectories for training, with 500 each for validation and testing.

### E.2 CE-KH

The CE-KH dataset is designed to benchmark operator learning models on nonlinear, compressible fluid dynamics with instability-driven structure formation. It is based on two-dimensional compressible Euler equations with Kelvin–Helmholtz-type initial conditions, leading to complex vortex roll-up and shock interactions.

**Governing Equations.** The dynamics are governed by the two-dimensional compressible Euler equations

$$\partial_t \mathbf{u} + \nabla \cdot \mathbf{F}(\mathbf{u}) = 0, \tag{18}$$

where the conserved variables and fluxes are given by

$$\mathbf{u} = \begin{bmatrix} \rho \\ \rho \mathbf{v} \\ E \end{bmatrix}, \quad \mathbf{F}(\mathbf{u}) = \begin{bmatrix} \rho \mathbf{v} \\ \rho \mathbf{v} \otimes \mathbf{v} + p\mathbf{I} \\ (E + p)\mathbf{v} \end{bmatrix}. \tag{19}$$

Here, $\rho$ denotes the density, $\mathbf{v} = (v_x, v_y)$ the velocity field, $p$ the pressure, and $E$ the total energy. The system is closed by the ideal gas equation of state

$$E = \frac{1}{2}\rho |\mathbf{v}|^2 + \frac{p}{\gamma - 1}, \tag{20}$$

with the adiabatic constant $\gamma = 1.4$.

**Initial Conditions.** The CE-KH dataset employs Kelvin–Helmholtz-type initial conditions characterized by a shear layer in the horizontal velocity component with superimposed perturbations. Specifically, the initial state is given by

$$(\rho, v_x, v_y, p)|_{t=0} = (\rho_0(\mathbf{x}), v_{x,0}(\mathbf{x}), v_{y,0}(\mathbf{x}), p_0(\mathbf{x})), \tag{21}$$

where the velocity field contains a discontinuous or smoothed shear profile in $v_x$ and a small-amplitude transverse perturbation in $v_y$, triggering Kelvin–Helmholtz instability and subsequent vortex roll-up. The density and pressure fields are initialized to maintain approximate pressure equilibrium while allowing compressibility effects.

**Dataset Information.** The dataset is generated by POSEIDON Herde et al. (2024). The dataset contains 4,290 trajectories, each with spatiotemporal resolution $20 \times 128 \times 128$. We use 3,790 trajectories for training, with 250 each for validation and testing. Our training objective is to use the horizontal velocity, vertical velocity, and pressure from the first ten frames as input to autoregressively predict the subsequent four frames.

### E.3 CE-CRP

The CE-CRP dataset is designed to benchmark operator learning models on compressible flow dynamics with spatially complex, curved, and multi-partitioned initial configurations. It extends the

classical compressible Euler Riemann problem by introducing non-axis-aligned, smoothly deformed subdomains, leading to intricate shock interactions and nonlinear wave propagation.

**Governing Equations.** The underlying dynamics of the CE-CRP dataset are governed by the same two-dimensional compressible Euler equations as in the CE-KH dataset (see Equations 18 and 20).

**Initial Conditions.** The CE-CRP dataset corresponds to a curved and multi-partitioned variant of the compressible Euler Riemann problem. Let $\{x\} = x - \lfloor |x| \rfloor \operatorname{sgn}(x)$ denote the fractional part of $x \in \mathbb{R}$. We define spatial deformation functions

$$\sigma_x(x,y) = \sum_{i,j=1}^{p} \alpha_{x,i,j} \sin\big(2\pi i x + j y + \beta_{x,i,j}\big), \tag{22}$$

$$\sigma_y(x,y) = \sum_{i,j=1}^{p} \alpha_{y,i,j} \sin\big(2\pi i x + j y + \beta_{y,i,j}\big), \tag{23}$$

where $\alpha_{k,i,j} \sim \mathcal{U}[-0.1, 0.1]$ and $\beta_{k,i,j} \sim \mathcal{U}[0,1]$. These functions are used to generate curved partitions of the spatial domain. Specifically, each subdomain $D_{i,j} \subset \mathbb{T}^2$ is defined as

$$D_{i,j} = \Big\{ (x,y) \in \mathbb{T}^2 \;\Big|\; x_{\min} \le \{x + \sigma_x(x,y) + 1\} < x_{\max}, \quad y_{\min} \le \{y + \sigma_y(x,y) + 1\} < y_{\max} \Big\}, \tag{24}$$

with

$$x_{\min} = \frac{i}{p+1}, \quad x_{\max} = \frac{i+1}{p+1}, \quad y_{\min} = \frac{j}{p+1}, \quad y_{\max} = \frac{j+1}{p+1}.$$

The initial state is defined piecewise over each subdomain $D_{i,j}$ as

$$(\rho, v_x, v_y, p)\big|_{t=0} = (\rho_{i,j}, u_{i,j}, v_{i,j}, p_{i,j}), \quad (x,y) \in D_{i,j}, \tag{25}$$

where $\rho_{i,j} \sim \mathcal{U}[0.1, 1]$, $u_{i,j}, v_{i,j} \sim \mathcal{U}[-1, 1]$, and $p_{i,j} \sim \mathcal{U}[0.1, 1]$. This construction results in curved, multi-region Riemann-type initial conditions that induce complex shock interactions and nonlinear wave dynamics.

**Dataset Information.** The CE-CRP dataset, generated by POSEIDON Herde et al. (2024), consists of 3,600 trajectories with a spatiotemporal resolution of $20 \times 128 \times 128$. We split this dataset into 3,000 trajectories for training, 300 for validation, and 300 for testing. For training, the objective is to autoregressively predict the next four frames using the first ten frames of horizontal velocity, vertical velocity, and pressure as input.

### E.4 NSE

The NSE dataset is designed to benchmark operator learning models on two-dimensional incompressible Navier–Stokes dynamics exhibiting isotropic turbulence. It captures long-range spatiotemporal dependencies induced by nonlinear advection and viscous dissipation, and is widely used to evaluate trajectory-to-trajectory and tensor-to-tensor neural operators in turbulent flow regimes.

**Governing Equations.** The dynamics are governed by the two-dimensional incompressible Navier–Stokes equations

$$\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f}, \tag{26}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{27}$$

where $\mathbf{u} = (u, v)$ denotes the velocity field, $p$ the pressure, $\nu$ the kinematic viscosity, and $\mathbf{f}$ an external forcing term. The equations are solved on a periodic square domain, modeling two-dimensional incompressible flow. The resulting dynamics exhibit an inverse energy cascade consistent with Kolmogorov turbulence theory.

**Dataset Information.** The NSE dataset is generated by SPECTRAL-REFINER Cao et al. (2025), which employs a pseudo-spectral method for spatial discretization. The dataset contains 1,280 trajectories, each with spatiotemporal resolution $30 \times 128 \times 128$. We use 1,000 trajectories for training, with 140 each for validation and testing. Our training objective is to use the velocity from the first ten frames as input to autoregressively predict the subsequent four frames.

### E.5 ERA5

ERA5 is the fifth-generation global atmospheric reanalysis produced by the European Centre for Medium-Range Weather Forecasts (ECMWF) and distributed through the Copernicus Climate Data Store Hersbach et al. (2020). It combines numerical weather prediction with global observations to provide spatially complete and physically consistent atmospheric fields.

**Dataset Information.** We use hourly ERA5 data from January 2000 to April 2004, retaining the zonal and meridional wind components, $u_{10}$ and $v_{10}$, at 10 m above the surface. These two variables constitute both the model inputs and prediction targets. The data are provided on a global $721 \times 1440$ regular latitude–longitude grid with a native resolution of $0.25°$. To obtain a fixed $256 \times 256$ spatial grid without interpolation, we crop the region spanning $0.25°$–$64°$N and $60°$–$123.75°$E. The resulting hourly sequence is partitioned chronologically into 1,216 non-overlapping trajectories of 30 consecutive frames; the first 1,116 trajectories are used for training, the next 50 for validation, and the final 50 for testing.

### E.6 MHD

The MHD-64 dataset from The Well Burkhart et al. (2020) describes three-dimensional, isothermal, compressible magnetohydrodynamic turbulence without self-gravity. It covers diverse physical regimes, including subsonic and supersonic flows as well as sub- and super-Alfvénic turbulence.

**Governing Equations.** The dynamics are governed by the ideal MHD equations:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0, \tag{28}$$

$$\frac{\partial (\rho \mathbf{v})}{\partial t} + \nabla \cdot (\rho \mathbf{v}\mathbf{v} - \mathbf{B}\mathbf{B}) + \nabla p^* = 0, \tag{29}$$

$$\frac{\partial \mathbf{B}}{\partial t} - \nabla \times (\mathbf{v} \times \mathbf{B}) = 0, \tag{30}$$

where $\rho$, $\mathbf{v}$, and $\mathbf{B}$ denote the density, velocity, and magnetic field, respectively, and $p^* = c_s^2 \rho + \frac{1}{2}\|\mathbf{B}\|^2$ is the total pressure under the isothermal closure.

**Dataset Information.** The original simulations are generated at a resolution of $256^3$, anti-aliased using an ideal low-pass filter, and downsampled to $64^3$. Although the underlying dynamics are governed by the full MHD system, our processed benchmark retains only the three Cartesian velocity components, $\mathbf{v} = (v_x, v_y, v_z)$, as input and prediction variables. The dataset contains 679 sequences, each comprising 14 time steps on a periodic $64^3$ grid. We use 619 sequences for training, 30 for validation, and 30 for testing. The first ten velocity fields are used as input, and the subsequent four are predicted autoregressively.

## F Visualization Results

This section presents DESSNO's four-step autoregressive rollout results on five two-dimensional datasets. In each figure, the rows show the ground-truth fields, DESSNO predictions, and pointwise absolute errors, respectively.

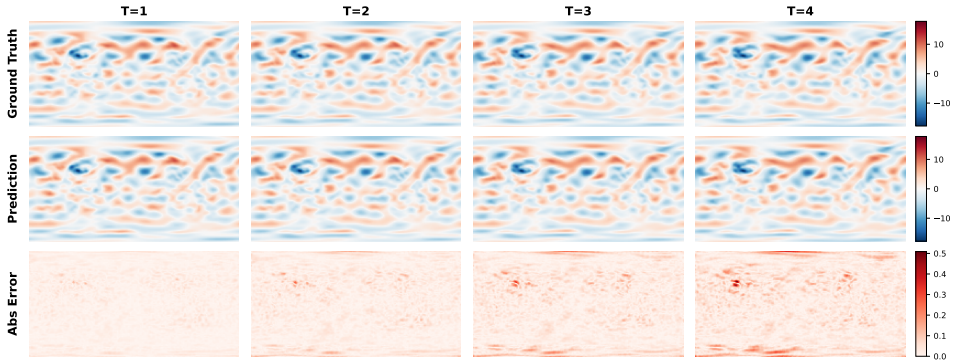


Figure 4: SWE.

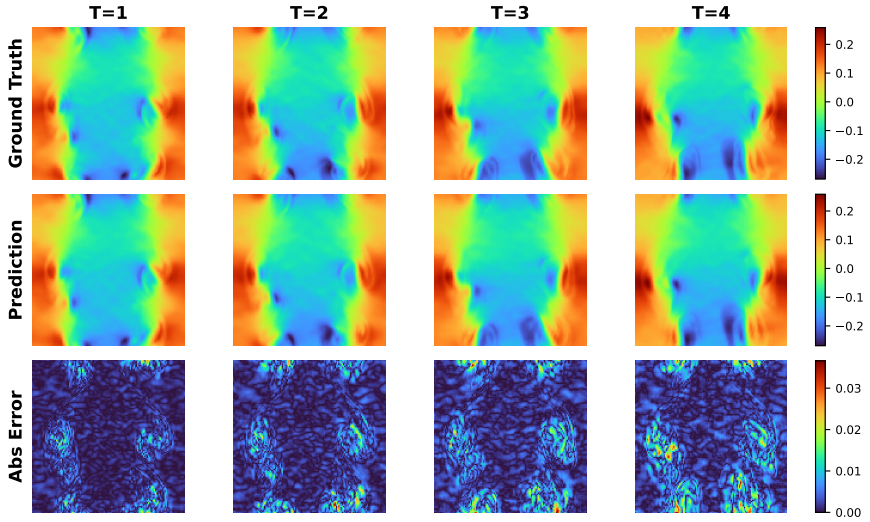


Figure 5: CE-KH.

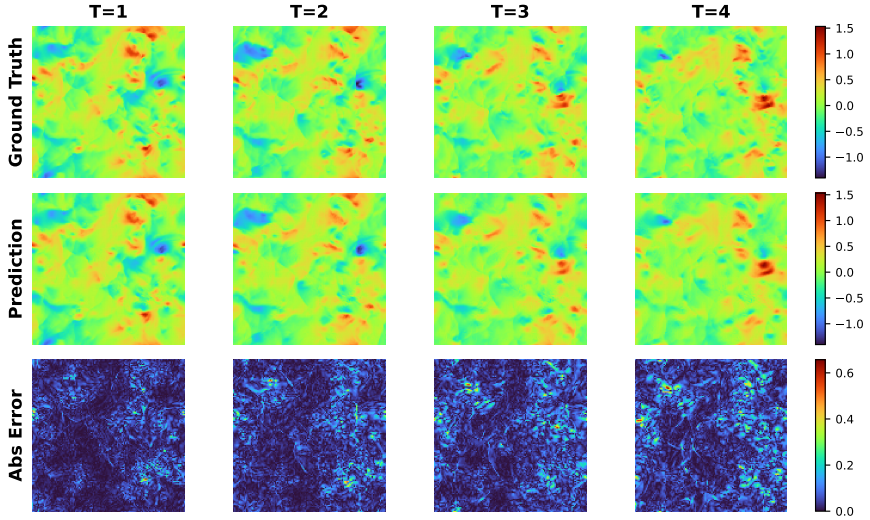


Figure 6: CE-CRP.

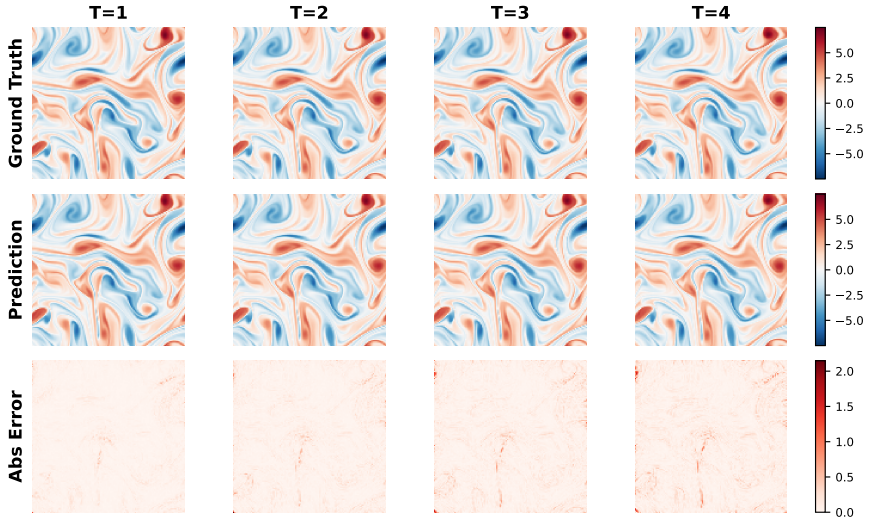


Figure 7: NSE.

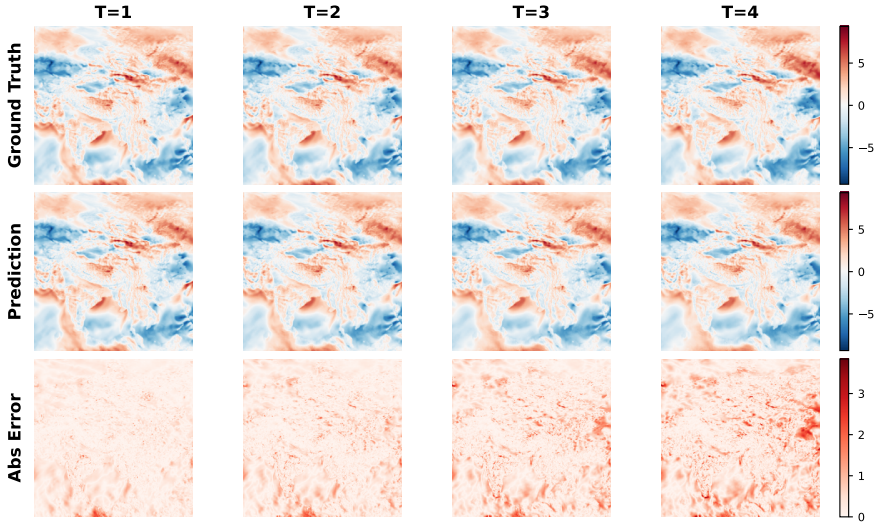


Figure 8: ERA5.