Wavelet Flow Matching for Multi-Scale Physics Emulation

Gabriele Accarino[1]* **Juan Nathaniel**[1] **Carla Roesch**[2] **Pierre Gentine**[1,3]
Sara Shamekh[4] **Duncan Watson-Parris**[5] **Viviana Acquaviva**[6,7]

[1]Department of Earth and Environmental Engineering, Columbia University, New York, NY, USA
[2]University of Edinburgh, Edinburgh, Scotland, UK
[3]École Normale Supérieure – PSL, Paris, France
[4]Courant Institute of Mathematical Sciences, New York University, New York, NY, USA
[5]Scripps Institution of Oceanography & Halıcıoğlu Data Science Institute, UC San Diego, La Jolla, CA, USA
[6]CUNY New York City College of Technology, Brooklyn, NY, USA
[7]Lamont-Doherty Earth Observatory, Columbia University, Palisades, NY, USA

Abstract

Accurate emulation of multi-scale physical systems governed by PDEs demands models that remain stable over long autoregressive rollouts while preserving fine-scale structures. Deterministic emulators produce overly smoothed predictions, while generative approaches better capture details but are costly. Latent-space generative models have emerged as a compromise but with the additional cost of separately pre-trained autoencoders. We propose **Wavelet Flow Matching** (WFM), a generative emulator that overcomes current trade-offs between cost and skill by performing optimal-transport flow matching directly in the multi-scale wavelet space. Rather than learning a latent compression, WFM leverages the hierarchical structure of a U-Net to jointly predict the transport velocities of a prescribed, lossless wavelet representation. On three challenging systems of chaotic fluid dynamics, WFM achieves superior spectral coherence and competitive or better long-horizon accuracy compared to state-of-the-art models. Our results position the wavelet space as an effective training-free representation for generative emulation of complex physical dynamics.

1 Introduction

Simulating complex physical systems governed by partial differential equations (PDEs) lies at the heart of scientific computing, with applications spanning turbulence modeling [1], weather and climate prediction [2–4], and astrophysics [5]. Classical numerical solvers are accurate but computationally prohibitive when thousands of trajectories need to be generated [6, 7]. This has motivated a growing body of work on data-driven emulators trained to predict future system states from past ones at a fraction of the cost of high-fidelity simulations [8, 9].

The canonical formulation treats emulation as an autoregressive prediction task: given the current state x^t (and optionally a context window of L past states $x^{t-L:t-1}$), a neural network f_θ is trained to predict the next state x^{t+1}. Deterministic emulators such as variants of Neural Operators (NOs) [8, 10, 11], U-Nets [12], and transformers [13], are trained by regression loss and are extraordinarily fast at inference. However, squared-error training encourages over-smoothed predictions, and, crucially, small errors compound over autoregressive rollouts. At each rollout step, the model receives input that can be out of the data manifold on which it has been trained, progressively drifting from the true

*Corresponding author: ga2673@columbia.edu

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

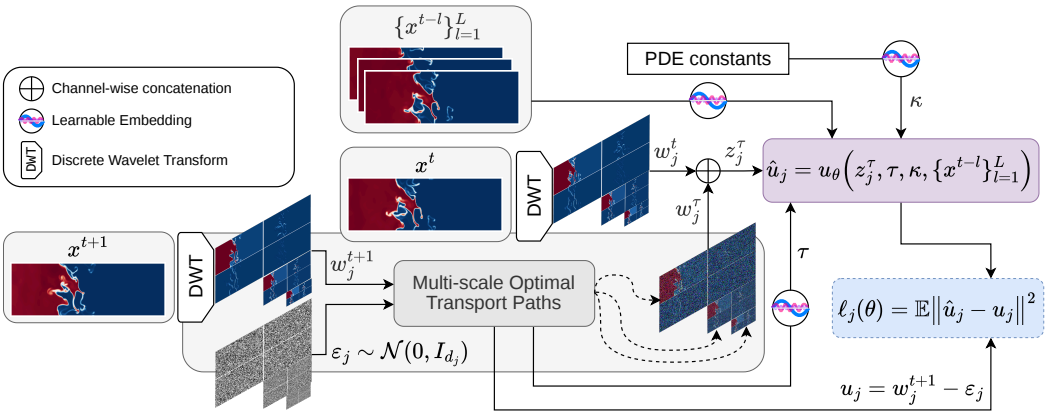


Figure 1: Illustration of the Wavelet Flow Matching architecture. During training, Gaussian noise is sampled to match the shape of wavelet coefficients at each scale. OT-CFM interpolation is applied between the next-state wavelet coefficients w_j^{t+1} and noise at every scale with a shared interpolation time τ, yielding J per-scale flows that are coupled through the shared U-Net. The velocity-predicting U-Net u_θ (purple box) is conditioned on the interpolation step τ, the channel-wise concatenation z_j^τ of w_j^τ and the current-state coefficients w_j^t, PDE constants κ, and embedding of past L frames. The illustration of u_θ is shown in Figure S1.

dynamics [14, 15]. This effect is especially severe for chaotic systems, where trajectories diverge exponentially [16].

Generative models address this failure mode by learning the full conditional distribution $p(x^{t+1} \mid x^t)$ rather than its mean [17]. Sampling from it introduces structured stochasticity and mitigates rollout drift [18, 15, 16, 19–22]. Among generative frameworks, Flow Matching (FM) [23–26] is particularly attractive, as it learns a continuous flow by regressing a conditional vector field. This approach achieves high sample quality on turbulent PDE systems while maintaining high inference efficiency [27], making it a competitive approach for generative modeling [28].

Recent works on representation-based FM for physics emulation have shown great success. Latent-space formulations apply FM in the space of learned autoencoders [29], reducing computational cost but introducing additional training overhead and are less interpretable. This has motivated the use of physics-based representations, most prominently Fourier approaches [10, 30], where FM is performed in a fixed spectral basis that can generalize across resolutions. However, such representations are inherently global and do not explicitly capture localized multi-scale structure, which is central to many physical systems. As such, we ask the following question:

Are there representations in which the generative process can be carried out naturally across multiple scales, while preserving accuracy, spectral coherence, and physical plausibility of the emulated dynamics?

As an answer to this question, in this work we present **Wavelet Flow Matching** (WFM), a framework that performs physical emulation in the wavelet space. WFM leverages Optimal-Transport Conditional Flow Matching (OT-CFM) [26] that performs autoregressive rollouts on the multi-scale wavelet representation of physical fields. Discrete Wavelet Transform (DWT) representations are grounded in classical harmonic analysis, require no training, and are simultaneously localized in space (or time) and scale [31], i.e., capturing both where frequencies occur and at what resolution. Each scale of the DWT reduces the spatial resolution, so that coarser coefficient maps live on progressively smaller grids. Training and sampling are carried out entirely in wavelet space, while an Inverse DWT (IDWT) is applied to the generated coefficients only at the end of the sampling procedure to reconstruct the next physical state (Figure 1). The framework is highly configurable, for instance users can choose the desired number of scales, as well as different types of prescribed wavelet representations.

Our **main contributions** are (Figure 1):

- We introduce WFM, a framework performing autoregressive emulation in the wavelet space, leveraging the hierarchical structure of U-Nets to predict multi-scale flow velocities.
- We benchmark WFM on three challenging PDE datasets, showing that WFM outperforms other baselines in terms of spectral coherence RMSE and overall long-range skill. The advantage of multi-scale wavelet representation is especially evident in the most chaotic regimes.
- WFM offers a favourable accuracy–cost trade-off: it needs no pre-trained compression and samples 1.1–1.5$\times$ faster than pixel-space flow matching at comparable or better accuracy.

To facilitate further research in this direction, we publicly release our code at `https://github.com/gabrieleaccarino/wavelet_flow_matching`, along with the pre-trained model weights at `https://huggingface.co/gabrieleaccarino/wavelet-flow-matching`.

2 Background

2.1 Optimal-Transport Conditional Flow Matching

Let $p_0 = \mathcal{N}(0, I_d)$ denote a simple source distribution and let p_1 be the target distribution on $\mathbb{R}^d$, accessed through n samples $x_1^{(1)}, \dots, x_1^{(n)} \sim p_1$. The goal of FM is to learn a time-dependent velocity field $u : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$ that transports samples from p_0 to p_1.

This transport is defined through the ordinary differential equation (ODE):

$$\begin{cases} x(0) = x_0 \sim p_0 \\ \dot{x}(\tau) = u\big(x(\tau), \tau\big) \end{cases} \qquad \tau \in [0,1] \quad . \tag{1}$$

whose solution induces a trajectory $x(\tau)$ starting from p_0. For every τ in [0, 1], the intermediate marginals of $x(\tau)$ trace a probability path $p(\cdot \mid \tau)$ that continuously deforms p_0 into p_1. Once u is learned, sampling reduces to drawing $x_0 \sim p_0$ and numerically integrating Equation (1) (e.g., with the Euler method).

Conditional Flow Matching. Rather than learning u directly, conditional flow matching (CFM) conditions on a target sample $x_1 \sim p_1$ (independent of τ), and introduces a conditional probability path

$$p(\cdot \mid x_1) = \mathcal{N}\big(\tau x_1,\ (1-\tau)^2 I_d\big), \qquad x_\tau = \tau x_1 + (1-\tau)\epsilon, \qquad \epsilon \sim \mathcal{N}\big(0, I_d\big)$$

As τ moves from 0 to 1, the mean $\mathbb{E}[x_\tau \,|\, x_1] = \tau x_1$ drifts linearly from the origin to x_1 while the covariance $(1-\tau)^2 I_d$ contracts to zero, so that the path collapses to the point mass δ_{x_1} at $\tau = 1$. However, other choices of conditional path are possible and we refer the reader to [24, 32, 33]. Given the continuity equation [24, Section 3.5], this construction induces a conditional velocity field with the closed-form expression, whose conditional expectation recovers the optimal velocity field $u^\star(x, \tau)$ [23, Theorem 1]:

$$u^{\text{cond}}(x, x_1, \tau) \ = \ \frac{x_1 - x}{1-\tau} \quad . \tag{2}$$

Training Objective. The optimal velocity can be approximated by a neural network $u_\theta : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$ with parameters θ minimizing

$$\mathcal{L}_{\text{FM}}(\theta) \ = \ \mathbb{E}_{\substack{\tau \sim \mathcal{U}([0,1]) \\ x_\tau \sim p(\cdot|\tau)}} \Big\| u_\theta(x_\tau, \tau) - u^\star(x_\tau, \tau) \Big\|^2 \quad . \tag{3}$$

A direct regression onto $u^\star$ would require evaluating the conditional expectation in Equation (2), which is generally intractable. The key observation of Lipman et al. [23] is that regressing against $u^\star$ is equivalent, up to an additive constant, to minimizing

$$\mathcal{L}_{\text{CFM}}(\theta) \ = \ \mathbb{E}_{\substack{\tau \sim \mathcal{U}([0,1]) \\ x_0 \sim p_0,\, x_1 \sim p_1}} \left[\Big\| u_\theta(x_\tau, \tau) - \underbrace{\frac{x_1 - x_\tau}{1-\tau}}_{= x_1 - x_0} \Big\|^2 \right], \qquad x_\tau := \tau x_1 + (1-\tau) x_0 \quad . \tag{4}$$

This objective is straightforward to estimate: τ and x_0 are sampled directly, and p_1 is replaced by the empirical distribution over the training set. When pairs (x_0, x_1) are drawn from an optimal-transport coupling between p_0 and p_1, the same linear interpolant x_τ defines the displacement interpolation underlying OT-CFM [26]. Under the independent sampling $x_0 \sim p_0$, $x_1 \sim p_1$ used above, the objective remains a valid CFM loss but the coupling is not optimal.

2.2 Discrete Wavelet Transform

The DWT decomposes a signal into components simultaneously localized in space (or time) and scale [34, 35]. Formally, for a one-dimensional signal $f \in L^2(\mathbb{R})$, a family of discrete wavelets is generated from a single function $\psi \in L^2(\mathbb{R})$, called the *mother wavelet*, through dyadic dilations and discrete translations:

$$\psi_{j,n}(x) = 2^{-j/2}\,\psi\big(2^{-j}x - n\big)\,, \quad (j,n) \in \mathbb{Z}^2 \quad . \tag{5}$$

where j controls the scale (fine-to-coarse) and n the spatial location. For suitably constructed ψ, the family $\{\psi_{j,n}\}_{j,n}$ forms an *orthonormal basis* of $L^2(\mathbb{R})$, so that f admits the unique multi-scale expansion:

$$f \;=\; \sum_n a_J[n]\,\phi_{J,n} + \sum_{j=1}^{J}\sum_n d_j[n]\,\psi_{j,n} \quad . \tag{6}$$

where $a_J[n]$ are coarse-scale *approximation coefficients*, $d_j[n]$ are *detail coefficients* at scale 2^j, and ϕ is the scaling function associated to the mother wavelet [34, 36, 37]. For a signal $x \in \mathbb{R}^{C\times H\times W}$, where C is the number of channels (or physical variables) and $H \times W$ is the spatial grid, the 2-dimensional DWT is applied separately along rows and columns. We indicate the DWT of the signal x across scales $j \in \{1, \dots, J\}$, induced by a mother wavelet ψ as:

$$w_j \;=\; \mathcal{W}_j(x) \;\in\; \mathbb{R}^{C\times 4\times H_j\times W_j}, \qquad H_j \approx \frac{H}{2^j}, \quad W_j \approx \frac{W}{2^j}. \tag{7}$$

At each scale, the DWT produces four coefficient maps (or sub-bands), one approximation (LL) and three oriented detail matrices (LH (horizontal), HL (vertical), HH (diagonal)), computed on a progressively down-sampled grid of size $H_j \times W_j$, where $j = 1$ corresponds to the finest resolution. The full multi-scale representation is the collection $w = \mathcal{W}(x) = \{w_j\}_{j=1}^{J}$, and exact reconstruction is guaranteed by the inverse DWT (IDWT) by $x = \mathcal{W}^{-1}(w)$. Different mother wavelets exist, and in this work we consider four mother wavelets, $\mathrm{db1}$ (haar), $\mathrm{db2}$, $\mathrm{db4}$, and $\mathrm{db6}$, with filter lengths 2, 4, 8, and 12, respectively (refer to Appendix D and Table S10 for additional details).

3 Wavelet Flow Matching for Physics Emulation

Rather than learning the conditional distribution $p(x^{t+1}\,|\,x^t)$ directly in pixel space, **Wavelet Flow Matching** (WFM) re-frames the OT-CFM problem (Section 2.1) in the wavelet domain and learns $p(w^{t+1}\,|\,w^t)$, where $w^{(\cdot)} = \mathcal{W}(x^{(\cdot)})$ denotes the multi-scale wavelet representation of the state of the system at a specific point in time $(\cdot)$ (Section 2.2). This yields J scale-specific flows, sharing one interpolation time τ and one velocity network, that together define the generative task over the full wavelet representation of the target field. The illustration of the wavelet-space emulation process is depicted in Figure 1.

Problem statement. Let $x^t \in \mathbb{R}^{C\times H\times W}$ denote the state of the system at time t. The goal is to learn the conditional distribution $p(w^{t+1} \mid w^t, \{x^{t-l}\}_{l=1}^{L}, \kappa)$, where $\{x^{t-l}\}_{l=1}^{L}$ is a context window of L past states and κ is a set of scalar variables encoding static physical properties of the system. We distinguish throughout between physical time t and the OT-CFM interpolation time $\tau \in [0, 1]$.

Multi-scale Optimal Transport Paths. Rather than constructing a single flow in pixel space, we define J flows, one per wavelet scale, each transporting scale-matched Gaussian noise to the corresponding four wavelet sub-bands (see Section 2.2) of the target state:

$$w_j^\tau \;=\; \tau\, w_j^{t+1} \;+\; (1-\tau)\,\varepsilon_j, \quad \tau \sim \mathcal{U}([0,1]), \quad \varepsilon_j \sim \mathcal{N}(0, I_{d_j}), \tag{8}$$

where $w_j^{t+1} = \mathcal{W}_j(x^{t+1}) \in \mathbb{R}^{C\times 4\times H_j\times W_j}$ are the wavelet coefficients of the target state at scale j, and ε_j is Gaussian noise drawn independently at each scale. This is merely a practical design choice:

since the wavelet families used in this work are orthonormal, applying the DWT to Gaussian noise sampled in pixel space yields independent standard Gaussian coefficients at each scale, so the two sampling procedures are equivalent in distribution, $\mathcal{W}_j(\varepsilon) \overset{d}{=} \varepsilon_j$ [38, 39]. In both cases, the condition on the source distribution in OT-CFM [23, 24, 26] is satisfied. Therefore, we draw ε_j directly in wavelet space for sake of simplicity and to avoid additional DWT for the noise.

Velocity-predicting Neural Network. The velocity network u_θ is a U-Net [12] whose encoder–decoder hierarchy naturally mirrors the multi-scale structure of the DWT (see Figure S1). At each scale j, it receives as input the channel-wise concatenation of the wavelet coefficients of the current state $w_j^t = \mathcal{W}_j(x^t)$ and the noisy interpolant w_j^τ, obtained by interpolating between Gaussian noise and the wavelet coefficients of the subsequent state:

$$z_j^\tau \;=\; \big[\, w_j^t,\;\; w_j^\tau \,\big]_C \;\in\; \mathbb{R}^{2C\times 4\times H_j\times W_j}, \tag{9}$$

where the current-state coefficients w_j^t serve as a static, scale-consistent conditioning signal throughout integration. Note that this concatenation does not violate the FM framework: at $\tau = 0$ the interpolated coefficients w_j^τ coincide with pure Gaussian noise ε_j, meaning the model always begins from an unstructured noise field at sampling time. The concatenated wavelet coefficients z_j^τ at each scale j are injected directly into the corresponding encoder level via a dedicated stem convolution, so that each resolution stage of the U-Net processes the frequency sub-bands it is spatially matched to, as illustrated in Figure S1. Starting from a shared backbone, the decoder employs J prediction heads, one per scale, each producing the corresponding velocity estimate $\hat{u}_j$ at the appropriate spatial resolution and for all the four sub-bands. Beyond the input in Equation (9), the U-Net is conditioned on three sources of information, fused into a single vector $c \in \mathbb{R}^d$ and injected into every residual block [40] via adaptive group normalization (AdaGN) [41], a feature-wise scale and shift [42] applied after group normalization: *(i)* the FM interpolation time τ embedded via a sinusoidal Fourier encoding followed by a small MLP; *(ii)* the static physical parameters κ, embedded linearly; and *(iii)* the temporal context $\{x^{t-l}\}_{l=1}^{L}$, embedded via cross-attention over the L frames prior to x^t. Because every head $\hat{u}_j$ is computed from the shared encoder–decoder, it depends on the interpolants at all scales, fine-to-coarse through the encoder injections and coarse-to-fine through the decoder, so the J flows are coupled at every integration step (see Appendix B.1).

Formulation of Training Objective in Wavelet Space. As introduced in Section 2.1 Equation (2), the target velocity at scale j along the straight path connecting ε_j to w_j^{t+1} is constant and is obtained by differentiating the interpolant in Equation (8) with respect to τ:

$$u_j \;=\; \frac{d\, w_j^\tau}{d\tau} \;=\; w_j^{t+1} - \varepsilon_j, \qquad u_j \in \mathbb{R}^{C\times 4\times H_j\times W_j}. \tag{10}$$

We train a velocity-predicting neural network u_θ to match Equation (10) at every scale simultaneously, where $\hat{u}_j$ denotes the network prediction at scale j. To account for the large differences in energy across scales, we extend Equation (4) by normalizing the per-scale loss by the spatial variance of the target velocity:

$$\ell_j(\theta) \;=\; \mathbb{E}_{\substack{\tau\sim\mathcal{U}[0,\,1] \\ (x^t,x^{t+1})\sim q,\; \varepsilon_j\sim\mathcal{N}(0,I_{d_j})}} \left[\frac{1}{BC} \sum_{b,c} \frac{\big\langle \big(\hat{u}_j - u_j\big)^2 \big\rangle}{\big\langle \big(u_j - \langle u_j \rangle\big)^2 \big\rangle + \epsilon} \right], \tag{11}$$

where q denotes the data distribution, B and C are the batch size and number of channels respectively, $\langle\cdot\rangle$ denotes spatial averaging over the $H_j \times W_j$ grid, and $\epsilon = 10^{-4}$ is a small constant for numerical stability. The denominator is the per-sample, per-channel spatial variance of the target velocity. This normalization ensures that all scales contribute equally to the gradient, preventing the energy-dominant LL sub-band from overwhelming the loss contribution of the detail sub-bands. Variance-normalized objectives of this kind are increasingly common in physics emulation [21]; in wavelet space the normalization is essential, because approximation and detail coefficients differ in energy by orders of magnitude. The total training objective is then a weighted sum across all J scales:

$$\mathcal{L}(\theta) \;=\; \sum_{j=1}^{J} \frac{\lambda_j}{\sum_j \lambda_j}\, \ell_j(\theta), \tag{12}$$

In all experiments we use uniform weighting, setting $\lambda_j = 1$ for all j, so that each scale contributes equally to $\mathcal{L}(\theta)$.

Inference and Autoregressive Rollout. At inference time, sampling is performed entirely in the wavelet space. Given the current physical state x^t, we first compute the multi-scale wavelet representation $\mathcal{W}(x^t)$ and independently sample a noise field at each scale $\varepsilon_j \sim \mathcal{N}(0, I_{d_j})$ directly in wavelet space. Starting from $w_j^{\tau=0} = \varepsilon_j$ at each scale, the predicted wavelet coefficients of the next state are obtained by integrating the learned velocity field via an Euler solver with uniform step size $\Delta\tau = 1/N$ over $\tau \in [0, 1]$:

$$w_j^{\tau_{n+1}} \;=\; w_j^{\tau_n} \;+\; \Delta\tau \cdot u_\theta\big(\big[w_j^t,\, w_j^{\tau_n}\big]_C,\; \tau_n,\; \kappa,\; \{x^{t-l}\}_{l=1}^{L}\big)\,, \qquad n = 0, \ldots, N-1, \tag{13}$$

where at each step the network u_θ produces multi-scale velocity estimates that are scaled by $\Delta\tau$ and accumulated into the current wavelet state. After N steps, $w_j^{\tau_N} = \hat{w}_j^{t+1}$ gives the multi-scale wavelet representation of the next state at physical time $t+1$. The corresponding pixel space state is then recovered in a single call to the IDWT: $\hat{x}^{t+1} = \mathcal{W}^{-1}\big(\{\hat{w}_j^{t+1}\}_{j=1}^{J}\big)$ (Section 2.2). For autoregressive rollout, $\hat{x}^{t+1}$ is fed back as the new current state, its wavelet coefficients $\{\mathcal{W}_j(\hat{x}^{t+1})\}$ are recomputed via the DWT, the context window is updated, and the procedure repeated across many steps.

4 Results

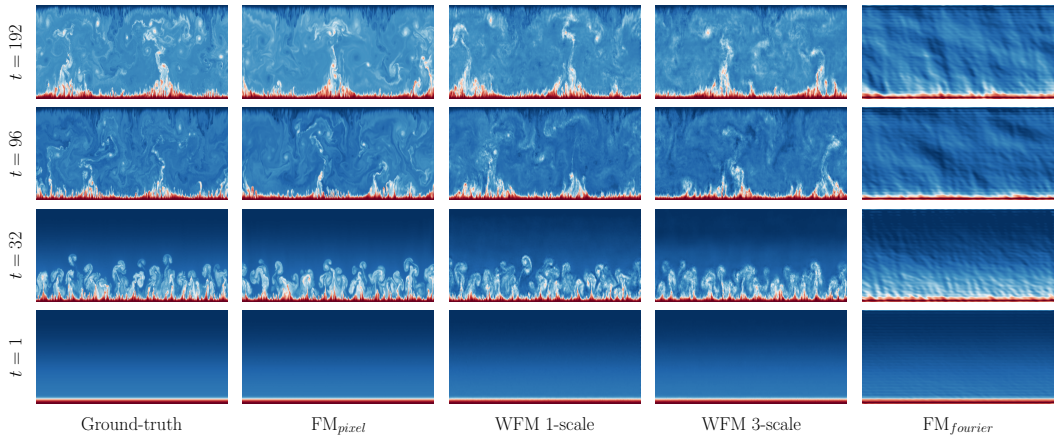
4.1 Experimental Setup

In this section, we empirically evaluate **WFM**'s performance and its ability to accurately emulate complex physical systems across a range of dynamical regimes. We compare WFM against a representation-based suite of baselines. The only exception is FM$_{pixel}$, which operates directly in pixel space and therefore serves as a control reference of generation quality in the original data space. Unless otherwise stated, generative models produce $\mathcal{M} = 8$ trajectories from the same initial condition, each drawing fresh Gaussian noise at every autoregressive step, and all scores are averaged over three inference seeds (42, 84, 168) shared across models; deterministic baselines produce a single prediction. Each table caption reports the seed-to-seed variability.

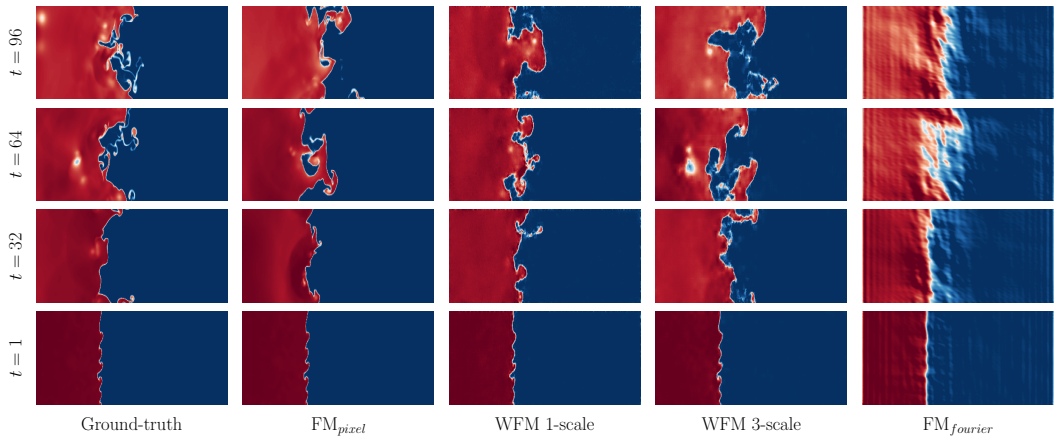
Datasets. To study the effect of the generation across multiple levels of decomposition in wavelet space, we select three PDE datasets with different dynamical characteristics from The Well [5], including Turbulent Radiative Layer 2D (TRL) [43], Rayleigh–Bénard (RB) convection [44], and Active Matter (AM) [45]. These datasets represent distinct fluid regimes that cover many key challenges in dynamical systems emulation, including non-linearities, multi-scale interactions, and complex spatio-temporal patterns. Additional details are provided in Appendix A.

WFM variants. WFM is built on a U-Net encoder–decoder [12] that operates across wavelet scales (Section 3). To isolate the effect of multi-scale generation, we train both a 1-scale ($J = 1$) and 3-scale WFM ($J = 3$) for each PDE system. We set the maximum number of wavelet scales to $J = 3$, as this is the largest value compatible with all grid resolutions considered. For grids with a minimum spatial dimension of 128 (i.e., TRL and RB), the coarsest approximation sub-band at $J = 3$ has size $(\cdot) \times 16$; increasing to $J = 4$ would reduce the bottleneck dimension in the U-Net to 8, which is insufficient for meaningful convolutional feature extraction. WFM conditions on $L = 3$ past frames, alongside the current state x^t, yielding a total temporal context of four states, consistent with baseline models in The Well [5]. Past frames are provided in pixel space, as they serve only to inform the temporal context and need not be decomposed into frequency sub-bands. We fix the embedding dimension to $d = 256$ throughout all experiments. Additional information is provided in Appendix B.

Baselines. To assess the quality of WFM's wavelet-space generation, we compare against flow matching in pixel space (FM$_{pixel}$), flow matching in Fourier space (FM$_{fourier}$) as described in [46], Wavelet Diffusion Neural Operator (WDNO) [47], Fourier Neural Operator (FNO) [48], a Tensorized FNO (TFNO) [10], and a Wavelet Neural Operator (WNO) [49], which represent the main families of generative and deterministic PDE surrogates. All flow-matching models use the same U-Net architecture, flow-matching objective and sampling budget ($N = 50$ Euler steps, $\mathcal{M} = 8$ trajectories rolled out from the same initial condition); apart from learning rate and batch size, they differ only in the space where generation is performed. Capacity is matched pairwise: WFM 1-scale and FM$_{pixel}$ are architecturally identical (12.7M parameters), and WFM 3-scale and FM$_{fourier}$ have the same capacity (≈30.5M parameters; Appendix B). Differences within each pair can therefore be



(a) Rayleigh–Bénard Convection



(b) Turbulent Radiative Layer 2D

Figure 2: Qualitative results at different rollout lead-times (t).

attributed to the representation rather than to architecture or parameter count. On Rayleigh–Bénard we also compare against a latent-space model, $\text{LatentFM}_{\div 64}$, which performs flow matching in the latent space of the pre-trained autoencoder of Rozet et al. [21] (64× compression) with the same flow-matching objective and sampler as WFM (Appendix B.8).

Metrics. To guarantee a comprehensive comparison of WFM across scales we consider multiple metrics. Similar to [5, 21], our analysis is either performed at a lead time $t = i \times \Delta$ or averaged across lead times $a{:}b$. Specifically, we include the variance-normalized root mean squared error (VRMSE), which compares better across scales and in low-variance regions; the continuous ranked probability score (CRPS) [50], which generalizes the mean absolute error to probabilistic forecasts and, computed over $\mathcal{M}$ independent trajectories rolled out from the same initial condition, rewards both accuracy and calibrated spread; and the spectral coherence RMSE [21], which measures the error across frequency bands. Full definitions are provided in Appendix C.

4.2 WFM Evaluation

We benchmark WFM 1- and 3-scale with four prescribed mother wavelets (haar, db2, db4, db6) on each PDE system and compare them to other generative models. The wavelet is selected by a single criterion, the spectral coherence RMSE averaged over the *Low*, *Mid* and *High* bands on the test trajectories (Table 1), which provides a holistic measure of spectral fidelity across the full frequency spectrum. Tables S12–S14 report every wavelet on all metrics, including per-band results, so the choice can be checked against VRMSE and CRPS.

Table 1: Spectral Coherence RMSE averaged across *Low*, *Mid* and *High* frequency bands and rollout horizons per PDE system. For each WFM variant, **bold** denotes the optimal wavelet representation used in all subsequent experiments. LatentFM$_{\div 64}$ is evaluated on Rayleigh–Bénard only. Mean over three inference seeds (42, 84, 168); the standard deviation across seeds is below 5% of the mean for all entries. Here, – indicates that the model was not available for the corresponding system.

	Turbulent Radiative Layer 2D	Rayleigh–Bénard	Active Matter
WFM-1 haar	**0.147**	0.140	**0.241**
WFM-1 db2	0.154	0.132	0.288
WFM-1 db4	0.149	**0.121**	0.303
WFM-1 db6	0.150	0.140	0.290
WFM-3 haar	**0.149**	0.131	0.247
WFM-3 db2	0.154	0.111	**0.220**
WFM-3 db4	0.154	**0.109**	0.249
WFM-3 db6	0.153	0.112	0.238
FM$_{\text{pixel}}$	0.153	0.131	0.244
FM$_{\text{fourier}}$	0.379	0.266	0.376
WDNO	0.271	0.467	0.304
LatentFM$_{\div 64}$	–	0.162	–

Both WFM 1- and 3-scale consistently outperform all generative baselines across every PDE system and frequency band. For WFM 1-scale, haar achieves the lowest average spectral coherence RMSE on TRL and AM, while $\mathrm{db4}$ is optimal for RB. For WFM 3-scale, haar yields the best representation on TRL, $\mathrm{db4}$ on RB, and $\mathrm{db2}$ on AM. All subsequent experiments adopt these PDE-specific wavelet selections, which we summarize in Table S11.

Qualitative results are presented in Figures 2a–2b. Both WFM variants produce physically plausible states at extended rollout steps, remaining comparable to pixel-space generation. The difference between variants is most evident on AM (Figure S5), where WFM 1-scale loses coherent structure as early as $t = 16$, whereas WFM 3-scale preserves it throughout.

WFM performance relative to deterministic and generative baselines is shown in Figure 3. WFM 3-scale outperforms all deterministic baselines (Figure 3a) and the other generative baselines (Figures 3b and 3c) across all systems. It is also competitive with the quality-control model FM$_{pixel}$, achieving a lower spectral coherence RMSE on RB in all frequency bands and on AM in the high-frequency band.

Beyond competitive accuracy, WFM consistently delivers faster inference than FM$_{pixel}$ across all three systems, as reported in Table 2. Since all experiments are conducted under identical conditions, we attribute this result to the spatially smaller wavelet representation space in which the generative process operates. We report the inference profiling results for the selected WFM variants in Table S23, and provide a comprehensive profiling analysis covering all considered mother wavelets in Table S24.

Table 2: Inference speedup of WFM variants relative to FM$_{pixel}$ (higher is better).

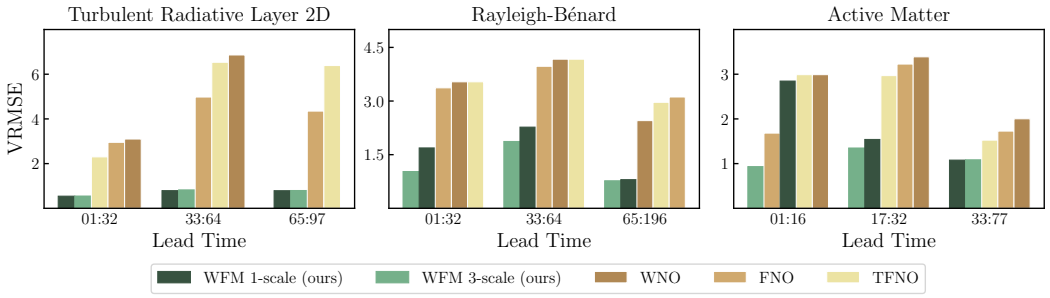
PDE	**WFM 1-scale**	**WFM 3-scale**
TRL	1.25×	1.14×
RB	1.52×	1.38×
AM	1.46×	1.26×

We further evaluate WFM across turbulent regimes. For TRL, we vary the cooling time t_{cool} and report CRPS (Tables 3a and S15) and spectral coherence RMSE (Table S16). WFM is most advantageous in the small t_{cool} regime, where rapid condensation produces a highly corrugated interface and sharp density contrasts that are particularly challenging to emulate. For RB, we vary the Ra and Pr numbers and report CRPS (Tables 3b, S17 and S18) and spectral coherence RMSE (Tables S19a and S19b). Here, WFM achieves the lowest CRPS across the full Ra–Pr parameter space, including the most chaotic regime ($Ra = 10^{10}$, $Pr = 10^{-1}$).

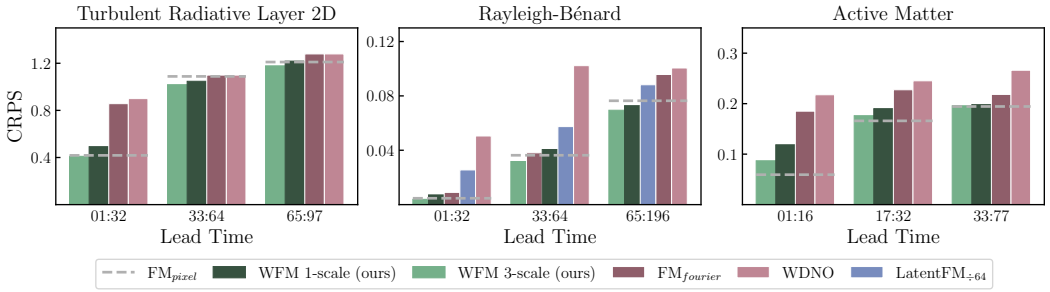
Cross-scale coupling, physical consistency and robustness. Although WFM defines one flow per scale, the flows are coupled through the shared U-Net: perturbing one scale changes the velocity predicted at all others, which independent per-scale flows would not (Appendix B.1). WFM also

preserves the kinetic energy of the reference trajectories, ranking first or second at every rollout horizon on all systems (Table S20). Finally, its rollouts are robust to perturbations of the initial conditions, with CRPS changing by at most 2% (Appendix E).

(a) VRMSE for WFM variants and deterministic baselines.



(b) CRPS for WFM variants and generative baselines.



(c) Spectral Coherence RMSE for WFM variants and generative baselines.

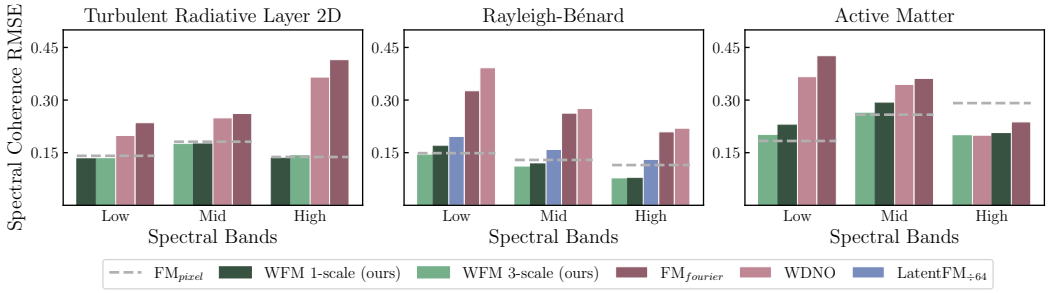


Figure 3: VRMSE (a) and CRPS (b) averaged over rollout horizons, and Spectral coherence RMSE (c) per frequency band, for the three PDE systems. Generative models are averaged over three inference seeds; FNO, TFNO and WNO are deterministic. FM$_{pixel}$ is shown as a dashed gray line and serves as a quality-control reference.

5 Discussion

A key perspective offered by our approach is its connection to multi-scale closure modeling in PDEs. Classical closure schemes parameterize unresolved small-scale processes as functions of resolved variables, often assuming a unidirectional dependence that neglects cross-scale feedbacks such as backscatter and intermittency [51, 52]. WFM instead models the joint evolution of all scales in a unified generative framework, where multi-scale representations are induced by the choice of mother wavelet. This provides a principled multi-scale decomposition aligned with the locality and cascade structure of many physical systems, without requiring learned compression and loss of information.

The resulting separation of representation and dynamics introduces a structured inductive bias that may improve interpretability [36] and generalization [53], as suggested by our consistent performance across the wide range of Ra–Pr regimes in RB and t_{cool} times in TRL.

Table 3: CRPS averaged over rollout steps, for the three shortest cooling times t_{cool} in TRL (a) and per Rayleigh number Ra, averaged over Prandtl numbers, in RB (b). Mean over three inference seeds (42, 84, 168); the standard deviation across seeds is below 5% of the mean for all entries except those marked *. **Bold**: best; underline: second best.

(a) TRL per t_{cool}.

t_{cool}	0.03	0.06	0.1
WFM-1 haar	1.165	1.135	0.865*
WFM-1 db2	1.112	1.078	0.878
WFM-1 db4	1.170	1.102	0.847*
WFM-1 db6	1.076	1.074	0.831
WFM-3 haar	1.107	1.117	0.868
WFM-3 db2	1.248	1.141	0.894
WFM-3 db4	**1.048**	**1.006**	**0.817***
WFM-3 db6	1.075	1.034	0.882
$\mathrm{FM}_{\mathrm{pixel}}$	1.220	1.126	0.875*
$\mathrm{FM}_{\mathrm{fourier}}$	1.439	1.424	1.195
WDNO	3.330	3.292	3.183

(b) Rayleigh–Bénard per Ra, averaged over Pr.

Ra	10^6	10^8	10^{10}
WFM-1 haar	0.059	0.060*	0.051
WFM-1 db2	**0.052**	0.062*	0.052*
WFM-1 db4	0.059*	0.060	0.051
WFM-1 db6	0.060*	0.069*	0.075
WFM-3 haar	0.060*	0.060	0.055*
WFM-3 db2	0.054*	0.055	0.049
WFM-3 db4	0.053*	0.058	**0.048***
WFM-3 db6	0.065*	**0.055**	0.057
$\mathrm{FM}_{\mathrm{pixel}}$	0.057*	0.066*	0.051
$\mathrm{FM}_{\mathrm{fourier}}$	0.071	0.081	0.065
WDNO	0.120	0.117	0.116
$\mathrm{LatentFM}_{\div 64}$	0.087*	0.067	0.065

Related work. Hybrid representations that combine data-driven and physics-informed constructions, such as operator-theoretic embeddings [22] or Fourier-based transformations [54], still rely on learning and often on empirical mode truncation or handcrafted basis selection, while multi-scale structure is typically enforced indirectly through hierarchical architectures, e.g. pyramidal flow matching [55], rather than by the representation itself [37]. Wavelets offer such a representation: analytically grounded, spatially localized, orthogonal and energy-compacting, with no training required. In generative modeling they have served as cascades of per-scale score models [56], tokens for autoregressive image generation [57], a training signal for latent diffusion [58], and components of adversarial neural operators [59]. In PDE emulation, WDNO [47] denoises space–time blocks of wavelet coefficients, Fourier flow matching [46] generates a truncated global spectrum, and latent models [21] learn a lossy representation that requires pre-training; WFM instead generates all wavelet coefficients of the next state, coupled across scales through a shared network.

Limitations. This study considers only the Daubechies family of orthonormal wavelets; other families, such as *symlets* and *coiflets* [35], remain unexplored, and the best mother wavelet is currently selected per dataset rather than learned or chosen a priori. The DWT also assumes regular grids and handles non-periodic boundaries through padding, which may affect boundary-dominated dynamics; extending WFM to irregular meshes and other geometries, such as spherical domains [60, 61], would broaden its applicability, including to weather and climate emulation.

Acknowledgments and Disclosure of Funding

We acknowledge funding from NSF through the Learning the Earth with Artificial intelligence and Physics (LEAP) Science and Technology Center (STC) (Award #2019625). VA acknowledges support from a PIVOT Research award (Award #12871) from the Simons Foundation. This work used Derecho-GPU system at NCAR through allocation CIS251353 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, which is supported by U.S. National Science Foundation grants #2138259, #2138286, #2138307, #2137603, and #2138296.

References

[1] Stephen B. Pope. *Turbulent Flows*. Cambridge University Press, 2000.

[2] Kaifeng Bi, Lingxi Xie, Hengheng Zhang, Xin Chen, Xiaotao Gu, and Qi Tian. Accurate medium-range global weather forecasting with 3D neural networks. *Nature*, 619(7970):533–538, July 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-06185-3. URL https://doi.org/10.1038/s41586-023-06185-3.

[3] Juan Nathaniel, Yongquan Qu, Tung Nguyen, Sungduk Yu, Julius Busecke, Aditya Grover, and Pierre Gentine. Chaosbench: A multi-channel, physics-based benchmark for subseasonal-to-seasonal climate prediction. *Advances in Neural Information Processing Systems*, 37: 43715–43729, 2024.

[4] Remi Lam et al. Learning skillful medium-range global weather forecasting. *Science*, 2023.

[5] Ruben Ohana, Michael McCabe, et al. The well: a large-scale collection of diverse physics simulations for machine learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[6] Tim N. Palmer. Stochastic weather and climate models. *Nature Reviews Physics*, 2019.

[7] 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*, 2021.

[8] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *International Conference on Learning Representations (ICLR)*, 2021.

[9] Michael McCabe, Bruno Regaldo-Saint Blancard, et al. Multiple physics pretraining for spatiotemporal surrogate models. *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[10] Jean Kossaifi, Nikola Kovachki, Kamyar Azizzadenesheli, and Anima Anandkumar. Multi-grid tensorized fourier neural operator for high-resolution PDEs. *arXiv preprint arXiv:2310.00120*, 2023.

[11] Bogdan Raonic, Roberto Molinaro, Tim De Ryck, Tobias Rohner, Francesca Bartolucci, Rima Alaifari, Siddhartha Mishra, and Emmanuel De Bézenac. Convolutional neural operators for robust and accurate learning of pdes. *Advances in Neural Information Processing Systems*, 36: 77187–77200, 2023.

[12] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

[13] Zijie Li, Kazem Meidani, and Amir Barati Farimani. Transformer for partial differential equations' operator learning. In *TMLR*, 2023.

[14] Michael McCabe, Peter Harrington, Shashank Subramanian, and Jed Brown. Towards stability of autoregressive neural operators. *Transactions on Machine Learning Research (TMLR)*, 2023.

[15] Phillip Lippe, Bastiaan S. Veeling, Paris Perdikaris, Richard E. Turner, and Johannes Brandstetter. PDE-Refiner: Achieving accurate long rollouts with neural pde solvers. In *NeurIPS*, 2023.

[16] Georg Kohl, Li-Wei Chen, and Nils Thuerey. Benchmarking autoregressive conditional diffusion models for turbulent flow simulation. In *ICML AI4Science Workshop*, 2024.

[17] Gabriele Accarino, Mohammad M. Amirian, Thea H. Heimdal, Amanda R. Fay, Galen A. McKinley, and Viviana Acquaviva. Learning calibrated and samplable predictive distributions of surface-ocean ΔpCO_2 using tabular diffusion models. In *NeurIPS 2026 Workshop on Representations for the Physical Sciences*, 2026. URL https://openreview.net/forum?id=dpSnxP8j9Y.

[18] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

[19] Aliaksandra Shysheya et al. On conditional diffusion models for PDE simulations. In *NeurIPS*, 2024.

[20] Jiahe Huang, Guandao Yang, Zichen Wang, and Jeong Joon Park. Diffusionpde: Generative pde-solving under partial observation. *Advances in Neural Information Processing Systems*, 37: 130291–130323, 2024.

[21] François Rozet, Ruben Ohana, Michael McCabe, Gilles Louppe, François Lanusse, and Shirley Ho. Lost in latent space: An empirical study of latent diffusion models for physics emulation. *NeurIPS*, 2025.

[22] Juan Nathaniel and Pierre Gentine. Generative emulation of chaotic dynamics with coherent prior. *Computer Methods in Applied Mechanics and Engineering*, 448:118410, 2026.

[23] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *ICLR*, 2023.

[24] Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky T. Q. Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code, 2024. URL `https://arxiv.org/abs/2412.06264`.

[25] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2023.

[26] Alexander Tong, Kilian Fatras, Nikolay Malkin, et al. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research (TMLR)*, 2024.

[27] Zijie Li, Anthony Zhou, and Amir Barati Farimani. Generative latent neural PDE solver using flow matching. *arXiv preprint arXiv:2503.22600*, 2025.

[28] Srishti Gupta and Yashasvee Taiwade. Efficiency vs. fidelity: A comparative analysis of diffusion probabilistic models and flow matching on low-resource hardware, 2025. URL `https://arxiv.org/abs/2511.19379`.

[29] Quan Dao, Hao Phung, Binh Nguyen, and Anh Tran. Flow matching in latent space. *arXiv preprint arXiv:2307.08698*, 2023.

[30] Xin Li, Jingdong Zhang, Qunxi Zhu, Chengli Zhao, Xue Zhang, Xiaojun Duan, and Wei Lin. From Fourier to neural ODEs: Flow matching for modeling complex systems. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 29390–29405. PMLR, 21–27 Jul 2024. URL `https://proceedings.mlr.press/v235/li24cn.html`.

[31] Stéphane Mallat. *A Wavelet Tour of Signal Processing: The Sparse Way*. Academic Press, 3rd edition, 2009.

[32] Anne Gagneux, Ségolène Martin, Rémi Emonet, Quentin Bertrand, and Mathurin Massias. A visual dive into conditional flow matching. In *ICLR Blogposts 2025*, 2025. URL `https://iclr-blogposts.github.io/2025/blog/conditional-flow-matching/`. https://iclr-blogposts.github.io/2025/blog/conditional-flow-matching/.

[33] Ruiqi Gao, Emiel Hoogeboom, Jonathan Heek, Valentin De Bortoli, Kevin Patrick Murphy, and Tim Salimans. Diffusion models and gaussian flow matching: Two sides of the same coin. In *The Fourth Blogpost Track at ICLR 2025*, 2025. URL `https://openreview.net/forum?id=C8Yyg9wy0s`.

[34] Stéphane G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989.

[35] Ingrid Daubechies. *Ten Lectures on Wavelets*. SIAM, Philadelphia, PA, 1992.

[36] Wooseok Ha, Chandan Singh, Francois Lanusse, Srigokul Upadhyayula, and Bin Yu. Adaptive wavelet distillation from neural networks through interpretations. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

[37] Gabriele Accarino, Viviana Acquaviva, Sara Shamekh, Duncan Watson-Parris, and David Lawrence. Wavesim: A wavelet-based multi-scale similarity metric for weather and climate fields. *arXiv preprint arXiv:2512.14656*, 2025.

[38] Eero P. Simoncelli and Edward H. Adelson. Noise removal via bayesian wavelet coring. *Proceedings of the IEEE International Conference on Image Processing*, 1:379–382, 1996.

[39] David L. Donoho and Iain M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994.

[40] 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*, pages 770–778, 2016.

[41] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 8780–8794. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper_files/paper/2021/file/49ad23d1ec9fa4bd8d77d02681df5cfa-Paper.pdf`.

[42] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. Film: visual reasoning with a general conditioning layer. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018. ISBN 978-1-57735-800-8.

[43] Drummond B Fielding, Eve C Ostriker, Greg L Bryan, and Adam S Jermyn. Multiphase gas and the fractal nature of radiative turbulent mixing layers. *The Astrophysical Journal Letters*, 894(2):L24, 2020.

[44] Keaton J Burns, Geoffrey M Vasil, Jeffrey S Oishi, Daniel Lecoanet, and Benjamin P Brown. Dedalus: A flexible framework for numerical simulations with spectral methods. *Physical Review Research*, 2(2):023068, 2020.

[45] Suryanarayana Maddu, Scott Weady, and Michael J Shelley. Learning fast, accurate, and stable closures of a kinetic theory of an active fluid. *Journal of Computational Physics*, 504:112869, 2024.

[46] Haixin Wang, Jiashu Pan, Hao Wu, Fan Zhang, and Tailin Wu. Fourierflow: Frequency-aware flow matching for generative turbulence modeling, 2025. URL `https://arxiv.org/abs/2506.00862`.

[47] Peiyan Hu, Rui Wang, Xiang Zheng, Tao Zhang, Haodong Feng, Ruiqi Feng, Long Wei, Yue Wang, Zhi-Ming Ma, and Tailin Wu. Wavelet diffusion neural operator. *arXiv preprint arXiv:2412.04833*, 2024.

[48] 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.

[49] 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.

[50] Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

[51] Shinya Maeyama and Tomo-Hiko Watanabe. Extracting and modeling the effects of small-scale fluctuations on large-scale fluctuations by mori–zwanzig projection operator method. *Journal of the Physical Society of Japan*, 89(2):024401, 2020.

[52] Benjamin Sanderse, Panos Stinis, Romit Maulik, and Shady E Ahmed. Scientific machine learning for closure models in multiscale problems: A review. *arXiv preprint arXiv:2403.02913*, 2024.

[53] Juan Nathaniel, Carla Roesch, Jatan Buch, Derek DeSantis, Adam Rupe, Kara D Lamb, and Pierre Gentine. Deep koopman operators for causal discovery. *Communications Physics*, 8(1): 513, 2025.

[54] Vivek Oommen, Aniruddha Bora, Zhen Zhang, and George Em Karniadakis. Integrating neural operators with diffusion models improves spectral representation in turbulence modeling. *arXiv preprint arXiv:2409.08477*, 2024.

[55] Jeremy Andrew Irvin, Jiaqi Han, Zikui Wang, Abdulaziz Alharbi, Yufei Zhao, Nomin-Erdene Bayarsaikhan, Daniele Visioni, Andrew Y Ng, and Duncan Watson-Parris. Spatiotemporal pyramid flow matching for climate emulation. *arXiv preprint arXiv:2512.02268*, 2025.

[56] Florentin Guth, Simon Coste, Valentin De Bortoli, and Stephane Mallat. Wavelet score-based generative modeling. *Advances in neural information processing systems*, 35:478–491, 2022.

[57] Wael Mattar, Idan Levy, Nir Sharon, and Shai Dekel. Wavelets are all you need for autoregressive image generation. *arXiv preprint arXiv:2406.19997*, 2024.

[58] Luigi Sigillo, Shengfeng He, and Danilo Comminiello. Latent wavelet diffusion for ultra high-resolution image synthesis. In *International Conference on Learning Representations*, volume 2026, pages 132796–132825, 2026.

[59] Jyoti Rani, Tapas Tripura, Hariprasad Kodamana, and Souvik Chakraborty. Generative adversarial wavelet neural operator with applications to fault detection and isolation of multivariate time series data. *Control Engineering Practice*, 165:106548, 2025. ISSN 0967-0661. doi: https://doi.org/10.1016/j.conengprac.2025.106548. URL `https://www.sciencedirect.com/science/article/pii/S0967066125003107`.

[60] Matthew A. Price, Alicja Polanska, Jessica Whitney, and Jason D. McEwen. Differentiable and accelerated wavelet transforms on the sphere and ball, 2024. URL `https://arxiv.org/abs/2402.01282`.

[61] Davide Donno, Italo Epicoco, Massimo Cafaro, Gabriele Accarino, Mohammad M. Amirian, Viviana Acquaviva, Paola Nassisi, Doroteaciro Iovino, Annalisa Bracco, Simona Masina, and Pierre Gentine. Neptune: An AI model for global ocean subseasonal prediction, 2026. URL `https://arxiv.org/abs/2609.08606`.

[62] Lord Rayleigh. Lix. on convection currents in a horizontal layer of fluid, when the higher temperature is on the under side. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 32(192):529–546, 1916.

[63] Baole Wen, David Goluskin, and Charles R Doering. Steady rayleigh–bénard convection between no-slip boundaries. *Journal of Fluid Mechanics*, 933:R4, 2022.

[64] Christopher A. T. Ferro. Fair scores for ensemble forecasts. *Quarterly Journal of the Royal Meteorological Society*, 140(683):1917–1923, 2014.

[65] Ingrid Daubechies. Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 41(7):909–996, 1988. doi: 10.1002/cpa.3160410705.

[66] Gregory R. Lee, Ralf Gommers, Filip Waselewski, Kai Wohlfahrt, and Aaron ;Leary. Pywavelets: A python package for wavelet analysis. *Journal of Open Source Software*, 4(36):1237, 2019. doi: 10.21105/joss.01237. URL `https://doi.org/10.21105/joss.01237`.

A Datasets

For all PDE datasets, each field is standardized using the mean and standard deviation computed over the training set. Each dataset also provides constant scalar parameters (see Table S1), denoted κ, which are linearly embedded and injected into every residual block of the velocity-predicting U-Net via AdaGN conditioning (Appendix B.1). For the Rayleigh–Bénard dataset, the Rayleigh and Prandtl numbers are log-transformed prior to embedding.

Table S1: Characteristics of the PDE systems considered in this study. Additional details can be found in [5].

	Turbulent Radiative Layer 2D	**Rayleigh–Bénard**	**Active Matter**
Fields	density, pressure, velocities (x, y)	buoyancy, pressure, velocities (x, y)	concentration, velocities (x, y), D_{xx}, D_{xy}, D_{yx}, D_{yy}, E_{xx}, E_{xy}, E_{yx}, E_{yy}
Scalar constants (κ)	t_{cool}	Ra, Pr	α, ζ
Resolution (pixel)	384×128	512×128	256×256
# Channels (C)	4	4	11
# Trajectories	90	1750	225
# Steps per trajectory	101	200	81
Size (GB)	6.9	342	51.3
Software	Athena++	Dedalus	Python
Reference	[43]	[44]	[45]

Turbulent Radiative Layer 2D. This PDE [43] simulates two-dimensional radiative turbulent mixing layers, a ubiquitous phenomenon in astrophysical environments such as the circum-galactic medium, where cold dense gas and hot dilute gas move relative to each other at subsonic velocities. The velocity shear drives the Kelvin–Helmholtz instability, populating intermediate-temperature gas through turbulent mixing; this mixed gas cools rapidly, driving a net mass flux from the hot to the cold phase. The evolution is governed by the compressible Euler equations with a radiative cooling source term,

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho v) = 0 \tag{14}$$

$$\frac{\partial \rho v}{\partial t} + \nabla \cdot (\rho v v + P) = 0 \tag{15}$$

$$\frac{\partial E}{\partial t} + \nabla \cdot [(E + P)v] = -E/t_{\text{cool}} \tag{16}$$

$$E = P/(\gamma - 1), \qquad \gamma = 5/3 \tag{17}$$

with ρ the density, v the 2D velocity, P the pressure, E the total energy, and t_{cool} is the radiative cooling time. The dataset comprises 90 trajectories spanning nine values of $t_{\text{cool}} = \{0.03, 0.06, 0.1, 0.18, 0.32, 0.56, 1.00, 1.78, 3.16\}$ (ten random seeds each), simulated on a 384×128 uniform Cartesian grid and stored at 101 snapshots. The fields available are density, pressure, and velocity components, providing a challenging testbed for emulators across a wide range of cooling-to-mixing dynamical regimes.

Rayleigh–Bénard Convection. Rayleigh–Bénard Convection (RBC) [62, 63] describes the dynamics of a horizontal fluid layer heated from below and cooled from above, where buoyancy-driven instabilities generate convection currents with rising warm fluid and sinking cold fluid, governed by

$$\frac{\partial b}{\partial t} - \kappa \Delta b = -u \cdot \nabla b, \tag{18}$$

$$\frac{\partial u}{\partial t} - \nu \Delta u + \nabla p - b \mathbf{e}_z = -u \cdot \nabla u, \tag{19}$$

where b denotes buoyancy, u the velocity field, and p the pressure. The thermal diffusivity κ and viscosity ν are parameterized by the Rayleigh and Prandtl numbers as

$$\kappa = (\text{Ra} \cdot \text{Pr})^{-1/2}, \quad \nu = (\text{Ra}/\text{Pr})^{-1/2}. \tag{20}$$

The system exhibits strongly nonlinear and chaotic behavior, transitioning between laminar and turbulent regimes, making it a challenging benchmark for PDE emulation. The 2D state is represented by two scalar fields (buoyancy and pressure) and one vector field (velocity components), discretized on a 512×128 grid ($C = 4$ channels). We gather data from The Well, selecting all the 200 timesteps ($\Delta t = 1$) available to evaluate the models along long rollouts. Each trajectory is associated with a distinct combination of Rayleigh and Prandtl numbers, providing a diverse range of dynamical regimes for training and evaluation.

Active Matter. Active matter [45] describes the dynamics of N elongated active particles of length ℓ and thickness b ($\ell/b \gg 1$) immersed in a Stokes fluid of volume V. Individual particles convert chemical energy into mechanical work, generating persistent active stresses that give rise to complex, non-equilibrium dynamics. In the continuum limit, these systems are described by a kinetic theory for the distribution function $\Psi(\mathbf{x}, \mathbf{p}, t)$ governed by the Smoluchowski equation,

$$\frac{\partial \Psi}{\partial t} + \nabla_{\mathbf{x}} \cdot (\dot{\mathbf{x}}\Psi) + \nabla_{\mathbf{p}} \cdot (\dot{\mathbf{p}}\Psi) = 0, \tag{21}$$

which ensures particle number conservation. The conformational fluxes $\dot{\mathbf{x}}$ and $\dot{\mathbf{p}}$ arise from single-particle dynamics in a background flow $\mathbf{u}(\mathbf{x}, t)$ and, for dense suspensions, take the form

$$\dot{\mathbf{x}} = \mathbf{u} - d_T \nabla_{\mathbf{x}} \log \Psi, \tag{22}$$

$$\dot{\mathbf{p}} = (\mathbf{I} - \mathbf{pp}) \cdot (\nabla \mathbf{u} + 2\zeta \mathbf{D}) \cdot \mathbf{p} - d_R \nabla_{\mathbf{p}} \log \Psi, \tag{23}$$

where d_T and d_R are translational and rotational diffusion coefficients, ζ controls alignment through steric interactions, and $\mathbf{D} = \langle \mathbf{pp} \rangle$ is the second-moment tensor. Macroscopic fields are obtained as moments of Ψ: the concentration $c = \langle 1 \rangle$, polarity $\mathbf{n} = \langle \mathbf{p} \rangle / c$, and nematic order parameter $\mathbf{Q} = \langle \mathbf{pp} \rangle / c$, with $\langle f \rangle = \int_{|\mathbf{p}|=1} f \Psi \, d\mathbf{p}$. The particle dynamics are coupled to the surrounding incompressible Stokes flow,

$$-\Delta \mathbf{u} + \nabla P = \nabla \cdot \mathbf{\Sigma}, \qquad \nabla \cdot \mathbf{u} = 0, \tag{24}$$

where the stress tensor

$$\mathbf{\Sigma} = \alpha \mathbf{D} + \beta \mathbf{S} : \mathbf{E} - 2\zeta\beta(\mathbf{D} \cdot \mathbf{D} - \mathbf{S} : \mathbf{D}) \tag{25}$$

captures contributions from active dipoles (α, the dipole strength), particle interactions (β, the particle density), and steric torques. Here $\mathbf{E} = (\nabla \mathbf{u} + \nabla \mathbf{u}^\top)/2$ is the rate-of-strain tensor and $\mathbf{S} = \langle \mathbf{pppp} \rangle$ the fourth-moment tensor. The dataset comprises 225 trajectories (5 random initializations for each of the 45 parameter sets) on a uniform 256×256 Cartesian grid with domain size $L_x = L_y = 10$ and periodic boundary conditions, stored at 81 snapshots separated by $\Delta t = 0.25$ seconds over a total window of $t \in [0, 20]$ s. The available fields are concentration, velocity components, orientation tensor, and strain-rate tensor. The parameter space covers active dipole strengths $\alpha \in \{-1, -2, -3, -4, -5\}$ and alignment coefficients $\zeta \in \{1, 3, 5, 7, 9, 11, 13, 15, 17\}$ at fixed $\beta = 0.8$, spanning regimes from disordered isotropic motion to coherent nematic collective flows, and constituting a challenging benchmark for surrogate modelling of long-horizon non-equilibrium dynamics.

B Models

This section details the model settings and hyperparameters used for training and inference. Unless stated otherwise, hyperparameters are shared across all PDE benchmarks; for WFM, the only system-specific choice is the mother wavelet (Table S11). FNO and TFNO were retrained with the hyperparameters pre-tuned by The Well [5]. WNO and WDNO follow their original papers, with the network width enlarged to approach the parameter budget of WFM and the adaptations to autoregressive emulation described in the corresponding paragraphs below; for WNO, all fine-scale wavelet coefficients are retained rather than discarded. At inference time, generative models produce $\mathcal{M} = 8$ trajectories ($\mathcal{M} = 5$ for WDNO) from the same initial condition; at every autoregressive step, each trajectory draws fresh Gaussian noise and is integrated with the learned ODE for $N = 50$ steps (flow-matching models) or sampled with deterministic DDIM for $N = 4$ steps (WDNO).

How the baselines differ. The generative baselines isolate the choice of representation. FM_{pixel} uses the same flow-matching objective, sampler and U-Net as WFM-1 (12.7M parameters each) but generates in physical space. $\text{FM}_{\text{fourier}}$ generates the leading Fourier modes and discards the rest. WDNO denoises a whole space–time block of wavelet coefficients with a diffusion model. $\text{LatentFM}_{\div 64}$ generates in the latent space of a pre-trained autoencoder with a 64-fold compression. WFM-3 (30.6M) is compared with $\text{FM}_{\text{fourier}}$ (30.3M) at a matched parameter budget.

B.1 Velocity-predicting U-Net

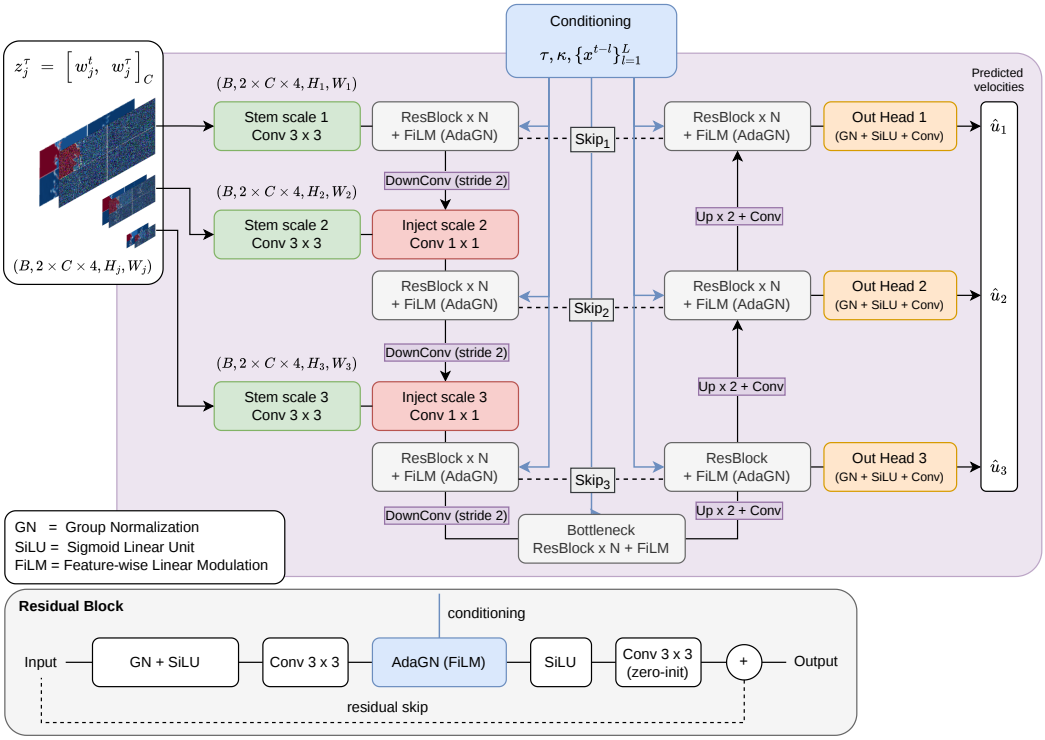


Figure S1: Illustration of the velocity-predicting U-Net u_θ.

The velocity field u_θ is parameterized by a U-Net (depicted in Figure S1) whose depth and wavelet injection points are controlled independently by the number of wavelet scales (n_{scales}), and the number of encoder/decoder levels (n_{levels}), with $n_{\text{levels}} \geq n_{\text{scales}}$. This decoupling allows the same architecture to be used across all experimental conditions without structural changes.

Stems and injection. Each wavelet scale j is processed by a dedicated *stem*, a single 3×3 convolution that projects the channel-wise concatenation of the current-state coefficients w_j^t and the noisy interpolant w_j^τ into the encoder feature space. The stem at scale $j = 1$ initializes the encoder,

while stems at scales $j > 1$ are fused into the encoder via a 1×1 convolution after the corresponding downsampling step, injecting multi-scale structure progressively into the representation.

Encoder, bottleneck, and decoder. Each encoder level applies three pre-activation residual blocks followed by a stride-2 convolution that halves the spatial resolution and doubles the channel width, up to a maximum of $8 \times C_{\text{base}}$. The bottleneck operates at the coarsest resolution with at least two residual blocks, providing a wide effective receptive field. The decoder mirrors the encoder with bilinear upsampling followed by a 3×3 convolution, and restores resolution by concatenating skip connections from the corresponding encoder level. A lightweight output head (GroupNorm–SiLU–Conv) is applied at each decoder level that corresponds to a wavelet scale, producing one velocity prediction per scale ($\hat{u}_j$).

Conditioning. All residual blocks use AdaGN [41]: the conditioning vector $\mathbf{c} \in \mathbb{R}^d$, obtained by summing a sinusoidal flow-time embedding, constant parameters of the PDE (κ), and the spatiotemporal context embedding of the past $L = 3$ states, modulates the feature activations via learned scale and shift parameters. Throughout the experiments we set $d = 256$.

Special cases. Setting $n_{\text{scales}} = n_{\text{levels}}$ recovers a standard multi-scale wavelet U-Net in which every encoder level receives the wavelet representation at the appropriate scale. Setting $n_{\text{scales}} = 1$ disables all injections beyond the input stem, reducing the model to a plain U-Net operating on the finest wavelet scale only ($j = 1$). This is the configuration used for WFM 1-scale variants. Instead, FM_{pixel} uses the same architecture with wavelet injection disabled, and the stem operating directly on the full pixel grid.

Cross-scale coupling. Because every head $\hat{u}_j$ is computed by the shared encoder–decoder, the J flows are not independent. We quantify this coupling for WFM 3-scale (haar) on TRL. If the flows were independent, each head would depend only on its own scale, i.e. $\partial \hat{u}_j / \partial w^\tau_{j'} = 0$ for $j \neq j'$. We therefore perturb the interpolant w^τ at a single scale j' and measure the relative change of the velocity predicted by every head j:

$$S_{jj'} = \frac{1}{P} \sum_{p=1}^{P} \frac{\left\| \hat{u}_j\big(w^\tau + \delta\, v^{(p)}_{j'}\big) - \hat{u}_j\big(w^\tau\big) \right\|_{\text{RMS}}}{\delta \left\| \hat{u}_j\big(w^\tau\big) \right\|_{\text{RMS}}}. \tag{26}$$

Here $v^{(1)}_{j'}, \ldots, v^{(P)}_{j'}$ are $P = 16$ random perturbation directions, each drawn from $\mathcal{N}(0, I)$ on the coefficients of scale j', set to zero on all other scales, and rescaled to the RMS of $w^\tau_{j'}$. The scalar δ sets the perturbation size relative to that RMS; we use $\delta = 10^{-2}$. $S_{jj'}$ is thus the relative change of head j per unit relative perturbation of scale j'. We average it over 8 time windows of a test trajectory and five interpolation times $\tau \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, and summarize the resulting matrix by its off-diagonal mass $\rho = \sum_{j \neq j'} S_{jj'} / \sum_{j,j'} S_{jj'}$, which is exactly zero for independent flows.

We obtain $\rho = 15.1\%$ (Table S2), and between 14.5% and 15.3% at each individual τ. At $\tau = 0.5$, repeating the measurement with $\delta = 10^{-3}$ and $\delta = 10^{-1}$ changes ρ by at most 0.3 percentage points, so the result does not depend on the perturbation size. Evaluating the network twice on identical inputs gives a difference of zero to machine precision, so the off-diagonal entries are not numerical noise. The coupling runs in both directions: coarse-to-fine entries (above the diagonal) average 0.271, and fine-to-coarse entries (below the diagonal) 0.138. This measures the local sensitivity of the learned velocity field; it does not show that the coupling reproduces the correct joint statistics across scales.

B.2 Wavelet Flow Matching

To ensure a fair comparison across scale configurations, we normalize model capacity. The 3-scale WFM employs a shared U-Net backbone with three output heads, making it an inherently harder multi-target task compared to single-scale prediction. In U-Net architectures, capacity is primarily driven by the convolutions in residual blocks [40], which scale as $\mathcal{O}(\text{init_dim}^2)$, where init_dim denotes the number of feature channels in the first layer, assuming input and output channels grow proportionally to init_dim. We therefore equalize capacity on a per-head basis. In particular, for init_dim $= 64$, the per-head capacity is proportional to $64^2/3 \approx 1365$, which corresponds to init_dim $= 64/\sqrt{3} \approx 37$

Table S2: Cross-scale sensitivity $S_{jj'}$ (Equation 26) of WFM 3-scale (haar) on TRL, with $\delta = 10^{-2}$, averaged over test windows and τ. Rows: affected head $\hat{u}_j$; columns: perturbed scale $w_{j'}^{\tau}$ ($j = 1$ finest). Independent per-scale flows would give zero off-diagonal entries; the measured off-diagonal mass is $\rho = 15.1\%$.

	w_1^{τ} (fine)	w_2^{τ} (mid)	w_3^{τ} (coarse)
$\hat{u}_1$ (fine)	2.292	0.225	0.277
$\hat{u}_2$ (mid)	0.135	2.366	0.311
$\hat{u}_3$ (coarse)	0.095	0.183	2.243

for single-head models (e.g., WFM 1-scale). We round this value to init_dim $= 40$, the nearest multiple of 8 above 37, to satisfy divisibility by 8 of GroupNormalization layers. WFM 1-scale and FM_{pixel} therefore use init_dim $= 40$, while $\text{FM}_{fourier}$ uses init_dim $= 64$ to match WFM 3-scale.

Table S3: Hyperparameters for the WFM configurations.

	WFM 1-scale	WFM 3-scale
Architecture	U-Net	U-Net
Initial dimension	40	64
Residual blocks	3	3
Embedding dimension	256	256
Parameters	12.7×10^6	30.6×10^6
Wavelet scales	1	3
Mother wavelet	PDE-specific	PDE-specific
Padding mode	periodic	periodic
Scales weighting scheme	1.0 for all scales	1.0 for all scales
Optimizer	AdamW	AdamW
Learning rate	3×10^{-4}	3×10^{-4}
Weight decay	1×10^{-2}	1×10^{-2}
Betas	$(0.9, 0.999)$	$(0.9, 0.999)$
Scheduler	CosineAnnealingLR	CosineAnnealingLR
Eta min	1×10^{-7}	1×10^{-7}
Epochs	200	200
Warm-up epochs	20	20
Batch size	32	32
Gradient clipping	1.0	1.0
Precision	`bf16-mixed`	`bf16-mixed`
Matmul precision	`high`	`high`
Strategy	DDP	DDP
Framework	PyTorch Lightning	PyTorch Lightning
GPUs	$4\times$ A100 (40GB)	$4\times$ A100 (40GB)

B.3 Flow Matching in Pixel Space

Table S4: Hyperparameters for the FM_{pixel} configuration.

	FM_{pixel}
Architecture	U-Net
Initial dimension	40
Residual blocks	3
Embedding dimension	256
Parameters	12.7×10^6
Optimizer	AdamW
Learning rate	6×10^{-4}
Weight decay	1×10^{-2}
Betas	$(0.9, 0.999)$
Scheduler	CosineAnnealingLR
Eta min	1×10^{-7}
Epochs	200
Warm-up epochs	20
Batch size	16
Gradient clipping	1.0
Precision	`bf16-mixed`
Matmul precision	`high`
Strategy	DDP
Framework	PyTorch Lightning
GPUs	$4\times$ A100 (40GB)

B.4 Flow Matching in Fourier Space

Table S5: Hyperparameters for the $\text{FM}_{fourier}$ configuration.

	$\text{FM}_{fourier}$
Architecture	U-Net
Initial dimension	64
Residual blocks	3
Embedding dimension	256
Parameters	30.3×10^6
Fourier Modes (H, W)	(32, 32)
Optimizer	AdamW
Learning rate	1×10^{-4}
Weight decay	1×10^{-2}
Betas	$(0.9, 0.999)$
Scheduler	CosineAnnealingLR
Eta min	1×10^{-7}
Epochs	200
Warm-up epochs	20
Batch size	32
Gradient clipping	1.0
Precision	`bf16-mixed`
Matmul precision	`high`
Strategy	DDP
Framework	PyTorch Lightning
GPUs	$4\times$ A100 (40GB)

B.5 Wavelet Neural Operator

Our implementation keeps the core WNO operator structure [49], adapting the wrapper to the WFM benchmark: we flatten history and field channels as model input, append available physical scalar conditions through the benchmark data interface, and evaluate using identical WFM normalization and rollout pipeline. For Rayleigh–Bénard we apply a logarithmic transform to the physical parameters; for TRL and active matter we use the default scalar representation. Hyperparameters are reported in Table S6.

Table S6: Hyperparameters for the WNO configuration.

	WNO
Hidden width	32
Layers	4
Wavelet scales	3
Mother wavelet	$\mathrm{db2}$
Padding mode	periodic
Input steps	4
Output steps	1
Optimizer	AdamW
Learning rate	1×10^{-3}
Weight decay	1×10^{-6}
Betas	$(0.9,\ 0.999)$
Scheduler	CosineAnnealingLR
Eta min	1×10^{-6}
Epochs	200
Early-stopping patience	20
Batch size	16
Framework	PyTorch Lightning
GPUs	$1\times$ A100 (40GB)

B.6 Wavelet Diffusion Neural Operator

Our WDNO baseline [47] is a diffusion model that operates in wavelet space. We adapt it to the $4 \to 1$ setting used here as follows. The input to the model is the full spatiotemporal block $[t_1, t_2, t_3, t_4, t_5]$, transformed via a 3D DWT over time and space; a 3D U-Net is then trained to denoise in this wavelet state space. For conditioning, the observed steps $[t_1, t_2, t_3, t_4]$ are provided as fixed channels, with known components re-imposed at each sampling step. We use the $\mathrm{db2}$ mother wavelet with periodic padding, u-prediction, deterministic DDIM sampling, and EMA weights at inference. Full hyperparameters are reported in Table S7.

Table S7: Hyperparameters for the WDNO configuration.

	WDNO
Architecture	U-Net
Initial dimension	64
Residual blocks	1
U-Net resolution levels	3
Conditioning dimension	128
Input steps	4
Output steps	1
Mother wavelet	$\mathrm{db2}$
Padding mode	periodic
Wavelet scales	1
Diffusion steps	4
Sampling steps	4
Sampler	DDIM ($\eta = 0$)
Beta schedule	Cosine
x_0 clipping	$[-1, 1]$
EMA decay	0.995
Optimizer	AdamW
Learning rate	1×10^{-4}
Weight decay	1×10^{-6}
Betas	$(0.9, 0.999)$
Scheduler	CosineAnnealingLR
Eta min	1×10^{-6}
Epochs	200
Early-stopping patience	20
Batch size	4
Precision	`fp32`
Framework	PyTorch Lightning
GPUs	$1\times$ A100 (40GB)

B.7 Fourier-based Neural Operators

Table S8: Hyperparameters for the FNO and TFNO configurations.

	FNO	TFNO
Architecture	FNO	TFNO
Hidden dimension	128	128
Fourier blocks	4	4
Parameters	19×10^6	7.9×10^6
Fourier Modes (H, W)	(16, 16)	(16, 16)
Tucker rank	-	0.42
Learning rate	5×10^{-3}	5×10^{-3}
Epochs	500	500
Batch size	16	16
Early stopping patience	20	20
Early stopping min delta	1×10^{-6}	1×10^{-6}
Precision	`32-true`	`32-true`
Matmul precision	`high`	`high`
Strategy	DDP	DDP
Framework	PyTorch Lightning	PyTorch Lightning
GPUs	2× A100 (40GB)	2× A100 (40GB)

B.8 Latent Flow Matching

LatentFM$_{\div 64}$ follows Rozet et al. [21] and is evaluated on Rayleigh–Bénard only, the system for which their pre-trained autoencoder is available. The autoencoder (DC-AE f32c64, 313.1×10^6 parameters) compresses each $512 \times 128 \times 4$ state into a $16 \times 4 \times 64$ latent, a 64-fold reduction, and is kept frozen; its encoder and decoder are called once per autoregressive step. Only the latent flow model is trained (8.4×10^6 parameters), with the same OT-CFM objective as WFM and the hyperparameters in Table S9. In terms of cost, the autoencoder required about one day of pre-training on 8 H100 GPUs [21], whereas WFM needs no pre-training stage and trains in about 6 hours on 4 A100 GPUs.

Table S9: LatentFM$_{\div 64}$ hyperparameters.

Autoencoder	DC-AE f32c64, pre-trained [21], frozen
Latent shape	$16 \times 4 \times 64$
U-Net	base channels 64, 2 levels, 3 res. blocks, cond. dim 256
Parameters	8.4×10^6 trainable (flow model); 313.1×10^6 frozen (autoencoder)
Optimizer	AdamW, lr 10^{-4}, weight decay 10^{-2}
Schedule	20 warm-up epochs, cosine decay to 10^{-7}
Epochs / batch size	200 / 32
Precision	bf16, DDP
Sampler	Euler, $N = 50$ steps, $\mathcal{M} = 8$ trajectories

C Metrics

Let $u \in \mathbb{R}^{H\times W}$ denote the ground-truth field at a given rollout step, and $\langle\cdot\rangle$ the mean over the spatial dimensions. A generative model returns the predictions $v_1, \ldots, v_{\mathcal{M}}$ of its $\mathcal{M}$ sampled trajectories at that step ($\mathcal{M} = 8$, or $\mathcal{M} = 5$ for WDNO). For simplicity of notation, we write $v \in \mathbb{R}^{H\times W}$ for the prediction: for deterministic models it is the model output, and for generative models it is the sample mean $\frac{1}{\mathcal{M}}\sum_m v_m$. CRPS is computed from the individual samples v_m, and for spectral coherence the power and cross-spectra are computed for each v_m and then averaged over the samples.

Variance-normalized Root Mean Squared Error (VRMSE). The RMSE of the prediction v, normalized by the spatial standard deviation of the reference field,

$$\mathrm{VRMSE}(u, v) = \sqrt{\frac{\big\langle (u - v)^2 \big\rangle}{\big\langle (u - \langle u\rangle)^2\big\rangle + \epsilon}} \tag{27}$$

where $\epsilon = 10^{-6}$ ensures numerical stability. A prediction equal to the spatial mean of u yields $\mathrm{VRMSE} \approx 1$, providing a meaningful baseline [5, 21]. Unlike the normalised RMSE (NRMSE), which divides by $\langle u^2\rangle$, VRMSE avoids down-weighting errors in non-negative fields (e.g. pressure and density) [21].

Fair Continuous Ranked Probability Score (CRPS). The CRPS generalizes the mean absolute error (MAE) to probabilistic forecasts by scoring an entire predictive distribution against an observation [50]. For $\mathcal{M}$ samples $\{v_m\}$, the unbiased ("fair") estimator of Ferro [64] is

$$\mathrm{CRPS}_{\mathrm{fair}}(u, \{v_m\}) \;=\; \frac{1}{\mathcal{M}}\sum_{m=1}^{\mathcal{M}} \langle |v_m - u|\rangle \;-\; \frac{1}{\mathcal{M}(\mathcal{M}-1)} \sum_{1\le m<m'\le \mathcal{M}} \langle |v_m - v_{m'}|\rangle, \tag{28}$$

The first term penalizes deviations of individual samples from the truth; the second rewards the spread between samples. Equation (28) is an unbiased estimator of the CRPS of the distribution from which the samples are drawn [64]; the unadjusted $1/\mathcal{M}^2$ weight favours under-dispersed samples. Because the CRPS is a strictly proper scoring rule, its expected value is minimized when the samples are drawn from the true conditional distribution of u [50]. For turbulent flows, CRPS measures whether the model correctly represents the range of possible realizations rather than just their mean, making it essential when trajectories diverge chaotically after a finite predictability horizon.

Spectral Coherence RMSE. Let $P_u(k)$ and $P_v(k)$ denote the isotropic power spectral densities (PSDs) of fields u and v respectively, at radial wavenumber k (cycles pixel^{-1}), obtained by azimuthal averaging of the 2-D DFT magnitude squared. Let $C_{uv}(k) = |\hat{u}(k)\,\overline{\hat{v}(k)}|$ be the corresponding cross-spectrum magnitude. The spectral coherence at wavenumber k is

$$\gamma(k) = \frac{C_{uv}(k) + \varepsilon}{\sqrt{P_u(k)\,P_v(k)} + \varepsilon}, \tag{29}$$

where $\varepsilon = 10^{-6}$. By the Cauchy–Schwarz inequality, $\gamma(k) \in [0, 1]$. We distribute the resulting wavenumbers evenly in log-space and divide into three frequency bands (*Low*, *Mid*, *High*), then for each band:

$$\mathrm{RMSE}_c(\mathcal{B}) = \sqrt{\frac{1}{|\mathcal{B}|}\sum_{k\in\mathcal{B}} \big(1 - \gamma(k)\big)^2}. \tag{30}$$

$\mathrm{RMSE}_c \approx 0$ indicates that v reproduces the correct phase and amplitude structure of u at each resolved scale within the band; $\mathrm{RMSE}_c = 1$ (the maximum, by Cauchy–Schwarz) indicates the two fields are spectrally uncorrelated across the band. For turbulent flows energy cascades from large to small scales, so coherence typically degrades first at high wavenumbers; RMSE_c therefore quantifies whether the model respects the inertial-range cascade structure at each resolved scale, complementing the point-wise VRMSE which loses sensitivity once trajectories have diverged chaotically. We prefer Spectral Coherence RMSE over Power Spectrum RMSE, which is defined as in Equation (30) but replacing $\gamma(k)$ with the ratio $P_v(k)/P_u(k)$. This choice is motivated by the fact that Power Spectrum RMSE is unbounded and numerically unstable when $P_u(k) \approx 0$, which is the case of near-zero-mean fields in some PDEs considered in this study.

D Mother Wavelets

All four wavelets used in this study belong to the Daubechies family [65], which constructs compactly supported orthonormal bases of $L^2(\mathbb{R})$. A family $\{\psi_{j,n}\}_{(j,n)\in\mathbb{Z}^2}$ is *orthonormal* when

$$\langle \psi_{j,n},\, \psi_{j',n'} \rangle = \delta_{jj'}\, \delta_{nn'}, \tag{31}$$

with δ_{ij} the Kronecker delta (1 if $i = j$, 0 otherwise). This means that the basis functions are both mutually orthogonal and of unit norm. This is strictly stronger than mere orthogonality, where cross inner products vanish but norms need not equal one. Because the basis is orthonormal, the L^2 energy of any signal is exactly preserved in its coefficient sequence (Parseval's identity), so energy-based metrics such as RMSE carry the same meaning in both the signal and the wavelet domain without any additional rescaling.

Vanishing moments and polynomial cancellation. The Daubechies $\mathrm{db}p$ wavelet is the unique compactly supported, orthonormal wavelet that maximises the number of vanishing moments p for a support of length $2p-1$ (equivalently, a filter of length $2p$). Having p vanishing moments means

$$\int_{-\infty}^{\infty} t^k\, \psi(t)\, \mathrm{d}t = 0, \qquad k = 0, \ldots, p-1, \tag{32}$$

so the wavelet transform exactly annihilates polynomials of degree less than p: detail coefficients are identically zero wherever the signal is locally well approximated by such polynomials. Increasing p therefore yields sparser coefficient representations for smooth signals, at the cost of a wider spatial support and a slightly larger convolution per decomposition level.

The Haar wavelet (equivalent to $\mathrm{db}1$) is the simplest member of this family: its scaling function is piecewise constant on $[0, 1)$, giving a support of length one and a single vanishing moment. It is the only Daubechies wavelet that is symmetric and discontinuous, making it computationally cheap but poorly suited to capturing smooth transitions. Instead, $\mathrm{db}2$, $\mathrm{db}4$, and $\mathrm{db}6$ progressively widen the support and increase regularity, suppressing polynomial trends of degree up to 1, 3, and 5, respectively. Table S10 summarises the key properties of the four mother wavelets considered in this work.

Table S10: Key properties of the mother wavelets used in this study. All four belong to the Daubechies ($\mathrm{db}p$) family [65, 35] and form orthonormal bases of $L^2(\mathbb{R})$. Filter length equals $2p$ and support length equals $2p-1$ for $\mathrm{db}p$.

Wavelet	Filter length	Support	Vanishing moments	Polynomials annihilated	Symmetry
db1 (haar)	2	[0, 1]	1	constants ($\deg \leq 0$)	symmetric
db2	4	[0, 3]	2	linear ($\deg \leq 1$)	asymmetric
db4	8	[0, 7]	4	cubic ($\deg \leq 3$)	asymmetric
db6	12	[0, 11]	6	quintic ($\deg \leq 5$)	asymmetric

Discrete Wavelet Transform Implementation. Our DWT is implemented in PyTorch using the `PyWavelets` (`pywt`) [66] backend, using the `periodic` padding mode. The transform operates on the spatial dimensions (H, W) only, leaving the batch, channel, and temporal axes untouched. To enable batched processing, input tensors of shape (B, C, T, H, W) are transiently reshaped to $(B \cdot C \cdot T,\, H, W)$ before each single-level DWT call and restored afterwards.

E Additional Experimental Results

This section provides additional material and analyses complementing the main text. Figures S2, S3, and S4 show WFM variants based on the wavelet selection reported in Section 4.2 and summarized in Table S11. Bold and underlined entries denote the best and second-best results, respectively. All metrics reported in this section are lower-is-better.

Table S11: Optimal mother wavelet per WFM variant and PDE system, selected by lowest spectral coherence RMSE averaged across *Low*, *Mid*, and *High* frequency bands and rollout horizons (Table 1). In parentheses: the selected wavelet's score, mean ± std over three inference seeds (42, 84, 168); the selection is made on the seed mean.

	Turbulent Radiative Layer 2D	Rayleigh–Bénard	Active Matter
WFM 1-scale	haar ($0.147 \pm <0.001$)	db4 (0.121 ± 0.002)	haar (0.241 ± 0.001)
WFM 3-scale	haar ($0.149 \pm <0.001$)	db4 (0.109 ± 0.001)	db2 ($0.220 \pm <0.001$)

VRMSE across autoregressive rollout steps.

Table S12: VRMSE across rollout horizons for all models and PDE systems. Deterministic baselines (WNO, FNO, TFNO) are single runs. LatentFM$_{\div 64}$ is evaluated on Rayleigh–Bénard only. Mean over three inference seeds (42, 84, 168); the standard deviation across seeds is below 5% of the mean for all entries except those marked *. Here, inf indicates that the rollout diverged, while – indicates that the model was not available for the corresponding system.

	Turbulent Radiative Layer 2D			Rayleigh–Bénard			Active Matter		
Rollout	01:32	33:64	65:97	01:32	33:64	65:196	01:16	17:32	33:77
WFM-1 haar	<u>0.544</u>	0.791	0.789	1.782	2.020	0.788	2.847	1.541	1.079
WFM-1 db2	0.550	0.750	0.850	1.333	1.844*	0.791	5.652	2.930	1.622
WFM-1 db4	0.550	<u>0.730</u>	0.811	1.689	2.271	0.802	14.227	5.574	2.488
WFM-1 db6	**0.536**	**0.703**	<u>0.786</u>	1.441	1.949	1.025	6.732	2.927	1.553
WFM-3 haar	0.549	0.824	0.797	1.559	2.318	0.807	<u>0.848</u>	<u>1.327</u>	**1.022**
WFM-3 db2	0.655	0.893	0.981	1.051	1.729	**0.766**	0.933	1.349	1.085
WFM-3 db4	0.586	0.759	0.803	<u>1.029</u>*	1.869	<u>0.770</u>	3.242	2.271	1.418
WFM-3 db6	0.591	0.879	0.831	**0.755**	<u>1.598</u>	0.834	2.341	2.118	1.276
FM_{pixel}	0.574	0.891	0.830	1.319	1.962	0.821	**0.580**	1.380	1.028
$\text{FM}_{\text{fourier}}$	0.619	0.771	**0.782**	1.098	**0.810**	0.839	1.302	**1.135**	<u>1.027</u>
WDNO	1.517	1.834	1.953	19.262	8.171	1.392	4.475	2.181	1.400
LatentFM$_{\div 64}$	–	–	–	9.515	4.080	0.917	–	–	–
WNO	inf	inf	inf	44.226	24.463	2.452	10.669	4.321	2.002
FNO	2.952	4.978	4.349	3.368	3.972	inf	1.682	3.227	1.727
TFNO	2.300	6.534	6.387	102.678	31.842	2.965	4.443	2.968	1.525

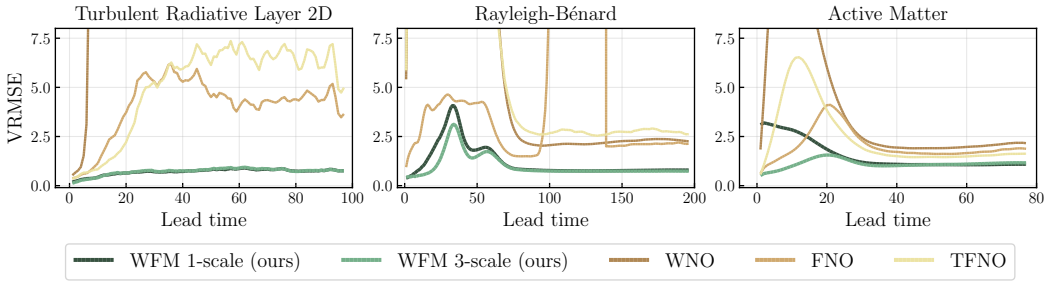


Figure S2: VRMSE across rollout steps for WFM and deterministic baselines per PDE system.

CRPS across autoregressive rollout steps.

Table S13: CRPS across rollout horizons for all models and PDE systems. LatentFM$_{\div 64}$ is evaluated on Rayleigh–Bénard only. Mean over three inference seeds (42, 84, 168); the standard deviation across seeds is below 5% of the mean for all entries except those marked *. Here, – indicates that the model was not available for the corresponding system.

	Turbulent Radiative Layer 2D			Rayleigh–Bénard			Active Matter		
Rollout	01:32	33:64	65:97	01:32	33:64	65:196	01:16	17:32	33:77
WFM-1 haar	0.492	1.047	1.218	0.005	0.037	0.074	0.119	0.190	0.198
WFM-1 db2	0.447	1.097	1.316	0.005	0.036	0.073	0.173	0.239	0.261
WFM-1 db4	0.521	1.052	1.241	0.007	0.041	0.073	0.326	0.388	0.387
WFM-1 db6	0.451	**0.983**	1.188	0.005	0.040	0.090	0.175	0.222	0.231
WFM-3 haar	**0.410**	1.018	1.178	0.004	0.037	0.077	0.080	0.169	**0.191**
WFM-3 db2	0.465	1.133	1.179	**0.003**	0.032	0.069	0.087	0.176	0.196
WFM-3 db4	0.427	0.984	**1.130**	0.004	0.032	**0.069**	0.109	0.209	0.239
WFM-3 db6	0.451	1.026	1.200	0.004	**0.031**	0.079	0.100	0.197	0.213
FM$_{\text{pixel}}$	0.418	1.088	1.210	0.005	0.036	0.076*	**0.059**	**0.166**	0.194
FM$_{\text{fourier}}$	0.857	1.418	1.560	0.009	0.038	0.096	0.185	0.227	0.218
WDNO	2.460	3.374	4.284	0.050	0.102	0.138	0.218	0.245	0.266
LatentFM$_{\div 64}$	–	–	–	0.026	0.057	0.088	–	–	–

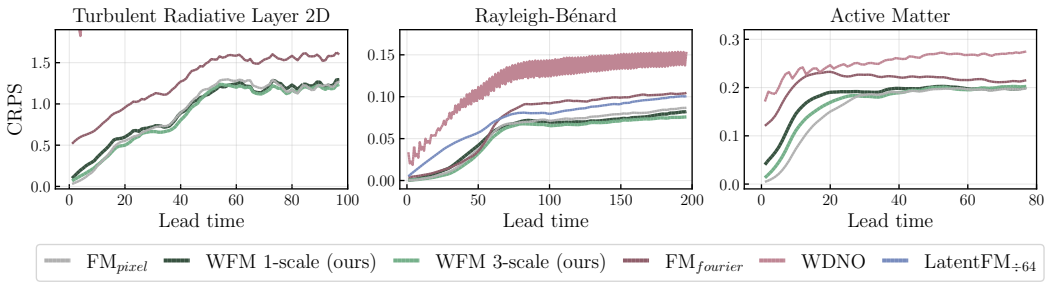


Figure S3: CRPS across rollout steps for WFM and generative baselines per PDE system.

Spectral Coherence RMSE.

Table S14: Spectral coherence RMSE across *Low*, *Mid*, and *High* frequency bands for all models and PDE systems, averaged over rollout horizons. LatentFM$_{\div 64}$ is evaluated on Rayleigh–Bénard only. Mean over three inference seeds (42, 84, 168); the standard deviation across seeds is below 5% of the mean for all entries. Here, – indicates that the model was not available for the corresponding system.

	Turbulent Radiative Layer 2D			Rayleigh–Bénard			Active Matter		
Frequency Bands	Low	Mid	High	Low	Mid	High	Low	Mid	High
WFM-1 haar	<u>0.132</u>	<u>0.175</u>	**0.134**	0.171	0.132	0.116	0.229	0.291	0.204
WFM-1 db2	0.139	0.181	0.142	0.167	0.126	0.105	0.291	0.336	0.236
WFM-1 db4	**0.132**	0.175	0.140	0.168	0.118	<u>0.077</u>	0.306	0.351	0.251
WFM-1 db6	0.133	0.175	0.141	0.196	0.131	0.093	0.283	0.346	0.242
WFM-3 haar	0.133	**0.174**	0.141	0.149	0.130	0.114	0.205	0.280	0.256
WFM-3 db2	0.139	0.188	<u>0.136</u>	**0.141**	0.113	0.079	0.199	0.262	**0.198**
WFM-3 db4	0.141	0.178	0.144	<u>0.143</u>	**0.109**	**0.075**	0.219	0.292	0.235
WFM-3 db6	0.136	0.175	0.148	0.147	<u>0.110</u>	0.080	<u>0.196</u>	**0.258**	0.260
FM$_{\text{pixel}}$	0.141	0.181	0.138	0.149	0.129	0.115	**0.183**	<u>0.258</u>	0.291
FM$_{\text{fourier}}$	0.235	0.487	0.415	0.327	0.263	0.209	0.426	0.463	0.238
WDNO	0.199	0.249	0.366	0.392	0.474	0.535	0.367	0.345	<u>0.200</u>
LatentFM$_{\div 64}$	–	–	–	0.196	0.159	0.131	–	–	–

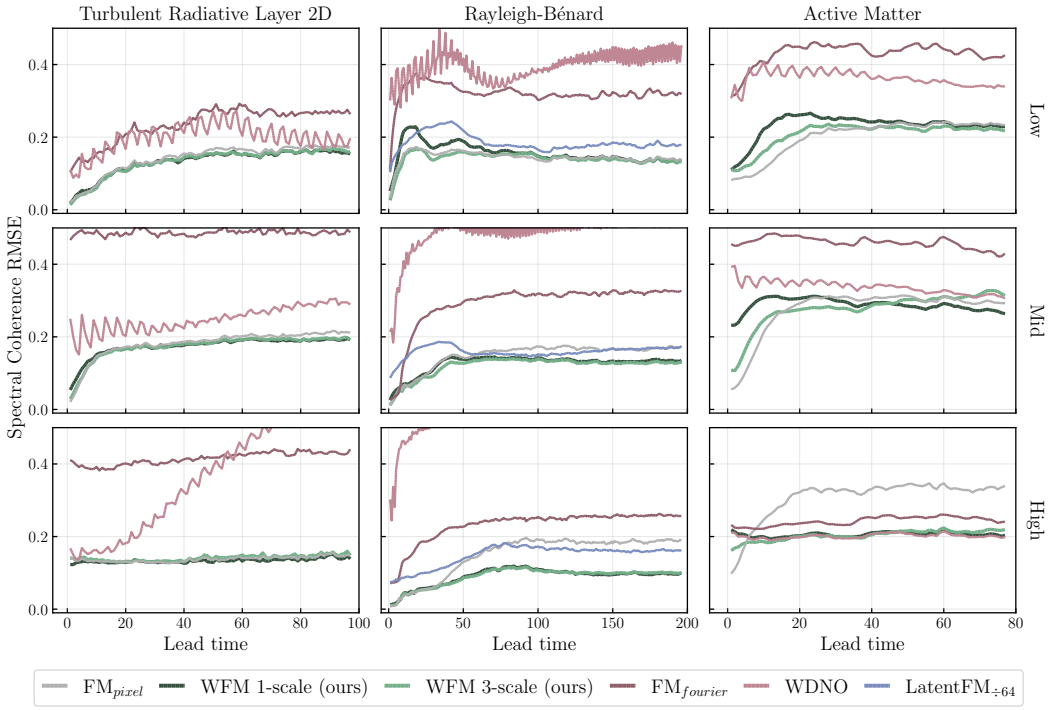


Figure S4: Spectral coherence RMSE across frequency bands and rollout steps for WFM and generative baselines per PDE system.

Turbulent Radiative Layer 2D: sensitivity to cooling time t_{cool}

Table S15: CRPS per t_{cool} on TRL, averaged over rollout steps. Mean over three inference seeds (42, 84, 168); the standard deviation across seeds is below 5% of the mean for all entries except those marked *.

t_{cool}	0.03	0.06	0.1	0.18	0.32	0.56	1	1.78	3.16
WFM-1 haar	1.165	1.135	0.865*	1.044	0.830*	0.798	0.806*	0.983	0.673
WFM-1 db2	1.112	1.078	0.878	1.057	0.880*	0.779	0.853*	0.976*	1.002*
WFM-1 db4	1.170	1.102	0.847*	1.084	0.875	0.849	0.879*	0.946	0.719
WFM-1 db6	1.076	1.074	0.831	1.041	0.823	0.773	0.741	0.943	0.594*
WFM-3 haar	1.107	1.117	0.868	0.988*	**0.773**	0.765*	**0.675**	0.961	0.595
WFM-3 db2	1.248	1.141	0.894	1.027	0.836	0.783*	0.751*	1.019	0.656
WFM-3 db4	**1.048**	**1.006**	**0.817***	**0.982**	0.817	**0.749**	0.692	**0.906**	0.632
WFM-3 db6	1.075	1.034	0.882	1.000*	0.863	0.850	0.770*	0.969	0.617
$\mathrm{FM}_{\mathrm{pixel}}$	1.220	1.126	0.875*	1.061	0.855*	0.783	0.731	0.978	**0.551***
$\mathrm{FM}_{\mathrm{fourier}}$	1.439	1.424	1.195	1.419	1.278	1.226	1.170	1.347	1.032
WDNO	3.330	3.292	3.183	3.419	3.369	3.409	3.459	3.552	3.424

Table S16: Spectral coherence RMSE per t_{cool} and frequency band on TRL. Mean over three inference seeds (42, 84, 168); the standard deviation across seeds is below 5% of the mean for all entries.

Model	t_{cool}	0.03	0.06	0.1	0.18	0.32	0.56	1	1.78	3.16
WFM-1 haar	Low	0.138	0.139	0.138	0.138	**0.132**	0.125	0.127	0.136	0.121
	Mid	0.190	0.194	0.189	0.189	**0.180**	0.172	0.157	0.155	0.150
	High	0.138	0.138	**0.136**	**0.140**	0.147	0.147	0.133	0.120	0.106
WFM-1 db2	Low	0.141	**0.139**	0.140	0.137	0.143	0.134	0.136	0.134	0.149
	Mid	0.192	0.195	0.189	0.193	0.189	0.179	0.170	0.163	0.164
	High	0.142	0.143	0.139	0.145	0.153	0.155	0.145	0.134	0.122
WFM-1 db4	Low	**0.136**	0.141	**0.134**	**0.134**	0.136	0.130	0.127	**0.129**	0.123
	Mid	**0.189**	**0.192**	**0.185**	**0.188**	0.184	0.177	0.162	0.154	0.148
	High	**0.136**	**0.138**	0.137	0.144	0.151	0.158	0.147	0.135	0.112
WFM-1 db6	Low	0.138	0.143	0.139	0.136	0.136	**0.124**	0.128	0.134	0.120
	Mid	0.190	0.196	0.191	0.191	0.183	0.172	0.158	0.151	0.146
	High	0.139	0.140	0.141	0.147	0.154	0.160	0.145	0.131	0.112
WFM-3 haar	Low	0.143	0.144	0.139	0.136	0.133	0.127	0.123	0.132	**0.118**
	Mid	0.194	0.199	0.188	0.190	0.181	**0.167**	**0.153**	**0.150**	**0.144**
	High	0.152	0.152	0.148	0.148	0.157	0.152	0.135	0.123	0.103
WFM-3 db2	Low	0.150	0.146	0.145	0.148	0.141	0.136	**0.123**	0.141	0.123
	Mid	0.193	0.196	0.191	0.201	0.195	0.188	0.174	0.180	0.175
	High	0.152	0.151	0.146	0.143	**0.141**	**0.140**	0.127	0.120	0.105
WFM-3 db4	Low	0.151	0.146	0.141	0.144	0.146	0.137	0.132	0.140	0.131
	Mid	0.195	0.197	0.189	0.192	0.189	0.176	0.162	0.155	0.147
	High	0.149	0.152	0.150	0.154	0.159	0.157	0.141	0.129	0.109
WFM-3 db6	Low	0.145	0.147	0.144	0.139	0.141	0.125	0.132	0.136	0.119
	Mid	0.193	0.198	0.188	0.191	0.181	0.168	0.158	0.152	0.147
	High	0.148	0.150	0.151	0.156	0.170	0.164	0.144	0.134	0.112
$\mathrm{FM}_{\mathrm{pixel}}$	Low	0.144	0.142	0.142	0.143	0.141	0.143	0.137	0.154	0.126
	Mid	0.193	0.199	0.191	0.196	0.186	0.187	0.169	0.163	0.151
	High	0.153	0.157	0.149	0.150	0.153	0.142	**0.126**	**0.113**	**0.098**
$\mathrm{FM}_{\mathrm{fourier}}$	Low	0.242	0.245	0.228	0.249	0.238	0.229	0.229	0.243	0.217
	Mid	0.496	0.497	0.493	0.502	0.493	0.487	0.481	0.468	0.464
	High	0.490	0.481	0.460	0.454	0.413	0.385	0.362	0.349	0.344
WDNO	Low	0.204	0.203	0.194	0.205	0.195	0.192	0.195	0.215	0.190
	Mid	0.253	0.252	0.250	0.246	0.241	0.244	0.247	0.251	0.262
	High	0.359	0.363	0.355	0.363	0.367	0.370	0.376	0.378	0.373

Rayleigh–Bénard: sensitivity to Rayleigh and Prandtl numbers.

Table S17: CRPS per Rayleigh number Ra on Rayleigh–Bénard, averaged over Prandtl numbers. Mean over three inference seeds (42, 84, 168); the standard deviation across seeds is below 5% of the mean for all entries except those marked *.

Ra	10^6	10^8	10^{10}
WFM-1 haar	0.059	0.060*	0.051
WFM-1 db2	**0.052**	0.062*	0.052*
WFM-1 db4	0.059*	0.060	0.051
WFM-1 db6	0.060*	0.069*	0.075
WFM-3 haar	0.060*	0.060	0.055*
WFM-3 db2	0.054*	<u>0.055</u>	<u>0.049</u>
WFM-3 db4	<u>0.053</u>*	0.058	**0.048***
WFM-3 db6	0.065*	**0.055**	0.057
FM_{pixel}	0.057*	0.066*	0.051
$\text{FM}_{\text{fourier}}$	0.071	0.081	0.065
WDNO	0.120	0.117	0.116
$\text{LatentFM}_{\div 64}$	0.087*	0.067	0.065

Table S18: CRPS per Prandtl number Pr on Rayleigh–Bénard, averaged over Rayleigh numbers. Mean over three inference seeds (42, 84, 168); the standard deviation across seeds is below 5% of the mean for all entries except those marked *.

Pr	10^{-1}	1	10^{1}
WFM-1 haar	0.091	0.050*	<u>0.029</u>
WFM-1 db2	0.088*	0.049	0.030
WFM-1 db4	0.088*	<u>0.048</u>	0.034
WFM-1 db6	0.087	0.066*	0.051
WFM-3 haar	0.092*	0.050	0.033
WFM-3 db2	<u>0.079</u>	**0.048**	0.031
WFM-3 db4	**0.079***	0.051*	**0.028**
WFM-3 db6	0.086*	0.050	0.040
FM_{pixel}	0.088*	0.054	0.033
$\text{FM}_{\text{fourier}}$	0.114	0.058	0.045
WDNO	0.146	0.112	0.096
$\text{LatentFM}_{\div 64}$	0.101*	0.062	0.056

Table S19: Spectral coherence RMSE per frequency band on Rayleigh–Bénard, broken down by Rayleigh number Ra (left, averaged over Pr) and Prandtl number Pr (right, averaged over Ra). Mean over three inference seeds (42, 84, 168); the standard deviation across seeds is below 5% of the mean for all entries except those marked *.

(a) vs Ra (averaged over Pr)

Model	Ra	10^6	10^8	10^{10}
WFM-1 haar	Low	0.174	0.166	0.138
	Mid	0.189	0.135	0.133
	High	0.110	0.186	0.163
WFM-1 db2	Low	0.154	0.176	0.140
	Mid	0.161	0.132	0.134
	High	0.092	0.163	0.150
WFM-1 db4	Low	0.160*	0.169	0.140
	Mid	0.121	0.130	0.131
	High	**0.049**	0.102	0.120
WFM-1 db6	Low	0.199	0.219	0.210
	Mid	0.160	0.138	0.178
	High	0.066	0.121*	0.183
WFM-3 haar	Low	0.154	0.163	0.130
	Mid	0.188	0.133	0.129
	High	0.110	0.181	0.156
WFM-3 db2	Low	0.138	0.159*	**0.127**
	Mid	0.130	0.120	**0.125**
	High	0.060	0.104	**0.120**
WFM-3 db4	Low	**0.137***	0.160	0.130
	Mid	**0.118**	**0.119**	0.125
	High	0.052	**0.097**	0.120
WFM-3 db6	Low	0.145	0.166	0.138
	Mid	0.122	0.120	0.130
	High	0.066	0.099	0.127
FM_{pixel}	Low	0.155	**0.150**	0.131
	Mid	0.188	0.130	0.128
	High	0.112	0.183	0.156
$\text{FM}_{\text{fourier}}$	Low	0.347	0.321	0.299
	Mid	0.253	0.308	0.314
	High	0.137	0.262	0.296
WDNO	Low	0.426	0.402	0.363
	Mid	0.549	0.484	0.448
	High	0.585	0.581	0.529
$\text{LatentFM}_{\div 64}$	Low	0.206	0.183	0.167
	Mid	0.177	0.149	0.151
	High	0.111	0.139	0.196

(b) vs Pr (averaged over Ra)

Model	Pr	10^{-1}	1	10^1
WFM-1 haar	Low	0.187	0.144	0.147
	Mid	0.158	0.164	0.136
	High	0.192	0.155	0.111
WFM-1 db2	Low	0.180*	0.145*	0.145
	Mid	0.150	0.153	0.123
	High	0.182	0.136	0.087
WFM-1 db4	Low	0.181*	0.141	0.148
	Mid	0.138	0.133	0.111
	High	0.120	**0.087**	**0.063**
WFM-1 db6	Low	0.200*	0.210	0.217
	Mid	0.148	0.162	0.165
	High	0.129	0.150	0.091
WFM-3 haar	Low	0.171	0.139	0.137
	Mid	0.155	0.159	0.136
	High	0.189	0.150	0.108
WFM-3 db2	Low	**0.160***	0.133	**0.131**
	Mid	0.128	0.134	0.112
	High	0.122	0.096	0.066
WFM-3 db4	Low	0.163	**0.130**	0.134
	Mid	**0.126**	**0.128**	**0.108**
	High	**0.112**	0.090	0.066
WFM-3 db6	Low	0.172	0.137	0.139
	Mid	0.129	0.132	0.111
	High	0.120	0.094	0.079
FM_{pixel}	Low	0.169	0.132	0.136
	Mid	0.153	0.160	0.133
	High	0.197	0.149	0.106
$\text{FM}_{\text{fourier}}$	Low	0.342	0.314	0.311
	Mid	0.309	0.292	0.275
	High	0.262	0.228	0.205
WDNO	Low	0.390	0.388	0.413
	Mid	0.477	0.498	0.505
	High	0.574	0.575	0.545
$\text{LatentFM}_{\div 64}$	Low	0.199*	0.173	0.185
	Mid	0.160	0.160	0.157
	High	0.163	0.151	0.132

Physical consistency: kinetic energy. To check that the gains in VRMSE and CRPS do not come at the expense of physical realism, we compare the kinetic energy $E_k = \langle \frac{1}{2}\rho\,|\mathbf{u}|^2 \rangle$ of each generated sample with that of the reference field at the same step ($\rho = 1$ for RB and AM). The absolute error $|E_k(v_m) - E_k(u)|$ is averaged over the samples and divided by the time-mean reference energy of the trajectory rather than by the instantaneous one, which approaches zero during the early steps of RB. Table S20 reports this error per rollout horizon. On TRL, WFM 3-scale matches FM_{pixel} up to step 64 and is more accurate at the longest horizon (0.147 vs. 0.177). On RB and AM, FM_{pixel} has the lowest late-horizon error (0.251 and 0.320, vs. 0.348 and 0.531 for WFM 3-scale), while WFM 3-scale remains below $\text{FM}_{\text{fourier}}$ and WDNO on both systems, and below $\text{LatentFM}_{\div 64}$ on RB.

Table S20: Kinetic-energy relative MAE across rollout horizons (error normalised by the trajectory time-mean of the true kinetic energy). Mean over three inference seeds (42, 84, 168); the standard deviation across seeds is below 5% of the mean for all entries except those marked *. WFM rows use the selected wavelets (Table S11). Here, – indicates that the model was not available for the corresponding system.

	Turbulent Radiative Layer 2D			Rayleigh–Bénard			Active Matter		
Rollout	01:32	33:64	65:97	01:32	33:64	65:196	01:16	17:32	33:77
WFM 1-scale	0.047	0.113	0.221	0.007	0.410	0.379	0.502	0.538	<u>0.508</u>
WFM 3-scale	<u>0.017</u>	<u>0.067</u>	**0.147**	**0.003**	<u>0.217</u>	<u>0.348</u>*	<u>0.112</u>	<u>0.391</u>	0.531
FM_{pixel}	**0.016***	**0.067***	0.177	<u>0.004</u>	0.235	**0.251**	**0.051**	**0.288**	**0.320**
$\text{FM}_{\text{fourier}}$	0.037*	0.082	<u>0.150</u>*	0.006	**0.153**	1.018	0.440	0.513	0.642
WDNO	0.356	0.225	0.457	1.014	2.506	2.309	0.747	0.890	1.052
$\text{LatentFM}_{\div 64}$	–	–	–	0.252	0.574	0.913	–	–	–

Robustness to initial-condition perturbations. We perturb the initial conditions of all TRL test trajectories and roll out WFM 3-scale (haar) as before. Each of the $\mathcal{M}$ trajectories receives independent Gaussian noise on its whole context window, in physical space, with standard deviation $\eta\,\sigma_c$, where σ_c is the standard deviation of channel c over the window and $\eta \in \{10^{-3}, 10^{-2}, 10^{-1}\}$. The flow-matching noise is identical across η, so differences come from the perturbation alone. CRPS changes by at most 2.1% (horizon 65:97, $\eta = 0.1$; Table S21), and spectral coherence RMSE is unchanged at three decimals (Table S22).

Table S21: TRL, WFM 3-scale (haar): CRPS across rollout horizons under Gaussian perturbations of the initial conditions with standard deviation $\eta\,\sigma_c$. Mean over three inference seeds (42, 84, 168); the standard deviation across seeds is below 5% of the mean for all entries.

IC noise	01:32	33:64	65:97
$\eta = 0$	0.410	1.018	1.178
$\eta = 0.001$	0.410	1.036	1.180
$\eta = 0.01$	0.410	1.030	1.190
$\eta = 0.1$	0.413	1.026	1.203

Table S22: TRL, WFM 3-scale (haar): Spectral coherence RMSE across bands under Gaussian perturbations of the initial conditions with standard deviation $\eta\,\sigma_c$. Mean over three inference seeds (42, 84, 168); the standard deviation across seeds is below 5% of the mean for all entries.

IC noise	Low	Mid	High
$\eta = 0$	0.133	0.174	0.141
$\eta = 0.001$	0.133	0.174	0.141
$\eta = 0.01$	0.133	0.174	0.141
$\eta = 0.1$	0.133	0.174	0.141

Qualitative Assessment for Active Matter.

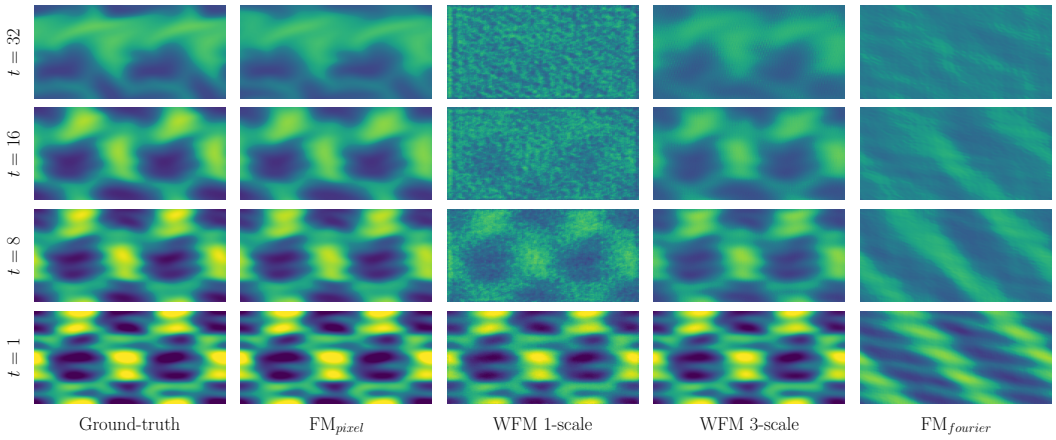


Figure S5: Qualitative rollout comparison for Active Matter: ground truth vs. model predictions across autoregressive steps.

F Profiling Analysis

Profiling metrics. Inference wall-clock times are measured by recording system timestamps immediately before the first rollout step and immediately after the last, on the same hardware without concurrent workloads. WFM 1-scale, WFM 3-scale, and FM_{pixel} are profiled under identical conditions: $\mathcal{M} = 8$ trajectories rolled out for the full test horizon, with each autoregressive step integrating the learned ODE with the Euler method for $N = 50$ neural function evaluations (NFEs), for a total of $\mathcal{M} \times T \times N$ NFEs per run. All runs use 4 NVIDIA A100 GPUs on the NSF NCAR Derecho system (doi:10.5065/qx9a-pg09).

Timing is collected in a distributed setting with $\mathcal{R} = 4$ parallel processes (ranks), one per GPU, each responsible for generating $\lfloor \mathcal{M}/\mathcal{R} \rfloor$ trajectories. To reflect true end-to-end latency, the reported wall-clock time T_{wall} is taken as the maximum across all ranks:

$$T_{\text{wall}} = \max_{r \in \{0,\ldots,\mathcal{R}-1\}} T_r, \tag{33}$$

where T_r is the local elapsed time on rank r, measured with a synchronization barrier both before and after the inference loop to ensure all ranks begin and end counting simultaneously. DataFrame construction, file I/O, and shard merging are explicitly excluded from this measurement.

From T_{wall}, two throughput metrics are derived. The *step throughput* (Γ_{steps}) measures the total number of rollout steps generated across all ranks per unit time:

$$\Gamma_{\text{steps}} = \frac{\sum_r n_{\text{steps},r}}{T_{\text{wall}}} \quad [\text{steps/s}], \tag{34}$$

where $n_{\text{steps},r}$ is the number of rollout steps generated on rank r, summed over its trajectories. The *frame throughput* (Γ_{frames}) further accounts for spatial resolution by weighting each step by the number of spatial elements $H \times W$:

$$\Gamma_{\text{frames}} = \frac{(\sum_r n_{\text{steps},r}) \times H \times W}{T_{\text{wall}}} \quad [\text{frames/s}]. \tag{35}$$

Peak GPU memory consumption (in GiB) is recorded per rank using `torch.cuda.max_memory_allocated` and reported individually to diagnose load imbalance across devices. The *speedup* of a WFM variant relative to FM_{pixel} is defined as the ratio of their respective wall-clock times under the same experimental conditions:

$$\text{Speedup} = \frac{T_{\text{wall}}^{\text{FM}_{\text{pixel}}}}{T_{\text{wall}}^{\text{WFM}}}. \tag{36}$$

Since all hyperparameters (number of trajectories, NFEs, ODE solver, hardware) are held fixed across models, any observed speedup is attributable to the multi-scale wavelet space in which the generative process operates. The speedup is moderate and is a by-product rather than the main goal: $\text{FM}_{fourier}$, which generates only a truncated set of Fourier modes, has a substantially higher error (Table 1), and latent-space models require pre-training an autoencoder (one day on 8×H100 GPUs for the Rayleigh–Bénard autoencoder of Rozet et al. [21], compared with about six hours on 4×A100 GPUs to train WFM). WFM therefore offers accuracy comparable to or better than pixel-space generation at a lower cost.

Table S23: Inference performance across PDEs and models. The reported WFM variants are selected based on the procedure in Section 4.2, and summarized in Table S11.

PDE	Model	Wall Clock (s)	sec / step	GPU Mem (GiB)	Throughput (Mfr/s)	Speedup ×	Mem ratio ×
TRL	FM_{pixel}	247.8	0.0355	0.59	1.39	—	—
	WFM 1-scale (haar)	198.1	0.0284	0.59	1.73	1.25×	1.00×
	WFM 3-scale (haar)	218.3	0.0313	0.66	1.57	1.14×	0.89×
RB	FM_{pixel}	633.9	0.0449	1.44	1.46	—	—
	WFM 1-scale (db4)	418.1	0.0296	1.44	2.21	1.52×	1.00×
	WFM 3-scale (db4)	457.9	0.0324	1.51	2.02	1.38×	0.95×
AM	FM_{pixel}	310.8	0.0459	1.63	1.43	—	—
	WFM 1-scale (haar)	213.3	0.0315	1.64	2.08	1.46×	1.00×
	WFM 3-scale (db2)	247.4	0.0365	1.71	1.80	1.26×	0.95×

Profiling different WFM representations. In this section we ablate the profiling results of WFM variants (1- and 3-scale) across different representations induced by a different choice of mother wavelet.

Table S24: Complete inference profiling across all WFM variants (scale and choice of the mother wavelet) relative to FM_{pixel}.

PDE	Model	Wavelet	Wall Clock (s)	sec / step	GPU Mem (GiB)	Throughput (Mfr/s)	Speedup ×	Mem ratio ×
TRL	FM_{pixel}	—	247.8	0.0355	0.59	1.39	—	—
	WFM 1-scale	**haar**	198.1	0.0284	0.59	1.73	1.25×	1.00×
		db2	199.0	0.0285	0.59	1.72	1.24×	1.00×
		db4	208.1	0.0298	0.59	1.65	1.19×	1.00×
		db6	204.5	0.0293	0.59	1.68	1.21×	1.00×
	WFM 3-scale	**haar**	218.3	0.0313	0.66	1.57	1.14×	0.89×
		db2	224.0	0.0321	0.66	1.53	1.11×	0.89×
		db4	228.2	0.0327	0.66	1.50	1.09×	0.89×
		db6	221.1	0.0317	0.66	1.55	1.12×	0.89×
RB	FM_{pixel}	—	633.9	0.0449	1.44	1.46	—	—
	WFM 1-scale	haar	413.3	0.0293	1.44	2.24	1.53×	1.00×
		db2	417.4	0.0296	1.44	2.22	1.52×	1.00×
		db4	418.1	0.0296	1.44	2.21	1.52×	1.00×
		db6	417.4	0.0296	1.44	2.22	1.52×	1.00×
	WFM 3-scale	haar	453.0	0.0321	1.51	2.04	1.40×	0.95×
		db2	446.5	0.0316	1.51	2.07	1.42×	0.95×
		db4	457.9	0.0324	1.51	2.02	1.38×	0.95×
		db6	468.0	0.0332	1.51	1.98	1.35×	0.95×
AM	FM_{pixel}	—	310.8	0.0459	1.63	1.43	—	—
	WFM 1-scale	**haar**	213.3	0.0315	1.64	2.08	1.46×	1.00×
		db2	216.9	0.0320	1.64	2.05	1.43×	1.00×
		db4	222.6	0.0329	1.64	1.99	1.40×	1.00×
		db6	228.2	0.0337	1.64	1.95	1.36×	1.00×
	WFM 3-scale	haar	244.7	0.0361	1.71	1.82	1.27×	0.95×
		db2	247.4	0.0365	1.71	1.80	1.26×	0.95×
		db4	252.6	0.0373	1.71	1.76	1.23×	0.95×
		db6	261.2	0.0386	1.71	1.70	1.19×	0.95×

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We clearly outline our contribution to the paper and scope of the work in the abstract and introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We clearly outline our contribution to the paper and scope of the work in the abstract and introduction. We further separate our work from related work in Section 5.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not include any theoretical results. Where necessary published theoretical background has been included in the paper and properly referenced.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We give a detailed description of the methods in Section 2 and in Appendix A and B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We will open access our code upon publication. But the dataset is freely available and described in the Section A. The model description and baselines are provided in Section 3 and Appendix B.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: Yes hyperparameters for method reproducibility are provided in Appendix B.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: The number of experiments and ensemble members to justify the evaluation metrics are provided in Section B.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide computational information in the text such as Table 2, S23 and S24.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research conforms with the NeurIPS Code of Ethics, as we do not present any research involving human subjects or participants, follow norms of fair use and copyright for any algorithms or datasets we use, and present a method with no risk of potential harmful consequences or societal impact, such as privacy, surveillance, discrimination, human rights, etc.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: As we provide a methodological advance for foundational research, we do not see any opportunity for potential negative societal impacts such as disinformation, deepfakes, surveillance, unfair impacts of specific groups, privacy considerations, or security considerations.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We do not provide any data or models that are at high risk of misuse such as LLMs, image generators or from the internet scraped datasets.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Where necessary we cite the original authors of adapted open-source algorithms and refer to the respective open-sourced GitHub repositories for benchmarks. For instance, we acknowledge [5] for the dataset source.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: As our contribution is modeling advancement, no new asset is produced except for code to reproduce the results presented.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: Our paper does not include any crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Our paper does not include any crowdsourcing or research with human subjects.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: We do not use any LLM for the core methodology, scientific rigorousness, or originality of the research, except for copy edit to fix grammar, misspelling, and double word counting.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.