# Multimodal PDE Foundation Models and Applications to Physical Systems

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

## Abstract

We introduce MORPH, a multimodal foundation model for partial differential equations (PDEs) designed to learn across heterogeneous physics datasets with varying dimensionalities, resolutions, and fields. Its architecture combines component-wise convolutions, field-wise cross-attention, and factorized axial spatiotemporal attention, enabling scalable learning across diverse PDE modalities. We pretrain multiple model variants on a heterogeneous PDE corpus using a next-frame prediction objective and evaluate transfer across a broad set of downstream tasks. These include forward modeling tasks such as autoregressive rollouts and terminal key-frame prediction, as well as inverse tasks including composite material property estimation, damage detection in aerospace structures, inertial-confinement-fusion parameter estimation, and sparse sea-surface-temperature reconstruction. Across zero-shot evaluation, full-model fine-tuning, and parameter-efficient low-rank adaptation, pretrained MORPH models consistently outperform models trained from scratch. Our results demonstrate positive cross-modality transfer, improved data efficiency in low-data regimes, favorable model- and data-scaling behavior, and strong generalization to out-of-distribution physical systems. Collectively, these capabilities establish MORPH as a flexible backbone for learning from the heterogeneous and multimodal nature of scientific observations, charting a path toward scalable and data-efficient scientific machine learning.

## 1 Introduction

Many problems in physics are governed by partial differential equations (PDEs) where field variables exhibit complex spatiotemporal evolution. Machine learning-based surrogate models have made substantial progress in learning such spatiotemporal physical systems. Prominent standalone surrogates include neural operators such as DeepONet (Lu et al., 2021; Kontolati et al., 2024), Fourier Neural Operators (Li et al., 2020), the U-shaped Neural Operator (Rahman et al., 2022) and transformer-based models (Solera-Rico et al., 2024; Li et al., 2022). Inspired by the success of foundation models in natural language processing, recent research directions in scientific machine learning are transitioning from task-specific to task-agnostic modeling. We use the term "PDE foundation model" to denote a universal PDE surrogate or a universal PDE operator pretrained on large, diverse datasets spanning multiple physics. With task-specific fine-tuning, such models can outperform standalone surrogates and demonstrate utility in data- and compute-scarce scenarios.

*Data heterogeneity is a bottleneck in learning a PDE foundation model.* Two forces collide in practice. *Physics:* dynamical systems couple nonlinear, multiscale interactions across scalar and vector fields evolving in four-dimensional continuous space-time, often with widely separated characteristic scales. *Data:* unlike web-scale text or images, pretraining corpora for PDEs are scarce, costly to curate, and frequently provide multiple modalities. The existing benchmark spatiotemporal PDE datasets (Takamoto et al., 2022; Ohana et al., 2024; Herde et al., 2024) contain varying data modality (1D–3D) including trajectories, fields, components and resolutions curated through computationally costly simulations. In addition, such datasets can be terabytes in size, making naive homogenization or padding prohibitive. Experiments typically provide sparse, time series or low-dimensional slices of high-dimensional dynamics. For example, seismometers record 1D time series at sparse locations while seismic waves propagate as a (3+1)-D field; similarly, pressure probes and

velocimetry provide 1D/2D slices of fully spatiotemporal fluid dynamics. These constraints motivate models that learn from incomplete measurements and generalize across heterogeneous data or modalities.

*A modeling challenge under multiple modalities.* A foundation model with a transformer backbone provides flexibility in learning from partial physical information. Recent PDE foundation models have made substantial progress toward a task-agnostic universal PDE surrogate. However, most existing approaches (McCabe et al., 2024; Chen et al., 2024; Herde et al., 2024) implicitly assume near-homogeneous inputs typically 2D Cartesian grids with fixed scalar/vector components and limited multiphysics coverage. As a result, they scale poorly across modalities and resolutions: volumetric patching makes 3D prohibitively expensive by inflating sequence length and the quadratic attention cost $\mathcal{O}(L_{\text{seq}}^2)$, which also discourages high-resolution 2D pretraining. At the other extreme, 1D data are often forced into 2D pipelines via padding or tiling, wasting compute without adding information. We treat data heterogeneity or multi-modality as first-class requirements. This gap is highlighted by concurrent work (Ye et al., 2024; McCabe et al., 2024) and remains a key bottleneck, motivating a flexible, scalable PDE foundation model that can learn across diverse scientific modalities.

**Our Approach.** We introduce MORPH, a PDE foundation model designed to accommodate heterogeneous data across diverse physical phenomena. It is shape-agnostic, with physics-aware channel handling of PDE datasets. The architecture combines three mechanisms: (a) component-wise convolutions to capture local interactions while jointly processing scalar and vector channels; (b) inter-field cross-attention that models and selectively propagates information across physical fields while condensing them into a single fused representation; and (c) axial attention that factorizes full spatiotemporal self-attention along spatial and temporal axes, reducing compute while retaining expressivity. We define a Unified Physics Tensor Format (UPTF-7) for mini-batches that generalizes across PDE datasets while preserving the semantics of the physical observations. We pretrain MORPH on six heterogeneous spatiotemporal datasets spanning 1D–3D spatial domains with multiple field compositions. We perform full fine-tuning and parameter-efficient fine-tuning with low-rank adapters (LoRA) on seven additional heterogeneous datasets for downstream prediction tasks. We release four model variants: MORPH-Tɪ (7M), MORPH-S (30M), MORPH-M (126M), and MORPH-L (480M).

**Contributions.** Our specific contributions are:

- **Modality-agnostic universality.** We introduce a modality-agnostic model that operates across arbitrary spatiotemporal PDE dataset dimensionalities (1D–3D), resolutions and mixed scalar/vector fields without task-specific reconfiguration.
- **Expanded spatiotemporal context.** MORPH employs a larger transformer architecture with one cross-attention and four axial attention modules that attend over a *multi-fold* increase in spatiotemporal patches (i.e., a larger context window).
- **Diverse pretraining and transfer.** We pretrain and fine-tune on a broad heterogeneous suite drawn from three benchmarks including multi-physics datasets like magnetohydrodynamics (MHD), turbulent self-gravitating flows with cooling (TGC), high-resolution 2D compressible and incompressible Navier–Stokes and large-scale 3D datasets.
- **Parameter-efficient adaptation.** We adapt LoRA for fine-tuning our largest model (MORPH-L), showing that low-rank adapters recover most of the gains of full fine-tuning with substantially fewer trainable parameters.
- **Forward and inverse scientific applications.** We evaluate MORPH on autoregressive rollouts, terminal key-frame prediction, material property estimation, structural damage detection, inertial-confinement-fusion parameter estimation, and sparse sea-surface-temperature reconstruction, demonstrating broad utility beyond standard PDE forecasting benchmarks.

**Organization of the paper.** The manuscript is organized into two main parts. The first part, comprising Sections 3–8, establishes MORPH as a modality-agnostic PDE foundation model and evaluates its behavior

on heterogeneous spatiotemporal benchmark systems. *Section 3 presents the core method*: the Unified Physics Tensor Format (UPTF-7), the MORPH architecture, its component-wise convolutional encoder, field-wise cross-attention module, factorized spatiotemporal axial attention blocks, positional encodings, LoRA parameterization, model variants, and a controlled cross-modality transfer analysis. *Section 4 details the experimental setup*, including normalization, streaming and distributed sharding, pretraining and fine-tuning protocols, balanced task sampling, and compute settings. *Section 5 reports the main quantitative results* against standalone baselines and recent PDE foundation models, covering zero-shot evaluation, full-model fine-tuning, LoRA fine-tuning, and data-scarce adaptation. *Sections 6 and 7 analyze scaling behavior and ablations*, while *section 8 presents autoregressive rollout studies* that assess the stability of learned dynamics over multiple prediction steps. The second part begins in Section 9 and studies whether the pretrained MORPH backbone can transfer beyond canonical PDE benchmarks to practical physical-system tasks. These applications include terminal key-frame prediction for extreme material dynamics, inverse parameter estimation in inertial-confinement fusion, composite material property estimation, guided-wave damage detection in aerospace structures, and sparse sea-surface-temperature reconstruction.

## 2 Related Work

**Standalone PDE surrogate models.** A wide range of ML-based standalone surrogate models are deployed for solving PDEs. Some of these methods are 3D-CNN (Wandel et al., 2021), ConvLSTM (Shi et al., 2015), DCGAN (Cheng et al., 2020) and GNNs (Chen et al., 2021). Beyond purely data-driven surrogates, physics-informed neural networks (PINNs) have been applied to spatiotemporal problems (Raissi et al., 2019). Recently, neural operators like DeepONet (Goswami et al., 2022), Fourier Neural Operators (FNOs) (Li et al., 2020), wavelet neural operators (WNOs) (Tripura & Chakraborty, 2023), Laplace neural operators (Cao et al., 2024) and physics-informed neural operators (PINOs) (Li et al., 2024b) have gained attention for solving spatiotemporal problems. Another line of methods uses generative models to capture spatiotemporal data distributions (Buzzicotti et al., 2021; Zhuang et al., 2025; Li et al., 2024a; Gao et al., 2025). Researchers have investigated latent evolution models where spatial learning is performed through autoencoders (Oommen et al., 2022; Vlachas et al., 2022; Maulik et al., 2021; Kontolati et al., 2024) and variational autoencoders (Solera-Rico et al., 2024; Rautela et al., 2024; 2025). Temporal dynamics in the latent space have been modeled with recurrent neural networks including LSTMs (Maulik et al., 2021; Vlachas et al., 2022; Liu et al., 2022; Rautela et al., 2024), Koopman with non-linear forcing (Khodkar & Hassanzadeh, 2021), DeepONets (Oommen et al., 2022; Kontolati et al., 2024), ODE-Nets (Chen et al., 2018) and transformers with self-attention and easy-attention (Solera-Rico et al., 2024; Alkin et al., 2024).

**PDE foundation models.** Standalone PDE surrogates are typically specialized to one equation or parameterized family, which means they must be retrained from scratch whenever the PDE form or conditions change. This lack of generality leads to significant computational overhead without knowledge transfer across PDEs. Some initial works have successfully demonstrated the capability of transfer learning for PDEs (Goswami et al., 2022; Xu et al., 2023). Foundation models for PDEs have emerged to overcome these limitations by capturing the common patterns underlying many different PDE systems. A PDE foundation model is a task-agnostic generalist model pretrained on diverse data, followed by minimal task-specific fine-tuning (Menon et al., 2026). Recent works (Yang et al., 2023; Chen et al., 2024; Herde et al., 2024; Chen et al., 2018; Liu et al., 2024; Ye et al., 2024) have demonstrated that a PDE foundation model can outperform standalone models. The advantages in generality, data efficiency, transferability, and versatility motivate the development of foundation PDE models over traditional one-off surrogates (Subramanian et al., 2023).

## 3 Proposed Method

### 3.1 Datasets and Unified Physics Tensor Format

To demonstrate that MORPH accommodates data heterogeneity, we select datasets for both pretraining and fine-tuning. The pretraining and fine-tuning datasets and their shapes are listed in Table 1 and 2. These datasets come from popular benchmarks like PDEBench (Takamoto et al., 2022), The Well (Ohana et al., 2024), and PDEGym (Herde et al., 2024). Compared to existing PDE foundation models, our pretraining

and fine-tuning sets are larger and more diverse. The pretraining and fine-tuning datasets, along with their field compositions and tensor shapes, are summarized in Tables 1 and 2. Collectively, this suite spans 1D–3D modalities, resolutions from 1D grids of length 1024 to 2D grids up to 512×512 and 3D volumes up to $128^3$, and includes both scalar and vector fields across single- and multi-physics systems. For ablation studies, we use 3 more datasets, i.e., NS-SL, NS-PwC, and CE-RM from PDEGYM (Herde et al., 2024). More details about them are discussed later. For brevity, data details are deferred to Appendix A.

Table 1: Pretraining datasets with their physical fields (scalar or vector) and data shapes (N,T,D,H,W,C), together with the benchmark/source.

| PT sets | MHD-3D | DR-2D | CFD-1D | CFD2D-IC | CFD-3D | SW-2D |
|---|---|---|---|---|---|---|
| Shapes (fields) | $\rho$: $(97, 100, 64^3)$<br>$\mathbf{B}$: $(97, 100, 64^3, 3)$<br>$\mathbf{v}$: $(97, 100, 64^3, 3)$ | $u_1$: $(1k, 100, 128^2)$<br>$u_2$: $(1k, 100, 128^2)$ | $\rho$: $(10k, 100, 1024)$<br>$\mathbf{v}$: $(10k, 100, 1024)$<br>$P$: $(10k, 100, 1024)$ | $\mathbf{v}$: $(64, 1k, 512^2, 2)$ | $\rho$: $(200, 21, 128^3)$<br>$\mathbf{v}$: $(200, 21, 128^3, 3)$<br>$P$: $(200, 21, 128^3)$ | $h$: $(1k, 100, 128^2)$ |
| Source | The Well | PDEBench | PDEBench | PDEBench | PDEBench | PDEBench |

Table 2: Fine-tuning datasets with their physical fields (scalar or vector) and data shapes (N,T,D,H,W,C), together with the benchmark/source.

| FT sets | DR-1D | CFD-2D | CFD3D-TURB | BE-1D | GSDR-2D | TGC-3D | FNS-KF-2D |
|---|---|---|---|---|---|---|---|
| Shapes (fields) | $u_1$: $(10k, 100, 1024)$ | $\rho$: $(10k, 21, 512^2)$<br>$\mathbf{v}$: $(10k, 21, 512^2, 2)$<br>$P$: $(10k, 21, 512^2)$ | $\rho$: $(600, 21, 64^3)$<br>$\mathbf{v}$: $(600, 21, 64^3, 3)$<br>$P$: $(600, 21, 64^3)$ | $u$: $(10k, 200, 1024)$ | $A$: $(200, 1k, 128^2)$<br>$B$: $(200, 1k, 128^2)$ | $\mathbf{v}$: $(100, 50, 64^3, 3)$<br>$P$: $(100, 50, 64^3)$<br>$T$: $(100, 50, 64^3)$ | $u$: $(20k, 21, 128^2)$<br>$v$: $(20k, 21, 128^2)$ |
| Source | PDEBench | PDEBench | PDEBench | PDEBench | The Well | The Well | PDEgym |

The most direct way to handle data heterogeneity is to pad the datasets to a common shape. However, raw PDE datasets can occupy gigabytes to terabytes of storage. For instance, THE WELL alone contains ∼15 TB of physics simulations (Ohana et al., 2024). This makes naive padding both compute and storage-wise inefficient. We therefore seek a unified dataset format that generalizes across PDE datasets while preserving the semantics of the physical observations. We observe that popular benchmarks like PDEBENCH (Takamoto et al., 2022), THE WELL (Ohana et al., 2024), PDEGYM (Herde et al., 2024), PDEARENA (Gupta & Brandstetter, 2022), and CFDBENCH (Yining et al., 2023) can be represented via a generic 7D tensor. We denote it as $(b, t, F, C, D, H, W)$, where $N$ denotes number of trajectories, $T$ are the number of time steps, $N \times T$ becomes the total number of pretraining samples under next-step prediction (NSP) strategy, $F$ are the number of field variables, $C$ are the associated scalar/vector components per field, and $(D, H, W)$ are the spatial depth, height, and width. We retain each benchmark's native on-disk format and convert mini-batches on the fly to our *Unified Physics Tensor Format* (UPTF-7) when loading to GPUs. For a batch size of b with input temporal context window of t (set to 1), the tensor shape (coming to GPUs) for various datasets are

$$
\begin{aligned}
\text{1D-CFD} &= (b, t, 3, 1, 1, 1, 1024), & \text{2D-DR} &= (b, t, 2, 1, 1, 128, 128),\\
\text{2D-CFD-IC} &= (b, t, 1, 2, 1, 512, 512), & \text{2D-SW} &= (b, t, 1, 1, 1, 128, 128),\\
\text{3D-MHD} &= (b, t, 3, 3, 64, 64, 64), & \text{3D-CFD} &= (b, t, 3, 3, 128, 128, 128),\\
\text{1D-DR} &= (b, t, 1, 1, 1, 1, 1024), & \text{1D-BE} &= (b, t, 1, 1, 1, 1, 1024),\\
\text{2D-FNS-KF} &= (b, t, 1, 2, 1, 128, 128), & \text{2D-CFD} &= (b, t, 3, 2, 1, 512, 512),\\
\text{2D-GSDR} &= (b, t, 2, 1, 1, 128, 128), & \text{3D-CFD-TURB} &= (b, t, 3, 3, 64, 64, 64),\\
\text{3D-TGC} &= (b, t, 3, 3, 64, 64, 64).
\end{aligned}
$$

The datasets used for the forward and inverse application studies in the second part of the paper are described in their respective application subsections.

### 3.2 Overview of MORPH.

A schematic of the model is shown in Fig. 1. The shape-agnostic design of MORPH handles data heterogeneity via data-channel-specific operators. The architecture comprises three key mechanisms: (a) *component-wise convolutions*, which jointly process scalar and vector channels to capture local interactions; (b) *inter-field multi-head cross-attention*, which models and selectively propagates information across physical fields; and (c) *4D axial attentions*, which factorize full spatiotemporal self-attention along individual spatial and temporal

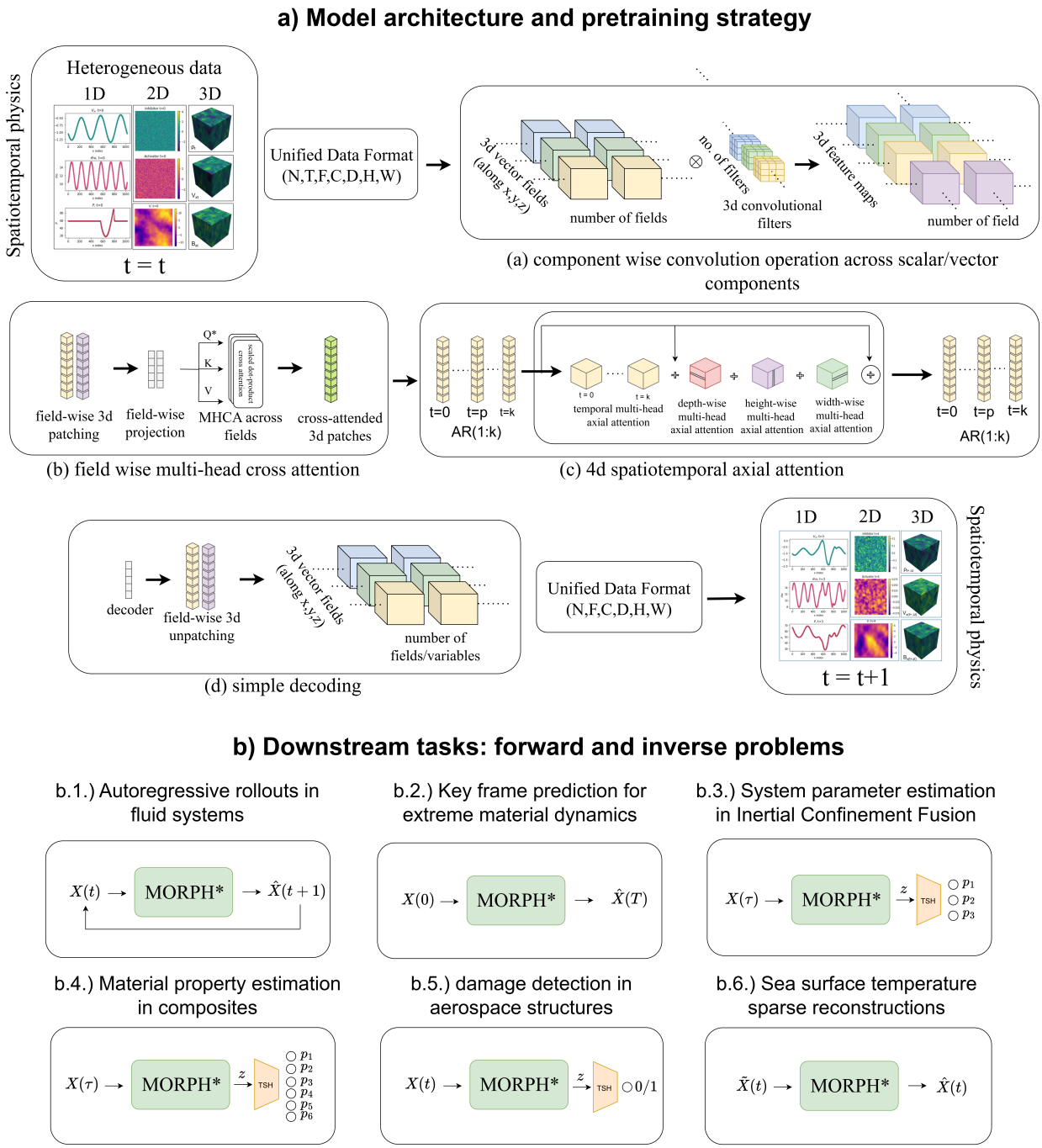



Figure 1: **MORPH architecture and downstream task adaptation.** (a) Heterogeneous 1D–3D spatiotemporal physics data are mapped to a unified tensor format and processed by MORPH through component-wise convolutions, field-wise cross-attention, factorized 4D axial attention, and a lightweight decoder for next-frame prediction. This design enables a single backbone to operate across different resolutions, spatial dimensionalities, and mixed scalar/vector field compositions. (b) The pretrained backbone is adapted to downstream forward and inverse scientific tasks, including autoregressive rollouts, terminal key-frame prediction, inertial-confinement-fusion parameter estimation, composite material property estimation, aerospace damage detection, and sparse sea-surface-temperature reconstruction. MORPH* denotes the task-adapted pretrained model, and TSH denotes a task-specific head.

axes to reduce computational burden while retaining expressivity. We validate the necessity of each design choice (component-wise convolution, field fusion, and axial attention) through ablations in Sec. 7.1.

**Convolutional operator.** It acts on the component channel(s) of the UPTF-7 batch, producing a set of feature maps determined by the number of filters. Adding convolutional inductive bias helps vision transformers capture locality and improves sample efficiency (Dosovitskiy et al., 2020). We can scale feature maps and ultimately MORPH via the number of convolutional filters (default: 8). We partition feature maps into patches of size 8. Across pretraining and fine-tuning datasets, we use at most 4,096 patches, yielding a 32,768-dimensional feature vector, which we linearly project to an embedding whose dimensionality ranges from 256 to 1,024 depending on the model variant. By default, we use 8 filters and a 2-layer network with LeakyReLU activations. Inputs with up to `max_in_ch` channels are zero-padded (if needed) and projected to $h$=8 channels by a 1×1×1 convolution (bias-free). We then apply $L$ blocks of 3×3×3 convolution followed by LeakyReLU ($\alpha$=0.2), doubling the channel width each block until reaching $F$ output channels: $h \to \min(2h, F) \to \cdots \to F$. All 3×3×3 layers use padding = 1 to preserve spatial size. Unless noted, $F$=8, yielding two convolutional layers in total (the 1×1×1 projection and one 3×3×3 layer).

**Field-wise multi-head cross-attention.** In most PDE datasets, multiple fields with different spatial scales evolve at various temporal rates. Field-fusion prioritizes relevant inter-field interactions and selectively propagates information, enabling careful feature fusion while suppressing spurious activations arising from scale mismatches among coupled fields. The cross-attention operator outputs a single fused field, significantly reducing computational cost (Nguyen et al., 2023). We scale MORPH further via the cross-attention dimension, which by default matches that of axial self-attention. After projecting each field patch to a common embedding size (`model_dim`), we form a length-$F$ sequence $\mathbf{X} \in \mathbb{R}^{F \times E}$ per patch and apply a single-layer, $H$-head cross-attention in which the *query* is learnable ($\mathbf{q} \in \mathbb{R}^{E}$), while $\mathbf{X}$ provides both keys and values. The cross-attention operation performs content-based pooling across the $F$ field variables via a learned query. The time complexity scales linearly with $F$ (a single query attends over $F$ keys). The module naturally accommodates a variable number of fields F at runtime and inference time. We intentionally omit field-wise positional encodings, making the field-fusion module permutation-invariant to field ordering and allowing any field to appear in any position.

Concretely, with head dimension $d_h = E/H$ and per-head projections $W_Q^{(h)}, W_K^{(h)}, W_V^{(h)} \in \mathbb{R}^{E \times d_h}$,

$$\alpha^{(h)} = \mathrm{softmax}\Big(\frac{(\mathbf{q}W_Q^{(h)})(\mathbf{X}W_K^{(h)})^\top}{\sqrt{d_h}}\Big) \in \mathbb{R}^{1 \times F}, \qquad \mathbf{z}^{(h)} = \alpha^{(h)}(\mathbf{X}W_V^{(h)}) \in \mathbb{R}^{1 \times d_h},$$

and the fused representation is

$$\mathbf{z} = \mathrm{Concat}_h(\mathbf{z}^{(h)})\, W_O \in \mathbb{R}^{1 \times E},$$

**Tokenization/Patchification.** By default, each sample is partitioned into non-overlapping **patches** of size 8 along the spatial axes $(D, H, W)$, although the model supports a range of patch sizes. Accordingly, the per-patch shapes are $1 \times 8 \times 8$ for 2D data, $1 \times 1 \times 8$ for 1D data, and $8 \times 8 \times 8$ for 3D data. In our pretraining corpus, the 2D-CFD-IC and 3D-CFD datasets yield the largest number of spatial patches per batch, i.e., `max_patches` = 4096. Practically, the choice of patch size is constrained by available compute. Reducing the patch size increases the sequence length and thus raises memory and runtime costs during training and inference.

**Axial attention (factorized 4D space–time).** We factorize full 4D spatiotemporal attention for a (3+1)D dynamical system into time-wise, depth-wise, height-wise, and width-wise *axial attentions*. For high-dimensional datasets, this limits the computational cost to $\mathcal{O}\big(tDHW\,(t + D + H + W)\big)$, in contrast to full spatiotemporal attention with cost $\mathcal{O}\big((TDHW)^2\big)$ (Ho et al., 2019). Time-wise axial attention supports flexible temporal context allowing fixed or variable order autoregression (default: 1 step).

Given patch embeddings $x \in \mathbb{R}^{B \times t \times N \times E}$ with $N$=$DHW$, we reshape to $x \in \mathbb{R}^{B \times t \times D \times H \times W \times E}$ and replace full space–time self-attention over $L$=$tDHW$ tokens with four 1D multi-head attention (MHA) operations applied along each axis i.e., time ($t$), depth ($D$), height ($H$), and width ($W$). For each axis we fold the

remaining axes into the batch, attend along the axis length, invert the reshape, and use a residual sum:

$$\begin{aligned}
\tilde{x}^{(t)}_{b,d,h,w} &= \mathrm{MHA}_t\big(x_{b,\cdot,d,h,w}\big) \in \mathbb{R}^{t\times E},\\
\tilde{x}^{(D)}_{b,t,h,w} &= \mathrm{MHA}_D\big(x_{b,t,\cdot,h,w}\big) \in \mathbb{R}^{D\times E},\\
\tilde{x}^{(H)}_{b,t,d,w} &= \mathrm{MHA}_H\big(x_{b,t,d,\cdot,w}\big) \in \mathbb{R}^{H\times E},\\
\tilde{x}^{(W)}_{b,t,d,h} &= \mathrm{MHA}_W\big(x_{b,t,d,h,\cdot}\big) \in \mathbb{R}^{W\times E},\\
y &= x \;+\; \tilde{x}^{(t)} \;+\; \tilde{x}^{(D)} \;+\; \tilde{x}^{(H)} \;+\; \tilde{x}^{(W)},
\end{aligned}$$

followed by flattening back to $\mathbb{R}^{B\times t\times N\times E}$. We adopt a pre-norm Transformer block: LN $\rightarrow$ axial attention $\rightarrow$ residual, then LN $\rightarrow$ MLP (GELU, Dropout) $\rightarrow$ residual. The temporal branch is enabled only when $t{>}1$, which supports an autoregressive curriculum (stage 1 uses spatial-only AR(1); stage 2 enables AR(2:$k$)).

Full space–time attention costs $\mathcal{O}\big((tDHW)^2\big)$ in sequence length. Axial factorization reduces this to

$$\mathcal{O}\big(tDHW\,(t+D+H+W)\big),$$

since each axis attends over its own length while other dimensions are folded into the batch, yielding substantial savings in compute and memory without sacrificing global receptive field.

**Positional encodings.** Let a learned absolute table $\mathbf{P} \in \mathbb{R}^{\texttt{max_ar}\times\texttt{max_patches}\times E}$ provide time–patch embeddings, and let token tensors be $x \in \mathbb{R}^{B\times t\times n\times E}$. At runtime we construct $\widehat{\mathbf{P}} \in \mathbb{R}^{t\times n\times E}$ to match the current autoregressive context $t$ and patch count $n$, and add it to $x$ (broadcast over the batch). Here, "patches" refer to the flattened patch sequence index obtained from a fixed rasterization of the $(D, H, W)$ patch grid.

In this, we *slice* the first $t$ time steps and *linearly* resample only along the patch axis:

$$\widehat{\mathbf{P}}_{\tau,\pi} \;=\; \sum_{j=1}^{\texttt{max_patches}} w^{(n)}_{\pi,j}\,\mathbf{P}_{\tau,j}, \qquad \tau \in \{1,\dots,t\},\;\; t \le \texttt{max_ar},$$

yielding $\widehat{\mathbf{P}} \in \mathbb{R}^{t\times n\times E}$. This avoids temporal smoothing, preserves the semantics of discrete AR time steps, and is lighter-weight computationally. We use this variant for the **Ti/S/M/L** models with $\texttt{max_ar} = 1$ (consistent with the $AR(1)$ setting used in this paper). We apply dropout after constructing $\widehat{\mathbf{P}}$ and then add it to token embeddings prior to attention.

**LoRA parameterization.** Low-Rank Adaptation (LoRA) is used for fine-tuning LLMs where the pretrained model weights are frozen while introducing trainable rank decomposition matrices into the transformer layers, dramatically reducing task-specific parameters during fine-tuning (Hu et al., 2022). The motivation is supported by evidence that over-parameterized networks possess low intrinsic dimensionality, suggesting that task-specific updates lie in a low-rank subspace (Aghajanyan et al., 2020).

All attention projections (Q, K, V, O) and the two MLP linear layers are equipped with low-rank adapters. When rank $r{=}0$ they reduce to plain linear maps. For input $x$ and base weight $W_0$, a LoRA-enhanced linear layer produces

$$y \;=\; xW_0^\top \;+\; \tfrac{\alpha}{r}\,\big(xA^\top\big)B^\top \;+\; b,$$

with $A \in \mathbb{R}^{r\times \mathrm{in}}$, $B \in \mathbb{R}^{\mathrm{out}\times r}$, scaling $\alpha/r$, and optional dropout $p$ on the LoRA path. This adds $r(\mathrm{in}+\mathrm{out})$ parameters per linear while leaving the base weights reusable/freezable.

In this paper, we introduce four different variants of MORPH presented in Table 3, based on the number of convolutional filters, dimensions of attention layers, number of attention heads and depth of the transformer.

### 3.3 Transfer across data modalities

A central question for a PDE foundation model is *whether representations learned on one modality (e.g., 2D incompressible flows) transfer across multiple data modalities such as 1D, 2D and 3D*. MORPH is explicitly

Table 3: Four Variants of MORPH with the associated hyperparameters.

| Size | Conv. filters | Attention dims | Heads | Depth | MLP dims | Model size |
|---|---|---|---|---|---|---|
| MORPH-Ti (Tiny) | 8 | 256 | 4 | 4 | 1024 | 7M |
| MORPH-S (Small) | 8 | 512 | 8 | 4 | 2048 | 30M |
| MORPH-M (Medium) | 8 | 768 | 12 | 8 | 3072 | 126M |
| MORPH-L (Large) | 8 | 1024 | 16 | 16 | 4096 | 480M |

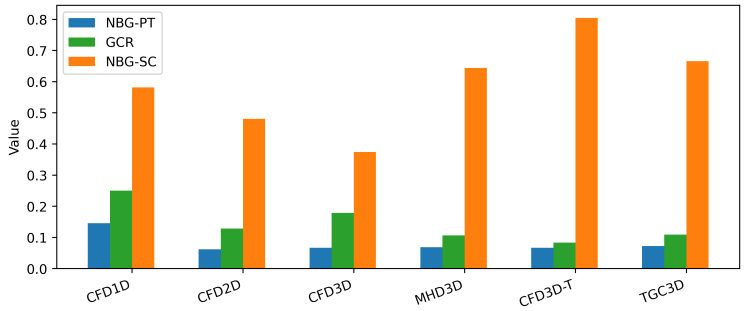



Figure 2: **Zero-shot transfer from CFD2D-IC pretraining.** Targets: CFD1D, CFD2D, CFD3D, MHD3D, CFD3D-Turb, TGC3D. Bars show the Naive Baseline Gain of the pretrained model (**NBG-PT**), the Naive Baseline Gain of a trained-from-scratch model (**NBG-SC**), and the resulting **Gap-Closure Ratio (GCR)**. GCR $> 0$ indicates cross-modality transfer.

designed to be modality-agnostic while maintaining generalization across diverse physics. To test this hypothesis, we pretrain MORPH exclusively on the 2D incompressible Navier–Stokes dataset (CFD2D-IC) under an autoregressive objective. We then perform *zero-shot* inference (no fine-tuning, all weights frozen) on six heterogeneous targets (fluids based) spanning different modalities: CFD1D, CFD2D, CFD3D, CFD3D-Turb, MHD3D, and TGC3D.

To quantify transfer, we define zero-shot *Gap-Closure Ratio* (GCR), which measures what fraction of the performance gap between a naive, training-free baseline and a fully trained-from-scratch model is closed by the pretrained model in the zero-shot setting. Formally,

$$\mathrm{GCR} = \frac{1 - (E_{\mathrm{PT}}/E_{\mathrm{NB}})}{1 - (E_{\mathrm{SC}}/E_{\mathrm{NB}})} \tag{1}$$

Here, $E_{\mathrm{PT}}$ is the zero-shot error of the CFD2D-IC pretrained model evaluated on a given target, $E_{\mathrm{SC}}$ is the error of an identical architecture trained fully from scratch on that target, and $E_{\mathrm{NB}}$ is the error of a training-free naive baseline (the same architecture with random initialization and no learning, informally corresponding to near–coin-toss performance), taken as the minimum over 25 random seeds. We define $1 - E_{\mathrm{PT}}/E_{\mathrm{NB}}$ as the naive-baseline gain of the pretrained model (NBG-PT), and $1 - E_{\mathrm{SC}}/E_{\mathrm{NB}}$ as the naive-baseline gain of the trained-from-scratch model (NBG-SC). In simpler terms, GCR is the ratio NBG-PT to NBG-SC. Our definition of zero-shot GCR is inspired by performance-gap-recovered (PGR) metric introduced in Burns et al. (2023).

*Interpretation*: GCR $= 0$ means zero-shot does no better than naive. GCR $= 1$ means zero-shot matches a fully trained scratch model; GCR $> 1$ indicates zero-shot surpasses a model trained from scratch; any GCR $> 0$ demonstrates positive transfer across modalities and physics. Fig. 2 summarizes transfer from CFD2D-IC to the six targets via three quantities: NBG-PT, NBG-SC and the resulting GCR. We observe consistent positive transfer (GCR $> 0$) across all fluid-specific targets (green bar), with the strongest effects on near-domain problems (e.g., compressible CFD1D, CFD2D and CFD3D) and attenuated but still nontrivial transfer to CFD3D (with turbulence) and multiphysics settings such as MHD3D and self-gravitating TGC3D. These results indicate that 2D pretraining learns reusable operators that carry to 1D/2D/3D data modalities, providing empirical support for MORPH's modality-agnostic design.

# 4 Experimental details

## 4.1 Normalization.

We use Reversible Instance Normalization (ReVIN) to counter covariate shift across heterogeneous datasets and training regimes. It was first proposed for time-series forecasting and validated to mitigate distribution shift (Kim et al., 2021). We normalize the data and cache the corresponding means and standard deviations, then use these statistics to exactly denormalize model outputs.

## 4.2 Data streaming and sharding setup.

MORPH is designed to handle data heterogeneity and diversity, which introduces system-level challenges. To utilize CPU, GPU, and RAM efficiently, we make several non-traditional choices. One issue is that compute-cluster resources impose RAM limits, making it difficult to load all datasets into memory simultaneously. As PDE benchmarks continue to grow, streaming becomes essential for building large PDE foundation models. Accordingly, we stream datasets from storage in chunks, which uses RAM efficiently and scales with the number and size of datasets. Another issue is the wide range of token lengths (128–4096) across pretraining datasets, which leads to varying computational burdens in parallel processing across batches. Instead of a single PyTorch `DataLoader` for all datasets, we use one dataloader per dataset. With six pretraining datasets, we run six DataLoaders. This enables dataset-specific choices of batch sizes and number of workers. Given this multi-dataloader setup, data sharding under `DistributedDataParallel`(DDP) is nontrivial, so we implement custom sharding across workers and ranks.

We deploy six PyTorch `DataLoader` instances and use `DistributedDataParallel` (DDP) to shard data across ranks and workers while streaming. The data chunks are loaded from storage into RAM, locally shuffled, and converted into AR pairs. We first count the total number of autoregressive (AR) samples $T$ across all HDF5 files, then balance the workload across $W$ ranks and $K$ `DataLoader` workers by setting $G = W \cdot K$ and dropping any remainder so $T^*$ is divisible by $G$. Each sub-worker receives exactly $m = T^*/G$ samples and is identified by $\texttt{my_id} = \texttt{rank} \cdot K + \texttt{worker_id}$. As we stream the data in chunks, we maintain a running global index $g$ for every candidate sample; a sub-worker processes a sample iff $g\%G = \texttt{my_id}$. We reseed the RNG each epoch with `base_seed`+`epoch` and only shuffle within each chunk before forming AR pairs, yielding a single-pass, low-memory pipeline with no duplicate samples and balanced work across all ranks and workers. More details of this setup is presented in Algorithm 1.

## 4.3 Pretraining settings.

**Pretraining objective.** Our model accommodates both fixed–order autoregressions, $AR(p)$, and time-varying orders, $AR(p_{1:T})$, where $p$ (or $p_t$) denotes the autoregressive order. In this paper, we train MORPH in a self-supervised $AR(1)$ setting where we learn a map $F_\theta : \mathcal{X} \to \mathcal{X}$ such that for all $t \geq 0$, $X_{t+1} \approx F_\theta(X_t)$ by minimizing the mean-squared error. High–order fixed AR models (e.g., $AR(16)$ in McCabe et al. (2024) and $AR(10)$ in Hao et al. (2024)) often yield stronger long-horizon rollouts by reducing exposure bias via a longer receptive field. However, this design departs from the initial-value-problem (IVP) semantics that PDE foundation models aim to respect. For these reasons we adopt a one-step ($AR(1)$) formulation aligned with IVP semantics that preserves the compositional structure fundamental to PDE time evolution. Nevertheless, MORPH remains flexible and supports multi-order (fixed or varying) autoregression to improve rollout quality.

**Balanced task sampling.** We train MORPH on multiple pretraining datasets with varying numbers of trajectories and time-steps. We perform balanced task sampling to avoid imbalanced learning and catastrophic forgetting. Each dataset is assigned a sampling weight inversely proportional to its number of snapshots (trajectories $\times$ time-steps), so datasets with fewer snapshots are randomly sampled more frequently. In our corpus, the 3D-MHD dataset has the fewest snapshots and is therefore sampled more often than the remaining five. For multi-dataset training, each dataset $i$ is assigned a sampling weight inversely proportional to its number of snapshots $N_i$, i.e., $w_i \propto 1/N_i$, and the weights are normalized so that $\sum_i w_i = 1$. The resulting

**Algorithm 1** Simple Streaming Sharding (DDP ranks × `DataLoader` workers)

**Require:** Ordered files `FILES`, autoregressive window `ar_order`, chunk size `chunk_size`, world size $W$, `rank`, num workers $K$, `worker_id`, `base_seed`, `epoch`
**Ensure:** Stream of $(X, y)$ pairs for this sub-worker

```
T ← Σ_{f∈FILES} N_sims(f) · max(0, T_steps(f) − ar_order)
G ← W · K
T* ← T − (T mod G)
m ← T*/G
my_id ← rank · K + worker_id
global_idx ← 0
Seed RNG ← base_seed + epoch
for all f ∈ FILES do
  Open f and iterate in chunks of size chunk_size
  Shuffle each chunk with the epoch seed and prepare AR pairs (X, y)
  for all (x_i, y_i) ∈ (X, y) do
    if global_idx ≥ T* then
      stop
    end if
    if (global_idx mod G) = my_id then
      yield (x_i, y_i)
      if this sub-worker has yielded m samples then
        stop
      end if
    end if
    global_idx ← global_idx + 1
  end for
end for
```

empirical per-batch sampling probabilities are approximately: 3D-MHD: 0.31, 2D-CFD-IC: 0.19, 3D-CFD: 0.18, 2D-DR: 0.12, 2D-SW: 0.12, and 1D-CFD: 0.08.

**LoRA layers.** We apply LoRA to the dense projections in attention and MLP blocks of the large model, with default ranks $r_{\text{attn}} = 16$ and $r_{\text{mlp}} = 12$, respectively, with dropout $p = 0.05$.

**Compute infrastructure.** We ran experiments across several GPU configurations, depending on the training stage and experiment type.

- **Pretrained foundation models:** All MORPH foundation-model variants were pretrained using 4× NVIDIA H100 GPUs with 80 GB memory each.
- **Standalone surrogates:** All 12 MORPH standalone surrogate models were trained using 2× NVIDIA A100 GPUs with 40 GB memory each.
- **Main fine-tuning experiments:** The MORPH foundation-model variants reported in Table 4 were fine-tuned using 1× NVIDIA H100 GPU.
- **SOTA foundation-model comparisons:** MPP, DPOT, and POSEIDON were fine-tuned using 1× NVIDIA H100 GPU.
- **Architecture ablations:** The model-architecture ablations reported in Table 6 were performed using 4× NVIDIA H100 GPUs.
- **Pretraining-corpus ablation:** For the pretraining-corpus ablation, MORPH-FM-S$^*$ was pretrained using 4× NVIDIA H100 GPUs and subsequently fine-tuned on each downstream task using 1× NVIDIA H100 GPU.

- **Application studies:** For the application studies described in Sec. 9, compute resources are specified in the corresponding subsections.

**Other pretraining details.**

- **Data splits**: Train/Val/Test: 0.8/0.1/0.1
- **Data pipeline**: Custom chunked streaming with deterministic sharding across workers.
- **Training duration**: Standalone and fine-tuned models are trained for approximately 100 true epochs. For foundation models, we pretrain all variants to 625K gradient steps.
- **Batch size**: Dataset heterogeneity necessitates dataset-specific batch sizes, implemented via per-dataset `DataLoader`. Accounting for our continuous data-streaming pipeline and targeting high GPU utilization on a 2× A100/A6000 setup, we use the following per-GPU batch sizes:

$$\begin{gathered}\text{1D-CFD} = 128, \quad \text{2D-DR} = 64, \quad \text{2D-CFD-IC} = 16, \quad \text{2D-SW} = 64 \\ \text{3D-MHD} = 16,\ \text{3D-CFD} = 4,\ \text{1D-DR} = 384,\ \text{1D-BE} = 384,\ \text{2D-FNS-KF} = 64 \\ \text{2D-CFD} = 8, \quad \text{2D-GSDR} = 64, \quad \text{3D-CFD-Turb} = 16, \quad \text{3D-TGC} = 16.\end{gathered}$$

  The batch sizes are reduced by 0.25x when pretraining foundation models.
- **Optimizer**: For foundation models, we use AdamW (Loshchilov & Hutter, 2017) with a learning rate of $5 \times 10^{-5}$. For standalone datasets, we use a learning rate of $10^{-4}$. The learning-rate schedule is `ReduceLROnPlateau` with a decay factor of 0.5 and a patience of 5 epochs, applied after a 20-epoch warmup.
- **Early stopping**: We trigger early stopping if the validation loss fails to improve for more than 10 epochs. In practice, early stopping never activates when pretraining the foundation models.

### 4.4 Fine-tuning settings

**Hyperparameters.** We use the same normalization and effective batch size as above. Fine-tuning runs for 100–200 true epochs with early stopping, and the learning rate is scheduled with `ReduceLROnPlateau`. We optimize with AdamW. We use a single PyTorch `DataLoader` per dataset. We support both single- and multi-GPU training via `DataParallel`.

**Fine-tuning levels.** MORPH supports four levels of fine-tuning (level-1 through level-4). Level-1 applies LoRA-based fine-tuning for the MORPH-FM-L model. We set the LoRA ranks on the linear layers in the axial-attention and MLP blocks via arguments. In level-2, we additionally unfreeze the encoder (convolutional, projection, and cross-attention layers). In level-3, we also unfreeze the decoder's projection layer. Level-4 fine-tunes all parameters. For the Ti, S, and M models, we use level-4. For level-1 LoRA layers, we use a learning rate of 1e−3 and weight decay 0.0. For level-4, we use a default learning rate of 1e−4 and weight decay 0.0.

## 5 Experiments

### 5.1 Baselines

**Models.** We compare MORPH against classical neural operators and recent PDE foundation models. As popular baselines, we use Fourier Neural Operator (FNO) (Li et al., 2020), a parameter-efficient spectral neural operator widely adopted for fluid surrogates, and UNet (Ronneberger et al., 2015; Rahman et al., 2022), a convolutional encoder–decoder with multiscale skip connections. Among foundation models, we include MPP (McCabe et al., 2024), which demonstrates gains from multi-physics pretraining; DPOT (Hao et al., 2024), a denoising operator transformer with Fourier attention; and Poseidon (Herde et al., 2024), a hierarchical multiscale operator transformer with shifted-window attention and lead-time-conditioned normalization for continuous-time querying.

Table 4: **Combined main results**: Single-step prediction NRMSE (N) and VRMSE (V) of the experiments (lower is better) across standalone surrogates, zero-shot foundation models, and full-shot finetuning. MORPH results are averaged over five runs. FNO and UNet values are reported from the corresponding benchmark sources or original papers. The *overall best* for each dataset across this combined table is **bold**. Standalone MORPH models are shown in blue and foundation MORPH models are shown in red. An asterisk (*) on the L model indicates LoRA fine-tuning.

| Model / evaluation | Pretraining sets | | | | | | Finetuning sets | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1D-CFD (N) | 2D-DR (N) | 2D-CFD-IC (N) | 2D-SW (N) | 3D-MHD (V) | 3D-CFD (N) | 1D-DR (N) | 1D-BE (N) | 2D-CFD (N) | 2D-GSDR (V) | 3D-CFD-Turb (N) | 3D-TGC (V) |
| **Standalone surrogates** | | | | | | | | | | | | |
| FNO Baseline | 0.095 | 0.12 | - | 0.0044 | 0.36 | 0.37 | 0.0014 | 0.029 | 0.28 | 0.1365 | 0.24 | 0.2429 |
| UNet Baseline | 0.36 | 0.84 | - | 0.083 | 0.1798 | 1.0 | 0.006 | 0.37 | 0.66 | 0.2252 | 0.23 | 0.6753 |
| MORPH-SS Ti | 0.06102 | 0.14637 | 0.08760 | 0.00445 | 0.3149 | 0.09819 | 0.00158 | 0.03961 | 0.05318 | 0.0086 | 0.08297 | 0.16366 |
| MORPH-SS S | 0.05719 | 0.12285 | 0.11385 | 0.0021 | 0.26845 | 0.10573 | 0.00134 | 0.07646 | 0.05712 | 0.01151 | 0.07941 | 0.07153 |
| **Foundation models: zero-shot** | | | | | | | | | | | | |
| MORPH-FM Ti | 0.05628 | 0.12588 | 0.09657 | 0.00475 | 0.29203 | 0.09163 | 0.02142 | 0.22246 | 0.09012 | 0.51230 | 0.31847 | 0.50845 |
| MORPH-FM S | 0.05375 | 0.11843 | 0.09284 | 0.00375 | 0.24276 | 0.08994 | 0.02104 | 0.21366 | 0.08446 | 0.50012 | 0.29081 | 0.42134 |
| MORPH-FM M | 0.05351 | 0.11551 | 0.08002 | 0.00217 | 0.22118 | 0.08447 | 0.01941 | 0.20133 | 0.08113 | 0.48239 | 0.28117 | 0.42196 |
| MORPH-FM L | **0.05056** | **0.10023** | **0.07678** | **0.00208** | **0.1787** | **0.07279** | 0.01553 | 0.19162 | 0.07019 | 0.46245 | 0.27345 | 0.45611 |
| **Foundation models: full-shot finetuning** | | | | | | | | | | | | |
| DPOT-FM S (30M) | - | - | - | - | - | - | - | - | 0.12357 | 0.00749 | - | - |
| DPOT-FM M (122M) | - | - | - | - | - | - | - | - | 0.103255 | 0.00710 | - | - |
| Poseidon-FM T (21M) | - | - | - | - | - | - | - | - | 0.25897 | 0.0210 | - | - |
| Poseidon-FM B (158M) | - | - | - | - | - | - | - | - | 0.15332 | 0.01116 | - | - |
| MORPH-FM Ti | - | - | - | - | - | - | 0.0012 | 0.0802 | 0.05294 | 0.00757 | 0.08192 | 0.05221 |
| MORPH-FM S | - | - | - | - | - | - | 0.00131 | 0.0584 | 0.05132 | 0.00711 | 0.0788 | 0.04074 |
| MORPH-FM M | - | - | - | - | - | - | **0.00091** | 0.04177 | 0.05123 | 0.00508 | 0.07513 | 0.03223 |
| MORPH-FM L* | - | - | - | - | - | - | 0.00111 | **0.04123** | **0.0419** | **0.00413** | **0.06921** | **0.03124** |

**Evaluation Metrics.** We follow the conventions of PDEBench (PB) and The Well (TW) when reporting errors on spatiotemporal PDE datasets. The *Normalized Root Mean Squared Error* (NRMSE) is the RMSE normalized by the $\ell_2$ norm of the ground truth, making it a *scale-independent* (dimensionless) measure and the default metric in PDEBench (Takamoto et al., 2022). In contrast, the *Variance-Scaled RMSE* (VRMSE) normalizes the RMSE by the target's per-snapshot standard deviation (i.e., the square root of its variance), thereby comparing errors to the intrinsic variability of the field. The Well adopts VRMSE as a primary metric (Ohana et al., 2024).

**Comparison Caveats.** A central challenge in fairly comparing PDE foundation models is their use of different pretraining corpora. In this work, we therefore fine-tune the FMs on the same downstream tasks and interpret the results independently of their pretraining corpus. Another difficulty in comparing MORPH to prior foundation models is the data modality. The existing models are trained on 2D benchmarks and are consequently constrained to 2D inputs, whereas MORPH is agnostic to data modality. To ensure fairness, we report head-to-head results only where the data modality aligns with a given baseline, i.e., table entries are marked "–" when a model cannot be evaluated due to such shape constraints. A further complication for fair comparison is the input context window employed by prior foundation models. MPP is trained with a 16-step context, while DPOT uses a 10-step context. In line with numerical solvers, we posit that PDE foundation models should initialize from an initial condition (or initial time step) and advance the state autoregressively into the future. Accordingly, we train with a single-step context, AR(1), for all models.

### 5.2 Main results

Our primary experiments are organized in Table 4. In this tables, we compare baseline and SOTA models using NRMSE and VRMSE as evaluation metrics.

**Standalone surrogates.** In Table 4, we compare performance of MORPH standalone surrogates (MORPH-SS) on pretraining and fine-tuning datasets against existing baseline (FNO and UNets) and SOTA models (MPP and DPOT). The results for baselines and SOTA models are inherited from the respective benchmark sources (Takamoto et al., 2022; Ohana et al., 2024) and articles (McCabe et al., 2024; Hao et al., 2024). Our MORPH-SS-Ti and S models outperform baseline and SOTA models on standalone datasets and remain competitive across the rest. Practically, this makes MORPH a viable standalone surrogate when the target application focuses on a single PDE family.

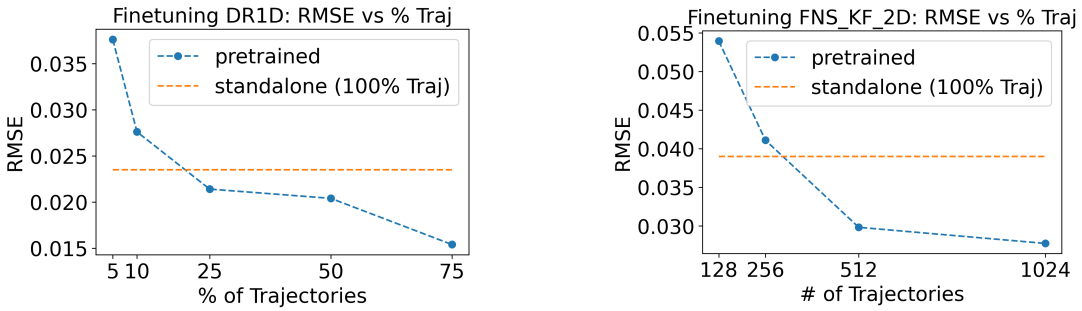



Figure 3: Scaling with respect to fine-tuning dataset size: (Left:) **MORPH-FM-Ti on 1D-DR**: RMSE vs percentage of different trajectories finetuned for *100 epochs*, (Right:) **MORPH-FM-S on 2D-FNS-KF**: RMSE vs number of trajectories of finetuned for *100 epochs*.

**Zero-shot performance.** In Table 4, we report the zero-shot performance of different MORPH foundation models. For this evaluation, we run the pretrained FMs in inference mode on the test sets of their corresponding datasets. We observe that MORPH-FM-L achieves the best performance on 11 out of 12 datasets. These results indicate favorable scaling behavior of MORPH under zero-shot adaptation.

**Full-shot fine-tuning performance.** Full-shot fine-tuning results are presented in Table 4. On the 2D datasets, we compare the fine-tuning performance of MORPH-FM to DPOT-FM and Poseidon-FM, and find that our models consistently surpass these existing FMs. We can quantify the adaptation-efficiency gains of MORPH-FM over MORPH-SS on the fine-tuning benchmarks: MORPH-FM outperforms MORPH-SS on 5 out of 6 fine-tuning datasets.

**Parameter-efficient fine-tuning with LoRA.** We investigate fine-tuning the MORPH-FM-L model using low-rank adapters (LoRA). In our setup, only 77M of the 480M parameters, comprising LoRA adapters on attention and MLP blocks, positional encodings and layer norms are fine-tuned, which is fewer trainable parameters than in the M model (126M). Notably, MORPH-FM-L matches or outperforms the M model across full-shot fine-tuning (Table 4) in 5 of the 6 fine-tuning datasets. To our knowledge, this is among the first studies to apply LoRA to PDE foundation models (see also: Holzschuh et al. (2025)), and our results indicate that LoRA-based fine-tuning is a promising direction given the expanding PDE datasets and increasing model sizes.

### 5.3 Data- and compute-scarce fine-tuning

Fig. 3 presents a dataset–size scaling study on two downstream tasks (1D-DR and 2D-FNS-KF). We fix the training protocol (optimizer, schedule, epochs, batch size) and vary the *number of fine-tuning trajectories*. We report RMSE as a function of the number of trajectories used to fine-tune MORPH-FM, and compare against the standalone MORPH-SS trained on 100% of trajectories. Errors consistently decrease as dataset size increases. On 1D-DR, the pretrained MORPH-FM-Ti already outperforms the standalone model with only 25% of the trajectories. On 2D-FNS-KF, with $\sim 256$ trajectories (less than 1% of the full set), MORPH-FM-S surpasses the standalone MORPH-SS-S trained on all trajectories.

To analyze scaling behavior across foundation models of varying capacities, we perform a comparative scaling study on the FNS-KF dataset. We fix the fine-tuning data size (*128 trajectories, $\leq$1% of full data*) and compute budget (*200 epochs*), and compare the performance of MORPH-FM-S against larger PDE foundation models (DPOT-FM-M and Poseidon-FM-B). As summarized in Table 5, MORPH-FM-S contains only $\sim$0.25$\times$ the parameters of DPOT-FM-M and $\sim$0.18$\times$ that of Poseidon-FM-B, yet achieves superior accuracy under identical data- and compute-scarce settings. This experiment highlights scaling efficiency, showing that MORPH attains competitive or better performance at significantly smaller model scales. Additional details on the hyperparameter settings and fine-tuning setup are provided in Appendix B.

We perform full fine-tuning of MORPH-FM-S using a learning rate of 1e-4, batch size 40, and weight decay 1e-4. We employ a ReduceLROnPlateau scheduler with a wait of 5 epochs and early stopping with a patience of 10 epochs. The hyperparameter settings and fine-tuning setup for Poseidon and DPOT are provided in Appendix B. Table 5 shows the comparison.

Table 5: Comparison of PDE foundation models on the FNS-KF task in a data- and compute-scarce setting with 128 fine-tuning trajectories and 200 training epochs. We report average MSE and RMSE over the full 20-step prediction horizon initialized from the initial frame, (t=0), on the test set. Lower values are better. MORPH-FM-S is evaluated autoregressively, Poseidon-FM-B is evaluated using direct lead-time-conditioned predictions, and DPOT-FM-M uses its 10-frame input context populated from the available rollout history.

| **Models** / **Metrics** | DPOT-FM-M 122M | Poseidon-FM-B 158M | **MORPH-FM-S 30M** |
|---|---|---|---|
| MSE | 0.0301 | 0.0017 | **0.00162** |
| RMSE | 0.176 | 0.0412 | **0.0401** |

## 6 Scaling studies

We study scaling behavior in both pretraining and fine-tuning, with respect to both pretraining dataset size and model capacity. For pretraining, Fig. 4 reports training and validation losses for MORPH-FM-S (30M parameters) as we vary the number of pretraining trajectories. The results are shown over the first 10 epochs, during which both losses decrease steadily as the pretraining dataset size increases. In terms of model capacity, MORPH-FM-L achieves the lowest pretraining validation loss, saturating at $\sim 0.02$, compared with $\sim 0.05$, $\sim 0.11$, and $\sim 0.14$ for the M, S, and Ti variants, respectively.

The scaling experiments are conducted on four H100 GPUs using Distributed Data Parallel (DDP). We use the per-GPU batch sizes (global batch size divided by 4) described in Sec. 4. To vary the effective dataset size, we subsample the training data by randomly retaining a fraction $p\%$ of the mini-batches produced by the training `DataLoader` and processing only these mini-batches in the model's forward pass.

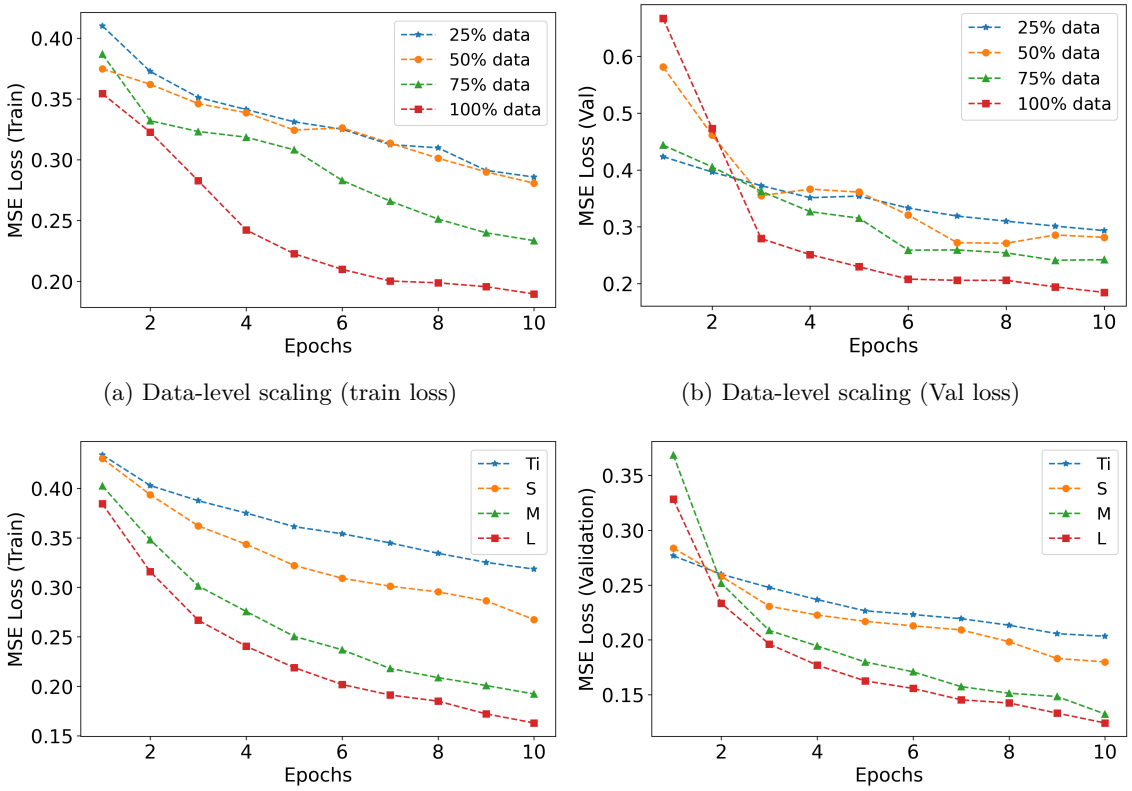
## 7 Ablation studies

### 7.1 Model architecture.

Table 6 summarizes ablations over the convolutional operator, field-fusion module, and attention operator in MORPH. Introducing a lightweight convolutional operator with 8 filters consistently reduces error on 2D/3D benchmarks with only a modest increase in GFLOPs, while having negligible effect on 1D. For field fusion, cross-attention outperforms a simple concat+MLP scheme across datasets, at nearly identical parameter counts and modest additional compute. Among attention variants, axial attention systematically offers the

Table 6: Ablation studies (Model architecture): Effect of convolution operator, cross-attention field fusion and axial attention on the total number of parameters (in M), floating point operations of the operator (in GFLOPs), and single-step prediction test loss (MSE) for different datasets. ps = patch size, Np - number of patches or tokens.

| Ablations | Variant | MHD3D | | | DR2D | | | CFD1D | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Param. | GFLOPs | Loss | Param. | GFLOPs | Loss | Param. | GFLOPs | Loss |
| Conv Filters | 0 (no conv) | 8.29 | 0 | 0.18376 | 7.04 | 0 | 0.00787 | 7.14 | 0 | 0.17185 |
| | **8** | 8.94 | 2.7556 | **0.16851** | 7.96 | 0.11376 | **0.00747** | 8.06 | 0.011 | 0.17348 |
| Field-fusion | Concat & Dense | 8.8 | 0.2013 | 0.17101 | 7.89 | 0.10066 | 0.00827 | 7.99 | 0.05 | 0.18141 |
| | **Cross-attention** | 8.94 | 0.4713 | **0.16851** | 7.96 | 0.16829 | **0.00747** | 8.06 | 0.1178 | **0.17348** |
| Attention | | CFD2D-IC (ps = 8, Np = 4096) | | | CFD2D-IC (ps = 16, Np = 1024) | | | SW2D (ps = 8, Np = 256) | | |
| | Full | 5.78 | 5.9056 | 0.01228 | 14.2 | 75.141 | 0.01208 | 4.67 | 0.6711 | 0.00291 |
| | Sparse | 5.78 | 1.8832 | 0.01324 | 14.2 | 8.6064 | 0.00856 | 4.67 | 0.4372 | 0.00208 |
| | **Axial** | 8.94 | 1.8832 | **0.00812** | 17.3 | 8.6064 | **0.00724** | 7.83 | 0.4372 | **0.00172** |




(a) Data-level scaling (train loss)

(b) Data-level scaling (Val loss)

(c) Model-level scaling (train loss)

(d) Model-level scaling (val loss)

Figure 4: Scaling studies: (a-b) Data level scaling showing train and validation loss as a function of epochs for MORPH-S, (c-d) Model level scaling showing train and validation loss as a function of epochs for full data

best accuracy–compute trade-off, outperforming both full and sparse self-attention. The gains are most pronounced at larger patch-token counts, where axial factorization yields large FLOP savings while still improving test loss.

Table 6 evaluates three key design choices in MORPH i.e., the component-wise convolutional preprocessor, the field-fusion module, and the attention operator. We report parameter count (M), operator FLOPs (GFLOPs), and test MSE on representative datasets (MHD3D, DR2D, CFD1D, CFD2D-IC, and SW2D; here $ps$ denotes patch size and $N_p$ the number of patches/tokens). The ablation studies are performed on 4x H100s GPUs with the same hyperparameters.

*Convolutional preprocessor.* Comparing "0 (no conv)" to the 8-filter variant shows that adding a lightweight convolutional operator consistently reduces error on 3D and 2D problems with modest compute overhead: on MHD3D the loss drops from 0.18376 to 0.16851 (about 8.3%) at the cost of 2.76 GFLOPs, and on DR2D from 0.00787 to 0.00747 (about 5.1%) with only 0.11 additional GFLOPs. For CFD1D the effect is small and slightly negative ($0.17185 \rightarrow 0.17348$), indicating that the convolutional inductive bias is most beneficial for higher-dimensional settings.

*Field-fusion.* For field fusion, cross-attention is compared against a simple concat+dense baseline. Cross-attention consistently improves accuracy with negligible parameter overhead and a moderate increase in FLOPs: on MHD3D the loss decreases from 0.17101 to 0.16851 ($\approx 1.5\%$), on DR2D from 0.00827 to 0.00747 ($\approx 9.7\%$), and on CFD1D from 0.18141 to 0.17348 ($\approx 4.4\%$), while GFLOPs roughly double but remain small in absolute terms. This supports the use of content-aware attention for fusing multiple fields instead of static mixing.

*Attention operator.* The bottom block compares axial, full, and sparse self-attention for different token counts. Axial attention offers the best accuracy–compute trade-off and its advantage grows with sequence length. On CFD2D-IC with $ps$=16 and $N_p$=1024, axial attention achieves a loss of 0.00724 using 8.61 GFLOPs, whereas full attention reaches 0.01208 at 75.14 GFLOPs (about 8.7× more compute and ≈ 40% higher error). For $ps$=8, $N_p$=4096, axial is again superior (0.00812 vs. 0.01228) while being ≈ 3.1× cheaper in FLOPs, and a similar pattern holds on SW2D where axial is ≈ 1.5× cheaper and ≈ 41% lower in loss than full attention. Compared to sparse attention, axial has essentially the same FLOPs but reduces loss by 15-40% across datasets. These trends are consistent with the theoretical complexity reduction from $O((tDHW)^2)$ for full attention to $O(tDHW\,(t{+}D{+}H{+}W))$ for axial factorization, and motivate axial attention as a near–Pareto-optimal choice for large spatiotemporal token grids.

### 7.2 Pretraining corpus.

To evaluate the effect of the pretraining set and to disentangle the influence of the dataset from that of the model architecture, we pretrain MORPH-FM-S*(27M) on the same datasets as Poseidon. We then fine-tune both MORPH and Poseidon for 200 epochs using 128 training trajectories and evaluate them on 240 held-out test trajectories across three downstream tasks: CE-RM, NS-PwC, and NS-SL. The results reported in Table 7 demonstrate that MORPH-FM-S*(27M) outperforms on CE-RM and is slightly better on NS-SL, but underperforms Poseidon on NS-PwC. Figs. 6, 7 and 8 show full autoregressive rollouts on representative test trajectories for NS-SL, NS-PwC, and CE-RM. These findings indicate that the MORPH architecture yields performance gains that are largely independent of the choice of pretraining corpus.

We employ a ReduceLROnPlateau scheduler with a wait of 5 epochs and early stopping with a patience of 10 epochs. We use a learning rate of 1e-4, batch size 40, and weight decay 1e-4, early stopping with patience of 25 during the fine-tuning. The hyperparameter settings for Poseidon are presented in Appendix B. Table 7 shows comparison. Full autoregressive rollouts are shown in Figs. 6, 7 and 8.

Table 7: Pretraining-corpus ablation comparing MORPH-FM-S* and Poseidon-FM-B after fine-tuning with 128 training trajectories for 200 epochs. We report average MSE over the full 20-step prediction horizon initialized from the initial frame, (t=0), and averaged over 240 held-out test trajectories. Lower values are better. MORPH-FM-S* is evaluated autoregressively, while Poseidon-FM-B is evaluated using direct lead-time-conditioned predictions.

| **FT datasets** / **Models** | CE-RM | NS-PwC | NS-SL |
|---|---|---|---|
| Poseidon-FM-B (158M) | 0.4181 | **0.0004** | 0.0163 |
| **MORPH-FM-S*** (27M) | **0.09044** | 0.00408 | **0.0159** |

## 8 Forward task: Autoregressive rollouts

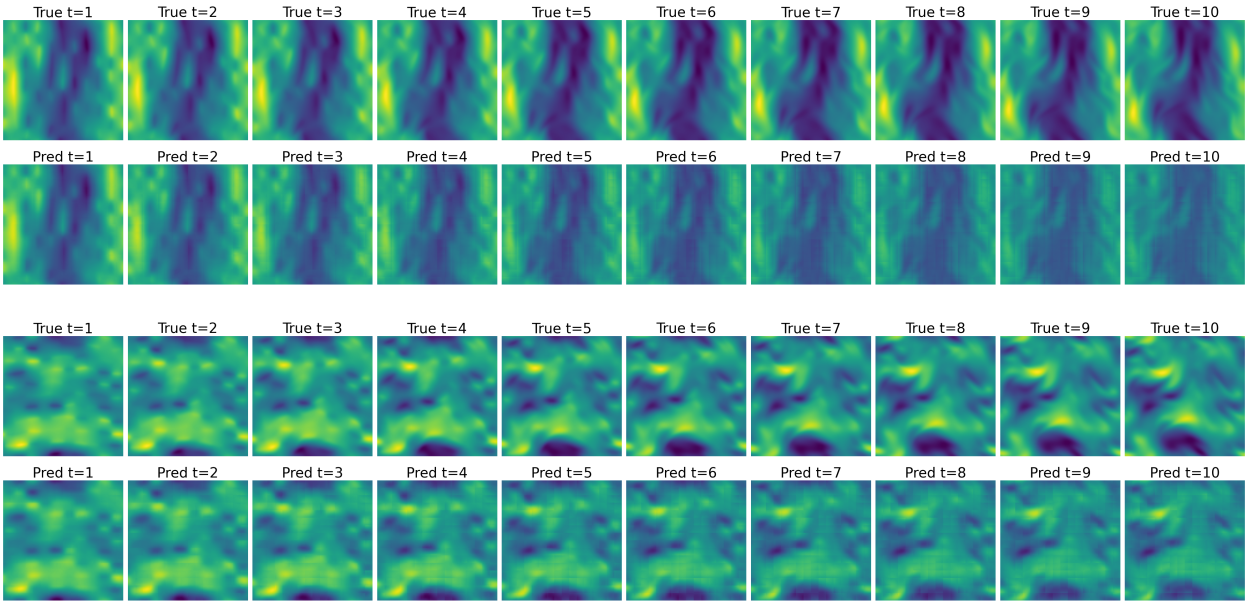
We have presented autoregressive rollouts results for the finetuned foundation models.

**FNS-KF.** Figs. 5a and 5b present 10-step autoregressive (AR) rollouts for MORPH-FM-Ti and MORPH-FM-S fine-tuned on the forced incompressible Navier–Stokes with Kolmogorov forcing (FNS-KF) dataset, initialized from the t=0 snapshot. Over the 10-step horizon, MORPH-FM-S achieves consistently lower error than MORPH-FM-Ti. Both models remain stable under rollouts with limited error accumulation and no concerning instability across the rollout.

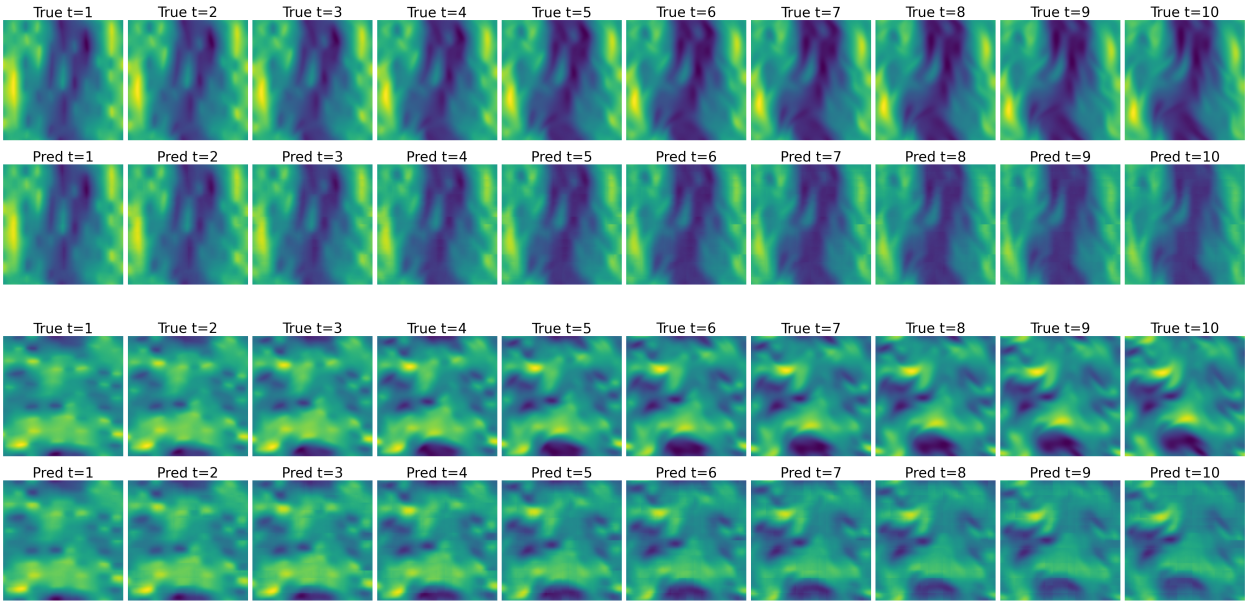
**NS-SL.** Fig. 6 presents full autoregressive (AR) rollouts for MORPH-FM-S*(27M), pretrained on PDE-Gym (NS-Sines, NS-Gaussians, CE-RP, CE-CRP, CE-KH, CE-Gauss) and subsequently fine-tuned on *128 trajectories of NS-SL* and evaluated on 240 trajectories over 200 epochs. We demonstrate stable rollouts with limited error accumulation even with a compact 27M-parameter MORPH model. These results suggest that MORPH can serve as an efficient foundation model, accurately capturing the underlying dynamics while remaining computationally lightweight.

**NS-PwC.** Fig. 7 presents full autoregressive (AR) rollouts for MORPH-FM-S*(27M), pretrained on PDEGym (NS-Sines, NS-Gaussians, CE-RP, CE-CRP, CE-KH, CE-Gauss) and subsequently fine-tuned on *128 trajectories of NS-PwC* and evaluated on 240 trajectories over 200 epochs. We obtain stable rollouts with limited error accumulation even using a compact 27M-parameter MORPH model. This indicates that MORPH can function as an efficient foundation model, capturing the essential dynamics while remaining computationally efficient.

**CE-RM.** Fig. 8 presents full autoregressive (AR) rollouts for MORPH-FM-S*(27M), pretrained on PDEGym (NS-Sines, NS-Gaussians, CE-RP, CE-CRP, CE-KH, CE-Gauss) and subsequently fine-tuned on *128 trajectories of CE-RM* and evaluated on 240 trajectories over 200 epochs. We observe stable rollouts, but note missing higher spectral components and finer details toward the end of the rollout. This behavior is expected given the relatively small model size (27M parameters) and the limited number of fine-tuning trajectories (128).

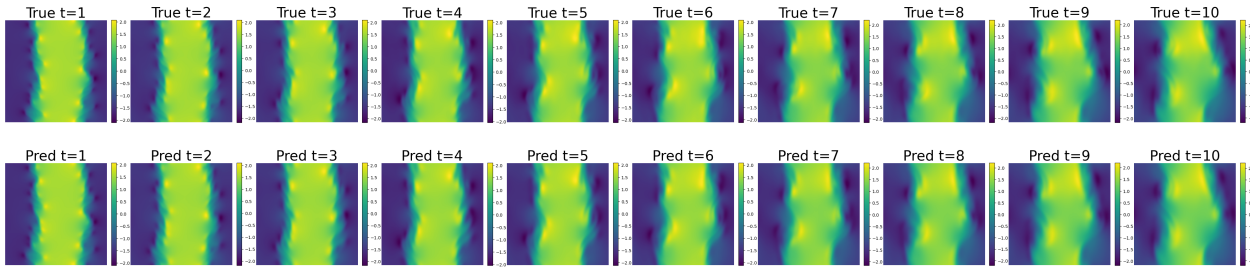



(a) **Fine-tuning MORPH-FM-Ti ($\sim$ 7M) for the FNS-KF prediction task**: 10-step autoregressive rollouts for $v_x$ and $v_y$ with $t = 0$ (initial frame) as input.

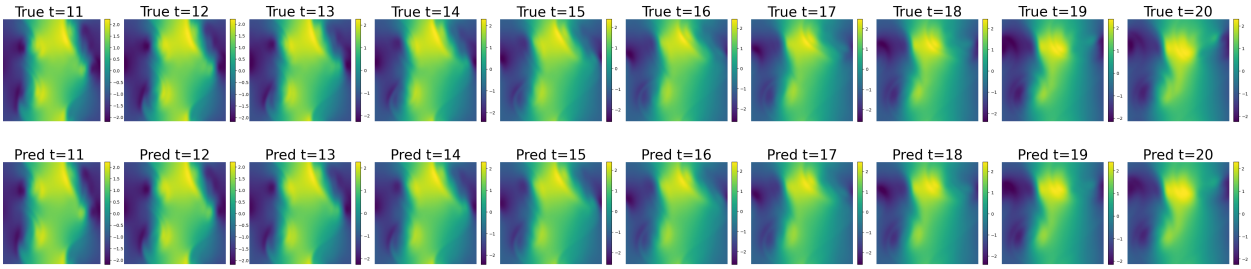



(b) **Fine-tuning MORPH-FM-S ($\sim$ 30M) for the FNS-KF prediction task**: 10-step autoregressive rollouts for $v_x$ and $v_y$ with $t = 0$ (initial frame) as input.

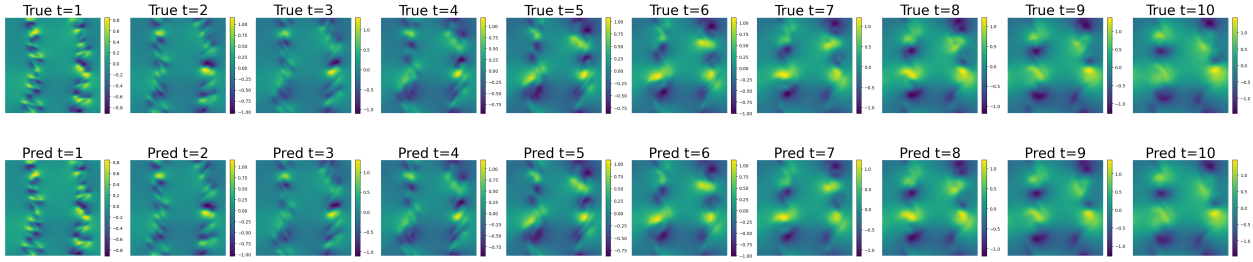
Figure 5: **Fine-tuning MORPH-FM models for the FNS-KF prediction task**: comparison of 10-step autoregressive rollouts for $v_x$ and $v_y$ using $t = 0$ as the initial input frame.



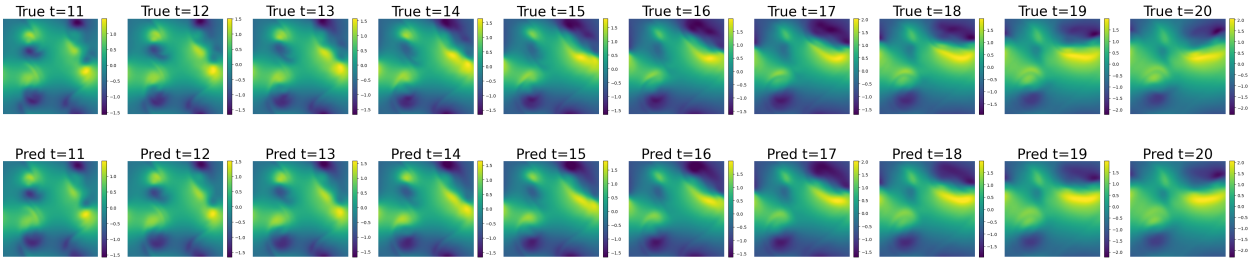

(a) First 10 time-steps of horizontal-velocity field ($v_x$).




(b) Last 10 time-steps of horizontal-velocity field ($v_x$).

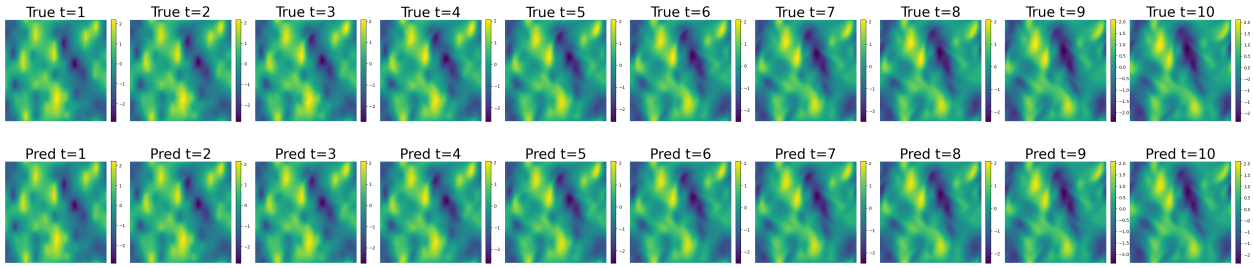



(c) First 10 time-steps vertical-velocity field ($v_y$).

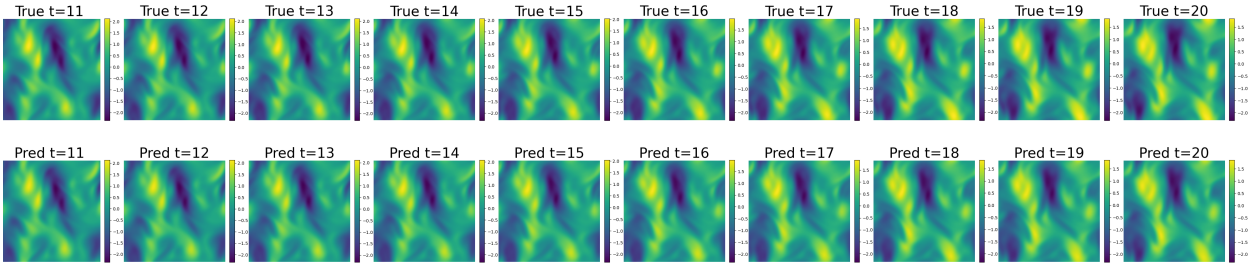



(d) Last 10 time-steps vertical-velocity field ($v_y$).

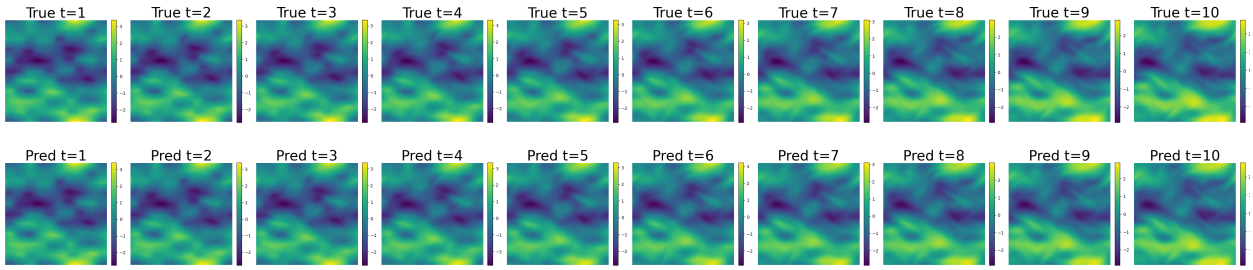
Figure 6: Qualitative 20-step autoregressive rollout for MORPH-FM-S* on the NS-SL task. The model is fine-tuned for 200 epochs using 128 training trajectories and visualized on a representative held-out test trajectory from the 240-trajectory test set. Panels show the first and last 10 predicted time steps for the horizontal- and vertical-velocity fields, alongside the corresponding ground truth. Aggregate quantitative errors over the full test set are reported in Table 7.



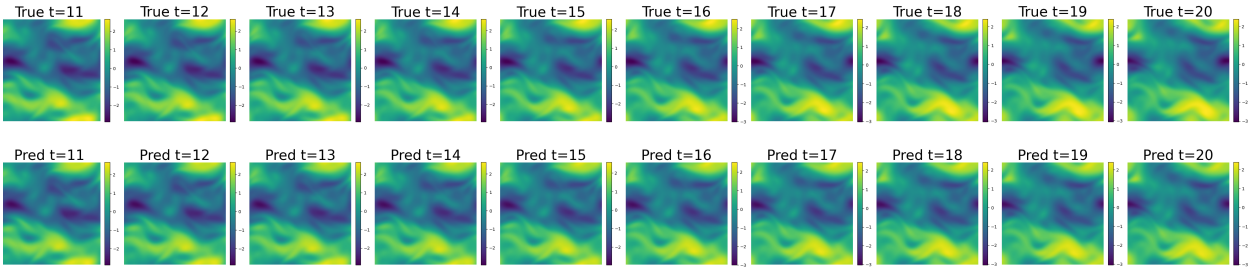

(a) First 10 time-steps of horizontal-velocity field ($v_x$).




(b) Last 10 time-steps of horizontal-velocity field ($v_x$).

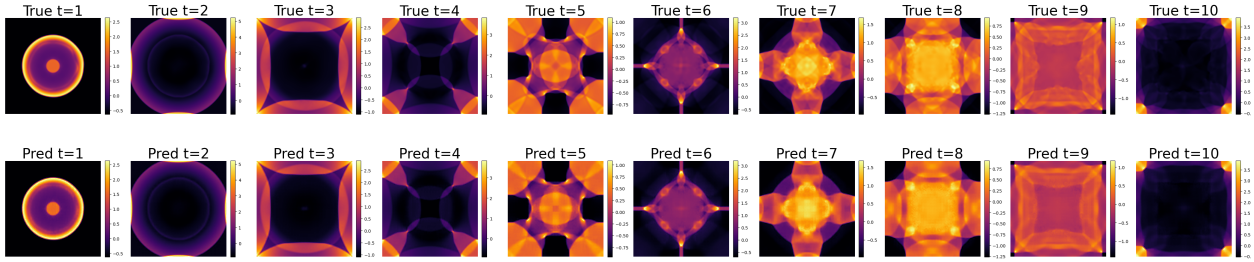



(c) First 10 time-steps vertical-velocity field ($v_y$).

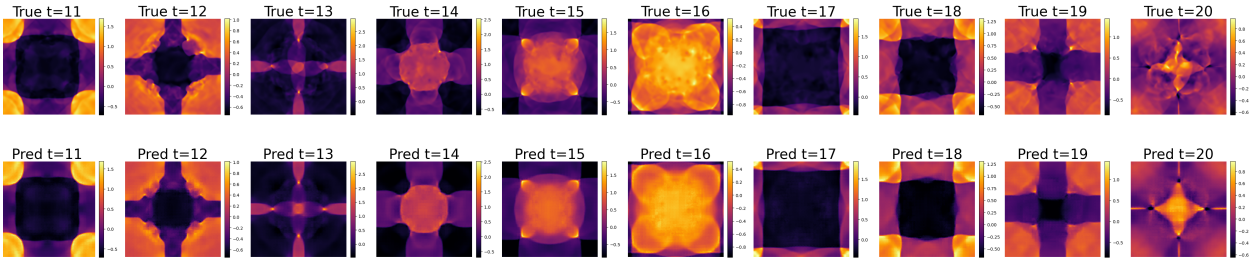



(d) Last 10 time-steps vertical-velocity field ($v_y$).

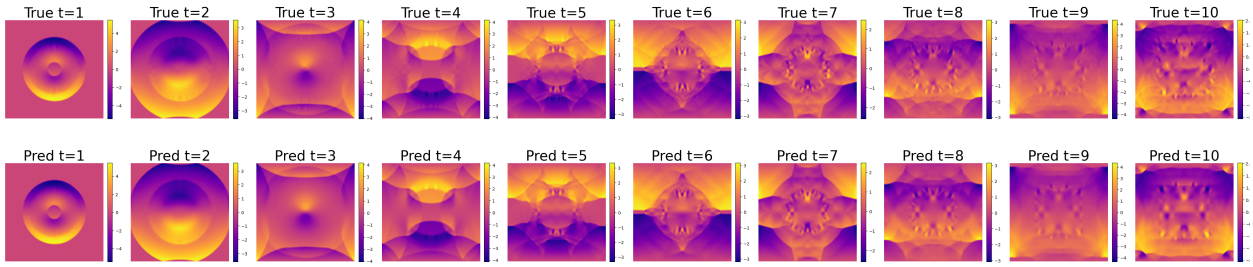
Figure 7: Qualitative 20-step autoregressive rollout for MORPH-FM-S* on the NS-PwC task. The model is fine-tuned for 200 epochs using 128 training trajectories and visualized on a representative held-out test trajectory from the 240-trajectory test set. Panels show the first and last 10 predicted time steps for the horizontal- and vertical-velocity fields, alongside the corresponding ground truth. Aggregate quantitative errors over the full test set are reported in Table 7.



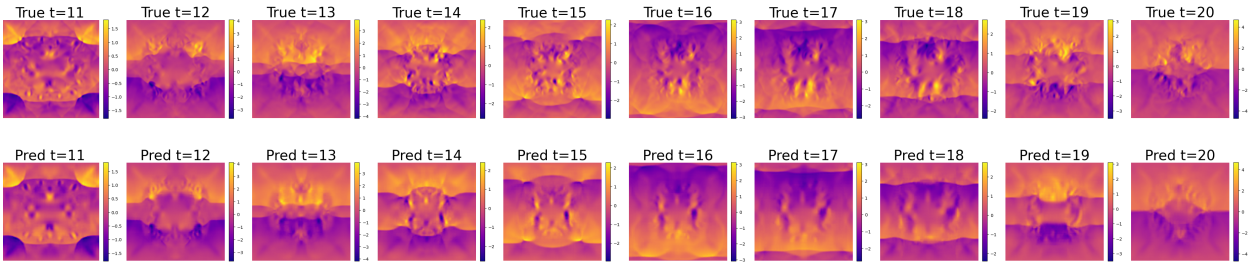

(a) First 10 time-steps of pressure field ($P$).




(b) Last 10 time-steps of pressure field ($P$).




(c) First 10 time-steps of horizontal-velocity field ($v_x$).




(d) Last 10 time-steps of horizontal-velocity field ($v_x$) .

Figure 8: Qualitative 20-step autoregressive rollout for MORPH-FM-S* on the CE-RM task. The model is fine-tuned for 200 epochs using 128 training trajectories and visualized on a representative held-out test trajectory from the test set. Panels show the first and last 10 predicted time steps for the pressure and horizontal-velocity fields, alongside the corresponding ground truth. Aggregate quantitative errors over the full test set are reported in Table 7.

# 9 Applications of PDE-FM to physical systems

## 9.1 Forward task: Key-frame prediction for material dynamics under extreme loading.

After evaluating autoregressive rollouts on benchmark fluid-based PDE systems, we next examine whether the pretrained MORPH backbone transfers to application-driven forward prediction tasks. We focus on material dynamics under extreme loading, which arise in mission- and safety-critical settings such as spacecraft structures, thermal protection systems, hypersonic vehicles, and explosive- or impact-driven experiments. These systems are characterized by shock waves, steep gradients, evolving multi-material interfaces, fracture fronts, and large morphology changes. In contrast to the rollout studies in Sec. 8, the goal here is terminal key-frame prediction: given an initial state, the model directly predicts the terminal state without access to intermediate frames.

**Datasets.** We use two open-source representative datasets for material-dynamics under extreme loading. PLI dataset contains high-strain rate multi-material-multi-physics simulations designed to capture high-explosive driven shock propagation through complex targets (Banesh et al., 2025). The underlying dynamics are strongly interface controlled, since shock and interface interactions can trigger Richtmyer-Meshkov-type instabilities and jetting. FRAC dataset focuses on fracture initiation, propagation, and interaction across multiple materials and loading conditions (Hill et al., 2025). The dataset covers diverse multiple materials with different boundary conditions, generated via phase-field modeling. The non-linear physics is governed by moving fronts (crack tips) and evolving discontinuities. More dataset-related details are deferred to Appendix A.4.1 and A.4.2.

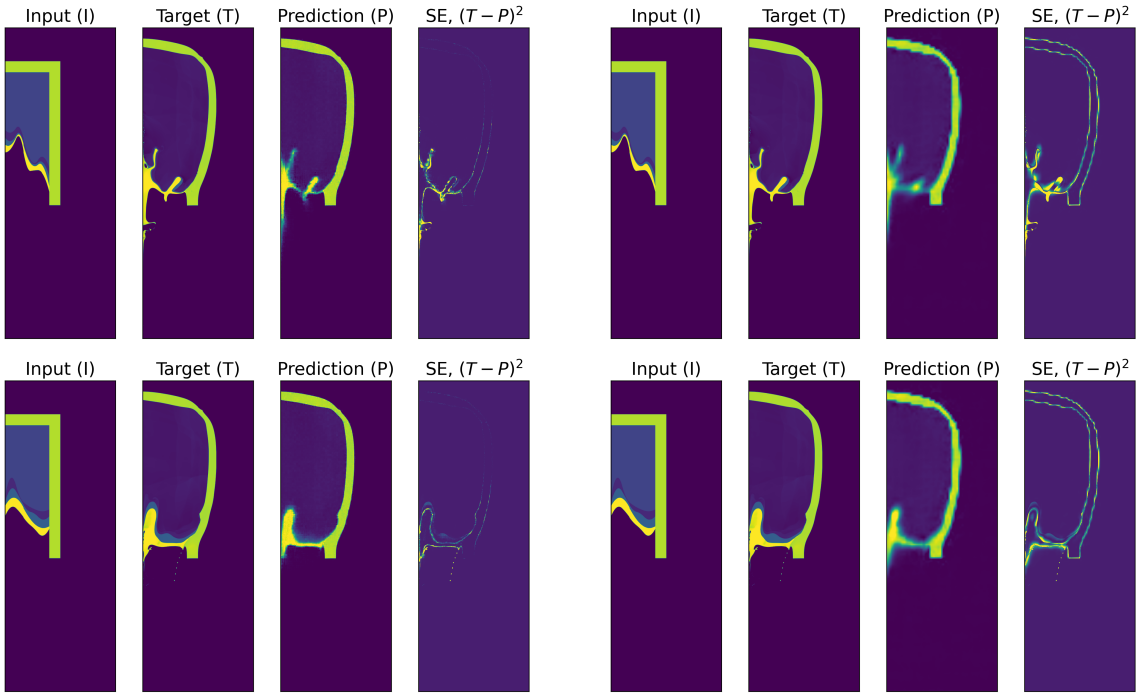
**Modeling.** For the PLI dataset, We finetune *MORPH-Ti (≈7M parameters)* and compare against *Poseidon-T (≈20M parameters)*. We fine-tune MORPH on the native spatial grids of PLI and FRAC with patch sizes of $8 \times 8$ and $4 \times 4$, respectively (7,000 and 1,024 patches). The details on Poseidon-T is deferred to Appendix B. For both datasets, we formulate the task as a long-horizon initial-to-terminal map, $F_\theta : X(t_0) \mapsto X(t_T)$, where $X(t_0)$ is the initial state and $X(t_T)$ is the terminal target state.

**Experimental details.** We train all models using the AdamW optimizer with weight decay $10^{-2}$ and an MSE objective. The learning rates are set to $10^{-4}$ for PLI and $10^{-3}$ for FRAC. In both cases, we use a learning-rate schedule with a 5-epoch linear warmup followed by cosine decay for the remaining epochs. All experiments are performed with data-parallel training on a single node with 8× NVIDIA H100 GPUs. We use effective batch sizes of 8 and 512 for PLI and FRAC, respectively. We evaluate two settings: (i) terminal-state prediction using the full training split for up to 200 epochs, and (ii) data-level scaling, where models are trained for up to 100 epochs on subsampled training sets to assess sample efficiency. As a preprocessing step, we apply RevIN normalization to all inputs and targets and report held-out test-set errors in the normalized scale (Kim et al., 2021).

**Results.** Figs. 9 and 10 show the initial input, target terminal state, model prediction, and squared error map $(T - P)^2$ for representative PLI and FRAC test examples. On PLI, MORPH-Ti accurately tracks the terminal interface geometry and captures the central jetting structures. On FRAC, MORPH-Ti recovers the global terminal damage field, with residual errors concentrated near crack tips, thin branches, and junctions where crack branching or coalescence occurs. Table 8 reports the held-out test MSE for both datasets. MORPH-Ti substantially outperforms Poseidon-T on PLI, reducing the MSE from 0.13892 to 0.05445. On FRAC, Poseidon-T achieves a modest advantage over MORPH-Ti, with MSE values of 0.03704 and 0.04103, respectively.

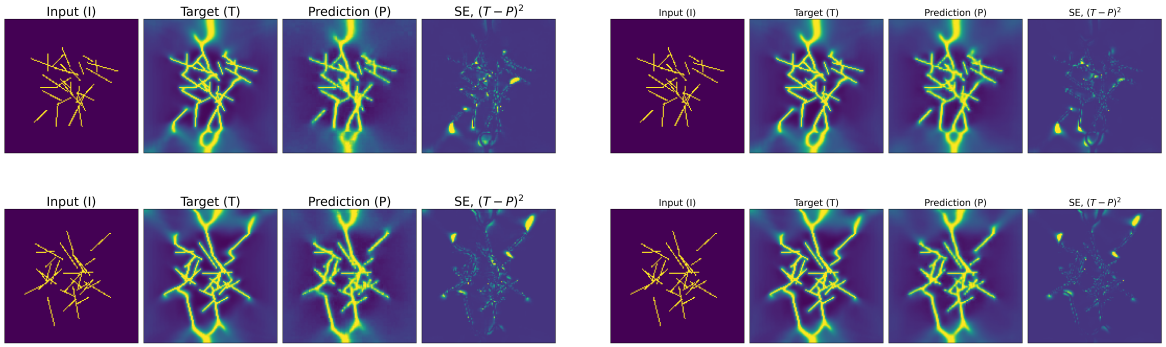
| Dataset/Models | Morph-Ti | Poseidon-T |
|---|---|---|
| PLI | 0.05445 | 0.13892 |
| FRAC | 0.04103 | 0.03704 |

Table 8: MSE on the test set




(a) MORPH-Ti (b) Poseidon-T

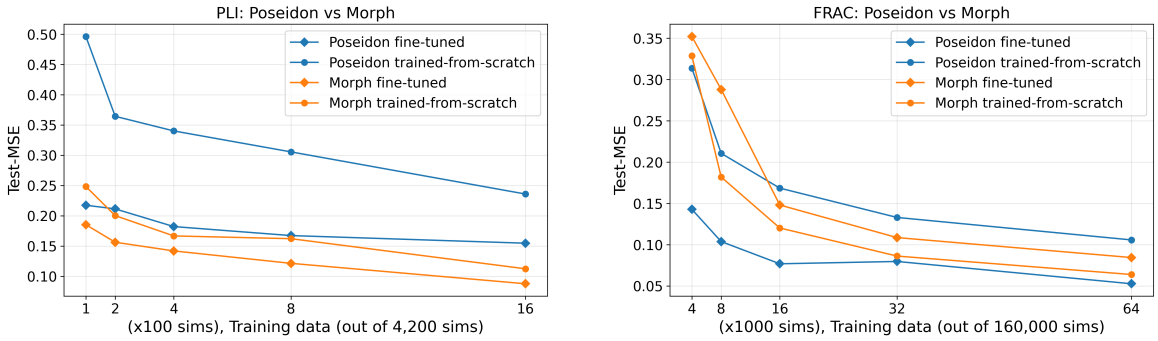
Figure 9: Terminal-state prediction on PLI comparing MORPH-Ti (a) and POSEIDON-T (b). Columns show the input, target terminal state, prediction, and squared error $(T-P)^2$.




(a) MORPH-Ti (b) Poseidon-T

Figure 10: Terminal-state prediction on FRAC comparing MORPH-Ti (a) and POSEIDON-T (b). For each test example, columns show the input (initial state), target terminal state, model prediction, and squared error map $(T-P)^2$.

**Transfer gain.** To quantify the benefit of pretrained initialization, we compare fine-tuning from pretrained weights against training from random initialization. Fig. 11 reports the held-out test MSE as a function of training-set size for PLI and FRAC. The pretrained models provide the largest gains in the low-data regime, where test errors are reduced by up to approximately $2\times$ relative to training from scratch. As more fine-tuning data become available, the performance gap naturally decreases, indicating that task-specific supervision progressively closes the gap to pretrained initialization. Overall, these results show that pretraining substantially improves sample efficiency for terminal-state prediction, particularly when long-horizon supervision is limited.




(a) PLI (b) FRAC

Figure 11: Sample efficiency study: Test MSE vs fine-tuning set size for PLI (a) and FRAC (b). We compare fine-tuning from pretrained weights versus training-from-scratch (random initialization) for MORPH and POSEIDON.

### 9.2 Inverse task: System parameter estimation in inertial confinement fusion

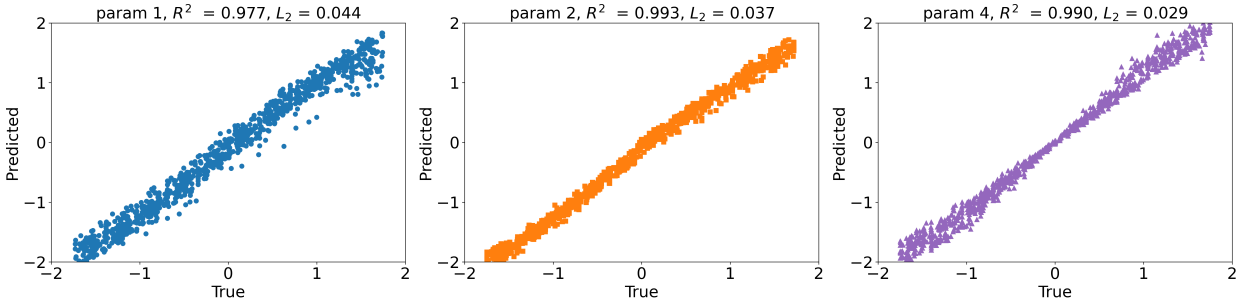
System parameter estimation is an inverse problem of considerable importance in inertial confinement fusion (ICF). During inversion, the quantities that govern implosion performance are often not directly observable and must instead be inferred from indirect diagnostic signatures (Spears et al., 2025). In the setting considered here, the available observations consist of multi-modal measurements, including hyperspectral X-ray images and scalar diagnostics such as yield, ion temperature, and pressure, while the goal is to recover the underlying design or simulator input parameters that produced them (Gutierrez et al., 2025). This capability is central to post-shot analysis, model calibration, and design optimization, since it enables researchers to interpret experimental outcomes, diagnose discrepancies between simulations and observations, and guide future implosion designs using physically meaningful latent variables rather than only observable outputs.

**Dataset.** LLNL `macc` repository provides open source dataset generated with a 1D semi-analytic ICF simulator that models capsule implosion dynamics and produces a multi-modal suite of synthetic diagnostics (Anirudh et al., 2020). The dataset contains 10,000 simulations where each simulation has a parameter vector by $x \in \mathbb{R}^3$, the hyperspectral images by $y_{\text{img}} \in \mathbb{R}^{64\times64\times4}$, and the scalar diagnostics by $y_{\text{sc}} \in \mathbb{R}^{15}$. The 10K-sample dataset is randomly split into train/val/test with an 80%/10%/10% ratio. The inversion task is $g : y = (y_{\text{img}}, y_{\text{sc}}) \mapsto x$. More dataset related details are deferred to Appendix A.4.3.

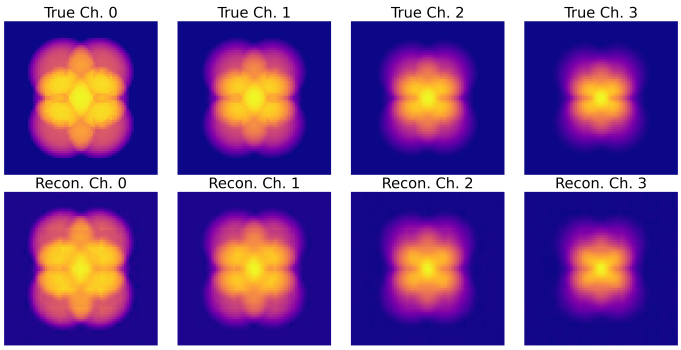
**Modeling.** We attach a lightweight task-specific head on the representation space of pretrained MORPH-Ti and estimate system parameters given hyperspectral images and scalar diagnostics. The MORPH backbone is finetuned with hyperspectral-image reconstruction objective (autoencoder-style) and the latent representations (at the end of transformer blocks) act as inputs (along with 15-dimensional scalar diagnostics) to the TSH for parameter regression. The TSH has two `Conv1D`→`GELU`→`Dropout` blocks, followed by two `Linear`→`GELU`→`Dropout` blocks.

**Experimental settings.** We fine-tune all parameters of MORPH-Ti and train the TSH (∼1M parameters) for 100 epochs on a single RTX-A6000 GPU. We use AdamW optimizers with learning rates of $10^{-4}$ for MORPH-Ti and $10^{-5}$ for TSH. For learning-rate scheduler, we use a 5-epoch linear warmup (minimum LR $10^{-8}$) followed by cosine decay. We use a batch size of 8 and optimize mean-squared error (MSE) losses for both the reconstruction and parameter-regression objectives.

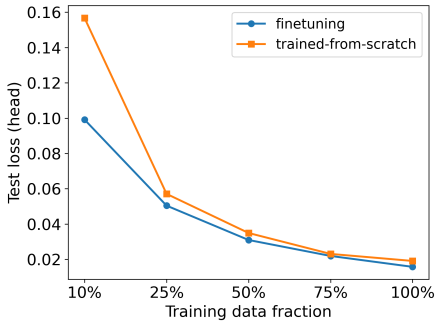
**Results.** Fig. 12 summarizes the performance of MORPH-Ti on the ICF inverse-parameter-estimation task. Fig. 12(a) compares the predicted and true values of the three normalized system parameters on the held-out test set. The predictions closely follow the diagonal, with high $R^2$ scores and low $L_2$ errors, indicating accurate recovery of the underlying simulator parameters from multimodal diagnostic observations. Fig. 12(b)




(a) Parameter estimation (normalized scale) on the full test set




(b) True vs reconstruction hyperspectral images




(c) Scaling study

Figure 12: Results: (a) Parameter estimation results on the full test set, (b) True vs reconstruction hyperspectral images on held-out test example, (c) Test-set loss of the TSH (finetuned vs trained-from-scratch) versus fine-tuning-data fraction.

shows representative hyperspectral X-ray reconstructions, where the reconstructed channels preserve the dominant spatial structure and intensity patterns of the corresponding targets. Finally, Fig. 12(c) compares fine-tuning from pretrained MORPH-Ti weights against training from random initialization across different fractions of the training set. The pretrained model provides a clear transfer advantage in the low-data regime, but this advantage gradually diminishes as the amount of fine-tuning data increases. This indicates that MORPH learns transferable representations for coupled multiphysics systems. In particular, pretraining on MHD dynamics may expose the model to coupled field interactions that are relevant to this inverse ICF problem.

### 9.3 Inverse task: Material property estimation of composite materials

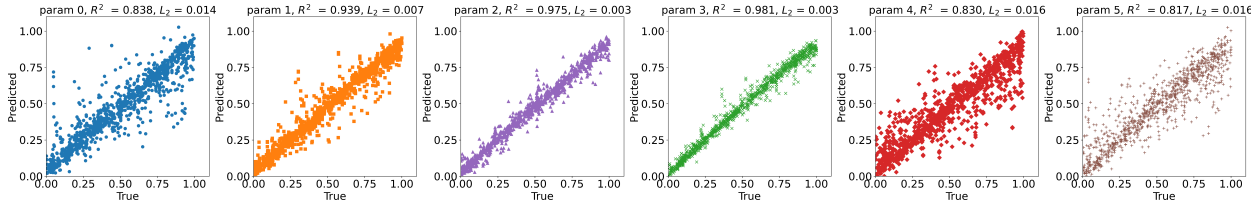
Inverse estimation of material properties in composites is a relevant and challenging problem because the mechanical response of these materials is governed by elastic parameters, which helps in non-invasive monitoring of structures (Mitra & Gopalakrishnan, 2016). Guided waves in laminated composites arise from the superposition of partial bulk waves that are trapped by free surfaces and internal interfaces; because the material is anisotropic and layered, both phase and group velocities become frequency-dependent and direction-dependent within the laminate (Huber & Sause, 2018).

**Dataset.** The dataset used is generated using the stiffness matrix method (SMM) together with group-velocity calculations, where six transversely isotropic CFRP material parameters: density ($\rho$), longitudinal and transverse Young's moduli ($E_1$ and $E_2$), in-plane shear modulus ($G_{12}$), and Poisson's ratios ($\nu_{12}$ and $\nu_{23}$) along with laminate layup are treated as inputs, and the corresponding polar group-velocity representations (PGVR) of the fundamental $A_0$ and $S_0$ Lamb modes are used as outputs. These responses are generated over propagation angles from 0° to 360° and excitation frequencies from 20 to 200 kHz in increments of 20 kHz for

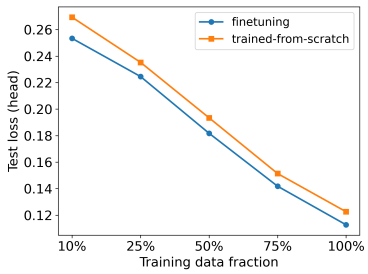
2 mm-thick, 16-layer symmetric laminates, and are subsequently converted into binary polar representations (Rautela et al., 2022a;b). More details are provided to Appendix A.4.4.

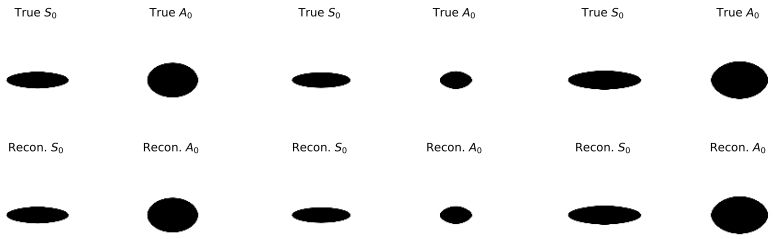
**Modeling.** We attach a lightweight task-specific head on the representation space of pretrained MORPH-S and estimate 6 material properties given polar group-velocity representations for $A_0$ and $S_0$ modes. The MORPH backbone is finetuned with image reconstruction objective (autoencoder-style) and the latent representations (at the end of transformer blocks) act as inputs to the TSH for regression. The TSH has two `Conv1D`→`GELU`→`Dropout` blocks, followed by two `Linear`→`GELU`→`Dropout` blocks.

**Experimental settings.** We fine-tune all parameters of MORPH-S and train the TSH (∼4M parameters) for 200 epochs on a single RTX-A6000 GPU. We use AdamW optimizers with learning rates of $10^{-4}$ for MORPH-S and $10^{-4}$ for TSH. For learning-rate scheduler, we linear warmup for 5 epochs (minimum LR $10^{-8}$) followed by cosine decay. We use a batch size of 8 and optimize mean-squared error (MSE) losses for both the reconstruction and regression objectives.




(a) Material property estimation (normalized scale) on the full test set



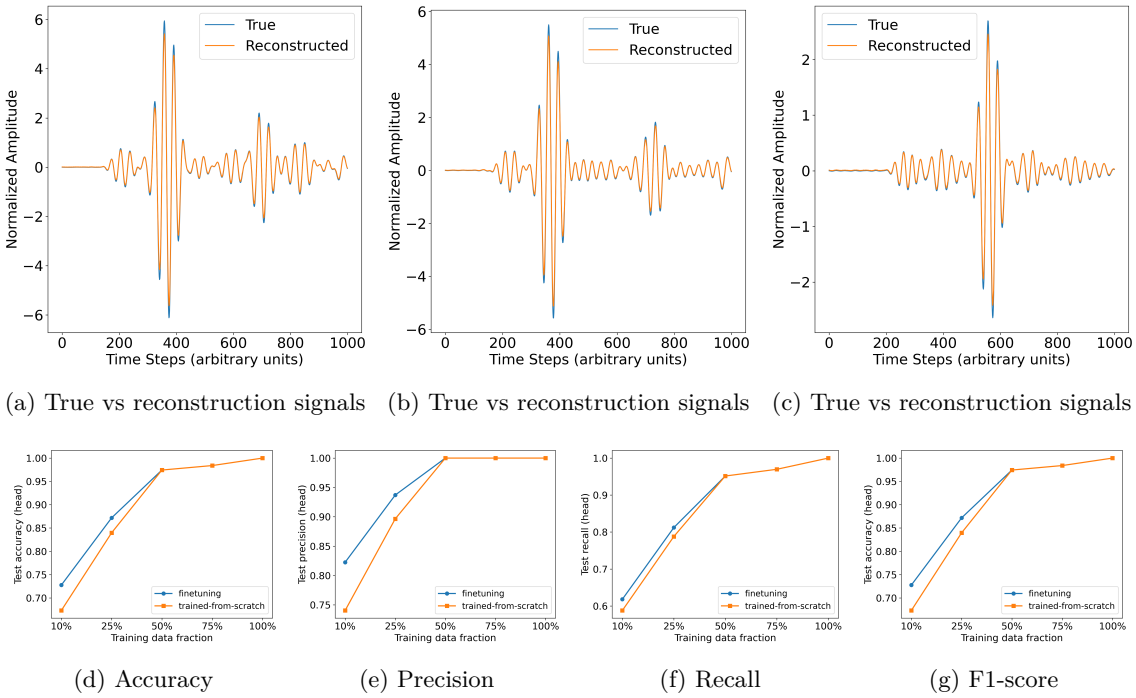

(b) (c) (d) (e)

Figure 13: Results: (a) Parameter estimation results on the full test set, (b) Test-set loss of the TSH (finetuned vs trained-from-scratch) versus fine-tuning-data fraction, (c-e) True vs reconstruction images on held-out test example

**Results.** Fig. 13 presents the results of inverse material-property estimation for composite materials. Fig. 13(a) compares the true and predicted values for six material properties: $\rho$, $E_1$, $E_2$, $G_{12}$, $\nu_{12}$, and $\nu_{23}$. The predictions achieve high $R^2$ values and low $L_2$ errors, indicating accurate property estimation. Fig. 13(c) shows the true and reconstructed polar group-velocity representations for the $A_0$ and $S_0$ modes across three test samples. Fig. 13(b) reports the transfer gain obtained by the finetuned model compared with a model trained from scratch. Notably, the finetuned model maintains a positive transfer gain even under out-of-distribution conditions. This suggests that the learned representations encode transferable features of wave propagation in composites, aided by exposure during pretraining to related wave dynamics present in shallow-water equations.

### 9.4 Inverse task: Damage detection in aerospace structures

Aerospace structures increasingly rely on composite materials because they offer high specific stiffness and strength, reduced mass, and greater design freedom, all of which are directly tied to fuel efficiency and structural performance. The difficulty is that composite airframes are also vulnerable to low-velocity




(a) True vs reconstruction signals (b) True vs reconstruction signals (c) True vs reconstruction signals

(d) Accuracy (e) Precision (f) Recall (g) F1-score

Figure 14: Results: (a-c) True vs reconstructed signals on the held out test, (d) Accuracy (higher is better), (e) Precision (higher is better), (f) Recall (higher is better), (g) F1-score (higher is better) on the held-out test set

impacts and other service events that can create subsurface, barely visible damage such as delamination and skin–stringer disbonding, with potentially large reductions in residual strength even when surface evidence is limited. This makes guided-wave-based structural health monitoring especially attractive: lightweight, permanently bonded piezoelectric transducers can interrogate relatively large areas, and the propagating waves are highly sensitive to discontinuities along their paths (Mitra & Gopalakrishnan, 2016). In this setting, damage detection is a fundamentally important *inverse problem* where we seek to recover the hidden damage state from a sparse set of noisy signals. For aerospace composites, this inverse mapping is particularly difficult because the signals are multimodal, dispersive, anisotropic, and strongly affected by reflections, thickness transitions, stiffeners, and environmental variability.

**Dataset.** In this work, we fine-tune MORPH-Ti using the UoNCT-GW dataset, which is derived from a representative composite wing-panel experiment originally introduced for guided-wave-based damage localization in a real-scale aerospace structure (Memmolo et al., 2018; Rautela et al., 2022c). The specimen is a tapered composite wing panel with three bays of different thicknesses, and each bay is instrumented with 13 piezoelectric transducers arranged in a circular array. The laminate stacking sequence is given as $[5H/B45/U/B/U/B45/U/B]_s$, and delamination is introduced by drop-weight impacts and verified by ultrasonic inspection. Following that study, the subset most commonly used for learning experiments is the 6 mm bay, interrogated in a round-robin pitch–catch scheme using a 4.5-cycle toneburst at 60 kHz and 80 V peak-to-peak amplitude. Signals are recorded both before and after impact, and the experiment is repeated ten times, yielding 1560 baseline signals and 1560 damaged signals. More dataset related details are deferred to Appendix A.4.5.

**Modeling.** We attach a lightweight task-specific head on the representation space of pretrained MORPH-Ti and classify signals into healthy and damaged classes. The MORPH backbone is finetuned with time-series reconstruction objective (autoencoder-style) and the latent representations (at the end of transformer blocks) act as inputs to the TSH for binary classification. The TSH has three `Conv1D`→`GELU`→`Dropout` blocks, followed by two `Linear`→`GELU`→`Dropout` blocks.

**Experimental settings.** We fine-tune all parameters of MORPH-Ti and train the TSH ($\sim$1M parameters) for 100 epochs on a single RTX-A6000 GPU. We use AdamW optimizers with learning rates of $10^{-4}$ for MORPH-Ti and $10^{-5}$ for TSH. For learning-rate scheduler, we linear warmup for 5 epochs (minimum LR $10^{-8}$) followed by cosine decay. We use a batch size of 8 and optimize mean-squared error (MSE) losses for both the reconstruction and classification objectives.

**Results.** We achieve full accuracy on the held-out test set. Fig. 14 shows reconstructions for three representative test examples in the top row. To study transfer gains, we sweep over different fractions of the training data and compare four classification metrics for models trained from scratch with random initialization and for the pretrained MORPH-Ti model. Our results match the full-dataset performance of CNN reported by Rautela et al. (2022c). We attribute this transfer capability to the time-series signals used during the pretraining phase.

### 9.5 Inverse task: Sea-Surface Temperature field reconstruction using sparse observations

Sea-surface temperature (SST) reconstruction is an inverse sparse-sensing problem in which a dense global temperature field must be inferred from only a small number of local ocean measurements. This setting is motivated by practical ocean monitoring, where observations from satellites, ships, buoys, or other sensing platforms may be sparse, irregular, and time-varying, while downstream climate and geophysical analyses require coherent global fields.

**Dataset.** We use the NOAA SST benchmark, which consists of weekly global SST observations collected from satellite and ship-based measurements on a $360 \times 180$ longitude–latitude grid, corresponding to a $1^\circ \times 1^\circ$ spatial resolution. Following prior work (Fukami et al., 2021; Santos et al., 2023), we use 1,040 weekly snapshots from 1981–2001 for training and evaluate on snapshots from 2001–2018. Sensors are sampled only over ocean locations, and land points are excluded since SST is not defined over continents. The input to the inverse task is therefore a sparse set of sensor readings and their spatial locations, or an equivalent Voronoi/mask representation, while the target is the dense valid-ocean SST field. This benchmark tests whether a model can reconstruct global temperature structure under very limited spatial coverage and remain robust to varying sensor counts and unseen sensor placements. More dataset related details are deferred to Appendix A.4.6.

**Modeling.** We formulate NOAA-SST reconstruction as a supervised dense-field regression problem from sparse observations. For each SST snapshot, we construct a two-channel input: the first channel contains the Voronoi-transformed sensor field, where each grid point is assigned the value of its nearest observed sensor, and the second channel contains a binary sensor-location mask. The target contains the full SST field together with the valid-ocean mask, which allows the reconstruction loss and evaluation to focus only on ocean points. During training, we sample the number of sensors from $\{1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$ and use different random seeds for each snapshot to expose the model to diverse sensor placements. For testing, we evaluate on unseen sensor placements with 10 and 100 sensors. We fine-tune MORPH-Ti for this inverse task using the two-field input at the native SST resolution of $180 \times 360$. We also pad the latitude dimension from 180 to 184 so that the spatial grid is divisible by the patch size.

**Experimental details.** We fine-tune all parameters of MORPH-Ti for 100 epochs on 2$\times$ RTX-A6000 GPU. We use AdamW optimizers with learning rates of $10^{-4}$. For learning-rate scheduler, we linear warmup for 5 epochs (minimum LR $10^{-8}$) followed by cosine decay. We use a batch size of 16 and optimize mean-squared error (MSE) loss over the valid SST channel. We use $L_2$ error norm ($\bar{\epsilon}$) as the success metric.

**Results.** Fig. 15 shows reconstruction results using 100 sensors at unseen placements for three test examples drawn from different months and years. Table 9 reports the $L_2$ error norms of the reconstructions across the full test set, demonstrating performance competitive with state-of-the-art methods. However, we do not observe transfer gains from the pretrained model in this study.

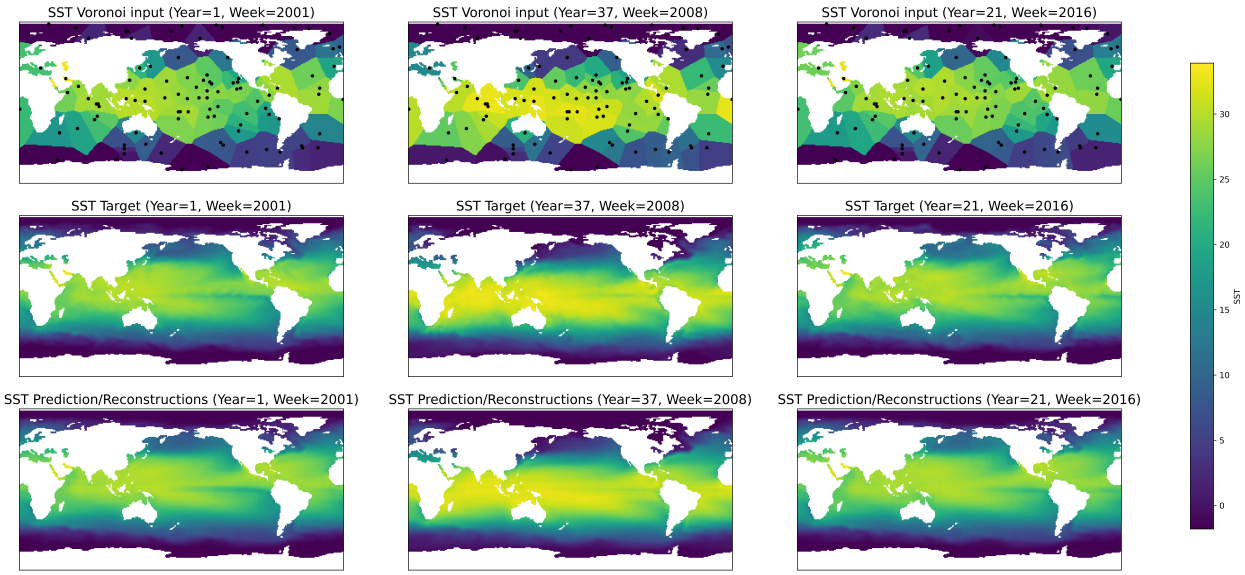



Figure 15: Reconstruction of SST field using 100 sensors (unseen placements): (top) Voronoi tessellation inputs, (middle) Targets, (bottom) Reconstructions. $L_2$ error norm ($\bar{\epsilon}$) across the test set is $0.038 \pm 8.73E-05$.

| **Sensors** | Voronoi-CNN | Senseiver | Morph (Ours) |
|---|---|---|---|
| 10 | 0.319 | 0.047 ± 3.07E-03 | 0.047 ± 1.09E-03 |
| 100 | 0.108 | 0.039 ± 1.54E-04 | 0.038 ± 8.73E-05 |

Table 9: Comparison of test-set reconstruction errors for MORPH-Ti-FT (ours), Voronoi-CNN (Fukami et al., 2021), and Senseiver (Santos et al., 2023) across different numbers of sensors. Errors are reported using the average $L_2$ norm ($\bar{\epsilon}$), with lower values indicating better accuracy.

## 10 Conclusion

We introduced MORPH, a modality-agnostic PDE foundation model for learning transferable representations across heterogeneous spatiotemporal physical systems. By combining the Unified Physics Tensor Format (UPTF-7) with component-wise convolutions, field-wise cross-attention, and factorized axial spatiotemporal attention, MORPH provides a single flexible backbone that operates across 1D–3D spatial domains, mixed scalar and vector fields, varying resolutions, and diverse physical regimes without task-specific architectural reconfiguration. Through large-scale pretraining on a heterogeneous multi-physics corpus and transfer to a broad suite of downstream tasks, we showed that pretrained MORPH models consistently improve over training from scratch under zero-shot evaluation, full-model fine-tuning, and parameter-efficient LoRA adaptation. The cross-modality transfer, scaling, and ablation studies further demonstrate that the proposed architectural components enable positive transfer across spatial dimensionalities and physics classes while improving data efficiency and model scalability. Beyond canonical PDE forecasting benchmarks, MORPH also transfers to practical forward and inverse physical-system tasks, including terminal key-frame prediction for extreme material dynamics, inertial-confinement-fusion parameter estimation, composite material property estimation, guided-wave damage detection, and sparse sea-surface-temperature reconstruction. These results indicate that pretrained PDE backbones can serve as reusable scientific representations for heterogeneous simulation, experimental, and sensing data, particularly in low-data and compute-constrained regimes. Overall, MORPH represents a step toward scalable and data-efficient scientific foundation models for heterogeneous PDE-governed systems.

## References


Armen Aghajanyan, Luke Zettlemoyer, and Sonal Gupta. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. *arXiv preprint arXiv:2012.13255*, 2020.

Benedikt Alkin, Andreas Fürst, Simon Schmid, Lukas Gruber, Markus Holzleitner, and Johannes Brandstetter. Universal physics transformers: A framework for efficiently scaling neural operators. *Advances in Neural Information Processing Systems*, 37:25152–25194, 2024.

Rushil Anirudh, Jayaraman J Thiagarajan, Peer-Timo Bremer, and Brian K Spears. Improved surrogates in inertial confinement fusion with manifold and cycle consistencies. *Proceedings of the National Academy of Sciences*, 117(18):9741–9746, 2020.

Divya Banesh, Sharmistha Chakrabarti, Soumi De, Gal Erez Egozi, Kyle Scott Hickmann, William Scott Hlavacek, Bryan Edward Kaiser, Sourabh Pandit, Jesus Jr Pulido, David James Schodt, and Christine Marie Sweeney. HEAT: The high explosives & affected targets (heat) dataset. Benchmark dataset, 2025. URL https://oceans11.lanl.gov/heat.

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.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, et al. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. *arXiv preprint arXiv:2312.09390*, 2023.

Michele Buzzicotti, Fabio Bonaccorso, P Clark Di Leoni, and Luca Biferale. Reconstruction of turbulent data with deep generative models for semantic inpainting from turb-rot database. *Physical Review Fluids*, 6(5): 050503, 2021.

Qianying Cao, Somdatta Goswami, and George Em Karniadakis. Laplace neural operator for solving differential equations. *Nature Machine Intelligence*, 6(6):631–640, 2024.

Junfeng Chen, Elie Hachem, and Jonathan Viquerat. Graph neural networks for laminar flow prediction around random two-dimensional shapes. *Physics of Fluids*, 33(12), 2021.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Wuyang Chen, Jialin Song, Pu Ren, Shashank Subramanian, Dmitriy Morozov, and Michael W Mahoney. Data-efficient operator learning via unsupervised pretraining and in-context learning. *Advances in Neural Information Processing Systems*, 37:6213–6245, 2024.

M Cheng, Fangxin Fang, Christopher C Pain, and IM Navon. Data-driven modelling of nonlinear spatio-temporal fluid flows using a deep convolutional generative adversarial network. *Computer Methods in Applied Mechanics and Engineering*, 365:113000, 2020.

Jungyeon Cho and A Lazarian. Compressible magnetohydrodynamic turbulence: mode coupling, scaling relations, anisotropy, viscosity-damped regime and astrophysical implications. *Monthly Notices of the Royal Astronomical Society*, 345(1):325–339, 2003.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

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

Han Gao, Sebastian Kaltenbach, and Petros Koumoutsakos. Generative learning of the solution of parametric partial differential equations using guided diffusion models and virtual observations. *Computer Methods in Applied Mechanics and Engineering*, 435:117654, 2025.

Somdatta Goswami, Katiana Kontolati, Michael D Shields, and George Em Karniadakis. Deep transfer operator learning for partial differential equations under conditional shift. *Nature Machine Intelligence*, 4 (12):1155–1164, 2022.

Peter Gray and Stephen K Scott. Autocatalytic reactions in the isothermal, continuous stirred tank reactor: Oscillations and instabilities in the system a+ 2b→ 3b; b→ c. *Chemical Engineering Science*, 39(6): 1087–1097, 1984.

Jayesh K Gupta and Johannes Brandstetter. Towards multi-spatiotemporal-scale generalized pde modeling. *arXiv preprint arXiv:2209.15616*, 2022.

Ricardo Luna Gutierrez, Vineet Gundecha, Rahman Ejaz, Varchas Gopalaswamy, Riccardo Betti, Sahand Ghorbanpour, Aarne Lees, and Soumyendu Sarkar. Im-lpg: Inverse modeling approach to laser pulse shape generation in inertial confinement fusion. In *NeurIPS 2025 AI for Science Workshop*, 2025.

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. *arXiv preprint arXiv:2403.03542*, 2024.

Maximilian Herde, Bogdan Raonic, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel de Bézenac, and Siddhartha Mishra. Poseidon: Efficient foundation models for pdes. *Advances in Neural Information Processing Systems*, 37:72525–72624, 2024.

Ryley G Hill, Kai Gao, Aleksandra Pachalieva, Agnese Marcato, Xiaoyu Wang, Pascal Grosset, Esteban Rougier, Zhou Lei, Javier E Santos, Vinamra Agrawal, et al. Material fracturing and failure simulation datasets. *Scientific Data*, 2025.

Keiya Hirashima, Kana Moriwaki, Michiko S Fujii, Yutaka Hirai, Takayuki R Saitoh, and Junichiro Makino. 3d-spatiotemporal forecasting the expansion of supernova shells using deep learning towards high-resolution galaxy simulations. *Monthly Notices of the Royal Astronomical Society*, 526(3):4054–4066, 2023.

Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. *arXiv preprint arXiv:1912.12180*, 2019.

Philipp Holl, Vladlen Koltun, Kiwon Um, and Nils Thuerey. phiflow: A differentiable pde solving framework for deep learning via physical simulations. In *NeurIPS workshop*, volume 2, 2020.

Benjamin Holzschuh, Qiang Liu, Georg Kohl, and Nils Thuerey. Pde-transformer: Efficient and versatile transformers for physics simulations. *arXiv preprint arXiv:2505.24717*, 2025.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Armin Huber and Markus GR Sause. Classification of solutions for guided waves in anisotropic composites with large numbers of layers. *The Journal of the Acoustical Society of America*, 144(6):3236–3251, 2018.

MA Khodkar and Pedram Hassanzadeh. A data-driven, physics-informed framework for forecasting the spatiotemporal evolution of chaotic dynamics with nonlinearities modeled as exogenous forcings. *Journal of Computational Physics*, 440:110412, 2021.

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International conference on learning representations*, 2021.

Katiana Kontolati, Somdatta Goswami, George Em Karniadakis, and Michael D Shields. Learning nonlinear operators in latent spaces for real-time predictions of complex dynamics in physical systems. *Nature Communications*, 15(1):5101, 2024.

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, 2024a.

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.

Zongyi Li, Hongkai Zheng, Nikola Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Kamyar Azizzadenesheli, and Anima Anandkumar. Physics-informed neural operator for learning partial differential equations. *ACM/JMS Journal of Data Science*, 1(3):1–27, 2024b.

Yuxuan Liu, Jingmin Sun, Xinjie He, Griffin Pinney, Zecheng Zhang, and Hayden Schaeffer. Prose-fd: A multimodal pde foundation model for learning multiple operators for forecasting fluid dynamics. *arXiv preprint arXiv:2409.09811*, 2024.

Yuying Liu, J Nathan Kutz, and Steven L Brunton. Hierarchical deep learning of multiscale differential equation time-steppers. *Philosophical Transactions of the Royal Society A*, 380(2229):20210200, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

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.

Romit Maulik, Bethany Lusch, and Prasanna Balaprakash. Reduced-order modeling of advection-dominated systems with recurrent neural networks and convolutional autoencoders. *Physics of Fluids*, 33(3), 2021.

Michael McCabe, Bruno Régaldo-Saint Blancard, Liam Parker, Ruben Ohana, Miles Cranmer, Alberto Bietti, Michael Eickenberg, Siavash Golkar, Geraud Krawezik, Francois Lanusse, et al. Multiple physics pretraining for spatiotemporal surrogate models. *Advances in Neural Information Processing Systems*, 37: 119301–119335, 2024.

Vittorio Memmolo, Natalino D Boffa, Leandro Maio, Ernesto Monaco, and Fabrizio Ricci. Damage localization in composite structures using a guided waves based multi-parameter approach. *Aerospace*, 5(4):111, 2018.

Sidharth S Menon, Trishit Mondal, Shuvayan Brahmachary, Aniruddha Panda, Subodh M Joshi, Kaushic Kalyanaraman, and Ameya D Jagtap. On scientific foundation models: Rigorous definitions, key applications, and a comprehensive survey. *Neural Networks*, pp. 108567, 2026.

Mira Mitra and S Gopalakrishnan. Guided wave based structural health monitoring: A review. *Smart Materials and Structures*, 25(5):053001, 2016.

Tung Nguyen, Johannes Brandstetter, Ashish Kapoor, Jayesh K Gupta, and Aditya Grover. Climax: A foundation model for weather and climate. *arXiv preprint arXiv:2301.10343*, 2023.

Ruben Ohana, Michael McCabe, Lucas Meyer, Rudy Morel, Fruzsina Agocs, Miguel Beneitez, Marsha Berger, Blakesly Burkhart, Stuart Dalziel, Drummond Fielding, et al. The well: a large-scale collection of diverse physics simulations for machine learning. *Advances in Neural Information Processing Systems*, 37: 44989–45037, 2024.

Vivek Oommen, Khemraj Shukla, Somdatta Goswami, Rémi Dingreville, and George Em Karniadakis. Learning two-phase microstructure evolution using neural operators and autoencoder architectures. *npj Computational Materials*, 8(1):190, 2022.

Md Ashiqur Rahman, Zachary E Ross, and Kamyar Azizzadenesheli. U-no: U-shaped neural operators. *arXiv preprint arXiv:2204.11127*, 2022.

M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. ISSN 0021-9991.

Mahindra Rautela, Armin Huber, J Senthilnath, and S Gopalakrishnan. Inverse characterization of composites using guided waves and convolutional neural networks with dual-branch feature fusion. *Mechanics of Advanced Materials and Structures*, 29(27):6595–6611, 2022a.

Mahindra Rautela, J Senthilnath, Armin Huber, and S Gopalakrishnan. Toward deep generation of guided wave representations for composite materials. *IEEE Transactions on Artificial Intelligence*, 5(3):1102–1109, 2022b.

Mahindra Rautela, J Senthilnath, Ernesto Monaco, and S Gopalakrishnan. Delamination prediction in composite panels using unsupervised-feature learning methods with wavelet-enhanced guided wave representations. *Composite Structures*, 291:115579, 2022c.

Mahindra Rautela, Alan Williams, and Alexander Scheinker. A conditional latent autoregressive recurrent model for generation and forecasting of beam dynamics in particle accelerators. *Scientific Reports*, 14(1): 18157, 2024.

Mahindra Rautela, Alan Williams, and Alexander Scheinker. Time-inversion of spatiotemporal beam dynamics using uncertainty-aware latent evolution reversal. *Phys. Rev. E*, 111:025307, Feb 2025. doi: 10.1103/PhysRevE.111.025307. URL https://link.aps.org/doi/10.1103/PhysRevE.111.025307.

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 R Fox, Arvind Mohan, Daniel O'Malley, Hari Viswanathan, and Nicholas Lubbers. Development of the senseiver for efficient field reconstruction from sparse observations. *Nature Machine Intelligence*, 5(11):1317–1325, 2023.

Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, 2015.

Alberto Solera-Rico, Carlos Sanmiguel Vila, Miguel Gómez-López, Yuning Wang, Abdulrahman Almashjary, Scott TM Dawson, and Ricardo Vinuesa. $\beta$-variational autoencoders and transformers for reduced-order modelling of fluid flows. *Nature Communications*, 15(1):1361, 2024.

Brian K Spears, Scott Brandon, Dan T Casey, John E Field, Jim A Gaffney, Kelli D Humbird, Andrea L Kritcher, Michael KG Kruse, Eugene Kur, Bogdan Kustowski, et al. Predicting fusion ignition at the national ignition facility with physics-informed deep learning. *Science*, 389(6761):727–731, 2025.

Shashank Subramanian, Peter Harrington, Kurt Keutzer, Wahid Bhimji, Dmitriy Morozov, Michael W Mahoney, and Amir Gholami. Towards foundation models for scientific machine learning: Characterizing scaling and transfer behavior. *Advances in Neural Information Processing Systems*, 36:71242–71262, 2023.

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Daniel MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. Pdebench: An extensive benchmark for scientific machine learning. *Advances in Neural Information Processing Systems*, 35:1596–1611, 2022.

Eleuterio F Toro, Michael Spruce, and William Speares. Restoration of the contact surface in the hll-riemann solver. *Shock waves*, 4(1):25–34, 1994.

Tapas Tripura and Souvik Chakraborty. Wavelet neural operator for solving parametric partial differential equations in computational mechanics problems. *Computer Methods in Applied Mechanics and Engineering*, 404:115783, 2023.

Bram Van Leer. Towards the ultimate conservative difference scheme. *Journal of computational physics*, 135 (2):229–248, 1997.

Pantelis R Vlachas, Georgios Arampatzis, Caroline Uhler, and Petros Koumoutsakos. Multiscale simulations of complex systems by learning their effective dynamics. *Nature Machine Intelligence*, 4(4):359–366, 2022.

Nils Wandel, Michael Weinmann, and Reinhard Klein. Teaching the incompressible navier–stokes equations to fast neural surrogate models in three dimensions. *Physics of Fluids*, 33(4), 2021.

Wuzhe Xu, Yulong Lu, and Li Wang. Transfer learning enhanced deeponet for long-time prediction of evolution equations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 10629–10636, 2023.

Liu Yang, Siting Liu, Tingwei Meng, and Stanley J Osher. In-context operator learning with data prompts for differential equation problems. *Proceedings of the National Academy of Sciences*, 120(39):e2310142120, 2023.

Zhanhong Ye, Xiang Huang, Leheng Chen, Hongsheng Liu, Zidong Wang, and Bin Dong. Pdeformer: Towards a foundation model for one-dimensional partial differential equations. *arXiv preprint arXiv:2402.12652*, 2024.

Luo Yining, Chen Yingfa, and Zhang Zhen. Cfdbench: A large-scale benchmark for machine learning methods in fluid dynamics. *arXiv preprint arXiv:2310.05963*, 2023.

Yilin Zhuang, Sibo Cheng, and Karthik Duraisamy. Spatially-aware diffusion models with cross-attention for global field reconstruction with sparse observations. *Computer Methods in Applied Mechanics and Engineering*, 435:117623, 2025.

# A Appendix: Data Details

## A.1 Benchmark: PDEBench

### A.1.1 1D Burgers' equation

**Description**: It is a canonical nonlinear advection–diffusion PDE that interpolates between smooth diffusion and shock-forming hyperbolic flow as viscosity decreases. By varying $\nu$, it effectively sweeps a Reynolds-like number and reveals steepening, shock formation, and dissipation—making it a clean test for nonlinearity and shock handling.

**PDE**: $\partial_t u(t,x) + \partial_x\big(\frac{1}{2}u^2(t,x)\big) = \frac{\nu}{\pi}\,\partial_{xx}u(t,x)$.

**Domain**: $x \in (0,1),\ t \in (0,2]$.

**BCs & ICs.** Periodic BCs on $(0,1)$. Initial condition is a random superposition of sines $u_0(x) = \sum_i A_i \sin(k_i x + \phi_i)$ with uniformly sampled amplitudes/phases/wavenumbers.

**Numerical scheme**: 2nd-order upwind finite difference scheme in space and time.

**Data specifications**: $N = 10{,}000$ trajectories, $T = 200$ time steps, $W = 1024$ spatial points, one scalar field variable $u$. We use dataset corresponding to $\nu = 0.001$.

**Category**: Fine-tuning

### A.1.2 1D Diffusion-reaction

**Description**: It combines linear diffusion with a local, potentially rapid (nearly exponential) source term, yielding stiff transients and front-like dynamics. It stresses models' ability to handle disparate time scales and locally amplified errors.

**PDE**: $\partial_t u - \nu\,\partial_{xx}u - \rho\,u(1-u) = 0$.

**Domain**: $x \in (0,1),\ t \in (0,1]$.

**BCs & ICs.** Periodic BCs. ICs drawn from the same random sinusoidal family as Burgers with rectification/normalization for well-posedness.

**Numerical scheme** Finite-difference time–space solver with the piecewise-exact solution (PES) method for the source term part.

**Data specifications**: $N = 10{,}000$ trajectories, $T = 100$ time steps, $W = 1024$ spatial points, one scalar field variable $u$. We use dataset corresponding to $\nu = 0.5, \rho = 1.0$.

**Category**: Fine-tuning

### A.1.3 2D Diffusion-reaction

**Description**: The FitzHugh–Nagumo (FHN) system is a prototypical reaction–diffusion equation with applications to modeling biological pattern phenomena. It consists of two nonlinearly coupled scalar fields i.e., activator and inhibitor.

**PDE**: $\partial_t u = D\nabla^2 u + R(u)$, where, $R_u = u - u^3 - k - v, R_v = u - v$

**Domain**: $(x,y) \in (-1,1)^2,\ t \in (0,5]$.

**BCs & ICs.** No–flux (Neumann) BCs. ICs are standard normal random noise.

**Constants**: $k = 5\times10^{-3}, D_u = 1\times10^{-3}, D_v = 5\times10^{-3}$.

**Numerical scheme** Finite-volume in space and classical RK4 in time.

**Data specifications**: $N = 1{,}000$ trajectories, $T = 100$ time steps, $H \times W = 128 \times 128$ spatial points, two scalar field variables $(u,v)$. We use dataset corresponding to $\nu = 0.5, \rho = 1.0$

**Category**: Pretraining

#### A.1.4 Compressible Navier-Stokes (CNS) system

**Description**: The CNS equations describe the motion of Newtonian fluids in regimes where density variations due to pressure are significant. This setting is fundamental in aerodynamics, gas dynamics, aeroacoustics, astrophysical flows, weather and climate dynamics etc. The governing PDEs are conservation laws for mass, momentum, and energy. The CNS solutions may exhibit discontinuities/shocks and turbulent cascades over a larger range of multiscale behavior.

**PDE**:

$$\begin{aligned}
\partial_t \rho + \nabla \cdot (\rho \mathbf{v}) &= 0, \\
\rho(\partial_t \mathbf{v} + \mathbf{v} \cdot \nabla \mathbf{v}) &= -\nabla p + \eta \Delta \mathbf{v} + (\zeta + \eta/3)\nabla(\nabla \cdot \mathbf{v}), \\
\partial_t \left(\epsilon + \tfrac{1}{2}\rho \|\mathbf{v}\|^2\right) + \nabla \cdot \left[(\epsilon + p + \tfrac{1}{2}\rho\|\mathbf{v}\|^2)\mathbf{v} - \mathbf{v} \cdot \sigma'\right] &= 0.
\end{aligned}$$

where $\rho$ is the density, $\mathbf{u}$ the velocity field, $p$ the pressure, $\epsilon$ the internal energy, $\eta$ the shear viscosity, $\zeta$ the bulk viscosity, and $\sigma'$ the viscous stress tensor.

**BCs & ICs**: Periodic BCs, Random ICs.

**Numerical scheme**: For advection/shock terms: HLLC + MUSCL (robust, shock-capturing, 2nd order). For viscous/diffusion terms: central differencing (smooth, consistent with Navier–Stokes viscosity). More details on the solver Toro et al. (1994); Van Leer (1997).

**Data specifications and category**:

1. 1d CNS (Pretraining): $N = 10{,}000$ trajectories, $T = 100$ time steps, $W = 1024$ spatial points, one vector field ($v$), two scalar fields ($\rho, P$). We use dataset corresponding to $\eta = 10^{-2}, \zeta = 10^{-2}$.
2. 2d CNS (Fine-tuning): $N = 10{,}000$ trajectories, $T = 21$ time steps, $H \times W = 512 \times 512$ spatial points, one vector field ($v$), two scalar fields ($\rho, P$). We use dataset corresponding to $\eta = 10^{-8}, \zeta = 10^{-8}, M = 0.1$.
3. 3d CNS (Pretraining): $N = 200$ trajectories, $T = 21$ time steps, $D \times H \times W = 128 \times 128 \times 128$ spatial points, one vector field ($v$), two scalar fields ($\rho, P$). We use dataset corresponding to $(\eta, \zeta, M)$=$(10^{-8}, 10^{-8}, 0.1)$ and $(10^{-8}, 10^{-8}, 1.0)$.
4. 3d CNS-Turbulent (Fine-tuning): $N = 600$ trajectories, $T = 21$ time steps, $D \times H \times W = 64 \times 64 \times 64$ spatial points, one vector field ($v$), two scalar fields ($\rho, P$). We use dataset corresponding to $(\eta, \zeta, M) = (10^{-8}, 10^{-8}, 1.0)$.

#### A.1.5 2D inhomogeneous, incompressible Navier-Stokes

**Description**: The inhomogeneous, forced incompressible Navier–Stokes (INS) system models Newtonian fluids in the low–Mach regime where density is effectively constant. It underpins a wide range of applications, including hydromechanics, environmental and geophysical flows, weather and climate components at resolved scales, and engineered processes such as mixing and cooling.

**PDE**: $\nabla \cdot \mathbf{v} = 0, \rho(\partial_t \mathbf{v} + \mathbf{v} \cdot \nabla \mathbf{v}) = -\nabla p + \eta \Delta \mathbf{v} + \mathbf{u}$ , where $\mathbf{v} = (v_x, v_y)$ is velocity, $p$ pressure, $\eta$ the (shear) viscosity, and $\mathbf{u}$ the inhomogeneous forcing.

**Domain**: $\Omega \in [0, 1]^2$.

**BCs & ICs**: Non-periodic Dirichlet (no-slip) BCs. Initial conditions $v_0$ and inhomogeneous forcing parameters $u$ are independently sampled from isotropic Gaussian random fields with truncated power-law power spectra.

**Constants**: Viscosity $\nu = \eta/\rho = 0.01$. Density is treated as constant.

**Numerical scheme**: Finite-difference simulations implemented in PhiFlow (Holl et al., 2020).

**Data specifications**: $N = 64$ trajectories, $T = 1000$ time steps, $H \times W = 512 \times 512$ spatial points, one vector field ($v$), one static vector field ($F$) (not used).

**Category**: Pretraining

### A.2 Benchmark: The Well

#### A.2.1 2D Gray-Scott diffusion reaction

**Description**: Two-species reaction–diffusion model whose scalar concentrations vary in space and time. It exhibits rich pattern formations and is used as a canonical test for nonlinear spatiotemporal dynamics and biological morphogenesis (Gray & Scott, 1984).

**PDE**: $\partial_t u = D_u \nabla^2 u - uv^2 + F(1-u); \partial_t v = D_v \nabla^2 v + uv^2 - (F+k)v$.

**Domain**: $\Omega \in [-1, 1]^2$.

**BCs & ICs**: Doubly periodic BCs

**Constants**: $\delta_A = 2 \times 10^{-5}$, $\delta_B = 1 \times 10^{-5}$.

**Numerical scheme**: Implicit-explicit exponential time-differencing RK4 in time and Fourier spectral method in space.

**Data specifications**: $N = 200$ trajectories, $T = 1000$ time steps, $H \times W = 128 \times 128$ spatial points, two scalar field variables $(u, v)$. We use dataset corresponding to "Bubbles" pattern with $f = 0.098, k = 0.057$.

**Category**: Fine-tuning.

#### A.2.2 3D Magnetohydrodynamics

**Description**: A multiphysics dataset coupling fluid dynamics and electromagnetism. MHD is a strongly nonlinear system in which coupled fluid–magnetic dynamics generate broadband turbulence, shocks and compressions, intermittent current sheets, and reconnection, yielding larger dynamics range of spatial and temporal scales. MHD modeling finds applications in understanding solar winds, interstellar medium, magnetospheres, fusion energy devices like tokamaks (Burkhart et al., 2020).

**PDE**: Ideal MHD with isothermal EOS $p = c_s^2 \rho$:

$$\partial_t \rho + \nabla \cdot (\rho \mathbf{v}) = 0, \quad \partial_t (\rho \mathbf{v}) + \nabla \cdot \left[ \rho \mathbf{v}\mathbf{v} + \left( p + \tfrac{B^2}{8\pi} \right) \mathbf{I} - \tfrac{1}{4\pi} \mathbf{B}\mathbf{B} \right] = \mathbf{f}, \quad \partial_t \mathbf{B} - \nabla \times (\mathbf{v} \times \mathbf{B}) = 0.$$

**Domain**: Periodic cube.

**BCs & ICs**: Periodic BCs, continuous large-scale solenoidal forcing at $k \approx 2.5$ (box units).

**Constants**: Isothermal sound speed $c_s$. Parameter sweeps over sonic Mach number $M_s \in \{0.5, 0.7, 1.5, 2.0, 7.0\}$ and Alfvénic Mach number $M_A \in \{0.7, 2.0\}$.

**Numerical scheme**: Third-order hybrid ENO shock-capturing scheme (Cho & Lazarian, 2003), isothermal EOS and forced turbulence. The high-res data were generated at $256^3$ and anti-aliased downsampled to $64^3$ for the MHD_64 split.

**Data specifications**: $N = 97$ trajectories, $T = 100$ time steps, $D \times H \times W = 64 \times 64 \times 64$ spatial points, two vector field variables $(v, B)$ and one scalar field ($\rho$). We use all the files present on the repository.

**Category**: Pretraining.

#### A.2.3 3D Turbulent cooling gravity

**Description:** Self-gravitating, compressible gas with radiative cooling/heating. It is used to model turbulent molecular clouds and multiphase interstellar medium where cooling drives filament formation and collapse.

**PDE:** Compressible hydrodynamics with gravity and energy source terms (monatomic gas $\gamma = 5/3$):

$$\frac{d\rho}{dt} = -\rho\nabla\cdot\mathbf{v}, \quad \frac{d\mathbf{v}}{dt} = -\frac{\nabla P}{\rho} - \nabla\Phi + \mathbf{a}_{\text{visc}}, \quad \frac{du}{dt} = -\frac{P}{\rho}\nabla\cdot\mathbf{v} + \frac{\Gamma - \Lambda}{\rho},$$

with Poisson gravity through $\Phi$; cooling/heating via metallicity-dependent $\Lambda, \Gamma$.

**Domain:** 3D volume.

**BCs & ICs:** Isolated self-gravitating gas sphere ($10^6\, M_\odot$) with turbulent velocity spectrum $\propto k^{-4}$;

**Constants:** Adiabatic index $\gamma = 5/3$.

**Numerical scheme:** Density-Independent SPH (DISPH) in `ASURA-FDPS`; radiative processes via tabulated cooling/heating (Hirashima et al., 2023).

**Data specifications:** $N = 100$ trajectories, $T = 50$ time steps, $D \times H \times W = 64 \times 64 \times 64$ spatial points, one vector field variable ($v$) and three scalar field ($\rho, P, T$). Out of 4 fields, we only take 3 independent fields ($v, P, T$). We use single dataset file corresponding to $T_0 = 10, \rho_0 = 0.445, Z = 0.1$.

**Category:** Fine-tuning

### A.3 Benchmark: PDEGYM

#### A.3.1 FNS-KF

**Description:** Two-dimensional Kolmogorov flow: an incompressible fluid driven by a steady sinusoidal body force.

**PDE:** Forced incompressible Navier–Stokes:

$$\partial_t \mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p - \nu\,\Delta\mathbf{u} = \mathbf{f}, \qquad \nabla\cdot\mathbf{u} = 0,$$

with time-independent forcing

$$f(x, y) = 0.1\,\sin\big(2\pi(x + y)\big), \qquad \partial_t f \equiv 0.$$

**Domain:** $D = [0, 1]^2$.

**BCs & ICs:** Periodic boundary conditions. Initial velocity is set exactly as in NS-PwC (Herde et al., 2024) by prescribing a piecewise-constant vorticity $\omega_0$ on a $p \times p$ uniform partition (with $p = 10$) and recovering a divergence-free velocity:

$$\omega_0(x, y) = c_{ij} \quad \text{for } (x, y) \in [x_{i-1}, x_i] \times [y_{j-1}, y_j], \qquad x_i = y_i = \frac{i}{p}, \quad c_{ij} \sim \mathcal{U}[-1, 1].$$

The velocity $\mathbf{u}_0$ is obtained from $\omega_0$ under $\nabla\cdot\mathbf{u}_0 = 0$.

**Constants:** Small effective viscosity from the spectral–hyperviscosity setup, $\nu = 4 \times 10^{-4}$, forcing amplitude 0.1 and wavenumber $2\pi$ along diagonal.

**Numerical scheme**: Pseudospectral Fourier method with artificial (hyper)viscosity and projection to the first N modes using AZEBAN spectral hyperviscosity code.

**Data specifications:** $N = 20{,}000$ trajectories, $T = 21$ time steps, $H \times W = 128 \times 128$, one vector field variable ($v$). We assemble all the '.nc' files in the data directory into a single combined file (Herde et al., 2024). However, we use limited trajectories.

**Category:** Fine-tuning

### A.3.2 NS-SL

**Description:** The Navier–Stokes shear-layer (NS-SL) dataset is a two-dimensional incompressible-flow benchmark in which two oppositely directed shear layers are perturbed by a random sinusoidal displacement. The resulting dynamics exhibit Kelvin–Helmholtz-type roll-up, vortex pairing, and fine-scale mixing, making the task a stringent benchmark for learning advection-dominated fluid evolution and stable autoregressive rollouts.

**PDE:** The trajectories solve the incompressible Navier–Stokes equations

$$\partial_t \mathbf{v} + (\mathbf{v} \cdot \nabla)\mathbf{v} + \nabla p = \nu \Delta \mathbf{v}, \qquad \nabla \cdot \mathbf{v} = 0,$$

where $\mathbf{v} = (v_x, v_y)$ is the velocity field and $p$ is the pressure.

**Domain:** $\Omega = [0, 1]^2$, with evaluation performed up to $t = 0.7$.

**BCs & ICs:** Periodic boundary conditions are imposed in both spatial directions. The initial condition is a perturbed double shear layer:

$$v_x^0(x, y) = \begin{cases} \tanh\left(2\pi \dfrac{y - 0.25}{\rho}\right), & y + \sigma_\delta(x) \le \dfrac{1}{2}, \\ \tanh\left(2\pi \dfrac{0.75 - y}{\rho}\right), & \text{otherwise}, \end{cases} \qquad v_y^0(x, y) = 0,$$

where the random interface perturbation is

$$\sigma_\delta(x) = \xi + \delta \sum_{k=1}^{m} \alpha_k \sin(2\pi k x - \beta_k).$$

**Constants:** The number of perturbation modes is sampled as $m \sim \mathcal{U}\{7, \dots, 12\}$. The random coefficients are $\alpha_k \sim \mathcal{U}[0, 1]$, $\beta_k \sim \mathcal{U}[0, 2\pi]$, with $\delta = 0.025$, $\rho \sim \mathcal{U}[0.08, 0.12]$, and $\xi \sim \mathcal{U}[-0.0625, 0.0625]$. The effective viscosity follows the PDEGym spectral-hyperviscosity setup, with $\nu \approx 4 \times 10^{-4}$ applied to high Fourier modes.

**Numerical scheme:** The data are generated using the AZEBAN pseudo-spectral Fourier solver with spectral projection and high-mode artificial viscosity. The discretization uses $N = 128$ Fourier modes, $m_N = \sqrt{N}$, and $\epsilon_N = 0.05/N$.

**Data specifications:** The full PDEGym dataset contains $N = 40{,}000$ trajectories, split into 39,640 training trajectories, 120 validation trajectories, and 240 test trajectories. Each snapshot is represented on a $128 \times 128$ grid with one vector field variable $\mathbf{v} = (v_x, v_y)$. The same temporal subsampling as NS-PwC is used, and evaluation is performed at the 14th time step, corresponding to $t = 0.7$. In our controlled ablation experiments, we fine-tune on 128 trajectories and evaluate autoregressive rollouts on the held-out test trajectories.

**Category:** Fine-tuning.

### A.3.3 NS-PwC

**Description:** The Navier–Stokes piecewise-constant-vorticity (NS-PwC) dataset is a two-dimensional incompressible-flow benchmark initialized by discontinuous vorticity patches. It can be interpreted as a multiple-Riemann-problem analogue in vorticity form, producing sharp vorticity interfaces, nonlinear advection, vortex interactions, and turbulent mixing.

**PDE:** The trajectories solve the incompressible Navier–Stokes equations

$$\partial_t \mathbf{v} + (\mathbf{v} \cdot \nabla)\mathbf{v} + \nabla p = \nu \Delta \mathbf{v}, \qquad \nabla \cdot \mathbf{v} = 0,$$

where $\mathbf{v} = (v_x, v_y)$ denotes the velocity field.

**Domain:** $\Omega = [0, 1]^2$, with evaluation performed up to $t = 0.7$.

**BCs & ICs:** Periodic boundary conditions are imposed. The initial condition is specified through the scalar vorticity $\omega_0 = \partial_x v_y^0 - \partial_y v_x^0$, which is constant on a uniform $p \times p$ partition:

$$\omega_0(x, y) = c_{ij}, \qquad (x, y) \in [x_{i-1}, x_i] \times [y_{j-1}, y_j],$$

where

$$x_i = y_i = \frac{i}{p}, \qquad i = 0, 1, \ldots, p, \qquad c_{ij} \sim \mathcal{U}[-1, 1].$$

The initial velocity field $\mathbf{v}^0 = (v_x^0, v_y^0)$ is then recovered from $\omega_0$ by enforcing incompressibility, for example through a periodic stream-function solve.

**Constants:** The number of cells per spatial direction is fixed to $p = 10$. The effective viscosity follows the PDEGym spectral-hyperviscosity setup, with $\nu \approx 4 \times 10^{-4}$ applied to high Fourier modes.

**Numerical scheme:** The trajectories are generated using the AZEBAN pseudo-spectral Fourier solver with projection onto the first $N = 128$ modes and artificial high-frequency damping. The hyperviscosity parameters are $m_N = \sqrt{N}$ and $\epsilon_N = 0.05/N$.

**Data specifications:** The full PDEGym dataset contains $N = 20{,}000$ trajectories, with the same split as the NS-Sines dataset: 19,640 training trajectories, 120 validation trajectories, and 240 test trajectories. Each snapshot is stored on a $128 \times 128$ grid with one vector field variable $\mathbf{v} = (v_x, v_y)$. Eight time steps are included by selecting every other snapshot from the initial state up to the 14th time step. Evaluation is performed at the 14th step, corresponding to $t = 0.7$. In our controlled ablation experiments, we fine-tune on 128 trajectories and evaluate autoregressive rollouts on the held-out test trajectories.

**Category:** Fine-tuning.

### A.3.4 CE-RM

**Description:** The compressible-Euler Richtmyer–Meshkov (CE-RM) dataset models shock-driven instability at a randomly perturbed material interface. A localized high-pressure region interacts with a density discontinuity, generating baroclinic vorticity, interface deformation, and multiscale compressible flow structures. This benchmark is therefore useful for testing whether a surrogate can capture shock-interface interactions and the subsequent growth of hydrodynamic instabilities.

**PDE:** The trajectories solve the compressible Euler equations

$$\partial_t U + \nabla \cdot F(U) = 0, \qquad U = (\rho, \rho\mathbf{v}, E)^\top,$$

with flux

$$F(U) = (\rho\mathbf{v}, \rho\mathbf{v} \otimes \mathbf{v} + pI, (E + p)\mathbf{v})^\top, \qquad E = \frac{1}{2}\rho\|\mathbf{v}\|_2^2 + \frac{p}{\gamma - 1},$$

where $\rho$ is density, $\mathbf{v} = (v_x, v_y)$ is velocity, $p$ is pressure, and $E$ is total energy.

**Domain:** $\Omega = [0, 1]^2$, with simulations evolved up to final time $T = 2$.

**BCs & ICs:** Periodic boundary conditions are imposed. The initial pressure, density, and velocity are

$$p_0(x, y) = \begin{cases} 20, & \sqrt{x^2 + y^2} < 0.1, \\ 1, & \text{otherwise}, \end{cases} \qquad \rho_0(x, y) = \begin{cases} 2, & |x| < I(x, y, \omega), \\ 1, & \text{otherwise}, \end{cases} \qquad v_x^0 = v_y^0 = 0.$$

The random material interface is defined as

$$I(x, y, \omega) = 0.25 + \epsilon \sum_{j=1}^{K} a_j(\omega) \sin\big(2\pi\big((x, y) + b_j(\omega)\big)\big).$$

**Constants:** The ideal-gas adiabatic index is $\gamma = 1.4$. The interface uses $K = 10$ sinusoidal perturbation modes, $\epsilon > 0$, and random variables $a_j, b_j \sim \mathcal{U}[0, 1]$. The amplitudes $a_j$ are normalized such that $\sum_j a_j = 1$.

**Numerical scheme:** Approximate solutions are generated using the FISH hydrodynamics code, which implements high-resolution finite-volume shock-capturing schemes for compressible flow.

**Data specifications:** The full PDEGym dataset contains $N = 1{,}260$ trajectories with a train/validation/test split of 1030/100/130. Each trajectory contains 21 evenly spaced snapshots in time. The state variables are density, velocity, and pressure, i.e., $(\rho, \mathbf{v}, p)$. Evaluation is performed at the 14th time step, corresponding to $t = 1.4$. In our controlled ablation experiments, we fine-tune on 128 trajectories and evaluate rollout accuracy on held-out trajectories.

**Category:** Fine-tuning.

### A.4 Datasets used for applications

This section provides additional details on the datasets used for the physical-system application studies in Sec. 9. Unlike the pretraining and benchmark fine-tuning datasets described in the previous sections, these datasets are drawn from application-driven physical settings and are used to evaluate the ability of MORPH to transfer to heterogeneous forward and inverse problems. They include discontinuity-dominated material dynamics, fracture and failure evolution, inertial-confinement-fusion diagnostics, ultrasonic-guided-wave inverse characterization, experimental guided-wave damage detection, and sparse geophysical field reconstruction. The datasets differ substantially in their dimensionality, data modality, governing physics, measurement structure, and target variable, thereby providing a broad test of the modality-agnostic design of MORPH.

#### A.4.1 Perturbed layered interface (PLI)

The perturbed layered interface (PLI) dataset is used to evaluate transfer to shock-driven, multi-material dynamics under extreme loading. The dataset consists of two-dimensional axisymmetric simulations of high-explosive-driven shock propagation through complex targets. Each simulation describes the interaction of shocks with layered material interfaces and contains several strongly coupled physical processes, including shock propagation, momentum transfer, plastic deformation, thermal effects, and interface-driven morphology changes. This makes PLI substantially different from the relatively smooth or fluid-centric systems commonly used to pretrain PDE foundation models.

The dynamics in PLI are dominated by discontinuities and material interfaces. Shock-interface interactions can trigger Richtmyer–Meshkov-type instabilities, jetting, and late-time interface deformation. Consequently, small variations in the initial geometry can lead to significant changes in the final morphology. The material set is fixed across the dataset and consists of copper, aluminum, stainless steel, a generic polymer, a generic high explosive, and an air background, while the target geometry varies across simulations. This separation between fixed material composition and variable geometric configuration makes the dataset particularly useful for testing whether a model can learn geometry-dependent terminal-state evolution.

Each simulation contains a spatiotemporal trajectory of multi-field state variables, including thermodynamic and kinematic quantities such as density, pressure, temperature, and velocity, together with additional derived mechanics-related channels. The full dataset contains 5293 simulations, each with 38 channels over 100 time steps on a native spatial grid of $1120 \times 400$. One more the representative trajectory from the dataset is shown in Fig. 16. For the application study, we formulate the task as terminal-state prediction rather than autoregressive rollout. Given the first frame of the trajectory, the model predicts the final frame directly:

$$\mathcal{F}_\theta : \mathbf{X}(t_0) \mapsto \mathbf{X}(t_T). \tag{2}$$

Unless otherwise stated, we use the `av-density` channel, corresponding to mixture-averaged density across materials. The dataset is split into train, validation, and test subsets using an 80/10/10 ratio. This first-frame-to-final-frame setting is intentionally challenging because it requires the model to learn a long-horizon map without supervision from intermediate states.

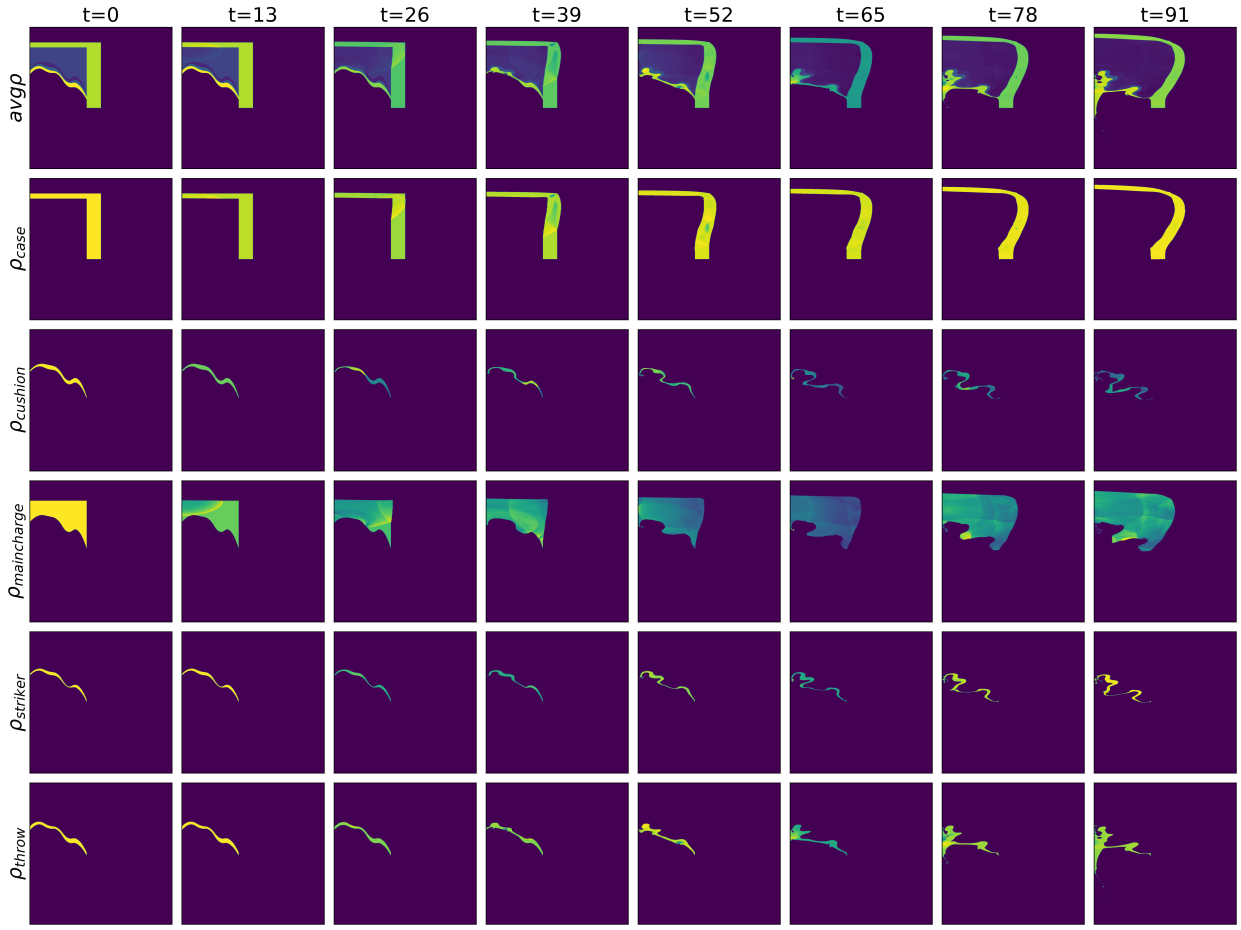



Figure 16: Representative PLI spatiotemporal evolution over uniformly spaced time-steps. Rows visualize selected per-material density channels.

### A.4.2 Dynamic fracture and failure evolution (FRAC)

The FRAC dataset is used to evaluate transfer to dynamic fracture and failure evolution in solid mechanics. Whereas PLI emphasizes shock-material-interface dynamics, FRAC focuses on fracture initiation, crack propagation, branching, coalescence, and damage localization. These processes are strongly nonlinear and path dependent: the late-time fracture topology depends on the initial crack network, the applied loading, material response, and the interaction of multiple moving crack fronts.

The complete FRAC dataset contains simulations generated using two families of solvers: phase-field fracture models and finite-discrete element method formulations such as HOSS/FDEM. These solvers provide complementary structured and unstructured representations of fracture processes. The dataset spans multiple material classes, including brittle and ductile responses, and includes different boundary and loading configurations such as uniaxial and biaxial tension. Randomized initial fracture patterns are used to generate diverse crack-network evolution. As a result, FRAC provides a stringent benchmark for learning discontinuity-dominated solid-mechanics fields, where the target contains thin crack branches, localized damage zones, and junctions formed through crack interaction.

In this work, we use the tungsten subset under the horizontal boundary-condition setting. This subset contains approximately $2 \times 10^5$ simulations with variable numbers of time steps across trajectories. One of the representative trajectory is shown in Fig. 17. To make the task consistent with the PLI application, each

trajectory is reduced to an initial-state/final-state pair:

$$\mathcal{F}_\theta : \mathbf{X}(t_0) \mapsto \mathbf{X}(t_T). \tag{3}$$

The dataset is partitioned into train, validation, and test sets using an 80/10/10 split. The task evaluates whether MORPH can represent long-horizon fracture evolution even though such crack-tip-dominated dynamics are far from the fluid-oriented systems used during pretraining.

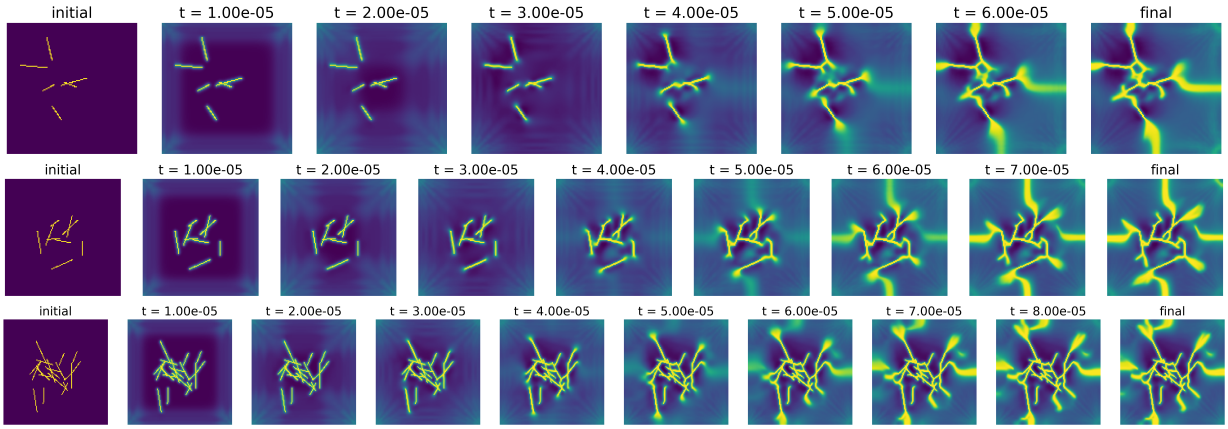



Figure 17: Representative FRAC trajectories (tungsten subset) showing the evolution of fracture/damage patterns from the initial condition to the terminal state. Each row corresponds to a distinct simulation, with frames annotated by characteristic time.

### A.4.3 Inertial-confinement-fusion diagnostics (ICF)

The ICF dataset is used for inverse system-parameter estimation from multimodal diagnostic observations. The data come from the open JAG benchmark, which is generated using a one-dimensional semi-analytic inertial-confinement-fusion simulator. Each simulation represents a capsule implosion configuration and produces synthetic diagnostic measurements. The inverse problem is to recover latent simulator/design parameters from diagnostic signatures rather than to predict a future physical field.

Each simulation is specified by a five-dimensional vector of design or physics parameters. The simulator output contains two complementary observation modalities. The first modality is a set of four spectrally resolved X-ray images, corresponding to four energy bands, each sampled on a $64 \times 64$ grid:

$$\mathbf{y}_{\text{img}} \in \mathbb{R}^{64\times 64\times 4}. \tag{4}$$

The second modality consists of 15 scalar diagnostic observables:

$$\mathbf{y}_{\text{sc}} \in \mathbb{R}^{15}, \tag{5}$$

including quantities such as yield, ion temperature, pressure, and related integrated diagnostics. The full dataset contains 10,000 simulated samples and defines a forward simulator map

$$f : \mathbf{x} \mapsto \mathbf{y} = (\mathbf{y}_{\text{img}}, \mathbf{y}_{\text{sc}}), \qquad \mathbf{x} \in \mathbb{R}^5. \tag{6}$$

For the MORPH application, the task is formulated as the inverse map

$$\mathcal{G}_\theta : (\mathbf{y}_{\text{img}}, \mathbf{y}_{\text{sc}}) \mapsto \mathbf{x}. \tag{7}$$

A data-driven sensitivity analysis is first used to examine parameter identifiability. The image modality is compressed using principal component analysis and combined with the scalar diagnostics in a ridge-regression analysis. This analysis shows that some parameters are weakly constrained by the available observables, motivating a reduced inverse task focused on the identifiable subset of three parameters. The dataset is randomly split into train, validation, and test subsets with an 80/10/10 ratio. This benchmark is useful for testing multimodal inverse reasoning because spatially resolved hyperspectral images and low-dimensional scalar diagnostics must be fused to infer latent physical parameters.

### A.4.4 Polar group-velocity representations (PGVR)

The PGVR dataset is used for inverse characterization of laminated composite materials from ultrasonic guided-wave representations. The dataset is synthetic and is generated using the stiffness matrix method (SMM) together with group-velocity calculations for Lamb-wave propagation in anisotropic composite plates. The forward model maps material properties and laminate layup information to the polar group-velocity responses of the two fundamental Lamb-wave modes: the symmetric $S_0$ mode and the antisymmetric $A_0$ mode.

The material system is modeled as a transversely isotropic carbon-fiber-reinforced polymer lamina. Each material realization is described by six parameters:

$$\boldsymbol{\theta} = (\rho, E_1, E_2, G_{12}, \nu_{12}, \nu_{23}), \tag{8}$$

where $\rho$ is density, $E_1$ and $E_2$ are the Young's moduli along the principal material directions, $G_{12}$ is the in-plane shear modulus, and $\nu_{12}$ and $\nu_{23}$ are Poisson's ratios. Two material sets are constructed. The first consists of commercially available CFRP materials, while the second is generated by uniformly sampling the six material properties over physically motivated ranges:

$$\begin{aligned} \rho &\in [1304, 1760]\ \text{kg/m}^3, \\ E_1 &\in [115, 184]\ \text{GPa}, \\ E_2 &\in [6, 14]\ \text{GPa}, \\ G_{12} &\in [3, 9]\ \text{GPa}, \\ \nu_{12} &\in [0.20, 0.52], \\ \nu_{23} &\in [0.23, 0.59]. \end{aligned} \tag{9}$$

For each material realization, the SMM forward model computes polar group velocities over propagation angles from 0° to 360° and excitation frequencies from 20 to 200 kHz in increments of 20 kHz. The frequency range is selected so that the fundamental Lamb modes remain the primary modes of interest for a 2 mm-thick laminate, thereby avoiding additional complexity from higher-order modes. The calculations are performed for 16-layer symmetric laminates and three layup classes: unidirectional, cross-ply, and quasi-isotropic. The resulting polar group-velocity curves are rasterized into binary images, called polar group-velocity representations.

The original PGVR study considers two related inverse problems: layup-sequence classification and material-property regression. For layup classification, the dataset contains 100 samples per layup class for each Lamb-wave branch, resulting in 300 samples overall. For material-property regression, a larger dataset is generated with 1000 randomly sampled materials evaluated at 10 excitation frequencies, yielding 10,000 samples per branch. In the MORPH application, the paired $S_0$ and $A_0$ polar representations are used as inputs, and the six material properties are treated as regression targets:

$$\mathcal{G}_\theta : (\mathbf{Y}_{S_0}, \mathbf{Y}_{A_0}) \mapsto (\rho, E_1, E_2, G_{12}, \nu_{12}, \nu_{23}). \tag{10}$$

This dataset is a controlled benchmark for evaluating whether MORPH can learn inverse relationships in wave-propagation data, where anisotropy, dispersion, multimodal wave behavior, and directional dependence are encoded geometrically in the polar representations.

### A.4.5 UoNCT-GW

The UoNCT-GW dataset is an experimental ultrasonic-guided-wave dataset collected on a representative aerospace composite wing panel. It is used for guided-wave-based damage detection, with the goal of identifying delamination from measured piezoelectric transducer signals. Unlike PGVR, which is generated from a numerical forward model, UoNCT-GW consists of experimental time traces and therefore includes measurement noise, structural reflections, actuator–sensor path variability, and damage-path interaction effects.

The specimen is a tapered composite wing panel with three bays of different thicknesses: 10 mm, 8 mm, and 6 mm. Each bay is instrumented with 13 piezoelectric transducers arranged in a circular array. The laminate

stacking sequence is

$$[5H/B_{45}/U/B/U/B_{45}/U/B]_s, \tag{11}$$

where $U$ denotes unidirectional 0° plies, $B$ denotes bidirectional 0°/90° plies, $B_{45}$ denotes ±45° bidirectional plies, and $5H$ denotes five-harness plies. Delamination damage is introduced by drop-weight impact using a 1-inch tip. The three bays are impacted with energies of 85, 110, and 150 J, respectively, and the presence of delamination is verified using C-scans.

The MORPH application uses the 6 mm bay. The panel is excited in a round-robin pitch–catch configuration using a 4.5-cycle toneburst with a center frequency of 60 kHz and an amplitude of 80 V peak-to-peak. Signals are recorded before and after impact, and the experiments are repeated ten times to ensure measurement fidelity. This produces 1560 baseline signals and 1560 damaged signals. Each signal corresponds to a specific actuator–sensor path and contains dispersive guided-wave packets, boundary reflections, and path-dependent damage interactions.

Prior work using this dataset converted the time traces into continuous-wavelet-transform representations. The CWT representation highlights time–frequency signatures of damage: damaged signals often show lower amplitude, reduced energy, and phase shifts relative to the baseline signal. However, some actuator–sensor paths interact only weakly with the damaged region, making certain damaged signals resemble healthy ones. This property makes the dataset nontrivial for binary damage detection.

In the MORPH application, the task is formulated as

$$\mathcal{G}_\theta : \mathbf{s}(t) \mapsto z, \qquad z \in \{0, 1\}, \tag{12}$$

where $\mathbf{s}(t)$ is the measured guided-wave signal and $z$ denotes the structural state, with $z = 0$ for baseline and $z = 1$ for damaged. The dataset tests whether a pretrained PDE backbone can transfer to experimental one-dimensional structural-health-monitoring signals that are dispersive, anisotropic, and path dependent.

#### A.4.6 NOAA-SST

The NOAA-SST dataset is used for sparse global field reconstruction from limited local measurements. The benchmark consists of weekly sea-surface-temperature observations collected from satellite and ship-based measurements. The data are provided on a global longitude–latitude grid with spatial resolution $1^\circ \times 1^\circ$, corresponding to a $360 \times 180$ grid. The training set contains 1040 weekly snapshots spanning 1981–2001, while the test snapshots are taken from 2001–2018. Sensors are sampled only over water locations, and land points are excluded using a valid-ocean mask.

The original reconstruction formulation addresses the practical problem that sparse sensors may be arbitrarily placed, may move, and may come online or offline over time. To handle arbitrary sensor locations with a structured-grid model, local sensor measurements are projected onto the full spatial domain using Voronoi tessellation. The input consists of two images:

$$\mathbf{X} = (\mathbf{S}_V, \mathbf{M}), \tag{13}$$

where $\mathbf{S}_V$ is the Voronoi-transformed sensor field and $\mathbf{M}$ is a binary mask indicating sensor locations. The reconstruction target is the dense sea-surface-temperature field over valid ocean points:

$$\mathcal{F}_\theta : (\mathbf{S}_V, \mathbf{M}) \mapsto \mathbf{Y}_{\mathrm{SST}}. \tag{14}$$

This dataset evaluates whether MORPH can recover coherent global geophysical structure from extremely sparse and irregular observations.

## B Appendix: Existing SOTA Models

To compare MORPH against recent PDE foundation models, we fine-tune two Poseidon variants (Poseidon-T (21M) and Poseidon-B (158M) (Herde et al., 2024)) and two DPOT variants (DPOT-S (30M) and DPOT-M (122M) (Hao et al., 2024)). Single-step prediction results are presented in Tables 4.

We use the official repositories: Poseidon `camlab-ethz/poseidon` at commit `b8fa28f` (retrieved 2025-08-01), and DPOT `HaoZhongkai/DPOT` at commit `dcd2f9a` (retrieved 2025-08-04), with default settings unless noted.

**DPOT.** DPOT is pretrained with a 10-step temporal context at 128×128 resolution. To adapt DPOT for single-step $t \to t{+}1$ prediction (Table 4), we follow the input-padding strategy analogous to Herde et al. (2024), feeding 10 copies of the state at time $t$ to satisfy the context window:

$$[\underbrace{x_t, \ldots, x_t}_{10 \text{ copies}}] \longmapsto \widehat{x}_{t+1}.$$

For rollouts (Table 5), when predicting early steps $k{+}1 \leq 10$, we left-pad the available prefix $[x_0, \ldots, x_k]$ with $x_0$ to length 10; e.g., to predict $x_4$,

$$[\mathrm{ts}_0, \mathrm{ts}_0, \mathrm{ts}_0, \mathrm{ts}_0, \mathrm{ts}_0, \mathrm{ts}_0, \mathrm{ts}_0, \mathrm{ts}_1, \mathrm{ts}_2, \mathrm{ts}_3] \longmapsto \widehat{\mathrm{ts}}_4.$$

For $k{+}1 > 10$, DPOT consumes the most recent 10 ground-truth states during fine-tuning and the most recent 10 predicted states during rollout evaluation. We use DPOT's repository defaults (AdamW with a One-Cycle learning-rate schedule; no input normalization), and fine-tune with base learning rate $1 \times 10^{-3}$ and batch size 32.

**Poseidon.** Poseidon takes a single input frame and a selectable $\Delta t$ and predicts $t{+}\Delta t$. For the single-step experiments (Table 4), we fine-tune Poseidon with $\Delta t{=}1$ on all $t \to t{+}1$ pairs in the training set. Results shown for Poseidon in Tables 5 and 7 are that of Poseidon's direct (non-autoregressive) prediction, where the model directly predicts every time-step in the trajectory without feeding any predictions back into the model. 2D-CFD data in Table 4 is of $(512 \times 512)$ resolution. To align with Poseidon's pretrained weights, we downsample these images to $128 \times 128$ with area averaging, run inference/training at $128 \times 128$, then upsample back to $(512 \times 512)$ with bilinear interpolation for evaluation against ground truth. All Poseidon models were fine-tuned with AdamW (LR $1 \times 10^{-4}$), batch size 40, cosine LR schedule, and gradient clipping (max grad norm = 5.0).

**Poseidon for key-frame prediction task.** POSEIDON-T (≈21M parameters) checkpoint and fine-tune it for terminal-state prediction by conditioning on the final lead time. We use the default patch-size of $4 \times 4$. The reference POSEIDON-T implementation expects inputs in a fixed tensor format of size $4 \times 128 \times 128$ (channels × height × width). To accommodate PLI, where each example is provided as a single-channel field at native resolution $1 \times 1120 \times 400$, we replicate the channel dimension to obtain $4 \times 1120 \times 400$. We find that this setting yields better results than the alternative options. We then downsample to $4 \times 128 \times 128$ using bilinear interpolation on-the-fly. During evaluation, we upsample the predicted terminal field back to the original $1120 \times 400$ grid and compute the test MSE at the native resolution. For FRAC, inputs are already $1 \times 128 \times 128$, therefore, we only replicate the single channel to four channels $(4 \times 128 \times 128)$ and apply no spatial rescaling. With a $4 \times 4$ patch size, this results in $32 \times 32 = 1{,}024$ input patches per batch.