MW-NET: MULTI-WAVE U-NET WITH CROSS-WAVE LINKS FOR MULTI-SCALE PHYSICAL DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose Multi-Wave Network (MW-Net), a novel deep learning architecture for modeling the temporal evolution of complex, multi-scale physical systems. MW-Net extends the U-Net architecture by stacking multiple encoder–decoder "waves" (U-Net modules). Unlike prior stacked U-Net variants such as SineNet, which restrict skip connections to within each wave, MW-Net introduces skip connections both within and across successive waves at matching spatial resolutions. This design enhances hierarchical representation learning by enabling repeated interactions between feature representations at the same and different spatial scales, supporting progressive refinement of learned dynamics and offering explicit control over network depth through the number of stacked waves. We evaluate MW-Net on a diverse set of physical systems: 2D Kolmogorov fluid turbulence, Hasegawa–Wakatani plasma turbulence, a shallow-water planetary atmosphere model, and buoyant smoke flows (2D and 3D). Across all cases, MW-Net consistently outperforms state-of-the-art baselines and achieves Pareto improvements in the accuracy–computational cost trade-off. While the best-performing baseline varied by task, MW-Net achieved substantially lower errors and up to 3× faster convergence in reaching low-error regimes under fixed learning schedules.

1 INTRODUCTION

Accurate prediction of the temporal evolution of complex, multi-scale physical systems is essential in many domains, including high Reynolds number fluid dynamics (Wilcox et al., 1998), magnetized plasma systems (Hasegawa, 2012), weather forecasting, and atmospheric modeling Lam et al. (2023). The dynamics of these systems are characterized by the emergence of interacting structures across a wide range of spatial scales, manifesting in nonlinear phenomena such as turbulent energy cascades (Kolmogorov, 1962). In such systems, the ratio between the largest and smallest relevant scales can span several orders of magnitude, making high-fidelity numerical simulations computationally expensive or infeasible. Moreover, resolving fast information propagation often necessitates implicit time integration schemes, further increasing computational cost.

This has motivated growing interest in machine learning (ML) approaches for accelerating simulations or building efficient surrogate models. ML can be used in several roles (Wang et al., 2024; Lino et al., 2023): (1) to accelerate traditional solvers by modeling sub-grid physics (Um et al., 2020; Kochkov et al., 2021a; Greif et al., 2023; Belbute-Peres et al., 2020; Beck et al., 2019; Lapeyre et al., 2019; Subel et al., 2021; Obiols-Sales et al., 2020) or learning effective initial conditions (sub, 2025); or (2) to fully replace solvers via learned surrogates (Bhatnagar et al., 2019; Brandstetter et al., 2023; Alkin et al., 2024; Li et al., 2020; Lu et al., 2021; Wang et al., 2021). Surrogate models can be further divided into physics-informed approaches, which incorporate known governing equations into the loss function (Raissi et al., 2019; Karniadakis et al., 2021; Donnelly et al., 2024; Zubov et al., 2021; Zhao et al., 2023; Jin et al., 2021; Eivazi et al., 2022; Li et al., 2024), and purely data-driven models that rely only on observed data (Clavier et al., 2025; Stachenfeld et al., 2022; Wang et al., 2020; Gahr et al., 2024; Lippe et al., 2023; Zhang et al., 2024; Kim et al., 2019).

In this work, we focus on the data-driven setting, motivated by scenarios where the underlying PDE may not be known (e.g., experimental observations) or where low-resolution data may not clearly satisfy the governing equations. While our experiments are conducted on high-fidelity numerical solutions of known PDEs, the data was sampled at relatively large output time steps — a common

practical setup (Stachenfeld et al., 2022) that accelerates surrogate evaluation but leads to violations of the original numerical constraints. This setup emulates aspects of reduced-order modeling and real-world data applications, and supports the development of architectures that do not rely on access to the underlying PDE.

To address these challenges, we introduce Multi-Wave Network (MW-Net), a deep learning architecture designed for efficient modeling of multi-scale physical dynamics. MW-Net builds upon the U-Net architecture by stacking multiple encoder-decoder "waves" (i.e., U-Nets), connected not only within each wave but also across waves via skip connections at matched spatial resolutions This design enables repeated interactions across spatial scales, progressive refinement of learned dynamics, and explicit control over network depth.

Our main contributions are:

- We present MW-Net, a new deep learning architecture for modeling multi-scale dynamics in complex physical systems.
- We evaluate MW-Net on four challenging physical systems — 2D Kolmogorov turbulence, Hasegawa–Wakatani plasma turbulence, buoyant smoke flow (2D and 3D), and a 2D shallow-water planetary atmosphere — demonstrating consistent improvements over strong state-of-the-art (SOTA) baselines. MW-Net achieves substantially lower prediction error and 3× faster convergence compared to best-performing baselines.
- We evaluate not only trajectory prediction but also statistical characteristics of the learned dynamics, specifically for the Hasegawa–Wakatani system, which exhibits chaotic behavior with a Lyapunov time shorter than the output sampling interval.

2 Surrogate Models for Multi Scale Physics

Below we summarize major architectural families used for learned surrogates in multi scale physics, highlighting their mechanisms, strengths, and limitations.

2.1 Fourier Neural Operators (FNOs)

FNOs (Poli et al., 2022; Tran et al., 2023; Helwig et al., 2023; Li et al., 2020; Rahman et al., 2022) use spectral convolution layers, where each layer: (1) applies a Fast Fourier Transform (FFT) to map the field to the frequency domain; (2) Truncates to a subset of Fourier modes and applies learnable complex-valued weights; and (3) uses an inverse FFT to return to the spatial domain.

Strengths: (1) Efficient access to global receptive fields; few spectral modes can capture large scale patterns effectively. (2) Offers a compact basis for smooth fields and long range correlations; complexity per layer scales roughly with the FFT cost (O(NlogN) for N grid points).

Limitations: (1) Frequency domain filtering can act as a low pass bias, making sharp local features (thin filaments, sharp fronts) harder to model such that fine scale fidelity may suffer (Liu et al., 2025; George et al., 2022; Roberts, 2025; Guan et al., 2023). (2) Application to non-periodic domains or complex geometries often requires windowing, or alternative bases (Qin et al., 2025). These limitations motivate architectures that preserve nonlocal coupling while explicitly modeling local interactions.

2.2 Transformer-based models

Vision style transformers (Vaswani et al., 2017; Dosovitskiy et al., 2020) adapted to 2D/3D physics (McCabe et al., 2024; Cachay et al., 2022; Chattopadhyay et al., 2020; Gao et al., 2022; Nguyen et al., 2023; Alkin et al., 2024) partition the domain into patches and embed each as a token. Self attention exchanges information across all tokens.

Strengths: Self-attention naturally captures multi-scale behavior; any patch can attend to any other.

Limitations: (1) Attention cost is quadratic in the number of tokens (area in 2D / volume in 3D), which limits scalability. Physics oriented works propose linear/near linear attention (Li et al., 2023b;

Cao, 2021; Hao et al., 2023; Li et al., 2023a), however these lead to a trade off in accuracy comparable to simpler architectures (e.g., FNO) (Li et al., 2023b). (2) Transformers are often data hungry (Abdel-Aty & Gould, 2022). (3) In periodic domains (common in physical modeling), learnable relative (periodic) encodings (Shaw et al., 2018; Wu et al., 2021) add model complexity. These trade-offs make convolutional approaches attractive for their efficiency and inductive biases.

2.3 Convolutional models

Convolutional models remain a popular choice for surrogates due to simplicity, inductive biases (translation equivariance), parameter efficiency (weight sharing), and linear scaling with the grid size for fixed kernel sizes.

2.3.1 Encode–Process–Decode (ResNet Processors)

An encoder downsamples to a latent grid with increased channel count; a processor operates at fixed spatial resolution (often a deep ResNet (He et al., 2016)) to model dynamics; a decoder upsamples back to the original resolution. This Encode–Process–Decode paradigm (Battaglia et al., 2018; Sanchez-Gonzalez et al., 2018; 2020) is frequently used in fluid and plasma surrogates (Cheng & Zhang, 2021; Stachenfeld et al., 2022; Kim et al., 2019).

Strengths: Simple, efficient, and effective for local modeling at the processor's resolution. Residual connections ease optimization and permit depth.

Limitations: The processor operates at a single resolution; cross scale interactions are learned only indirectly through the encoder/decoder pathway, which can limit fidelity under strong multi scale coupling (e.g., cascades, wave–eddy interactions). To expand receptive fields at constant resolution, dilations are often introduced.

2.3.2 Dilated ResNets (DilResNet)

DilResNet replaces/augments standard convolutions with dilated convolutions to enlarge the receptive field without pooling (Stachenfeld et al., 2022).

Strengths: Captures long range dependencies while preserving the native grid resolution and fine detail.

Limitations: Larger effective kernels substantially increase compute and memory; in practice, DilResNet can require up to an order of magnitude more compute than comparable convolutional models with similar parameter counts (Zhang et al., 2024; Gupta & Brandstetter, 2022; Li et al., 2023b). This motivates multi resolution designs that natively route information across scales.

2.3.3 U-Net and Variants

A U-Net couples a multi scale encoder and decoder via skip connections at matching resolutions, allowing similar scale features from the encoder to inform the decoder directly.

Strengths: Thanks to its strong multi scale inductive bias, computational efficiency, and robust training dynamics, U-Nets (and their many variants) are arguably the most widely adopted baseline for learned surrogates in multi scale physics, often achieving competitive—frequently state of the art—accuracy at a favorable accuracy–cost balance across diverse benchmarks (Zhang et al., 2024; Gupta & Brandstetter, 2022; Ohana et al., 2024). Common variants include Classic U-Net, Attention U-Net, ResUNet (Diakogiannis et al., 2020), and ConvNeXtU-Net (Ohana et al., 2024).

Limitations and remedies: (1) U-Nets perform only a single downsampling and upsampling pass, which limits the number of explicit cross-scale interactions during feature processing. (2) As a time-stepping predictor, a U-Net can exhibit temporal misalignment between encoder and decoder embeddings because skip-connected features correspond to different effective times when predicting the next step. SineNets (Zhang et al., 2024) mitigate this by stacking multiple U-Nets sequentially (Xia & Kulis, 2017; Shah et al., 2018), each advancing by a smaller sub-step to reduce misalignment. However, SineNet's skip connections remain confined within each U-Net wave, and information flows across waves only through composition at the highest-resolution level (Zhang et al., 2024).

Attention-augmented variants further strengthen cross-scale interactions but at the cost of higher training complexity and optimization difficulty.

3 MODELS

We selected baseline models that have demonstrated strong performance as physics surrogates on multiple systems in prior literature (Zhang et al., 2024; Gupta & Brandstetter, 2022; Ohana et al., 2024). Our focus is on U-Net variants, which consistently outperform other architectures across multiple benchmarks, including the two multi-scale systems evaluated in this paper. For instance, in 2D shallow water and buoyant smoke systems (described in Sections 5.1 and 5.2), U-Net variants outperformed Fourier Neural Operator (FNO) models by nearly an order of magnitude in terms of inverse error-to-computational-cost ratio Zhang et al. (2024). While we do not directly benchmark against FNOs and transformer-based models in this study, our evaluation indirectly reflects their performance through comparative analysis on shared systems.

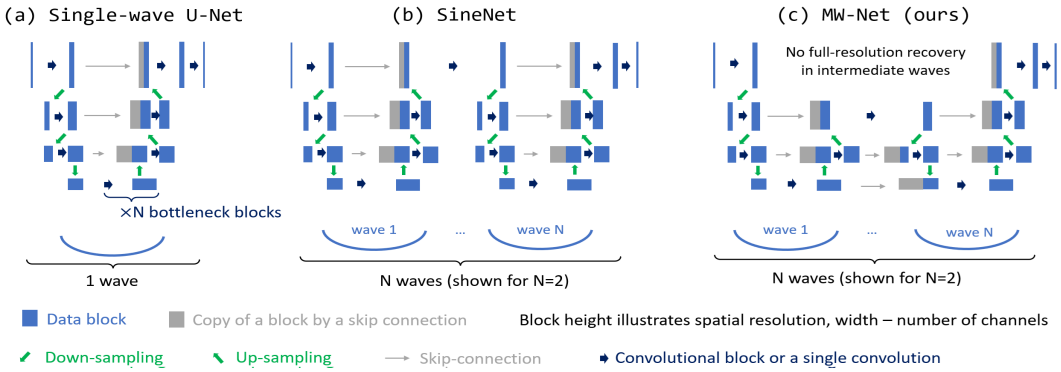

Figure 1: U-Net variants. Detailed description of convolutional blocks is given in Appendix A. The models are illustrated with 4 resolution levels for simplicity (but 5-level variants were used).

3.1 U-NET$_{\text{BASE}}$ (BASE VARIANT)

Our primary baseline is the U-Net architecture adapted from Gupta & Brandstetter (2022) (labeled U-Net$_{\text{base}}$). This model closely resembles the original U-Net (Ronneberger et al., 2015) as implemented in PDEBench Takamoto et al. (2022), with minor modifications inspired by modern variants, which include: (1) group normalization (Wu & He, 2018) instead of batch normalization, (2) enabling of bias parameters in convolutional layers, (3) a reduction of bottleneck block at the lowest resolution level to match the parameter count of the original U-Net (corresponding to N=1 in the Fig. 1a).

Each encoder and decoder level contains a convolutional block with two 3x3 convolutions (stride 1, padding 1), a widely adopted standard in convolutional architectures (Zagoruyko & Komodakis, 2016). At the highest-resolution level, the first convolution expands the number of channels to a specified width, while the final convolution restores it to the original count. Downsampling uses 2x2 max pooling, and upsampling uses transposed convolutions.

3.2 U-NET$_{\text{MOD}}$ (MODERNIZED VARIANT)

U-Net$_{\text{mod}}$ (also adapted from Gupta & Brandstetter (2022)), incorporates several enhancements from modern U-Net designs Ho et al. (2020); Nichol & Dhariwal (2021); Ramesh et al. (2021):

- Residual skip connections within convolutional blocks, similar to Wide ResNet (Zagoruyko & Komodakis, 2016) and ResUNet-a Diakogiannis et al. (2020).
- Learnable downsampling via 2×2 convolutions (stride 2) replacing max pooling.
- An enhanced bottleneck block (corresponding to $N = 3$ in the Fig. 1a).

Although some variants include spatial attention blocks Gupta & Brandstetter (2022), we observed no significant performance gains and encountered training instability and increased computational cost. Therefore, attention-based models were excluded from our baseline comparisons.

3.3 CONVNEXTU-NET (CNU-NET)

CNU-Net (adapted from Ohana et al. (2024) where it showed good performance on several physical systems) is another modern U-Net variant that integrates ConvNext blocks (Liu et al., 2022; Xie et al., 2017), which employ the 'divide and conquer' approach to broaden the spatial receptive fields and semantic representations without increasing computational cost. These blocks stack one channel-wise convolution (Chollet, 2017) with a 7×7 filter and two sequential 1×1 pointwise convolutions. The latter expand the channel dimensions by a factor of 4 and contract them back.

3.4 SINENET

SineNet (Zhang et al., 2024), Fig. 1b, addresses temporal misalignment in U-Net architectures by stacking multiple U-Nets sequentially, each operating at a reduced effective time step. Notably, each U-Net in the stack has internal skip connections, but no skip connections exist between encoder-decoder pairs across U-Nets. Thereby, deep (low-resolution) features are discarded between waves, limiting semantic continuity. Another feature is using average pooling for downsampling.

3.5 MW-NET (OURS)

3.5.1 ARCHITECTURAL MOTIVATION

In most U-Net variants, the network depth—defined as the total number of convolutional layers—is constrained by two factors. (1) The number of resolution levels is typically fixed at five based on empirical success across physical systems. (2) The number of convolutional layers per level is usually fixed at two. This leaves channel count (i.e., network width) as the primary tunable parameter. However, it is well-established that both depth and width must be carefully balanced to optimize performance. As shown in Bengio & LeCun (2007); Larochelle et al. (2007), and supported by circuit complexity theory, shallow networks may require exponentially more units to match the expressiveness of deeper architectures, which scale polynomially. This motivates architectural designs that allow explicit control over depth, especially for multi-scale physical modeling.

3.5.2 INSIGHTS FROM MULTI-SCALE PHYSICAL DYNAMICS

Features of various scales emerging in complex systems interact predominantly locally. Small-scale features that are spatially separated do not interact directly, but can interact indirectly through larger-scale structures that encompass them. This makes U-Nets well-suited for multi-scale systems: it efficiently creates multi-resolution embeddings and captures local interactions. However, it allows only a single pass of cross-scale interaction, limiting the ability to progressively refine representations through network propagation.

3.5.3 IMPROVING COMPUTATIONAL EFFICIENCY

In U-Net architectures (including SineNet), the highest-resolution layers—those with the largest spatial dimensions and lowest channel count—are typically the most computationally expensive. At the same time, these layers often encode less semantic information. MW-Net reduces their usage at intermediate waves, substantially improving efficiency without major sacrifices to accuracy.

3.5.4 DESIGN AND KEY COMPONENTS OF MW-NET

MW-Net, Fig. 1c, is constructed by stacking multiple U-Net modules (waves), each with its own encoder–decoder pair. The key innovation lies in introducing **cross-wave skip connections** at matching resolution levels, enabling features to persist and evolve across waves. This facilitates hierarchical learning, repeated multiscale interactions, and progressive refinement of representations.

To improve computational efficiency, MW-Net omits the highest-resolution layers in intermediate waves—these layers are costly yet contribute less semantic information. **Average pooling** is used for downsampling, and **transposed convolutions** for upsampling. Each convolutional block is independently parameterized, avoiding weight sharing and allowing flexible learning.

MW-Net shares structural similarities with **LadderNet** (Zhuang, 2018), originally proposed for medical image segmentation. Both architectures employ intra-wave and cross-wave skip connections. However, MW-Net introduces several key differences:

- **Aggregation method:** LadderNet uses summation; MW-Net uses channel-wise concatenation, preserving feature diversity.
- **Full-resolution recovery**: is avoided in intermediate waves, reducing cost.
- **Skip connectivity:** MW-Net includes skip connections at all resolution levels, including the deepest layers, which LadderNet omits.
- **Skip placement:** Skip connections are placed before concatenation, enhancing gradient flow and feature reuse.
- **Weight sharing:** MW-Net does not share weights across convolutional blocks, allowing more expressive learning.
- **Pooling:** Similar to SineNet, average pooling is used instead of stride-2 convolutions.

3.6 U-NET$_{\text{DEEP}}$ (DEEPER U-NET)

To isolate the effect of multi-scale interactions, we also implement DeeperU-Net—a single U-Net with deeper convolutional blocks (four layers per block) and internal skip connections. This variant does not use recurrent weight sharing, distinguishing it from R2U-Net (Alom et al., 2018) commonly used in vision applications.

4 BASIS FOR MODEL COMPARISON

4.1 HYPERPARAMETER SELECTION

To ensure fair comparisons, we standardized hyperparameters across models where possible:

- **Resolution levels**: All models use five levels, consistent with prior work (Gupta & Brandstetter, 2022; Zhang et al., 2024; Ohana et al., 2024).
- **Channel expansion**: A fixed ratio of 2 between resolution levels is used throughout.
- **Activation**: GELU activation (Hendrycks & Gimpel, 2016) is used throughout.
- **Boundary conditions (padding)**: consistently with Gupta & Brandstetter (2022), periodic padding was used for periodic domains; zero padding was used for other boundaries.

4.2 ACCURACY VS. COST TRADE-OFF

Rather than comparing best-performing variants alone, we reconstruct the Pareto frontier of accuracy vs. computational cost. Computational cost is defined as training / inference time on a single A100-80GB GPU. Width (channel count) is varied for all models, whereas depth is additionally varied in ablation studies for SineNet and MW-Net.

4.3 TRAINING SETUP

All models were trained using the ADAM optimizer with a custom learning schedule featuring exponential decay with sinusoidal annealing (see Appendix C). Batch size and epoch count were fixed per problem across models. Learning rate scaling was fine-tuned per model.

5 PHYSICAL SYSTEMS AND DATASETS

We evaluate all models on a diverse set of multi-scale physical systems. Here, we provide high-level description of the physical systems, details including data generation and model training strategies are given in Appendix B.

5.1 BUOYANT INCOMPRESSIBLE GAS FLOW WITH SMOKE

This system represents thermal convection of light species, e.g., smoke, in a closed rectangular domain. The flow is governed by the incompressible Navier-Stokes equations augmented with a transport equation for the smoke concentration (assuming pure advection). The flow is driven by the buoyancy force which is proportional to the smoke concentration. 2D and 3D systems were modeled. The datasets adopted from Gupta & Brandstetter (2022) and Li et al. (2023b) respectively, were generated using the ΦFlow solver (Holl et al., 2020). In the 2D case, a 128×128 grid was used and the Reynolds number was 100. In the 3D case, a 64×64x64 grid was used and the Reynolds number was 333. A trajectory example is shown in Figs. 9, 10, and 11 in Appendix.

5.2 SHALLOW-WATER PLANETARY ATMOSPHERE MODEL

The shallow water (SW) equations are derived by depth-integrating the incompressible Navier-Stokes equations (Vreugdenhil, 2013). One of their applications is for modeling planetary atmospheres, predicting evolution of the pressure field (scalar) and wind velocity field (vector). We adopted the dataset from Gupta & Brandstetter (2022) for a model planet which was generated using a modified SpeedyWeather.jl (Klöwer et al., 2022) solver on a cartesian 192 × 96 grid. A trajectory example is shown in Figs. 12, 13, and 14 in Appendix.

5.3 KOLMOGOROV FLOW TURBULENCE

The 2D Kolmogorov flow problem is a common benchmark for studying developed fluid turbulence in periodic domains. The sinusoidal flow of viscous liquid is induced by a unidirectional periodic force and the dynamics is governed by the incompressible Navier-Stokes equations. The dataset adopted from Li et al. (2023b) was generated using a modified pseudo-spectral solver with Re = 1000 and forcing factor $f_0 = 8$. The output resolution and time step were 256x256 and 1/16 respectively. A trajectory example is shown in Fig. 15 in Appendix.

5.4 HASEGAWA–WAKATANI (HW) PLASMA TURBULENCE

Hasegawa-Wakatani (HW) equations describe turbulence of fully-magnetized plasma in nuclear fusion devices. The model assumes that there a gradient in the plasma density transverse to an external uniform magnetic field. The dynamics is formulated for non-dimensional perturbations of plasma (ion) density n and the electric potential ϕ. Periodic boundary conditions are used. We have solved these equations for n and ϕ using the BOUT++ code (Dudson et al., 2009), for $\alpha = 0.01$ and $\kappa = 0.5$. Spatial resolution was 128x128 and the time step was 1. Trajectory example: Figs. 16 and 17.

6 EXPERIMENTS

Following Gupta & Brandstetter (2022); Zhang et al. (2024), all models were trained to predict one future state from a fixed number of past states (concatenated channel-wise), then rolled out autoregressively to generate trajectories (details on the number of time steps are in Appendix B).

Evaluation protocol. Following Li et al. (2023b); Zhang et al. (2024), for smoke, shallow-water, and the Kolmogorov flows, we use the *scaled L2* loss computed per time step and averaged across test trajectories:

$$L_{2,\mathrm{t}}(\hat{u}_t, u_t) = \frac{1}{M}\sum_{k=1}^{M}\frac{\|\hat{u}_t^k - u_t^k\|_2}{\|u_t^k\|_2},$$

where $\hat{u}_t$ is the predicted field at time step t, u_t is the ground truth field at time step t, M is the number of scalar fields, $\hat{u}_t^k$ and u_t^k are the k-th scalar fields of the prediction and ground truth, respectively, $\|\cdot\|_2$ denotes the L2 norm over spatial dimensions. Trajectory examples are presented in Appendix E.

For HW turbulence, where the Lyapunov time (≈ 0.5 (Pedersen et al., 1996)) is shorter than the output interval ($\Delta t = 1$), trajectory errors accumulate quickly (an example of a trajectory is shown in Figs. 16 and 17 in Appendix); thus, we evaluate statistical fidelity over a single 2000-step rollout using (i) time-averaged spatial FFT spectra and (ii) spatially averaged temporal autocorrelations, an example of which is given in Fig. 7 in Appendix. Here we present aggregated errors for these quantities normalized by ground-truth variance:

$$\mathrm{err}(y) = \frac{\langle (y - y_{\text{true}})^2 \rangle}{\mathrm{var}(y_{\text{true}})},$$

where y denotes the parameter of interest (FFT harmonics or autocorrelation), and $\langle \cdot \rangle$ denotes the average over the x-axis in Fig. 7 (k for spectra, time for autocorrelations).

Training details. Loss: scaled L2 loss was used for smoke, SW, Kolmogorov's flow; MSE for HW turbulence. Each model was trained with 3 initializations (6 for HW and Kolmogorov's flow) using fixed seeds $\{2, 12, 22, \dots\}$; we report the best-performing variant over the initializations.

6.1 Pareto Trade-off Between Accuracy and Cost

We vary model width starting from 4 channels at the highest-resolution level, doubling for each bigger model. The upper limit on the model width was dictated by the training budgets capped at 8 h (smoke, shallow-water), 5 h (Kolmogorov), and 2000 s (HW) wall clock time, all on a single A100–80GB GPU.

We benchmark the single-wave U-Net variants against SineNet (2 waves) and MW-Net (3 waves); SineNet was used with fewer waves due to higher computational cost. Limited 3D smoke experiments were run, with U-Net$_{\text{base}}$ being the only baseline.

Smoke, Kolmogorov's flow, and Shallow-Water model:

Figure 2a-d plots final-step relative L_2 error vs. training time for smoke, shallow-water, and Kolmogorov's turbulence. Error vs. inference-time trends are presented in Fig. 5 in Appendix. A clear Pareto frontier emerges: wider models improve accuracy but increase cost. Error drops steeply for narrow widths and saturates for wider ones.

MW-Net-3 consistently outperforms all baselines, reaching lower errors faster and maintaining its advantage as cost grows. The performance of base-lines is case-dependent with no clear winner. Notably, **U-Net$_{\text{deep}}$** performs best on 2D smoke and second-best on shallow-water and Kolmogorov, indicating that the model depth is important for accuracy but a single encoder–decoder pass is insufficient.

Relative to the best non-MW baseline, **MW-Net-3** reduces error by $\sim$30% (shallow-water), $\sim$20% (Kolmogorov), and $\sim$10% (smoke), translating to **2–3**$\times$ less training time to reach equivalent accuracy. These gains persist across model rollouts, starting from the first steps (first-step errors are reported in Fig. 5).

Cross-family comparison for Kolmogorov's flow (using literature data):

We also overlay published per-step rollout errors for the Kolmogorov flow by non-U-Net models (FactFormer, DilResNet, FNO, F-FNO, see Fig. 2f). Notably, U-Net-based models—especially **MW-Net-3**—achieve substantially lower errors at similar or shorter inference times ($<$ 1 s for U-Net-based models vs. $\sim$5 s for DilResNet vs. $\sim$1 s for other literature models). While optimization choices (e.g., batch size) may affect absolute values, they cannot account for the observed order-of-magnitude accuracy gap.

HW turbulence (statistical fidelity):

Figure 2e reports normalized spatial FFT errors for n. Spatial FFT errors for ϕ and temporal autocorrelation errors for both fields are presented in Fig. 6. **MW-Net-3** outperforms other models by

large margins, with almost an order of magnitude improvement for the FFT of n and more than an order-of-magnitude improvement for ϕ. Increasing width does not always help for the HW setup; most models degrade at large widths.

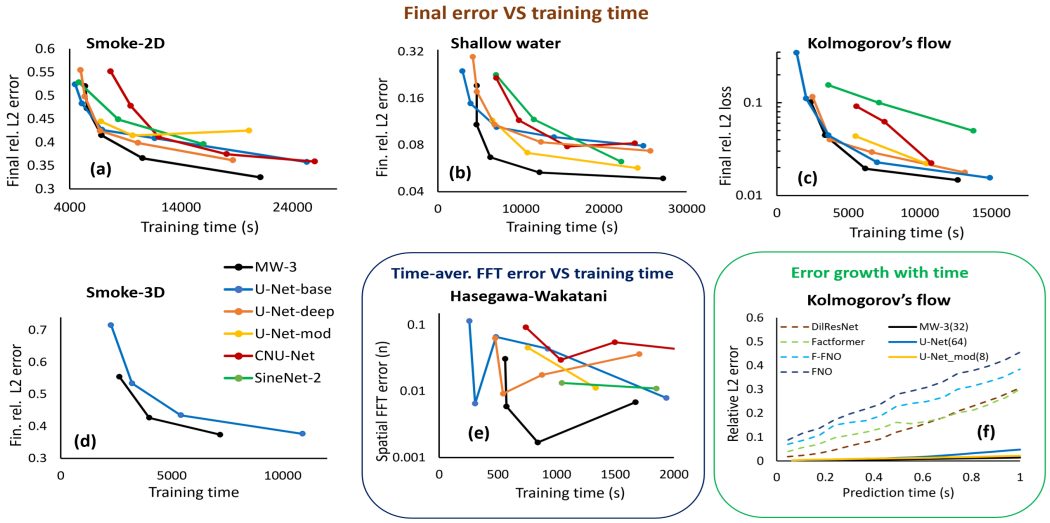

Figure 2: Accuracy vs. training time for all systems + accuracy vs. time step in a roll-out for Kolmogorov's flow.

6.2 Ablation on the number of waves

We conducted ablation studies on the smoke and shallow-water systems to assess the effect of stacking additional U-Net waves. Results in Figure 3 show no noticeable improvement beyond two waves for either MW-Net or SineNet. This suggests that, for these systems, two waves are sufficient to capture the relevant multi-scale dynamics.

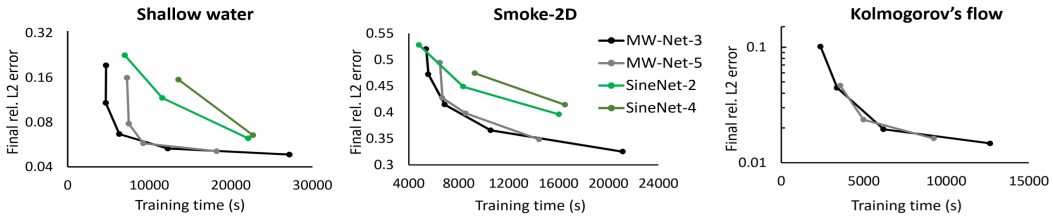

Figure 3: Errors for MW and SineNet models with various numbers of waves

7 Conclusion

We introduced MW-Net, a novel deep learning architecture designed for modeling multi-scale physical dynamics through stacked U-Net modules with cross-wave skip connections. MW-Net enables deeper and more efficient representation learning by facilitating repeated interactions across spatial scales while reducing reliance on high-resolution layers. Across a range of physical systems, including turbulent and transient regimes, MW-Net consistently outperformed strong baselines, achieving lower prediction errors and faster convergence. In particular, MW-Net demonstrated Pareto improvements in the accuracy–cost trade-off, with up to 3× reductions in training time to reach baseline-level accuracy and multifold improvements in statistical fidelity for chaotic systems. These results highlight MW-Net's potential as a robust and scalable surrogate model for complex physical simulations.

8 REPRODUCIBILITY

We provide the implementation of the experiments in the form of an anonymized OSF repository, available at:

`https://osf.io/ch6da/?view_only=8200f9eeeba743a694f4b1a707ed101c`

Please navigate to the 'Files' tab at the top of the screen to access the contents.

The repository includes: instructions in readme.txt, dependencies listed in requirements.txt, code for training models and analyzing results, and the dataset for Hasegawa-Wakatani turbulence.

Each experiment was conducted on a single A100-80GB GPU. A detailed description of the models, datasets, and training procedure can be found in Sections 3-6 of the paper, as well as in Appendices A-C.

9 LIMITATIONS

This work primarily focuses on autoregressive prediction tasks using high-fidelity simulated data. Extending MW-Net to settings involving noisy or experimental data and data assimilation remains an open direction for future research. While limited 3D tests indicate promising scalability, broader evaluation across high-dimensional systems is needed. Additionally, MW-Net has so far been applied only to regular Cartesian grids; adapting the architecture to unstructured meshes via graph-based convolutions (Pfaff et al., 2020; Kurz et al., 2025; Gruber et al., 2022; Grigorev et al., 2023; Fortunato et al., 2022), as demonstrated in U-Net GNNs (Gladstone et al., 2024; Deshpande et al., 2024), could enable broader applicability to complex geometries. Finally, although MW-Net shares structural similarities with architectures used in computer vision, its potential for tasks such as semantic segmentation has not yet been explored.

REFERENCES

Under review. 2025.

Hisham Abdel-Aty and Ian R Gould. Large-scale distributed training of transformers for chemical fingerprinting. *Journal of Chemical Information and Modeling*, 62(20):4852–4862, 2022.

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.

Md Zahangir Alom, Mahmudul Hasan, Chris Yakopcic, Tarek M Taha, and Vijayan K Asari. Recurrent residual convolutional neural network based on u-net (r2u-net) for medical image segmentation. *arXiv preprint arXiv:1802.06955*, 2018.

Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

Andrea Beck, David Flad, and Claus-Dieter Munz. Deep neural networks for data-driven les closure models. *Journal of Computational Physics*, 398:108910, 2019.

Filipe De Avila Belbute-Peres, Thomas Economon, and Zico Kolter. Combining differentiable pde solvers and graph neural networks for fluid flow prediction. In *international conference on machine learning*, pp. 2402–2411. PMLR, 2020.

Yoshua Bengio and Yann LeCun. Scaling learning algorithms towards AI. In *Large Scale Kernel Machines*. MIT Press, 2007.

Saakaar Bhatnagar, Yaser Afshar, Shaowu Pan, Karthik Duraisamy, and Shailendra Kaushik. Prediction of aerodynamic flow fields using convolutional neural networks. *Computational Mechanics*, 64:525–545, 2019.

Johannes Brandstetter, Rianne van den Berg, Max Welling, and Jayesh K Gupta. Clifford neural layers for PDE modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=okwxL_c4x84.

Salva Rühling Cachay, Peetak Mitra, Haruki Hirasawa, Sookyung Kim, Subhashis Hazarika, Dipti Hingmire, Phil Rasch, Hansi Singh, and Kalai Ramea. Climformer-a spherical transformer model for long-term climate projections. In *Proceedings of the Machine Learning and the Phys-ical Sciences Workshop, NeurIPS 2022*, 2022.

Shuhao Cao. Choose a transformer: Fourier or galerkin. *Advances in neural information processing systems*, 34:24924–24940, 2021.

Ashesh Chattopadhyay, Mustafa Mustafa, Pedram Hassanzadeh, and Karthik Kashinath. Deep spatial transformers for autoregressive data-driven forecasting of geophysical turbulence. In *Proceedings of the 10th international conference on climate informatics*, pp. 106–112, 2020.

Chen Cheng and Guang-Tao Zhang. Deep learning method based on physics informed neural network with resnet block for solving fluid flow problems. *Water*, 13(4):423, 2021.

François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251–1258, 2017.

B Clavier, D Zarzoso, D del Castillo-Negrete, and E Frénod. Generative-machine-learning surrogate model of plasma turbulence. *Physical Review E*, 111(1):L013202, 2025.

Saurabh Deshpande, Stéphane PA Bordas, and Jakub Lengiewicz. Magnet: A graph u-net architecture for mesh-based simulations. *Engineering Applications of Artificial Intelligence*, 133:108055, 2024.

Foivos I Diakogiannis, François Waldner, Peter Caccetta, and Chen Wu. Resunet-a: A deep learning framework for semantic segmentation of remotely sensed data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 162:94–114, 2020.

James Donnelly, Alireza Daneshkhah, and Soroush Abolfathi. Physics-informed neural networks as surrogate models of hydrodynamic simulators. *Science of the Total Environment*, 912:168814, 2024.

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.

B.D. Dudson, M.V. Umansky, X.Q. Xu, P.B. Snyder, and H.R. Wilson. Bout++: A framework for parallel plasma fluid simulations. *Computer Physics Communications*, 180(9):1467–1480, September 2009. ISSN 0010-4655. doi: 10.1016/j.cpc.2009.03.008. URL http://dx.doi.org/10.1016/j.cpc.2009.03.008.

Hamidreza Eivazi, Mojtaba Tahani, Philipp Schlatter, and Ricardo Vinuesa. Physics-informed neural networks for solving reynolds-averaged navier–stokes equations. *Physics of Fluids*, 34(7), 2022.

Meire Fortunato, Tobias Pfaff, Peter Wirnsberger, Alexander Pritzel, and Peter Battaglia. Multiscale meshgraphnets. *arXiv preprint arXiv:2210.00612*, 2022.

Constantin Gahr, Ionuţ-Gabriel Farcaş, and Frank Jenko. Scientific machine learning based reduced-order models for plasma turbulence simulations. *Physics of Plasmas*, 31(11), 2024.

Zhihan Gao, Xingjian Shi, Hao Wang, Yi Zhu, Yuyang Bernie Wang, Mu Li, and Dit-Yan Yeung. Earthformer: Exploring space-time transformers for earth system forecasting. *Advances in Neural Information Processing Systems*, 35:25390–25403, 2022.

Robert Joseph George, Jiawei Zhao, Jean Kossaifi, Zongyi Li, and Anima Anandkumar. Incremental spatial and spectral learning of neural operators for solving large-scale pdes. *arXiv preprint arXiv:2211.15188*, 2022.

Rini Jasmine Gladstone, Helia Rahmani, Vishvas Suryakumar, Hadi Meidani, Marta D'Elia, and Ahmad Zareei. Mesh-based gnn surrogates for time-independent pdes. *Scientific reports*, 14(1): 3394, 2024.

Robin Greif, Frank Jenko, and Nils Thuerey. Physics-preserving ai-accelerated simulations of plasma turbulence. *arXiv preprint arXiv:2309.16400*, 2023.

Artur Grigorev, Michael J Black, and Otmar Hilliges. Hood: Hierarchical graphs for generalized modelling of clothing dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16965–16974, 2023.

Anthony Gruber, Max Gunzburger, Lili Ju, and Zhu Wang. A comparison of neural network architectures for data-driven reduced-order modeling. *Computer Methods in Applied Mechanics and Engineering*, 393:114764, 2022.

Steven Guan, Ko-Tsung Hsu, and Parag V Chitnis. Fourier neural operator network for fast photoacoustic wave simulations. *Algorithms*, 16(2):124, 2023.

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

Zhongkai Hao, Zhengyi Wang, Hang Su, Chengyang Ying, Yinpeng Dong, Songming Liu, Ze Cheng, Jian Song, and Jun Zhu. Gnot: A general neural operator transformer for operator learning. In *International Conference on Machine Learning*, pp. 12556–12569. PMLR, 2023.

Akira Hasegawa. *Plasma instabilities and nonlinear effects*, volume 8. Springer Science & Business Media, 2012.

Akira Hasegawa and Masahiro Wakatani. Plasma edge turbulence. *Physical Review Letters*, 50(9): 682, 1983.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Jacob Helwig, Xuan Zhang, Cong Fu, Jerry Kurtin, Stephan Wojtowytsch, and Shuiwang Ji. Group equivariant fourier neural operators for partial differential equations. *arXiv preprint arXiv:2306.05697*, 2023.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

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.

Xiaowei Jin, Shengze Cai, Hui Li, and George Em Karniadakis. Nsfnets (navier-stokes flow nets): Physics-informed neural networks for the incompressible navier-stokes equations. *Journal of Computational Physics*, 426:109951, 2021.

George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

Byungsoo Kim, Vinicius C Azevedo, Nils Thuerey, Theodore Kim, Markus Gross, and Barbara Solenthaler. Deep fluids: A generative network for parameterized fluid simulations. In *Computer graphics forum*, volume 38, pp. 59–70. Wiley Online Library, 2019.

Milan Klöwer, Tom Kimpson, Alistair White, and Mosè Giordano. milankl/speedyweather. jl: v0. 2.1. *Version v0*, 2:181, 2022.

Dmitrii Kochkov, Jamie A. Smith, Ayya Alieva, Qing Wang, Michael P. Brenner, and Stephan Hoyer. Machine learning–accelerated computational fluid dynamics. *Proceedings of the National Academy of Sciences*, 118(21):e2101784118, 2021a. doi: 10.1073/pnas.2101784118. URL https://www.pnas.org/doi/abs/10.1073/pnas.2101784118.

Dmitrii Kochkov, Jamie A Smith, Ayya Alieva, Qing Wang, Michael P Brenner, and Stephan Hoyer. Machine learning–accelerated computational fluid dynamics. *Proceedings of the National Academy of Sciences*, 118(21):e2101784118, 2021b.

Andrey Nikolaevich Kolmogorov. A refinement of previous hypotheses concerning the local structure of turbulence in a viscous incompressible fluid at high reynolds number. *Journal of Fluid Mechanics*, 13(1):82–85, 1962.

Marius Kurz, Andrea Beck, and Benjamin Sanderse. Harnessing equivariance: Modeling turbulence with graph neural networks. *arXiv preprint arXiv:2504.07741*, 2025.

Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, et al. Learning skillful medium-range global weather forecasting. *Science*, 382(6677):1416–1421, 2023.

Corentin J Lapeyre, Antony Misdariis, Nicolas Cazard, Denis Veynante, and Thierry Poinsot. Training convolutional neural networks to estimate turbulent sub-grid scale reaction rates. *Combustion and Flame*, 203:255–264, 2019.

Hugo Larochelle, Dumitru Erhan, Aaron Courville, James Bergstra, and Yoshua Bengio. An empirical evaluation of deep architectures on problems with many factors of variation. In *Proceedings of the 24th international conference on Machine learning*, pp. 473–480, 2007.

Zijie Li, Kazem Meidani, and Amir Barati Farimani. Transformer for partial differential equations' operator learning. *Transactions on Machine Learning Research*, 2023a. ISSN 2835-8856. URL https://openreview.net/forum?id=EPPqt3uERT.

Zijie Li, Dule Shu, and Amir Barati Farimani. Scalable transformer for pde surrogate modeling. *Advances in Neural Information Processing Systems*, 36:28010–28039, 2023b.

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/IMS Journal of Data Science*, 1(3):1–27, 2024.

Mario Lino, Stathi Fotiadis, Anil A Bharath, and Chris D Cantwell. Current and emerging deep-learning methods for the simulation of fluid dynamics. *Proceedings of the Royal Society A*, 479 (2275):20230058, 2023.

Phillip Lippe, Bastiaan S. Veeling, Paris Perdikaris, Richard E Turner, and Johannes Brandstetter. PDE-refiner: Achieving accurate long rollouts with neural PDE solvers. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=Qv6468llWS.

Chaoyu Liu, Davide Murari, Chris Budd, Lihao Liu, and Carola-Bibiane Schönlieb. Enhancing fourier neural operators with local spatial features. *arXiv preprint arXiv:2503.17797*, 2025.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11976–11986, 2022.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Skq89Scxx.

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.

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.

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.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pp. 8162–8171. PMLR, 2021.

Octavi Obiols-Sales, Abhinav Vishnu, Nicholas Malaya, and Aparna Chandramowliswharan. Cfdnet: a deep learning-based accelerator for fluid simulations. In *Proceedings of the 34th ACM International Conference on Supercomputing*, ICS '20, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450379830. doi: 10.1145/3392717.3392772. URL https://doi.org/10.1145/3392717.3392772.

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.

Thomas Sunn Pedersen, Poul K Michelsen, and Jens Juul Rasmussen. Lyapunov exponents and particle dispersion in drift wave turbulence. *Physics of Plasmas*, 3(8):2939–2950, 1996.

Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter Battaglia. Learning mesh-based simulation with graph networks. In *International conference on learning representations*, 2020.

Michael Poli, Stefano Massaroli, Federico Berto, Jinkyoo Park, Tri Dao, Christopher Ré, and Stefano Ermon. Transform once: Efficient operator learning in frequency domain. *Advances in Neural Information Processing Systems*, 35:7947–7959, 2022.

Shaoxiang Qin, Dongxue Zhan, Dingyang Geng, Wenhui Peng, Geng Tian, Yurong Shi, Naiping Gao, Xue Liu, and Liangzhu Leon Wang. Modeling multivariable high-resolution 3d urban microclimate using localized fourier neural operator. *Building and Environment*, 273:112668, 2025.

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

Maziar Raissi, Paris Perdikaris, and George 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.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pp. 8821–8831. Pmlr, 2021.

Gareth G Roberts. Learning how landscapes evolve with neural operators. *Earth Surface Dynamics*, 13(4):563–570, 2025.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pp. 234–241. Springer, 2015.

Alvaro Sanchez-Gonzalez, Nicolas Heess, Jost Tobias Springenberg, Josh Merel, Martin Riedmiller, Raia Hadsell, and Peter Battaglia. Graph networks as learnable physics engines for inference and control. In *International conference on machine learning*, pp. 4470–4479. PMLR, 2018.

Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *International conference on machine learning*, pp. 8459–8468. PMLR, 2020.

Sohil Shah, Pallabi Ghosh, Larry S Davis, and Tom Goldstein. Stacked u-nets: a no-frills approach to natural image segmentation. *arXiv preprint arXiv:1804.10343*, 2018.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*, 2018.

Kim Stachenfeld, Drummond Buschman Fielding, Dmitrii Kochkov, Miles Cranmer, Tobias Pfaff, Jonathan Godwin, Can Cui, Shirley Ho, Peter Battaglia, and Alvaro Sanchez-Gonzalez. Learned coarse models for efficient turbulence simulation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=msRBojTz-Nh.

Adam Subel, Ashesh Chattopadhyay, Yifei Guan, and Pedram Hassanzadeh. Data-driven subgrid-scale modeling of forced burgers turbulence using deep learning with generalization to higher reynolds numbers via transfer learning. *Physics of Fluids*, 33(3), 2021.

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.

Alasdair Tran, Alexander Mathews, Lexing Xie, and Cheng Soon Ong. Factorized fourier neural operators. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=tmIiMPl4IPa.

Kiwon Um, Robert Brand, Yun Raymond Fei, Philipp Holl, and Nils Thuerey. Solver-in-the-loop: Learning from differentiable physics to interact with iterative pde-solvers. *Advances in neural information processing systems*, 33:6111–6122, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Cornelis Boudewijn Vreugdenhil. *Numerical methods for shallow-water flow*, volume 13. Springer Science & Business Media, 2013.

Haixin Wang, Yadi Cao, Zijie Huang, Yuxuan Liu, Peiyan Hu, Xiao Luo, Zezheng Song, Wanjia Zhao, Jilin Liu, Jinan Sun, et al. Recent advances on machine learning for computational fluid dynamics: A survey. *arXiv preprint arXiv:2408.12171*, 2024.

Rui Wang, Karthik Kashinath, Mustafa Mustafa, Adrian Albert, and Rose Yu. Towards physics-informed deep learning for turbulent flow prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, pp. 1457–1466, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450379984. doi: 10.1145/3394486.3403198. URL https://doi.org/10.1145/3394486.3403198.

Sifan Wang, Hanwen Wang, and Paris Perdikaris. Learning the solution operator of parametric partial differential equations with physics-informed deeponets. *Science advances*, 7(40):eabi8605, 2021.

David C Wilcox et al. *Turbulence modeling for CFD*, volume 2. DCW industries La Canada, CA, 1998.

Kan Wu, Houwen Peng, Minghao Chen, Jianlong Fu, and Hongyang Chao. Rethinking and improving relative position encoding for vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10033–10041, 2021.

Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19, 2018.

Xide Xia and Brian Kulis. W-net: A deep model for fully unsupervised image segmentation. *arXiv preprint arXiv:1711.08506*, 2017.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492–1500, 2017.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Xuan Zhang, Jacob Helwig, Yuchao Lin, Yaochen Xie, Cong Fu, Stephan Wojtowytsch, and Shuiwang Ji. Sinenet: Learning temporal dynamics in time-dependent partial differential equations. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=LSYhE2hLWG.

Zhiyuan Zhao, Xueying Ding, and B Aditya Prakash. Pinnsformer: A transformer-based framework for physics-informed neural networks. *arXiv preprint arXiv:2307.11833*, 2023.

Hongkai Zheng. kolmogorov_flow.py, 2023. URL https://github.com/neuraloperator/physics_informed/blob/master/solver/kolmogorov_flow.py.

Juntang Zhuang. Laddernet: Multi-path networks based on u-net for medical image segmentation. *arXiv preprint arXiv:1810.07810*, 2018.

Kirill Zubov, Zoe McCarthy, Yingbo Ma, Francesco Calisto, Valerio Pagliarino, Simone Azeglio, Luca Bottero, Emmanuel Luján, Valentin Sulzer, Ashutosh Bharambe, et al. Neuralpde: Automating physics-informed neural networks (pinns) with error approximations. *arXiv preprint arXiv:2107.09443*, 2021.

A U-NET VARIANTS

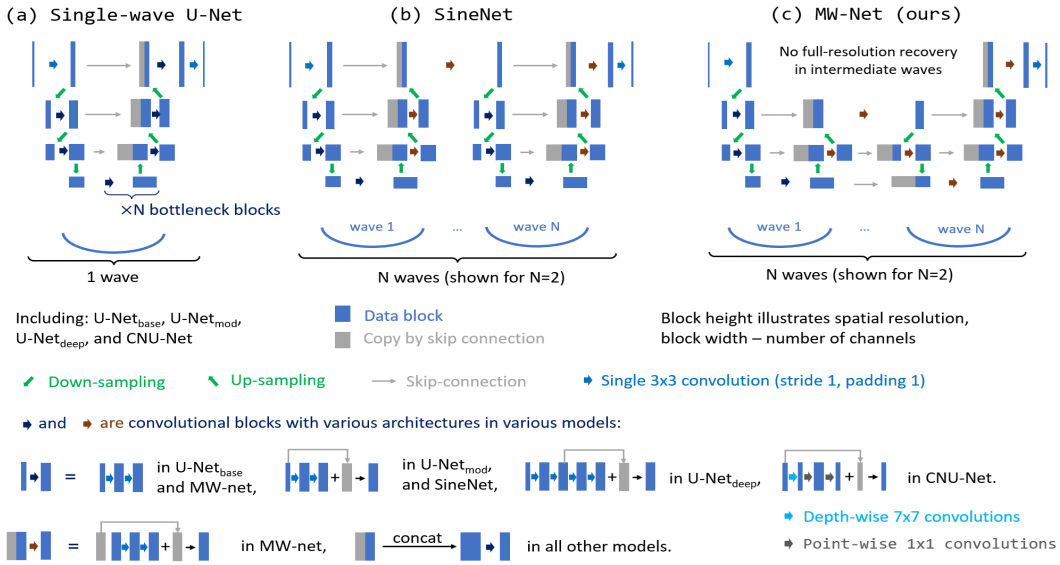

Figure 4: U-Net variants with details of the convolutional blocks.

B DETAILS OF THE SYSTEMS, DATASETS, AND MODEL TRAINING

B.1 THE KOLMOGOROV FLOW

The two-dimensional Kolmogorov flow problem is a benchmark for studying developed fluid turbulence in periodic domains. The sinusoidal flow of viscous liquid is induced by a unidirectional periodic force. The dynamics is governed by the incompressible Navier-Stokes equations in the vorticity form (non-dimensional):

$$\begin{aligned} \frac{\partial \omega(x,t)}{\partial t} + u(x,t)\cdot\nabla\omega(x,t) &= \frac{1}{\mathrm{Re}}\nabla^2\omega(x,t) + f(x), && x\in(0,2\pi)^2,\ t\in(0,T], \\ \nabla\cdot u(x,t) &= 0, && x\in(0,2\pi)^2,\ t\in[0,T], \\ \omega(x,0) &= \omega_0(x), && x\in(0,2\pi)^2. \end{aligned}$$

Here, x and y are spatial coordinates, u is velocity (directed along y), ω is vorticity, Re is the Reynolds number and f represents the driving force along the y-direction (Kochkov et al., 2021b):

$$f(x) = -f_0\cos(f_0 x) - 0.1\,\omega.$$

The dataset adopted from Li et al. (2023b) was generated using a modified pseudo-spectral solver (Zheng, 2023), for $Re = 1000$ and with forcing factor $f_0 = 8$. The dataset comprises 100 trajectories for training and 20 trajectories for testing, where each trajectory has 160 states. Initial condition ω_0 was sampled from a Gaussian random field following Li et al. (2024), ensuring a broad range of spatial scales in the flow.

The models were trained for 32 epochs to predict 1 time step ahead taking a single time step as input. The batch size was 20. Six initializations with fixed seeds were used for each model. The

models were then rolled-out auto-regressively to generate trajectories of 16 time steps on a batch of 10 trajectories with randomly selected starting time frames. Relative L2 loss was used for both training and presenting the results.

The following learning rate scaling factors were used for the models: U-Net$_{\text{base}}$ – 0.25, U-Net$_{\text{mod}}$ – 0.125, CNU-Net – 0.25, U-Net$_{\text{deep}}$ – 1, MW-Net – 0.25, SineNet – 0.25.

B.2 Buoyant Incompressible Gas Flow with Smoke (2D and 3D)

This system represents thermal convection of light species, e.g., smoke, in a closed domain. The flow is governed by the incompressible Navier-Stokes equations which assume that the flow velocity is too low to affect fluid density (Mach number $<< 1$, which is true for thermal convection). The equations are augmented by a transport equation for smoke concentration (assuming pure advection) and are solved in non-dimensional form:

$$
\begin{aligned}
\frac{\partial u(x,t)}{\partial t} + u(x,t)\cdot\nabla u(x,t) &= \frac{1}{\mathrm{Re}}\nabla^2 u(x,t) - \nabla p(x,t) + f(x,t), && x\in(0,L)^n,\ t\in(0,T],\\
\frac{\partial d(x,t)}{\partial t} + u(x,t)\cdot\nabla d(x,t) &= 0, && x\in(0,L)^n,\ t\in(0,T],\\
\nabla\cdot u(x,t) &= 0, && x\in(0,L)^n,\ t\in[0,T],\\
u(x,0) = 0, \quad d(x,0) &= d_0(x), && x\in(0,L)^n.
\end{aligned}
$$

Here, u is the velocity vector, p is the pressure, d is the concentration of light-weight species (e.g., smoke), f is the vertically-directed buoyancy force, which is proportional to the smoke concentration d with a factor 0.5. n denotes the number of spatial dimensions.

Dirichlet boundary conditions are applied to the velocity, and Neumann conditions to the smoke concentration.

B.2.1 2D case

We use the dataset from Gupta & Brandstetter (2022), generated using the ΦFlow solver (Holl et al., 2020) on a 128×128 grid with an output time step of 1.5. The domain size is 32×32, and the Reynolds number is 100. The dataset contains 5,200 training trajectories and 1,300 test trajectories, each with 14 time steps from randomly sampled initial conditions.

Following Gupta & Brandstetter (2022), the models are trained to predict one time step ahead using the previous four time steps (concatenated channel-wise) as input. Models are then rolled out autoregressively to predict time steps 5–14. Each model is trained with 3 fixed-seed initializations (the best-performing realization is used). The error metric was the scaled L2 loss computed per time step, used for both training and evaluation. The models were trained for 80 epochs with batches of 40 time steps. A batch of 30 test trajectories was randomly selected for evaluation. The errors for the first and the last time step are presented, averaged across the test trajectories.

The following learning rate scaling factors were used for the models: U-Net$_{\text{base}}$ – 1, U-Net$_{\text{mod}}$ – 0.25, CNU-Net – 1, U-Net$_{\text{deep}}$ – 1, MW-Net – 1, SineNet – 0.125.

B.2.2 3D case

To assess model scalability, we include a 3D version of the smoke flow system. The dataset, adopted from Li et al. (2023b), was also generated using the ΦFlow solver on a 64×64×64 grid with a time step of 0.75 and Reynolds number of 333. The relative L2 loss was also used for training and evaluation.

The dataset consists of 2,000 training trajectories and 200 test trajectories, each with 20 time steps. Models were trained for 10 epochs with a batch size of 20. 3D convolutions use the same filter

sizes and channel expansion ratio (i.e., ×2) as in 2D. Performance is compared against the U-Net$_{\text{base}}$ baseline.

A learning rate scaling factor of 1 was used for both models.

B.3 Shallow-Water Planetary Atmosphere Model

The shallow water (SW) equations are derived by depth-integrating the incompressible Navier-Stokes equations (Vreugdenhil, 2013). One of their applications is for modeling planetary atmospheres, predicting evolution of the pressure field (scalar) and wind velocity field (vector). We adopted the dataset from Gupta & Brandstetter (2022) for a model planet, generated using a modified SpeedyWeather.jl (Klöwer et al., 2022) solver. A cartesian grid 192 × 96 was used in combination with a fixed output time step of 48 h.

The training and test data consisted of 5,600 and 1,400 trajectories respectively, each trajectory having 11 time steps. The models were trained for 80 epochs to predict one time step ahead (for time steps 3-11) using two previous time steps as input. The batch size was 36. Three initializations with fixed seeds were used for each model. The model was then run autoregressively to predict 9 time steps (3 to 11). A batch of 30 trajectories were randomly selected from the test set to produce the presented results.

The following learning rate scaling factors were used for the models: U-Net$_{\text{base}}$ – 0.25, U-Net$_{\text{mod}}$ – 0.25, CNU-Net – 0.25, U-Net$_{\text{deep}}$ – 1, MW-Net – 1, SineNet – 0.25.

B.4 Hasegawa-Wakatani Plasma Turbulence

The Hasegawa-Wakatani (HW) equations Hasegawa & Wakatani (1983) describe turbulence relevant to fully-magnetized plasma in nuclear fusion devices. The model assumes a gradient in plasma density transverse to an external uniform magnetic field. The equations are formulated for normalized (non-dimensional) perturbations of plasma (ion) density n and electric potential ϕ (n is normalized to the background plasma density, and ϕ is normalized to the electron temperature):

$$\frac{\partial n}{\partial t} + \{\phi, n\} + \kappa \frac{\partial \phi}{\partial y} = \alpha(\phi - n) - D_n \nabla^4 n,$$
$$\frac{\partial}{\partial t}\Delta\phi + \{\phi, \Delta\phi\} = \alpha(\phi - n) - D_p \nabla^4 \phi.$$

Here, x and y are the spatial coordinates (the background density gradient is in the x direction). κ and α are non-dimensional parameters representing the density gradient and plasma adiabaticity. D_n and D_p are hyper-diffusivity parameters added for numerical stability. The Poisson bracket in the HW equations is defined as:

$$\{A, B\} = \frac{\partial A}{\partial x}\frac{\partial B}{\partial y} - \frac{\partial A}{\partial y}\frac{\partial B}{\partial x}.$$

Periodic boundary conditions are used.

We solve these equations for n and ϕ using the BOUT++ code (Dudson et al., 2009), for α = 0.01 and κ = 0.5. The hyper-diffusivity parameters were set to small values, $D_n = D_p = 0.0001$, to ensure numerical stability without affecting the results. Computations were performed on a high-performance computing cluster utilizing eight A100 GPUs. Spatial resolution was 128x128 with a time step of 1. The solver completed the task in approximately 3 hours. A single trajectory was modeled, initiated from white noise. The first 500 time steps corresponded to the warm-up stage, followed by instability growth and saturation. The subsequent 4,300 time steps corresponded to developed (quasi-steady) turbulence. Of those time steps, 4,000 were used for training and 300 for testing.

The models were trained to predict 1 time step ahead using 1 time step as an input. Batch size was 40, with 160 training epochs. The models were then rolled-out auto-regressively to generate

trajectories of 2000 time steps. Since the Lyapunov time for HW turbulence is about 0.5 (Pedersen et al., 1996), i.e., smaller than the output timestep, we compare statistical characteristics of the generated turbulence to the ground truth (no tracing of individual trajectories).

C Learning Schedule

A custom learning rate lr schedule was applied for all models, based on a warm-up stage followed by an exponential decay combined with cosine annealing Loshchilov & Hutter (2017), as determined by the following expression:

$$lr = 0.01 \cdot \alpha \cdot \exp\left(-5\frac{\max(i, N_{\text{warm}}) - N_{\text{warm}}}{N_{\text{total}}}\right) \cdot \left(0.8 + 0.5 \cdot \sin\left(2\pi\left(0.75 + \frac{i}{N_{\text{warm}}}\right)\right)\right).$$

Here, i is the epoch number, N_{total} is the total number of epochs reserved for learning, $N_{\text{warm}} = N_{\text{total}}/2$ corresponds to the linear warm-up stage, α is the scaling factor, which was fine-tuned in the range 0.125 - 1.0 for each model.

D Accuracy vs. Cost Trade-off (Additional Results)

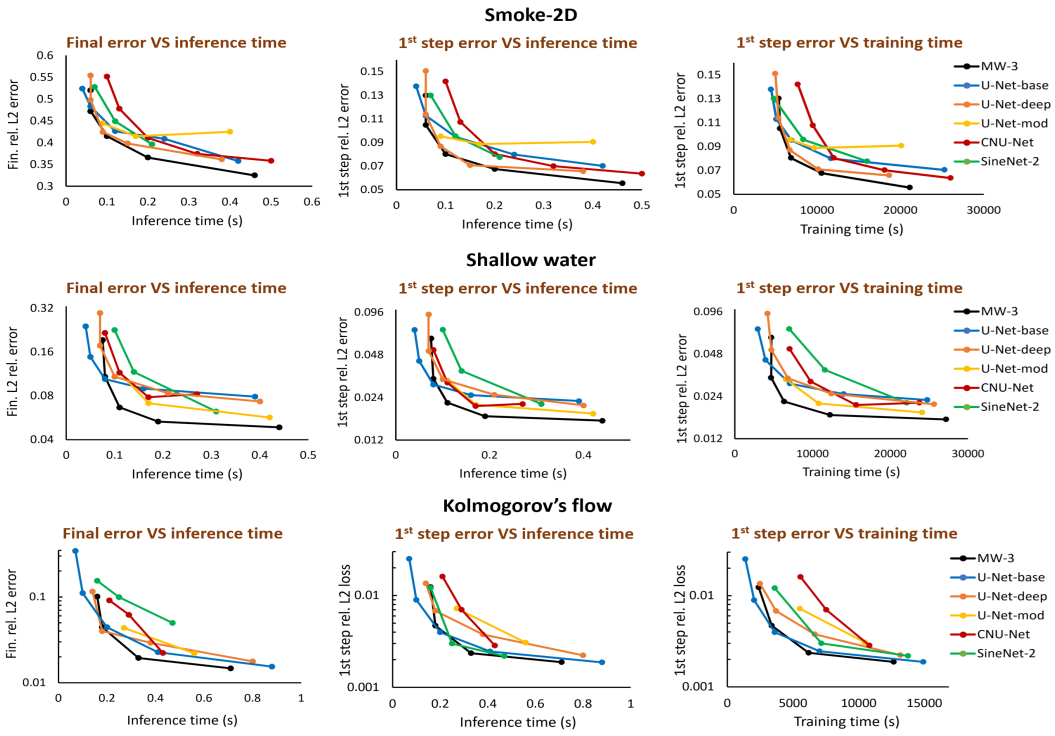

Figure 5: Additional results for the 2D systems where individual trajectories were traced.

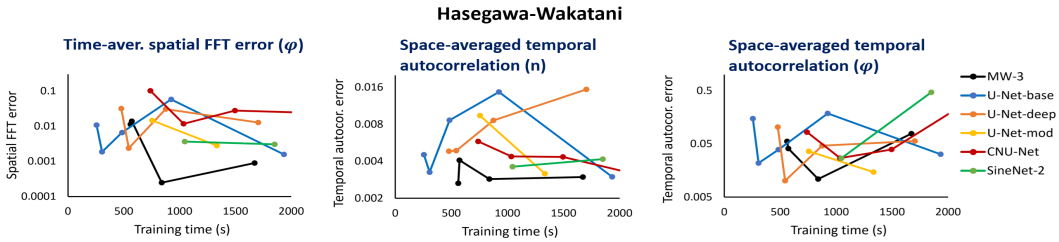

Figure 6: Additional results for the HW system. Examples of the FFT spectra and temporal autocorrelations for which aggregated errors are shown here are given in Fig. 7.

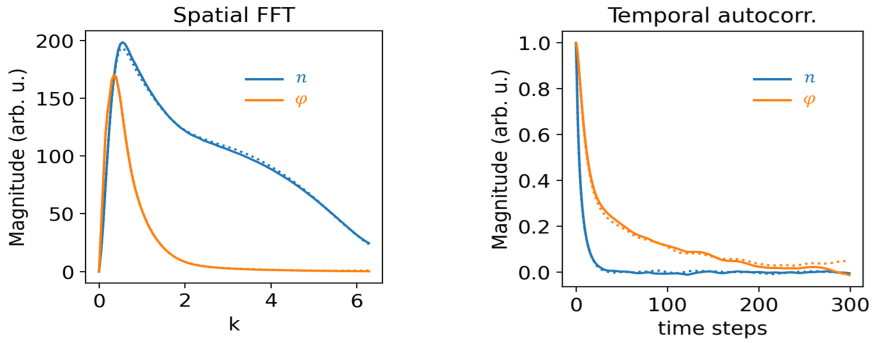

Figure 7: Time-averaged spatial FFT spectra for n and ϕ (left) and spatially-averaged temporal autocorrelation for n and ϕ (right). Solid lines – ground truth (simulation data), dotted lines – results of the best MW-3 model.

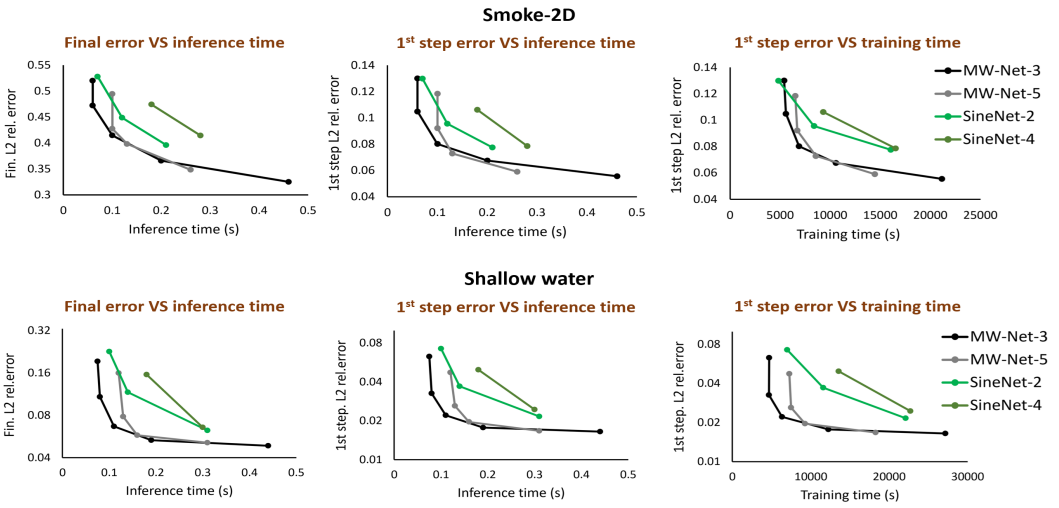

Figure 8: Additional comparison results of MW-Net and SineNet models with more waves on two systems.

E MODEL ROLLOUTS

E.1 BUOYANT FLOW OF SMOKE

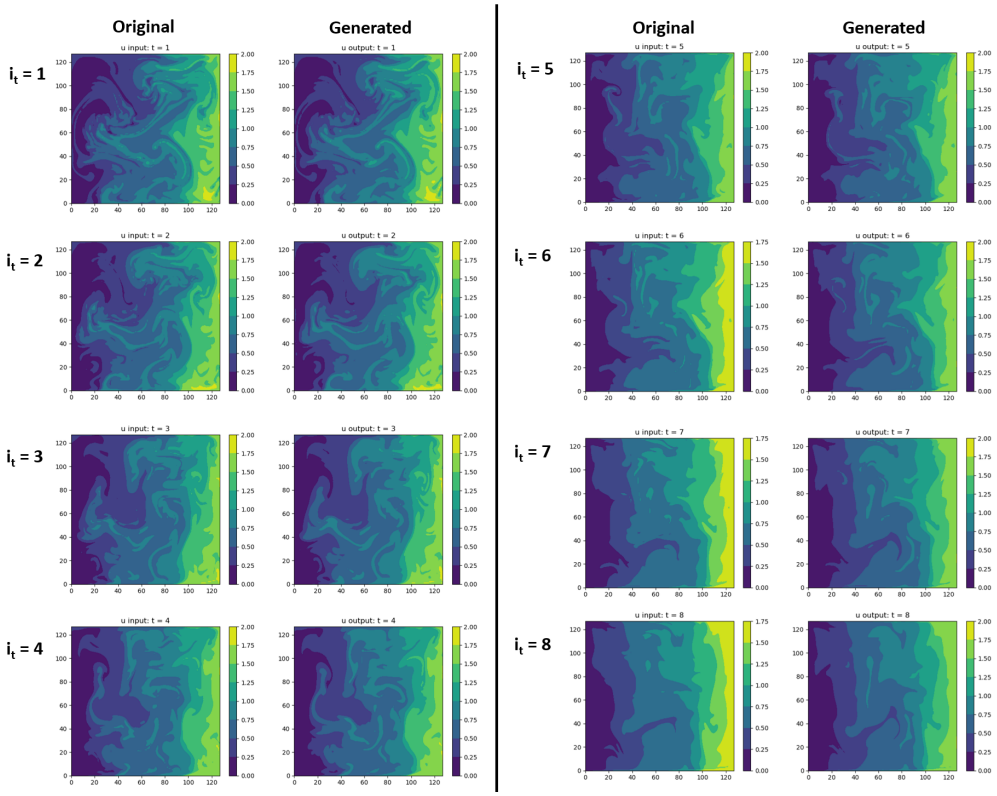

Figure 9: Example of a trajectory for the buoyant smoke flow generated by the MW-Net-3 model (best realization). The field of smoke density.

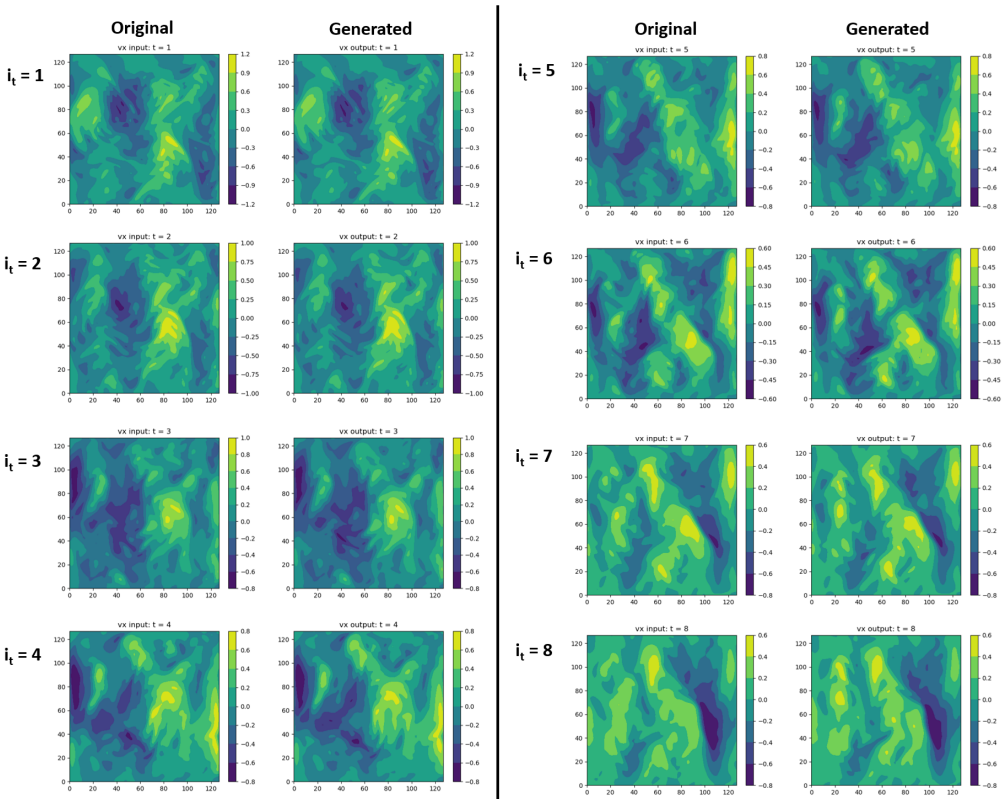

Figure 10: Example of a trajectory for the buoyant smoke flow generated by the MW-Net-3 model (best realization). The field of velocity (x-component).

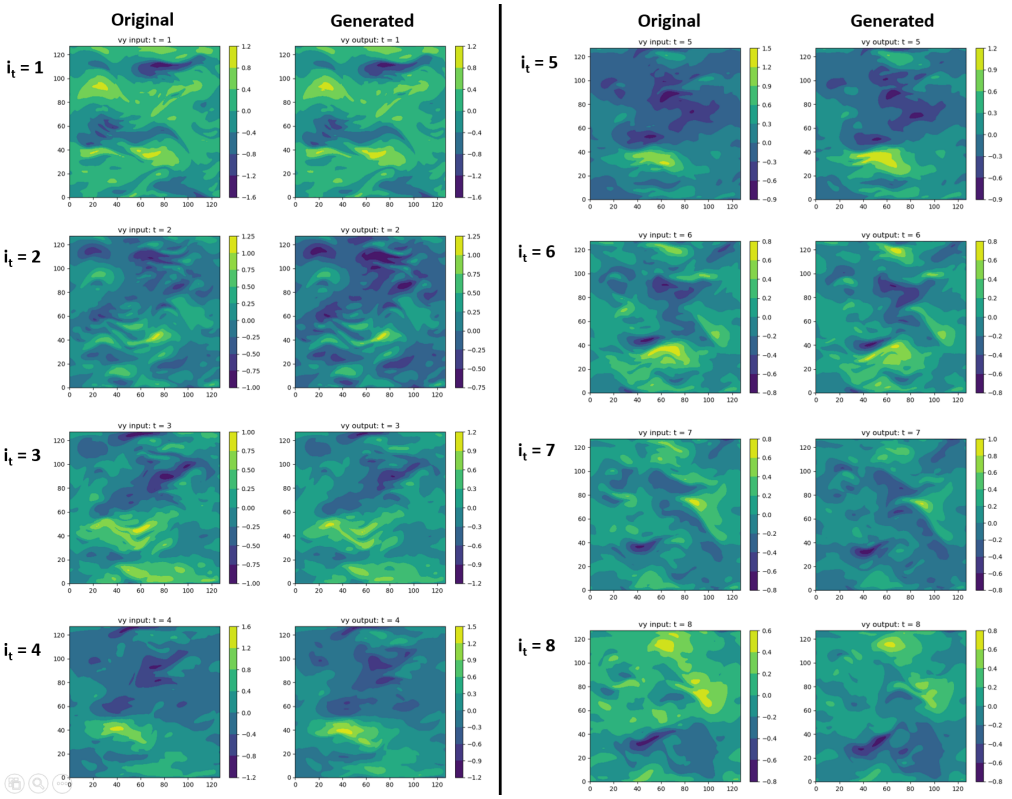

Figure 11: Example of a trajectory for the buoyant smoke flow generated by the MW-Net-3 model (best realization). The field of velocity (y-component).

E.2 THE SHALLOW WATER SYSTEM

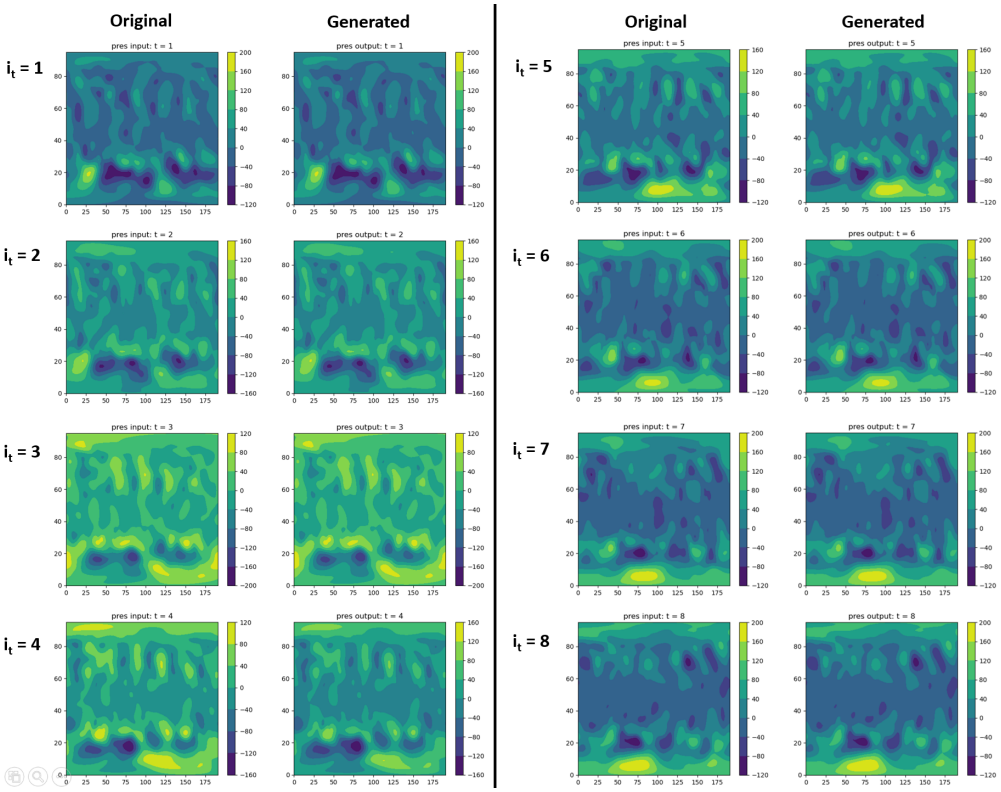

Figure 12: Example of a trajectory for the shallow water system generated by the MW-Net-3 model (best realization). The field of pressure.

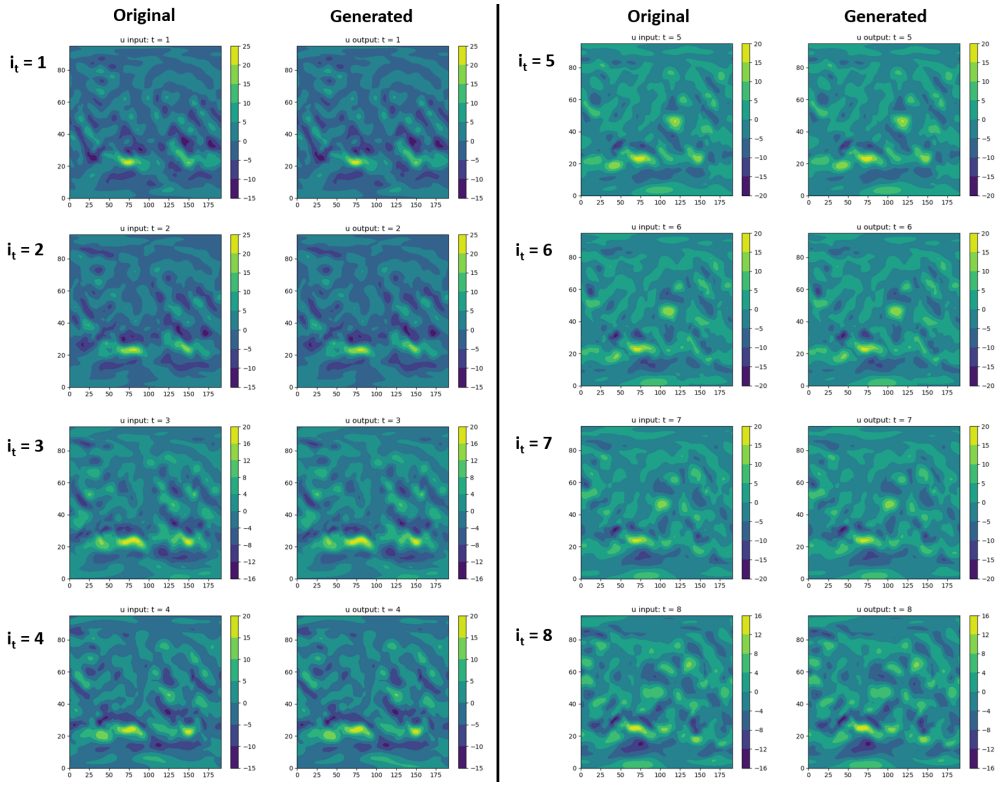

Figure 13: Example of a trajectory for the shallow water system generated by the MW-Net-3 model (best realization). The field of velocity (x-component).

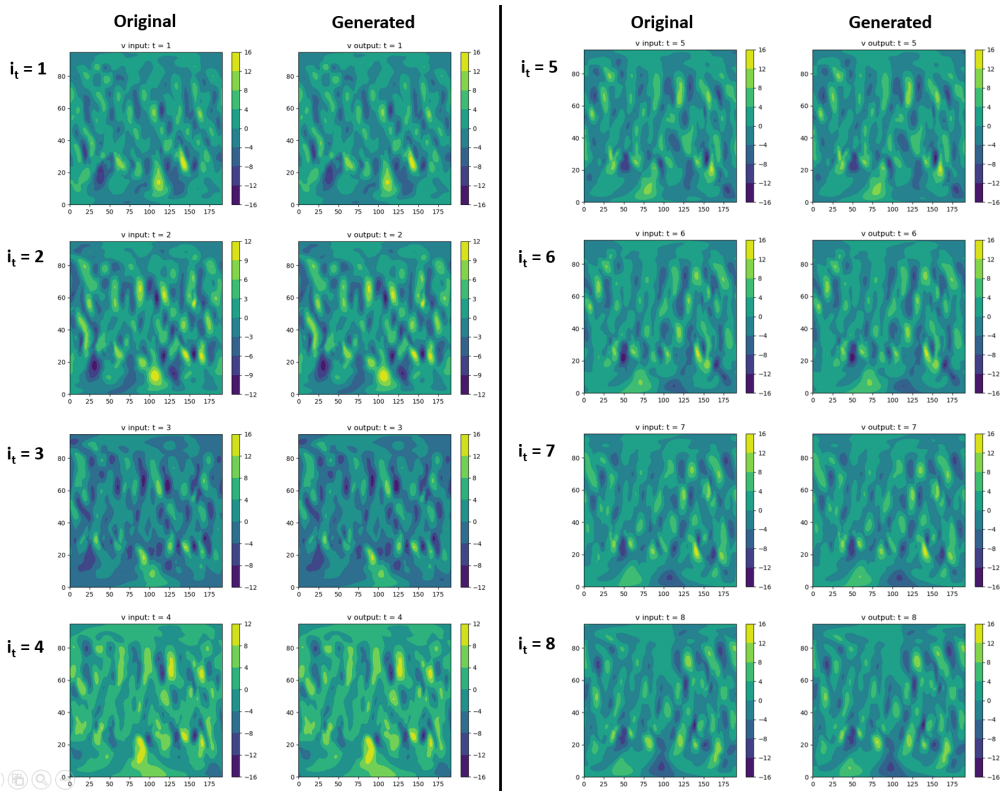

Figure 14: Example of a trajectory for the shallow water system generated by the MW-Net-3 model (best realization). The field of velocity (y-component).

E.3 KOLMOGOROV'S FLOW

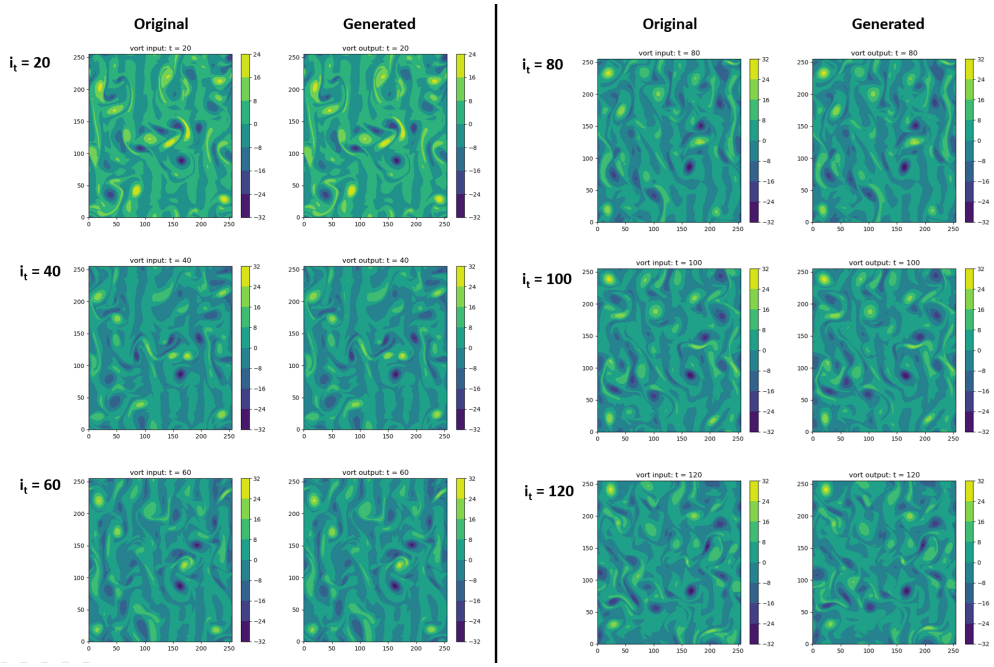

Figure 15: Example of a trajectory for the Kolmogorov turbulence generated by the MW-Net-3 model (best realization). The field of vorticity. Good agreement persists until the end of the trajectory of 120 time steps.

E.4 HASEGAWA-WAKATANI TURBULENCE

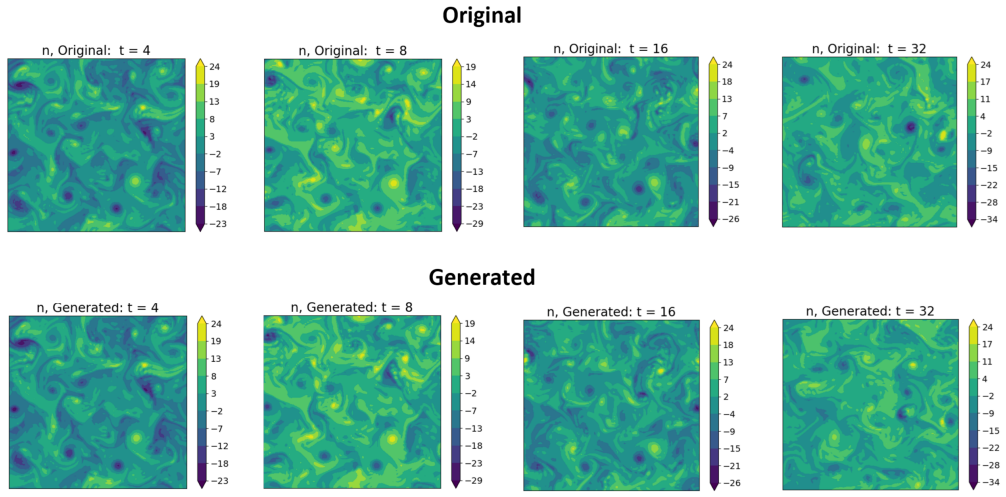

Figure 16: Beginning of a 2000 time step trajectory for HW turbulence generated by the MW-Net-3 model (best realization). The field of n: Generated data - bottom row vs. numerical simulation (using BOUT++) - top row. At first, the generated solution resembles the original (simulated) one quite closely. However, with time, the differences amplify and by the time step 32 become significant.

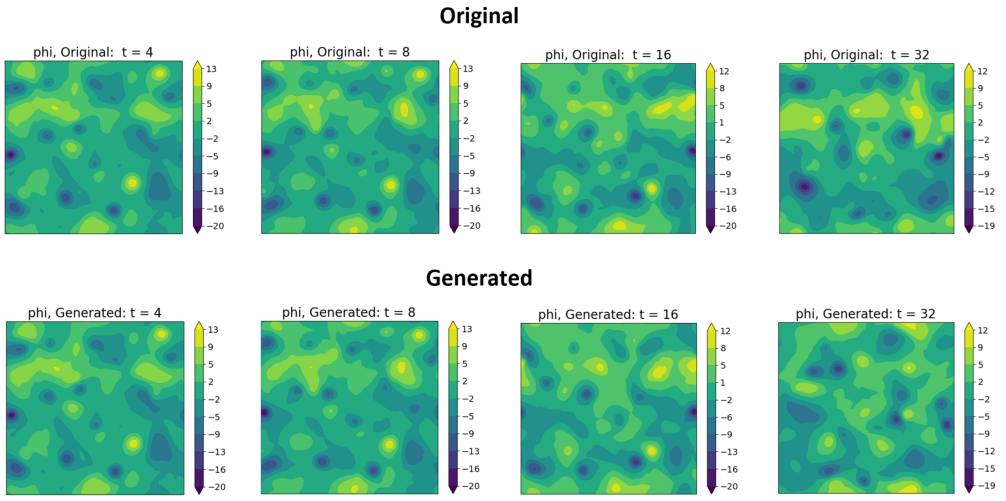

Figure 17: Beginning of a 2000 time step trajectory for HW turbulence generated by the MW-Net-3 model (best realization). The field of phi: Generated data - bottom row vs. numerical simulation (using BOUT++) - top row. At first, the generated solution resembles the original (simulated) one quite closely. However, with time, the differences amplify and by the time step 32 become significant.