RECURSIVE FLOW MATCHING

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative models have emerged as a powerful paradigm for solving physics systems and modeling complex spatiotemporal dynamics. However, achieving high physical accuracy without incurring high computational cost remains a fundamental challenge, as existing approaches face a critical speed-fidelity trade-off. In this work, we introduce **Recursive Flow Matching** (*RecFM*), a generative framework for forecasting complex spatiotemporal dynamics. RecFM enforces self-consistency to align trajectories across discretization scales, reducing discretization errors and improving performance across metrics for physics-based tasks. To our knowledge, this is the first method to achieve high-fidelity one- and few-step (*2-4 step*) dynamic generation for scientific systems, matching state-of-the-art multi-step solvers without relying on physics-based residuals. Across challenging scientific benchmarks, RecFM achieves up to a $20\times$ speedup over leading diffusion-based emulators while improving predictive accuracy. Furthermore, RecFM reduces mean squared error by over 15% compared to vanilla flow matching, offering a scalable and efficient solution for real-time scientific emulation.

1 INTRODUCTION

Predicting the evolution of physical systems is a fundamental challenge in scientific computing, with applications ranging from fluid dynamics to climate modeling and weather forecasting. Traditional numerical solvers provide high-fidelity solutions (Dhatt et al., 2012; Cantwell et al., 2015), but are typically computationally expensive and impractical for real-time or large-scale deployment. These limitations motivate the need for data-driven approaches that can efficiently model complex, high-dimensional dynamics. Advancements in scientific machine learning approaches like neural operators (Kovachki et al., 2023; Li et al., 2020; Lu et al., 2021) and PINNs (Raissi et al., 2019; Penwarden et al., 2022) are widely used to simulate systems described by partial differential equations (PDEs). However, in real-world applications, these governing equations are frequently incomplete, computationally prohibitive, or challenging to formulate for complex and stochastic systems such as climate dynamics.

Recent advances in generative modeling provide a powerful framework for learning high-frequency data distributions tailored to scientific applications, addressing key challenges in molecular design (Abramson et al., 2024; Shen et al., 2025), material generation (Zeni et al., 2025), and climate modeling (Duncan et al., 2025; Watt-Meyer et al., 2025). In these fields, the ability of generative models to quantify uncertainty and manage sparse or irregular measurements offers advantages over traditional deterministic methods. Especially in computational physics, generative methods have been shown to reconstruct spatiotemporal dynamics from limited observations, such as turbulent fluid flow or atmospheric models, effectively bridging the gap between inductive statistical learning and deductive physical laws (Cachay et al., 2025; Huang et al., 2024; Rühling Cachay et al., 2023; Zhuang et al., 2025). Nevertheless, deploying these models for accurate dynamical prediction remains challenging, as they must balance efficiency with the preservation of physical fidelity over time.

A key limitation of diffusion-based models is their inherently iterative inference procedure, which requires tens to hundreds of sequential denoising steps to produce high-quality predictions (Ho et al., 2020; Karras et al., 2022; Song et al., 2020; Nichol & Dhariwal, 2021). This results in significant computational overhead, especially for time-dependent simulations. To address this issue, continuous normalizing flows (CNFs) (Mathieu & Nickel, 2020) and flow matching (FM) (Chen & Lipman, 2023; Geng et al., 2025; Lipman et al., 2022) have emerged as efficient alternatives, learning continuous vector fields that define probability paths without requiring simulation during training. While these

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



(a) Flow Matching (b) Recursive Flow Matching

Figure 1: **Comparison of flow matching paradigms.** (a) Flow Matching (FM) learns a direct trajectory that transports samples from the data distribution (x_0) to the noise distribution (x_1). (b) Recursive Flow Matching (RecFM) augments this with recursively scaled trajectories (brown, blue, and red arrows) that intersect at shared spatial states (x_t), enabling cross-scale trajectory alignment and consistency training along the flow.

approaches reduce the number of required function evaluations, a fundamental trade-off remains: reducing the number of inference steps often leads to degraded accuracy and instability, particularly in long-term dynamical rollouts.

To further accelerate these systems, a wide range of approaches have been proposed, including consistency models and distillation-based methods (Tauberschmidt et al., 2025; Xu et al., 2023b). Consistency models, such as Shortcut Diffusion (Frans et al., 2024), introduce self-consistency constraints that enable direct mapping along the probabilistic path in a single step, while distillation techniques aim to compress multi-step generation into an efficient student model (Song et al., 2024). However, a key challenge in these approaches is preserving the spectral richness and spatiotemporal fidelity of physical fields, as aggressive step reduction often smooths out high-frequency structures that are critical for accurate scientific simulations (Nguyen et al., 2026; Xu et al., 2025). These limitations highlight the need for a framework that can achieve efficient few-step (typically *at most four steps*) generation while maintaining trajectory fidelity and stability.

To address these challenges, we introduce **Recursive Flow Matching** (*RecFM*), a generative framework for stable and efficient modeling of dynamical systems. Instead of relying on a single discretized trajectory, RecFM recursively models a family of trajectories spanning different inference-time traversal scales and enforces consistency among them. In particular, trajectories at different scales are coupled by aligning states that correspond to the same underlying point along the path, ensuring that predictions remain coherent across discretizations. This multi-scale coupling provides additional supervision and improves stability in one- or few-step regimes. Our main contributions include:

- **Recursive Flow Matching:** A novel flow matching framework for forecasting complex physical dynamics, enabling a unified treatment of systems governed either by explicit PDE formulations or by implicitly learned data-driven dynamics.
- **Recursive Trajectory Consistency:** A training mechanism that aligns predictions across recursively rescaled transport trajectories, improving stability under large integration steps and mitigating error accumulation during few-step genera
- **High-Efficiency Emulation:** We validate our approach on both simulated and real-world physical dynamics prediction benchmarks, achieving state-of-the-art accuracy with substantially fewer sampling steps.

2 Background

In this section, we introduce the necessary background for our proposed *RecFM*. We briefly review generative and trajectory flow matchings, which form the core building blocks of our framework.

2.1 Flow Matching

Flow Matching (FM) (Lipman et al., 2022) is a simulation-free paradigm for training Continuous Normalizing Flows by regressing onto a target vector field. Let p_0 denote the target data distribution and p_1 denote a tractable source distribution (*e.g.*, a standard Gaussian). FM seeks to learn a time-dependent vector field $v_t(x;\theta) : \mathbb{R}^d \to \mathbb{R}^d; t \in [0,1]$ that defines a probability path p_t connecting p_0 and p_1. The transformation of a sample $x_0 \sim p_0$ to $x_1 \sim p_1$ is governed by the ordinary differential equation (ODE):

$$\frac{d\psi_t(x)}{dt} = v_t(\psi_t(x), t), \quad \psi_0(x) = x_0 \tag{1}$$

where ψ_t represents the flow map. To ensure tractability, Conditional Flow Matching (CFM) utilizes a per-sample regression objective:

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t\sim U, x_0\sim p_0, x_1\sim p_1}[\|v_t(x_t, t; \theta) - u_t(x_t|x_0, x_1)\|^2] \tag{2}$$

where $u_t(x_t|x_0, x_1)$ is the conditional velocity field. A prevalent choice is the Optimal Transport (OT) path, which utilizes linear interpolation $x_t = (1-t)x_0 + tx_1$ to yield a constant target velocity $u_t(x_t|x_0, x_1) = x_1 - x_0$. Although this formulation fully specifies the generative process, its practical performance is largely determined by the structure of the induced trajectories, motivating a closer examination of trajectory design.

The choice of trajectory (*i.e.*, the transport map) plays a key role in determining sampling efficiency and stability. Approaches focusing on Trajectories of Flow Matching (Zhang et al., 2024; Islam et al., 2025), to parameterize the drift and diffusion terms to model stochastic and irregularly sampled time series. From a physical perspective, such trajectories can be interpreted as approximations of the underlying system dynamics, where geometric simplicity contributes to stable and accurate generation. Yet existing methods fail to maintain consistency across discretization scales, compromising both accuracy and physical fidelity.

2.2 Self-Consistency and the Flow Map

To overcome the iterative bottleneck of FM, recent work has introduced the self-consistency property (Frans et al., 2024; Xu et al., 2023a). For a flow map $\mathbf{X}_{s,t} : \mathbb{R}^d \to \mathbb{R}^d$ that transports a state from time s to time t, self-consistency requires that all points along a single trajectory map to the same endpoint. This is formally described by the semigroup condition:

$$\mathbf{X}_{u,t}(\mathbf{X}_{s,u}(x)) = \mathbf{X}_{s,t}(x)$$

for all s, u, t such that $0 \le s \le u \le t \le 1$ where u is an intermediate timestep. In consistency models (Song et al., 2023), a function $f_\theta(x_t, t)$ is trained to map every point of a trajectory to that trajectory's data endpoint, $f_\theta(x_t, t) = x_0$ for all $t \in [0, 1]$. A model satisfying this produces the same output whether the trajectory is traversed in one large step or in many small ones, which is what enables few-step sampling.

2.3 Discretization error of Few-Step Sampling

The accuracy of few-step sampling is governed by how strongly the sampled trajectories bend.

Lemma 2.1 (Euler discretization error; Benton et al., 2024)**.** *Let $v(\cdot, t)$ be L-Lipschitz for every $t \in [0, 1]$. Let ψ_t solve $\dot{\psi}_t = v(\psi_t, t)$ with $\psi_1 = x_1$, and let $\hat{\psi}_0$ be the result of K explicit Euler steps of size $h = 1/K$ from $t = 1$ to $t = 0$. Then*

$$\|\psi_0 - \hat{\psi}_0\| \;\le\; \frac{h}{2}\,\frac{e^L - 1}{L}\,\sup_{t\in[0,1]} \|a(\psi_t, t)\|, \tag{3}$$

where $a(x, t) = \partial_t v(x, t) + \big(\nabla_x v(x, t)\big)\, v(x, t)$.

The acceleration a is the rate of change of velocity along a trajectory. The bound vanishes when trajectories are straight lines traversed at constant speed.

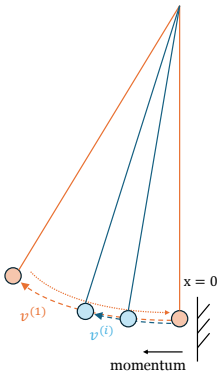


Figure 2: Pendulum trajectories and velocities for the primary trajectory ($v^{(1)}$, orange) and attenuated trajectories ($v^{(i)}$, $i > 1$, blue).

3 Recursive Flow Matching

We draw inspiration from the recursive movement of an ideal[1] wall-bouncing pendulum to design our method, *RecFM*. Below, we introduce the pendulum model, followed by the secondary trajectory formulation and the updated loss function for RecFM.

[1]We do not consider energy loss due to friction or drag forces.

3.1 PHYSICS INTUITION

Let's consider the classical physics toy problem of a 1D wall-bouncing pendulum, illustrated in Figure 2. Let $x(t)$ and $v(t)$ be the position and velocity of the pendulum at time t respectively. Away from the wall at $x = 0$, the pendulum travels at **constant speed** governed by:

$$\dot{x}(t) = v(t), \quad \dot{v}(t) = 0$$

A bounce occurs every time the pendulum strikes the wall ($x = 0$), resulting in a set of trajectories. At each collision, the velocity reverses direction and its magnitude is reduced, with a fraction $1 - \alpha^2$ of the kinetic energy lost, where $\alpha \in [0, 1]$ is the velocity retention coefficient. For simplicity, we consider velocities along a fixed direction (*e.g.*, from the wall toward the turning point), so that only their magnitudes are tracked across bounces. Let $v^{(i)}$ denote the velocity magnitude immediately after the i-th bounce. The collision update rule is:

$$v^{(i+1)} = \alpha\, v^{(i)}. \tag{4}$$

We assume a constant half-cycle duration across scales, consistent with small-angle dynamics, so that amplitude shrinks proportionally with velocity after each bounce. While not strictly physical, this yields a simple and tractable parameterization across trajectories.

After $D - 1$ collisions, we obtain a family of trajectories $\{v^{(i)}\}_{i=1}^{D}$ with progressively attenuated velocities. Writing $\boldsymbol{v}^* := v^{(1)}$ and $\alpha^{(i)} := \alpha^{i-1}$, we obtain the scaling relation

$$v^{(i)} = \alpha^{(i)}\, \boldsymbol{v}^*. \tag{5}$$

This velocity consistency defines a natural supervision signal for our multi-scale objective.

Figure 1 illustrates the key idea behind our approach. While vanilla flow matching learns a single trajectory between x_0 and x_1, we extend this formulation by introducing additional trajectories at different scales. Building on this intuition, we propose **Recursive Flow Matching** (***RecFM***), which enforces consistency across these trajectories.

3.2 RECFM ALGORITHM

Formulation. Given a data sample $x_0 \sim p_0$ and a noise sample $x_1 \sim p_1$, we define the standard linear interpolant

$$x_t = (1 - t)\, x_0 + t\, x_1, \qquad \boldsymbol{v}^* = x_1 - x_0. \tag{6}$$

The velocity network $v_\theta(x, t, \alpha)$ is conditioned on both time t and scale α, so that a single model represents the entire family of trajectories.

Consider a recursive formulation with depth D, where D trajectories are defined by time-scale pairs $\{(\tau^{(i)}, \alpha^{(i)})\}_{i=1}^{D}$. The rescaled time is defined as $\tau^{(i)} = t/\alpha^{(i)} \in [0, 1]$, with $\alpha^{(1)} = 1$ and $\tau^{(1)} = t$. Let $\hat{v}^{(i)} = v_\theta(x_t, \tau^{(i)}, \alpha^{(i)})$ denote the predicted velocity of the i-th trajectory. Under this alignment, all trajectories pass through the same spatial point x_t, yielding the cross-scale consistency relation

$$\hat{v}^{(i+1)} = \alpha\, \hat{v}^{(i)}. \tag{7}$$

This shared spatial point, which is visited by trajectories of different scales at correspondingly aligned times, is the structural property RecFM exploits.

Algorithm. We present Algorithm 1, which trains a velocity network on D recursive trajectories passing through the same point x_t: a primary trajectory (*i.e.*, $i = 1$) that learns the standard flow matching velocity $x_1 - x_0$, and $D - 1$ time-rescaled secondary trajectories parameterized by $\alpha^{(i)}$, whose target velocities are given by $\alpha^{(i)}(x_1 - x_0)$, inspired by the wall-bouncing dynamics in Section 3.1.

Training Objective. To enforce alignment across trajectory scales, we build on the recursive formulation above. The overall training objective aggregates supervision across all scales and enforces consistency with the primary trajectory:

$$\begin{gathered}\mathcal{L}_{\text{total}} = \sum_{i=1}^{D} \mathcal{L}_{\text{traj}}^{(i)} + \lambda \sum_{i=2}^{D} \mathcal{L}_{\text{cons}}^{(i)} \\ \text{where } \mathcal{L}_{\text{traj}}^{(i)} = \left\| \hat{v}^{(i)} - \alpha^{(i)} \boldsymbol{v}^* \right\|_2^2,\ \mathcal{L}_{\text{cons}}^{(i)} = \left\| \hat{v}^{(i)} - \alpha^{(i)} \hat{v}^{(1)} \right\|_2^2.\end{gathered} \tag{8}$$

Inference Sampling. Inference in RecFM is conducted by numerically solving the ODE defined by the learned velocity field $\hat{v}_\theta(x_t, t, \alpha)$. For single-step generation, using a first-order Euler step of size h, RecFM maps a noise sample $x_1 \sim p_1$ to the data manifold in one function evaluation:

$$x_0 \approx x_1 - h\,\hat{v}_\theta(x_1, 1, 1) \tag{9}$$

where $h = 1$, corresponding to integrating the trajectory over the full time horizon.

For multi-step generation, discretizing the trajectory into K steps $1 = t_0 > \cdots > t_K = 0$ with step sizes $h_k = t_{k-1} - t_k$, we iteratively update:

$$x_{t_k} = x_{t_{k-1}} - h_k\,\hat{v}_\theta(x_{t_{k-1}}, t_{k-1}, 1), \quad k = 1, \ldots, K. \tag{10}$$

By enforcing cross-scale velocity consistency during training, RecFM learns trajectories that remain stable under larger integration steps, enabling accurate few-step generation.

Why does RecFM work?

A given interpolated state x_t lies on infinitely many trajectories indexed by α. Vanilla FM exploits only one of them, providing a single regression target $\boldsymbol{v}^*$ per sample. RecFM uses every (τ, α) pair as an independent supervisory signal for the *same* underlying directional quantity $x_1 - x_0$ at the *same* spatial point, while following the marginal distribution (Theorem B.2). This functions as data augmentation in the conditioning space of the network and is particularly valuable in the one-step regime, where generation quality depends entirely on a single evaluation $v_\theta(x_0, 0, 1)$. By coupling predictions across scales, RecFM enriches the gradient signal at every training point and removes the warm-up phase typically required by shortcut or consistency-style training (Appendix H).

4 Related Work

Neural PDE Solvers and Physics-Informed Learning. Early advancements in scientific machine learning focused on directly embedding physical laws into neural architectures to solve boundary value problems with minimal data. Physics-Informed Neural Networks (PINNs) (Raissi et al., 2019) penalize PDE residuals at randomly sampled collocation points, while neural operators like Fourier Neural Operators (Li et al., 2020) and DeepONet (Lu et al., 2021), Equivariant Neural fields (Knigge et al., 2024) use functional mappings between infinite-dimensional spaces. To address the limitation of the availability of high-fidelity data, multi-fidelity PINNs (Penwarden et al., 2022) were introduced to utilize low-fidelity responses as regularizers. However, these deterministic methods often struggle in complex settings and with real-world observations. By producing point estimates rather than predictive distributions, they offer limited uncertainty quantification and are rarely evaluated using probabilistic metrics, which can lead to physically inconsistent outputs.

Algorithm 1 Recursive Trajectory Training with Consistency Alignment

Require: Data distribution p_0, Noise distribution p_1
Require: Velocity network $v_\theta(x, t, \alpha)$, recursion depth D
Require: Consistency weight λ, total training iterations N

for iteration $n = 1$ to N **do**
  Sample $x_0 \sim p_0$ and $x_1 \sim p_1$ ▷ Data and noise samples
  Sample $t \sim \mathcal{U}(0, 1)$ and $\alpha \sim \mathcal{U}(t, 1)$ ▷ Primary trajectory time and base recursion scale
  $\boldsymbol{v}^* \leftarrow x_1 - x_0$ ▷ Ground-truth primary velocity
  $x_t \leftarrow (1 - t)x_0 + tx_1$ ▷ Shared spatial point
  for $i = 1$ to D **do**
    $\alpha^{(i)} \leftarrow \alpha^{i-1}$ ▷ Recursive trajectory scale
    $\tau^{(i)} \leftarrow t/\alpha^{(i)}$ ▷ Aligned trajectory time
    $\hat{v}^{(i)} \leftarrow v_\theta(x_t, \tau^{(i)}, \alpha^{(i)})$ ▷ Predicted trajectory velocity
    $\mathcal{L}^{(i)}_{\text{traj}} \leftarrow \|\hat{v}^{(i)} - \alpha^{(i)}\boldsymbol{v}^*\|_2^2$ ▷ Trajectory supervision
  end for
  for $i = 2$ to D **do**
    $\mathcal{L}^{(i)}_{\text{cons}} \leftarrow \|\hat{v}^{(i)} - \alpha^{(i)}\hat{v}^{(1)}\|_2^2$ ▷ Cross-scale consistency
  end for
  $\mathcal{L}_{\text{total}} \leftarrow \sum_{i=1}^{D} \mathcal{L}^{(i)}_{\text{traj}} + \lambda \sum_{i=2}^{D} \mathcal{L}^{(i)}_{\text{cons}}$
  Update θ using $\nabla_\theta \mathcal{L}_{\text{total}}$
end for

Probabilistic Generative Modeling for Spatiotemporal Physics Systems. Probabilistic approaches quantify and calibrate uncertainty, providing a useful framework for learning physics-based systems DiffusionPDE (Huang et al., 2024) and FunDPS (Yao et al., 2025) unify the forward and backward problems through joint coefficient-solution state modeling, while VideoPDE (Li et al., 2025) regards various tasks as video restoration to preserve fine-grained spectral details. A notable advancement is DYffusion (Rühling Cachay et al., 2023), which replaces standard Gaussian perturbations with a dynamics-informed temporal interpolation. By avoiding the high memory overhead of video-based models like MCVD (Voleti et al., 2022), DYffusion leverages Monte Carlo dropout to produce probabilistic ensembles during inference. In physics and climate science, foundation models (Aich et al., 2026; Ohana et al., 2024; Tauberschmidt et al., 2025) can achieve high accuracy with simple finetuning. Similarly, Rolling Sequence Diffusion Models (Ruhe et al., 2024; Wu et al., 2023) and ERDM (Cachay et al., 2025) utilize adaptive noise schedules to reflect the growth of uncertainty, prioritizing the ability to transition from deterministic to random horizons.

Accelerated Inference and Consistency-Based Models. Diffusion's iterative bottleneck has spurred recent studies on inference acceleration of generative models. EDM (Karras et al., 2022) provides efficient sampling that reduces sampling time for various tasks like molecular design (Vadgama et al., 2025). Rectified Flow (Liu et al., 2022) reduces transportation costs by training new ODEs on the previous flow generation pairs, optimizing the generation to a one-step path, while Shortcut Model (Frans et al., 2024) stabilizes sampling through interval self-consistency. Recent innovations like MeanFlow (Geng et al., 2025) have introduced average velocity fields to characterize transitions, while Drifting Diffusion (Deng et al., 2026) performs few-step generation in feature space. Generalized flow maps (Davis et al., 2025) show few-step generation on arbitrary Riemannian manifolds. Physics-informed methods like PBFM (Baldan et al., 2026) further apply these ideas to physical dynamics by incorporating explicit PDE residuals into the objective. However, such methods are fundamentally constrained by their reliance on known physical formulas, making them unsuitable for complex systems where equations are unavailable or computationally prohibitive to implement. RecFM addresses these concerns by introducing a recursive framework in the data space to enforce flow trajectory across discretization scales. By adopting this approach without explicitly using PDE residual supervision, RecFM provides a robust solution for high-fidelity emulation in complex scientific domains.

5 Experiments

5.1 Datasets

We evaluate our methods on three different dynamic physics datasets characterized by non-linear evolution and diverse spectral features. Specific technical configurations and simulation details are provided in Appendix A.

Sea Surface Temperatures (SST). This real-world climate dataset is adapted from the DYffusion benchmark (Rühling Cachay et al., 2023), using daily global measurement data from the NOAA OISSTv2 (Huang et al., 2021) product. Its spatial resolution is $1/4°$. We utilized a regional 60×60 latitude and longitude grid in the eastern tropical Pacific to simulate the long-term time-dependent relationship of the ocean temperature field.

Navier-Stokes Flow. We follow the experimental setup of DYffusion (Otness et al., 2021; Rühling Cachay et al., 2023) to evaluate fluid dynamics rollouts. The environment consists of an incompressible channel flow past four randomly generated circular obstacles, inducing complex turbulence and vorticity patterns. The kinematic viscosity is set to $\nu = 10^{-3}$, and simulations are conducted on a 221×42 grid. The dataset comprises three channels: the velocity components in each spatial direction and the pressure field.

Helmholtz Staircase Equation. We follow the setup of The Well (Ohana et al., 2024). This benchmark corresponds to a higher-order analytical solution for acoustic scattering from a point source near an infinite, periodic "staircase" boundary. The simulated fields are discretized into 1024×256 grids to capture both the real and imaginary components of the pressure field. Accordingly, the dataset consists of two channels representing the real and imaginary parts.

5.2 Experiment Setup

5.2.1 Forecasting Configuration

We evaluate performance across varying temporal horizons:

Table 1: Quantitative forecasting results for Sea Surface Temperature, Navier-Stokes Flow, and Helmholtz Staircase Equation. Lower values are better for MSE and CRPS, while the optimal SSR is 1. Best results in **bold**, second best underlined, third best in gray.

Method	SST				Navier-Stokes			Helmholtz Staircase		
	CRPS	MSE	SSR	Time [s]	CRPS	MSE	SSR	CRPS	MSE	SSR
Perturbation*	0.281	0.180	0.411	0.4241	0.090	0.028	0.448	0.218	0.111	0.004
Dropout*	0.267	0.164	0.406	0.4241	0.078	0.027	0.715	0.099	0.049	0.631
DDPM*	0.246	0.177	0.674	0.3054	0.180	0.105	0.573	0.156	0.153	0.563
MCVD*	0.216	0.161	0.926	79.167	0.154	0.070	0.524	0.137	0.128	0.867
DYffusion*	0.224	0.173	1.033	4.6722	0.067	0.022	0.877	0.144	0.106	1.121
VideoPDE (Li et al., 2025)	0.216	0.162	0.746	19.753	0.033	0.0068	0.205	0.026	5.6e-4	4.334
Vanilla FM	0.260	0.232	0.914	1.5202	0.036	0.0076	0.911	0.030	6.5e-4	1.485
Shortcut (Frans et al., 2024)	0.249	0.297	0.798	0.4452	0.034	0.0079	0.717	0.0144	1.6e-4	0.467
RecFM (1-step)	0.217	0.162	0.984	0.4310	**0.031**	**0.0064**	**0.959**	0.0034	4.2e-5	**1.090**
RecFM (2-step)	**0.216**	**0.161**	**1.004**	0.7353	0.032	0.0068	0.932	**0.0027**	**2.7e-5**	1.440

*Results for SST and Navier-Stokes are reproduced from DYffusion (Rühling Cachay et al., 2023).

Temporal Horizons and Autoregressive Rollout. For SST, we predict 7 days ahead from a 1-day input. For Navier-Stokes and Helmholtz, we respectively perform complete trajectory reconstructions of 64 and 49 steps starting from the initial state. To manage these long-range sequences, models are applied autoregressively: Navier-Stokes models predict 16 frames each time, while Helmholtz models generate 7 frames, unless specified.

Ensemble Generation. For all probabilistic metrics (CRPS, SSR), we create $M = 50$ ensemble members per initial condition to ensure statistical reliability.

Model Architecture and Efficiency. We apply RecFM to a pixel-level temporal DiT backbone, following the design introduced in Li et al. (2025). All inference time measurements are performed on a single NVIDIA L40S GPU. RecFM is evaluated in both single- and multi-step regimes. More details of model architecture and implementation are included in Appendix D.

Hyperparameter Selection. We use $\lambda = 1$ for the consistency loss weight, with further analysis provided in Section 5.4. We adopt the depth-2 formulation for RecFM, corresponding to a primary trajectory with $\alpha^{(1)} = 1$ and a secondary trajectory with scale $\alpha^{(2)} = \alpha$, as it provides the best performance and efficiency (see Appendix C.2).

5.2.2 Baselines

We compare RecFM against a comprehensive suite of generative and stochastic benchmarks. For standard forecasting models, we adopt the experimental configuration and model suite from DYffusion (Rühling Cachay et al., 2023), which includes:

- **Stochastic Methods: Perturbation** and **Dropout** (Monte Carlo dropout at inference).
- **Iterative Models: DDPM** and **MCVD**, which utilize Gaussian noising processes.
- **Dynamics-Informed Solvers: DYffusion**, which directly couples diffusion steps with physical timesteps.

We further include benchmarks with state-of-the-art generative backbones:

- **VideoPDE (Li et al., 2025):** A unified solver that recasts PDE solving as hierarchical video inpainting using a pixel-space hierarchical transformer.
- **Vanilla FM:** Utilizes the identical architectural backbone to RecFM but is trained using a standard Flow Matching objective without the recursive feature.
- **Shortcut Model (Frans et al., 2024):** A method originally developed for single- and few-step image generation, which we carefully adapt to our dynamic backbone (Appendix H).

We exclude PBFM (Baldan et al., 2026) from the main comparisons as it requires explicit closed-form PDE residuals, which are impractical for complex, data-rich systems such as global SST. Results on PDE-governed data with physics-informed metrics are provided in Appendix F. One-step methods such as MeanFlow (Geng et al., 2025) and other benchmarks like Rectified Flow (Liu et al., 2022), which are primarily designed for static generation, are also omitted from the main comparisons. A comparison of MeanFlow and RecFM on Navier-Stokes equation is included in Appendix I.

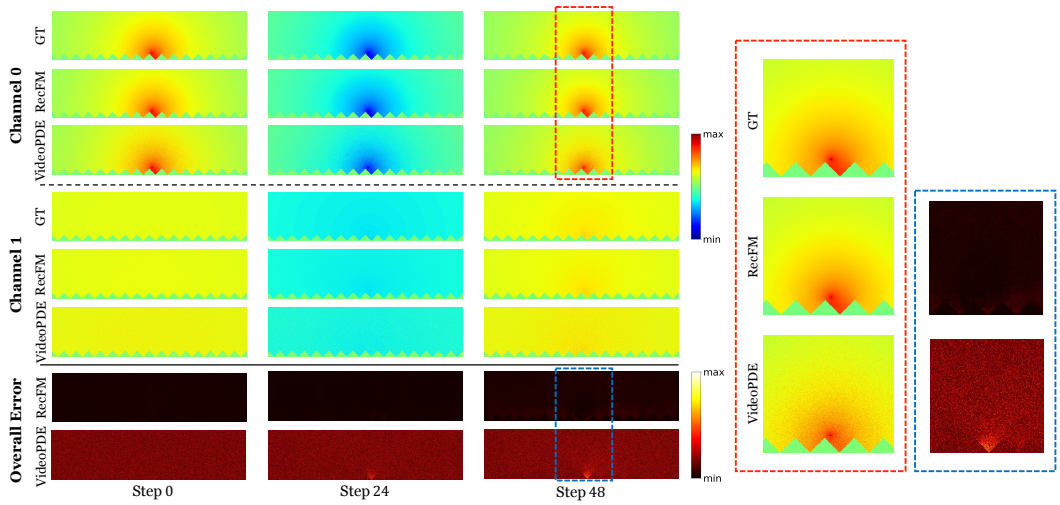


Figure 3: **Roll-out results of the Helmholtz Staircase equation.** Visual comparison of Ground Truth against RecFM and VideoPDE (best-performing baseline across datasets) for two channels, with the bottom rows indicating absolute errors. Columns correspond to dataset timesteps. The variation observed at Step 48 is displayed in an enlarged view on the right.

5.2.3 Evaluation Metrics

To evaluate the fidelity and calibration of probability prediction, we employ three standard metrics:

Continuous Ranked Probability Score (CRPS). A strictly proper scoring rule to measure the accuracy of the cumulative distribution function F relative to the observation y:

$$\text{CRPS}(F, y) = \int_{-\infty}^{\infty} (F(z) - \mathbf{1}[z \geq y])^2 dz \tag{11}$$

In practice, we use the unbiased "fair" estimator for M ensembles.

Mean Squared Error (MSE). Measures the deterministic accuracy of the ensemble mean prediction $\bar{x}$ against the ground truth y for the dataset of size S:

$$\text{MSE} = \frac{1}{S} \sum_{j=1}^{S} \|\bar{x}_j - y_j\|^2 \tag{12}$$

Spread-Skill Ratio (SSR). Evaluates the reliability of the ensemble by comparing the ensemble spread to the RMSE of the ensemble mean. An ideal ratio of 1.0 indicates a perfectly calibrated ensemble. Specifically, SSR values smaller than 1.0 indicate underdispersion, while values larger than 1.0 indicate overdispersion.

5.3 Forecasting Results

Quantitative results across all benchmarks are summarized in Table 1, with standard deviations in Appendix G. We evaluate RecFM using one- and two-step inference to highlight its flexibility, with additional analysis of step size in Appendix C.1. Overall, RecFM consistently achieves state-of-the-art performance in both fidelity and efficiency. In particular, it attains up to a $20\times$ speedup over the diffusion-based baseline VideoPDE while also improving predictive accuracy and calibration. This speedup is measured in terms of total rollout runtime, reflecting the reduced number of inference steps required by RecFM. On the Helmholtz Staircase equation, RecFM achieves a $10\times$ reduction in error compared to VideoPDE, which is the overall best-performing baseline across datasets. We further visualize roll-out snapshots for both channels of the Helmholtz equation, along with corresponding error maps, in Figure 3. RecFM produces predictions that closely match the ground truth, while VideoPDE struggles to capture the circular wave propagation patterns. RecFM also consistently outperforms Shortcut Model (see Appendix H). Additional visualizations are provided in Appendix E.

Moreover, compared to vanilla flow-matching methods, which typically require $\sim$5 inference steps, RecFM produces high-quality results with only 1-2 steps, achieving over 15% lower MSE and substantially better SSR scores. We also observe that multi-step RecFM models do not consistently outperform single-step variants, as errors can accumulate over successive iterations.

Table 2: Ablation study on the effect of λ for the Navier-Stokes equation using the 1-step model and 5-step model. Best results in **bold**.

	1-step model			5-step model		
λ	**CRPS** ($\downarrow$)	**MSE** ($\downarrow$)	**SSR** ($\rightarrow 1$)	**CRPS** ($\downarrow$)	**MSE** ($\downarrow$)	**SSR** ($\rightarrow 1$)
0.0	0.035	0.0074	0.957	0.039	0.0093	0.843
0.5	0.034	0.0071	1.024	0.040	0.0099	**0.855**
1.0	**0.031**	**0.0064**	0.959	**0.037**	**0.0083**	0.836
10.0	0.038	0.0089	**0.988**	0.039	0.0089	0.784
1000000	0.238	0.268	1.147	0.234	0.261	1.161

While RecFM performs best across all tasks, its advantage is more pronounced on deterministic problems, such as PDE prediction tasks governed by explicit physical constraints, than on more stochastic data such as SST. This behavior is expected, as few-step flow matching is inherently more deterministic, which introduces minimal randomness into the sampling process.

Training Stability. We measure training progress of Navier-Stokes Flow in terms of the number of function evaluations (NFE), defined as the total number of vector field evaluations (*i.e.*, forward passes) during optimization. As shown in Figure 4, RecFM converges faster than the diffusion-based baseline (VideoPDE) and consistently achieves lower validation error throughout training.

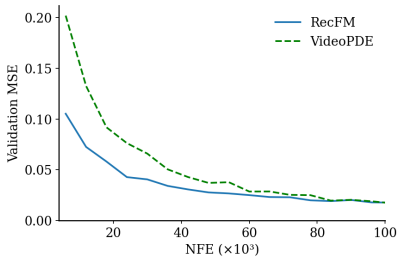


Figure 4: **Validation MSE versus NFE during training.** RecFM converges faster than the diffusion-based model VideoPDE and maintains consistently lower validation error.

5.4 Ablation Studies

We investigate the sensitivity of RecFM (see Equation 8) to the consistency loss weight λ on the Navier-Stokes equation. Table 2 reports results for both single-step and multi-step models over a wide range of λ values. A moderate setting (*e.g.*, $\lambda = 1.0$) consistently yields the best performance, suggesting that an appropriate balance between objectives is important. When λ is too small, the model places insufficient emphasis on trajectory consistency, which degrades performance. Interestingly, even without the self-consistency term ($\lambda = 0$), RecFM still surpasses the Vanilla FM baseline in Table 1, likely due to the presence of the secondary trajectory loss. In contrast, very large values (*e.g.*, $\lambda = 10^6$) cause the consistency term to overwhelm the primary flow-matching objective, leading to a marked drop in accuracy.

5.5 Image Generation Experiments

Although our primary focus is scientific emulation, RecFM also generalizes beyond physics-based systems, achieving competitive image quality with reduced training and inference cost (Appendix J).

6 Conclusion and Discussion

We introduced Recursive Flow Matching (RecFM), a framework that enforces consistency of generative trajectories across different sampling regimes. Our findings indicate that aligning trajectories across scales leads to an unexpected effect: using fewer inference steps can actually enhance both stability and accuracy, particularly in physics-based applications. This observation questions the usual trade-off between sampling efficiency and fidelity and suggests that the structure of the trajectory, rather than simply increasing the number of steps, is central to effective generation.

In experiments, RecFM delivers strong results on a range of scientific benchmarks, matching the performance of multi-step solvers while operating in the one- or few-step regimes and yielding notable speedups. These results highlight the potential of consistency-based approaches for real-time scientific emulation.

Limitations and Future Work. Despite these gains, extending RecFM to high-complexity real-world video remains challenging. Unlike physics-driven systems, natural videos involve rich semantic and temporal variations that may require modeling beyond standard flow matching trajectories. Although preliminary image-generation results are promising, scaling the framework to realistic video domains remains an open problem. Future work will further explore these settings and investigate RecFM as a general-purpose foundation model for multi-physics and real-world dynamical systems.

AI USE STATEMENT

In this work, we used generative AI tools to edit the manuscript to improve readability, provide feedback on the manuscript, and review and edit code. We reviewed all AI-assisted work and independently verified the suggestions for accuracy, relevance, and consistency with our intended content. We take full responsibility for the final content of this work, including any text or code produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We have provided all necessary details for reproducing experiments in the paper. We will release code upon acceptance.

REFERENCES

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500, 2024.

Michael Aich, Andreas Fürst, Florian Sestak, Carlos Ruiz-Gonzalez, Niklas Boers, and Johannes Brandstetter. Wind: Weather inverse diffusion for zero-shot atmospheric modeling. *arXiv preprint arXiv:2602.03924*, 2026.

Giacomo Baldan, Qiang Liu, Alberto Guardone, and Nils Thuerey. Physics vs distributions: Pareto optimal flow matching with physics constraints. In *The Fourteenth International Conference on Learning Representations*, 2026.

Joe Benton, George Deligiannidis, and Arnaud Doucet. Error bounds for flow matching methods. *Transactions on Machine Learning Research*, 2024.

Salva Rühling Cachay, Miika Aittala, Karsten Kreis, Noah Brenowitz, Arash Vahdat, Morteza Mardani, and Rose Yu. Elucidated rolling diffusion models for probabilistic forecasting of complex dynamics. *arXiv preprint arXiv:2506.20024*, 2025.

Chris D Cantwell, David Moxey, Andrew Comerford, Alessandro Bolis, Gabriele Rocco, Gianmarco Mengaldo, Daniele De Grazia, Sergey Yakovlev, J-E Lombard, Dirk Ekelschot, et al. Nektar++: An open-source spectral/hp element framework. *Computer physics communications*, 192:205–219, 2015.

Ricky TQ Chen and Yaron Lipman. Flow matching on general geometries. *arXiv preprint arXiv:2302.03660*, 2023.

Oscar Davis, Michael S Albergo, Nicholas M Boffi, Michael M Bronstein, and Avishek Joey Bose. Generalised flow maps for few-step generative modelling on riemannian manifolds. *arXiv preprint arXiv:2510.21608*, 2025.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Mingyang Deng, He Li, Tianhong Li, Yilun Du, and Kaiming He. Generative modeling via drifting. *arXiv preprint arXiv:2602.04770*, 2026.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Gouri Dhatt, Emmanuel Lefrançois, and Gilbert Touzot. *Finite element method*. John Wiley & Sons, 2012.

James PC Duncan, Elynn Wu, Surya Dheeshjith, Adam Subel, Troy Arcomano, Spencer K Clark, Brian Henn, Anna Kwa, Jeremy McGibbon, W Andre Perkins, et al. Samudrace: Fast and accurate coupled climate modeling with 3d ocean and atmosphere emulators. *arXiv preprint arXiv:2509.12490*, 2025.

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. *arXiv preprint arXiv:2410.12557*, 2024.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *arXiv preprint arXiv:2505.13447*, 2025.

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

Boyin Huang, Chunying Liu, Viva Banzon, Eric Freeman, Garrett Graham, Bill Hankins, Tom Smith, and Huai-Min Zhang. Improvements of the daily optimum interpolation sea surface temperature (doisst) version 2.1. *Journal of Climate*, 34(8):2923–2939, 2021.

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.

Mohammad Mohaiminul Islam, Thijs P Kuipers, Sharvaree Vadgama, Coen de Vente, Afsana Khan, Clara I Sánchez, and Erik J Bekkers. Longitudinal flow matching for trajectory modeling. *arXiv preprint arXiv:2510.03569*, 2025.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

David M Knigge, David R Wessels, Riccardo Valperga, Samuele Papa, Jan-Jakob Sonke, Efstratios Gavves, and Erik J Bekkers. Space-time continuous pde forecasting using equivariant neural fields. *Advances in Neural Information Processing Systems*, 37:76553–76577, 2024.

Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research*, 24(89):1–97, 2023.

Edward Li, Zichen Wang, Jiahe Huang, and Jeong Joon Park. Videopde: Unified generative pde solving via video inpainting diffusion models. *arXiv preprint arXiv:2506.13754*, 2025.

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.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

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

Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, pp. 23–40. Springer, 2024.

Emile Mathieu and Maximilian Nickel. Riemannian continuous normalizing flows. *Advances in neural information processing systems*, 33:2503–2515, 2020.

Anh Nguyen, Viet Nguyen, Duc Vu, Trung Dao, Chi Tran, Toan Tran, and Anh Tran. Improved training technique for shortcut models. *Advances in Neural Information Processing Systems*, 38: 92816–92846, 2026.

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

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

Karl Otness, Arvi Gjoka, Joan Bruna, Daniele Panozzo, Benjamin Peherstorfer, Teseo Schneider, and Denis Zorin. An extensible benchmark suite for learning to simulate physical systems. *arXiv preprint arXiv:2108.07799*, 2021.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Michael Penwarden, Shandian Zhe, Akil Narayan, and Robert M Kirby. Multifidelity modeling for physics-informed neural networks (pinns). *Journal of Computational Physics*, 451:110844, 2022.

Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

David Ruhe, Jonathan Heek, Tim Salimans, and Emiel Hoogeboom. Rolling diffusion models. *arXiv preprint arXiv:2402.09470*, 2024.

Salva Rühling Cachay, Bo Zhao, Hailey Joren, and Rose Yu. Dyffusion: A dynamics-informed diffusion model for spatiotemporal forecasting. *Advances in neural information processing systems*, 36:45259–45287, 2023.

Yuning Shen, Lihao Wang, Huizhuo Yuan, Yan Wang, Bangji Yang, and Quanquan Gu. Simultaneous modeling of protein conformation and dynamics via autoregression. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models, 2023. URL https://arxiv.org/abs/2303.01469.

Yanke Song, Jonathan Lorraine, Weili Nie, Karsten Kreis, and James Lucas. Multi-student diffusion distillation for better one-step generators. *arXiv preprint arXiv:2410.23274*, 2024.

Jan Tauberschmidt, Sophie Fellenz, Sebastian J Vollmer, and Andrew B Duncan. Physics-constrained fine-tuning of flow-matching models for generation and inverse problems. *arXiv preprint arXiv:2508.09156*, 2025.

Sharvaree Vadgama, Mohammad Mohaiminul Islam, Domas Buracas, Christian Shewmake, Artem Moskalev, and Erik Bekkers. Probing equivariance and symmetry breaking in convolutional networks. *arXiv preprint arXiv:2501.01999*, 2025.

Vikram Voleti, Alexia Jolicoeur-Martineau, and Chris Pal. Mcvd-masked conditional video diffusion for prediction, generation, and interpolation. *Advances in neural information processing systems*, 35:23371–23385, 2022.

Oliver Watt-Meyer, Brian Henn, Jeremy McGibbon, Spencer K Clark, Anna Kwa, W Andre Perkins, Elynn Wu, Lucas Harris, and Christopher S Bretherton. Ace2: accurately learning subseasonal to decadal atmospheric variability and forced responses. *npj Climate and Atmospheric Science*, 8(1): 205, 2025.

Tong Wu, Zhihao Fan, Xiao Liu, Hai-Tao Zheng, Yeyun Gong, Jian Jiao, Juntao Li, Jian Guo, Nan Duan, Weizhu Chen, et al. Ar-diffusion: Auto-regressive diffusion model for text generation. *Advances in Neural Information Processing Systems*, 36:39957–39974, 2023.

Sihan Xu, Yidong Huang, Jiayi Pan, Ziqiao Ma, and Joyce Chai. Inversion-free image editing with natural language. *arXiv preprint arXiv:2312.04965*, 2023a.

Sihan Xu, Ziqiao Ma, Yidong Huang, Honglak Lee, and Joyce Chai. Cyclenet: Rethinking cycle consistency in text-guided diffusion for image manipulation. *Advances in Neural Information Processing Systems*, 36:10359–10384, 2023b.

Zhi-Qin John Xu, Lulu Zhang, and Wei Cai. On understanding and overcoming spectral biases of deep neural network learning methods for solving pdes. *Journal of Computational Physics*, 530: 113905, 2025.

Jiachen Yao, Abbas Mammadov, Julius Berner, Gavin Kerrigan, Jong Chul Ye, Kamyar Azizzadenesheli, and Anima Anandkumar. Guided diffusion sampling on function spaces with applications to pdes. *arXiv preprint arXiv:2505.17004*, 2025.

Claudio Zeni, Robert Pinsler, Daniel Zügner, Andrew Fowler, Matthew Horton, Xiang Fu, Zilong Wang, Aliaksandra Shysheya, Jonathan Crabbé, Shoko Ueda, et al. A generative model for inorganic materials design. *Nature*, 639(8055):624–632, 2025.

Xi Zhang, Yuan Pu, Yuki Kawamura, Andrew Loza, Yoshua Bengio, Dennis L Shung, and Alexander Tong. Trajectory flow matching with applications to clinical time series modelling. *Advances in Neural Information Processing Systems*, 37:107198–107224, 2024.

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

A Dataset Details

In this section, we provide the formal governing equations and technical implementation details for the physics datasets used in our evaluation. For every dataset that includes boundary conditions, these conditions are provided as extra constraints to each model.

A.1 Sea Surface Temperatures (SST)

The SST dataset is a real-world climate benchmark representing the daily evolution of sea surface temperature fields $\mathbf{T}$ over the eastern tropical Pacific Ocean. While not governed by a single closed-form PDE, the dynamics arise from complex ocean-atmosphere interactions and large-scale climate variability. Each sample consists of 11 spatial boxes, each represented as a 60×60 latitude-longitude grid with a single scalar channel corresponding to temperature values (Rühling Cachay et al., 2023). Following prior work, we use data from 1982 to 2019 for training, 2020 for validation, and 2021 for testing.

A.2 Navier-Stokes (NS) Flow

The Navier-Stokes benchmark simulates incompressible channel flow past random circular obstacles. The dynamics are governed by the following momentum and continuity equations:

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} &= -\frac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f} \\ \nabla \cdot \mathbf{u} &= 0 \end{aligned} \tag{13}$$

where $\mathbf{u} = (u, v)$ is the velocity vector, p is the pressure field, ρ is the fluid density, and $\nu = 10^{-3}$ is the kinematic viscosity.

Technical Configuration:

- **Channels:** The dataset consists of 3 distinct channels: the x-velocity component (u), the y-velocity component (v), and the pressure field (p).
- **Preprocessing:** The raw simulation data is defined on a 221×42 grid. For the implementation of RecFM, Vanilla FM, Shortcut Model (Frans et al., 2024), and VideoPDE (Li et al., 2025), the input fields are bilinearly interpolated to a resolution of 220×40 before being passed to the model. During evaluation, the generated outputs are upsampled back to the original 221×42 resolution to compute metrics.

A.3 Helmholtz Staircase Equation

The Helmholtz benchmark evaluates acoustic scattering from a point source near a corrugated boundary. The steady-state pressure field u satisfies:

$$-(\Delta + \omega^2)u = \delta_{\mathbf{x}_0} \tag{14}$$

where Δ is the Laplacian, ω is the angular frequency, and $\mathbf{x}_0$ is the source position. The time-dependent pressure evolution is defined analytically as:

$$U(t, \mathbf{x}) = u(\mathbf{x})e^{-i\omega t} \tag{15}$$

Although the time dependence is analytically separable and purely periodic, such that the full trajectory is determined by the spatial field at a single time, the dataset remains physically meaningful as it encodes coherent wave propagation and phase dynamics. This setting therefore serves as a controlled benchmark for evaluating physical consistency (see Appendix F), complementing more dynamically complex systems.

Technical Configuration:

- **Channels:** To represent the complex-valued pressure fields, the model processes 2 primary channels: the real component $Re(U)$ and the imaginary component $Im(U)$ of the acoustic pressure. For all models, the constant domain masks provided in the dataset (Ohana et al., 2024) are used as a third input channel.

B Additional Theorems and Corollaries

Throughout, $D = 2$, $\mathcal{L}_{\text{pri}} = \mathcal{L}^{(1)}_{\text{traj}}$, $\mathcal{L}_{\text{sec}} = \mathcal{L}^{(2)}_{\text{traj}}$, and $v^\star(x,t) = \mathbb{E}[x_1 - x_0 \mid x_t = x]$. Training pairs are drawn as in Algorithm 1: $t \sim \mathcal{U}(0,1)$, $\alpha \mid t \sim \mathcal{U}(t,1)$, $\tau = t/\alpha$. Changing variables from (t,α) to (τ,α) gives the density $\alpha/(1-\alpha\tau)$, which is positive on $[0,1]\times(0,1]$, so every (τ,α) pair receives training signal. (It vanishes as $\alpha \to 0$, so small-scale secondary trajectories are rarely trained.)

Proposition B.1 (Population optimum)**.** *Let $\theta^\star$ minimize $\mathcal{L}_{total}$ for any $\lambda \geq 0$ over a sufficiently expressive function class. Then for all $(\tau,\alpha) \in [0,1]\times(0,1]$, the primary output is the vanilla flow matching optimum,*

$$v_{\theta^\star}(x,t,1) = v^\star(x,t) = \mathbb{E}[\, x_1 - x_0 \mid x_t = x\,], \tag{16}$$

The secondary output is a rescaling of it,

$$v_{\theta^\star}(x,\tau,\alpha) = \alpha\, v^\star(x,\alpha\tau), \tag{17}$$

and $\mathcal{L}_{cons} = 0$. In particular, RecFM and vanilla FM have the same population optimum at $\alpha = 1$, the only scale used at inference.

Proof. $\mathcal{L}_{\text{pri}}$ is a least-squares regression of $x_1 - x_0$ on (x_t, t), whose minimizer is the conditional expectation $v^\star$. $\mathcal{L}_{\text{sec}}$ regresses $\alpha(x_1 - x_0)$ on (x_t,τ,α) with $t = \alpha\tau$; its minimizer is $\alpha\,\mathbb{E}[x_1 - x_0 \mid x_{\alpha\tau} = x] = \alpha\, v^\star(x,\alpha\tau)$. Since the two losses act on disjoint conditioning inputs (almost surely $\alpha < 1$ for the secondary term), both minima are attained simultaneously, and the minimizer of $\mathcal{L}_{\text{pri}} + \mathcal{L}_{\text{sec}}$ is unique almost everywhere on the support. At this point $\mathcal{L}_{\text{cons}} = \|\alpha v^\star(x_t,t) - \alpha v^\star(x_t,t)\|^2 = 0$. Because $\mathcal{L}_{\text{cons}} \geq 0$, the same function minimizes $\mathcal{L}_{\text{total}}$ for every $\lambda \geq 0$. □

Theorem B.2 (Marginal Preservation of the Secondary Trajectory)**.** *Let v_{θ^*} be as in Proposition B.1, and assume that $v_{\theta^*}(\cdot,t,\alpha)$ is Lipschitz in x uniformly over t and α. For a fixed $\alpha \in (0,1]$, consider the secondary trajectory ODE:*

$$\frac{dx}{d\tau} = v_{\theta^*}(x,\,\tau,\,\alpha), \qquad x(0) \sim p_0. \tag{18}$$

Let $\{q^{(\alpha)}_\tau\}_{\tau\in[0,1]}$ denote the probability path induced by this ODE. Then for all $\tau \in [0,1]$, $q^{(\alpha)}_\tau$ coincides with the marginal distribution of $(1-\alpha\tau)\,x_0 + \alpha\tau\, x_1$, where $x_0 \sim p_0$ and $x_1 \sim p_1$. In particular,

$$q^{(\alpha)}_0 = p_0, \qquad q^{(\alpha)}_1 = p_\alpha, \tag{19}$$

where p_α is the marginal distribution of $(1-\alpha)\,x_0 + \alpha\, x_1$.

Proof. Define the candidate path $\tilde{x}_\tau := (1-\alpha\tau)\,x_0 + \alpha\tau\, x_1$. Differentiating: $\frac{\mathrm{d}\tilde{x}_\tau}{\mathrm{d}\tau} = \alpha\,(x_1 - x_0)$. Its marginal velocity field is

$$u^{(\alpha)}_\tau(x) := \mathbb{E}\left[\frac{\mathrm{d}\tilde{x}_\tau}{\mathrm{d}\tau} \;\middle|\; \tilde{x}_\tau = x\right] = \alpha\,\mathbb{E}[x_1 - x_0 \mid \tilde{x}_\tau = x]\,.$$

Since $\tilde{x}_\tau$ is the linear interpolant at fraction $\alpha\tau$, it has the same distribution as x_t with $t = \alpha\tau$. Substituting into equation 16 and comparing with equation 17:

$$u^{(\alpha)}_\tau(x) = \alpha\, v_{\theta^*}(x,\alpha\tau,1) = v_{\theta^*}(x,\tau,\alpha).$$

By uniqueness of solutions to the continuity equation under the Lipschitz assumption, the induced path $q^{(\alpha)}_\tau$ coincides with the marginal of $\tilde{x}_\tau$. Setting $\tau = 0$ and $\tau = 1$ gives equation 19. □

Theorem B.2 is a consistency guarantee: the secondary trajectories used during training describe valid transports between the correct marginals. It does not by itself imply better sampling, since inference uses only $\alpha = 1$.

Proof of Lemma 2.1. Let e_k be the error after k steps and $M = \sup_t \|a(\psi_t,t)\|$. Since $\ddot{\psi}_t = a(\psi_t,t)$, the local truncation error of one Euler step is at most $\frac{h^2}{2}M$, and the Lipschitz condition gives $e_{k+1} \leq (1+hL)\,e_k + \frac{h^2}{2}M$ with $e_0 = 0$. Unrolling, $e_K \leq \frac{hM}{2L}\big((1+hL)^K - 1\big) \leq \frac{h}{2}\,\frac{e^L-1}{L}\,M$, using $(1+hL)^K \leq e^L$ for $h = 1/K$. □

Remark B.3 (What the consistency loss constrains). *Set $\alpha = 1-\epsilon$, so $\tau = t/(1-\epsilon) = t+t\epsilon+O(\epsilon^2)$. A first-order expansion of the consistency residual gives*

$$v_\theta(x_t, \tau, \alpha) - \alpha\, v_\theta(x_t, t, 1) = \epsilon\big[\, t\,\partial_t v_\theta + v_\theta - \partial_\alpha v_\theta \,\big](x_t, t, 1) + O(\epsilon^2),$$

so $\mathcal{L}_{cons}$ penalizes, at leading order, $\|t\,\partial_t v_\theta + v_\theta - \partial_\alpha v_\theta\|^2$. This condition relates the network's dependence on α to its dependence on t; it does not constrain $\partial_t v_\theta$ itself. Indeed, for any primary field $g(x,t)$, the rescaled field $v(x,\tau,\alpha) = \alpha\, g(x, \alpha\tau)$ has $\partial_\alpha v\,|_{\alpha=1} = g + t\,\partial_t g$ and satisfies the condition identically, however curved g is.

Corollary B.4 (Consistent Few-Step Sampling). *Let v_{θ^*} be as in Proposition B.1.*

(i) *For any $\alpha \in (0, 1]$, a single Euler step of size α along the* primary *trajectory at $t = 0$ yields a sample whose distribution matches the endpoint of the secondary trajectory:*

$$x_0 + \alpha\, v_{\theta^*}(x_0, 0, 1) \;\overset{d}{=}\; x_0 + \int_0^1 v_{\theta^*}\big(x(\tau), \tau, \alpha\big)\, d\tau \;\sim\; p_\alpha. \tag{20}$$

Consequently, the family of secondary trajectories indexed by α provides a continuum of self-consistent few-step samplers, each producing a valid interpolant marginal p_α regardless of the discretization granularity.

(ii) *By Lemma 2.1, one-step generation is exact if and only if the trajectory acceleration $\mathbf{a} \equiv 0$. Since $\mathcal{L}_{\text{cons}}$ directly penalizes $\|\partial_t v_\theta\|$, a component of $\|\mathbf{a}\|$, RecFM actively drives the trajectory toward the zero-curvature regime where few-step Euler integration is accurate. Vanilla FM imposes no such constraint, leaving trajectory curvature uncontrolled.*

C Additional Results

C.1 Influence of Inference Steps

We further analyze the effect of the number of inference steps in RecFM. Figure 5 shows the MSE as a function of the number of inference steps on the Navier-Stokes task. RecFM achieves its best performance with one- or two-step generation. We hypothesize that this behavior is due to the largely deterministic nature of physics-governed systems, where longer sampling trajectories can introduce accumulated errors.

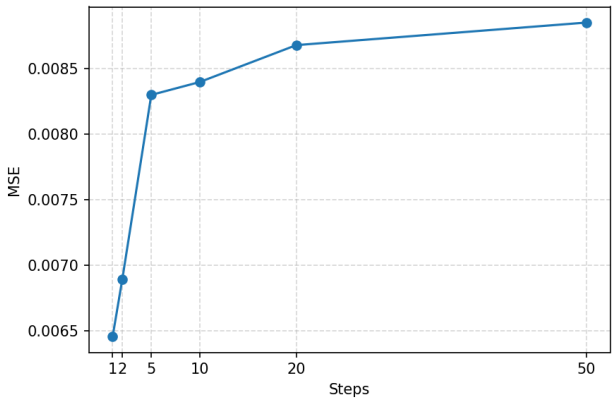


Figure 5: **MSE vs. inference steps on the Navier-Stokes benchmark.** RecFM achieves optimal performance with one- and two-step generation, while increasing the number of steps leads to error accumulation.

C.2 Influence of Recursion Depth D

We study the effect of recursion depth D on model performance. The depth D controls the number of trajectory scales used during training, thereby governing the strength of multi-scale supervision. While RecFM particularly corresponds to the depth-2 case in our main experiments, higher depths introduce additional consistency constraints across more trajectory scales.

Table 3 presents a comparison of different recursion depths on the Navier-Stokes benchmark. The vanilla FM ($D = 1$) needs multi-step inference (5 steps) to reach acceptable performance, whereas RecFM ($D = 2$) already achieves strong results with a single inference step. Raising the recursion depth to $D = 3$ leads to slightly inferior performance compared to $D = 2$, with small declines in MSE, SSR, and inference speed. In addition, training the depth-3 model requires larger memory than training the depth-2 RecFM due to the extra gradient terms introduced at depth 3. These observations indicate that enforcing pairwise consistency is sufficient to obtain the advantages of multi-scale alignment, and that further increasing the recursion depth yields diminishing returns.

Based on these results, we adopt $D = 2$ as the default configuration in the main experiments, as it achieves strong performance while maintaining a simple and efficient formulation.

Table 3: Ablation study (mean $\pm$ std) on recursion depth D for Navier-Stokes flow. Vanilla FM uses 5-step inference, while RecFM variants operate in the 1-step regime.

Depth D	**CRPS** ($\downarrow$)	**MSE** ($\downarrow$)	**SSR** ($\rightarrow 1$)	**Time [s]**
$D = 1$ (Vanilla FM, 5-step)	$0.036_{\pm 0.001}$	$0.0076_{\pm 0.0001}$	$0.911_{\pm 0.008}$	6.914
$D = 2$ (RecFM, 1-step)	$0.031_{\pm 0.001}$	$0.0064_{\pm 0.0001}$	$0.959_{\pm 0.003}$	1.588
$D = 3$ (extended RecFM, 1-step)	$0.031_{\pm 0.001}$	$0.0065_{\pm 0.0001}$	$1.091_{\pm 0.002}$	1.594

C.3 Additional Training Dynamics and Convergence

We further compare validation MSE versus NFE during training for flow matching methods. Figure 6 shows that RecFM converges faster than Vanilla FM and maintains lower validation error, demonstrating improved efficiency and stability.

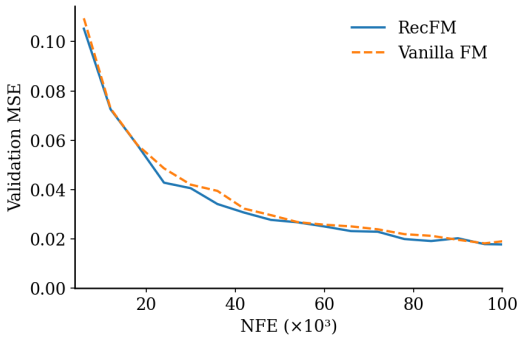


Figure 6: **Flow matching validation MSE versus NFE during training.** RecFM converges faster than Vanilla FM and maintains consistently lower validation error.

D Architecture and Implementation Details

We adopt the state-of-the-art Hierarchical Video Diffusion Transformer (HV-DiT) backbone from VideoPDE (Li et al., 2025), with the sole modification that the input mask channel is removed. Unlike DYffusion (Rühling Cachay et al., 2023), whose forecasting-oriented formulation is not naturally compatible with transformer-based diffusion backbones such as HV-DiT, RecFM operates directly on the learned velocity field and can be integrated into existing spatiotemporal diffusion architectures with minimal modification. This also makes direct comparison against the original DYffusion architecture a more appropriate and fair evaluation setting.

During training, our recursive formulation requires D forward/backward gradient evaluations per iteration due to multi-scale trajectory supervision. To maintain comparable overall training cost, we reduce the total number of training iterations by the same factor D relative to other models.

A detailed overview of the model architecture and its hyperparameters (*e.g.*, for the Navier-Stokes equation with $D = 2$) is provided in Table 4.

Table 4: **RecFM training and model hyperparameters of Navier-Stokes Flow.**

Hyperparameter	RecFM (NS Flow)
Parameters	116.2M
Training steps	40k
Batch size	64
GPUs	4 × L40S
Mixed Precision	bfloat16
Patch Size ($T \times H \times W$)	[2, 2, 1]
Neighborhood Attention Levels	1
Global Attention Level	1
Neighborhood Attention Depth	2
Global Attention Depth	11
Feature Dimensions	[384, 768]
Attention Head Dimension	64
Neighborhood Kernel Size ($T \times H \times W$)	[2, 7, 7]
Mapping Depth	1
Mapping Width	768
Dropout	0
Optimizer	AdamW
Learning Rate	5×10^{-4}
$[\beta_1, \beta_2]$	$[0.9, 0.95]$
λ	1
Epsilon	1×10^{-8}
Weight Decay	1×10^{-2}

E Additional Visualizations

Visualization of more timesteps of the Helmholtz Staircase equation is shown in Figure 7. Additionally, we visualize a representative Navier-Stokes rollout in Figure 8.

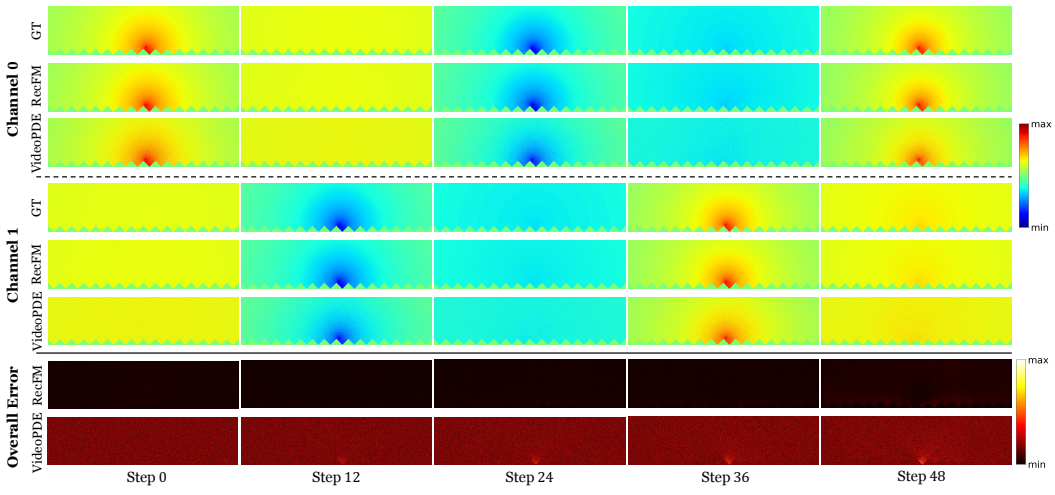


Figure 7: **More roll-out results of the Helmholtz Staircase equation.** Visual comparison of Ground Truth against RecFM and VideoPDE (best-performing baseline across datasets) for two channels, with the bottom rows indicating absolute errors. Columns correspond to dataset timesteps.

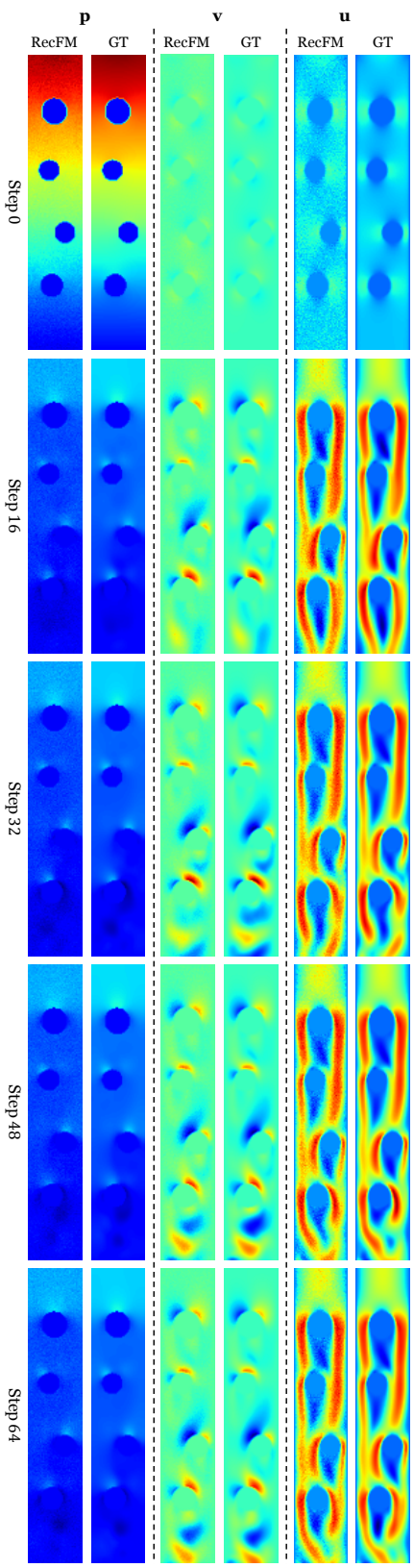


Figure 8: **Navier-Stokes rollout sample.**

F PHYSICS-INFORMED EVALUATION

We further compare RecFM with PBFM (Baldan et al., 2026) on PDE-governed datasets in Table 5, following the original setting of 20 inference steps. While PBFM consistently improves over vanilla Flow Matching, its iterative refinement procedure limits both efficiency and accuracy relative to RecFM. In contrast, RecFM achieves higher accuracy while requiring only 1-2 inference steps, resulting in substantially faster generation.

To assess physical consistency, we additionally report physics-informed evaluation metrics. For the Navier-Stokes dataset, the trajectories are transient and do not reach a statistically stationary regime, making standard long-time turbulence diagnostics inapplicable. Instead, we evaluate the average kinetic energy $\langle E(t) \rangle$, normalized by the initial ground-truth energy $\langle E_{\text{real}}^{t=0} \rangle$. In Table 5, we report KE Accuracy, which measures the relative agreement between predicted and ground-truth energy (values closer to 1 indicate better physical fidelity). The rollout of kinetic energy is shown in Figure 9. While PBFM enforces strong physical constraints at each autoregressive iteration, errors accumulate over time. In contrast, RecFM maintains stable dynamics and achieves lower overall error without noticeable accumulation.

For the Helmholtz Staircase equation, we further report the PDE residual of the wave equation $\partial^2 U/\partial t^2 + \omega^2 U = 0$ in the table, where values closer to zero indicate better adherence to the governing dynamics.

These results demonstrate that RecFM not only produces visually accurate predictions, but also more faithfully preserves the underlying physical dynamics due to its few-step nature.

Table 5: Physics-informed quantitative forecasting results for Navier-Stokes Flow, and Helmholtz Staircase Equation. Lower values are better for MSE, CRPS, and PDE Residual, while SSR and KE Accuracy are optimal when closer to 1. Best results in **bold**.

Method	Navier-Stokes					Helmholtz Staircase			
	CRPS	MSE	SSR	KE Accuracy	Time [s]	CRPS	MSE	SSR	PDE Residual
VideoPDE	0.033	0.0068	0.205	0.9670	72.64	0.026	5.6e-4	4.334	0.00841
Vanilla FM	0.036	0.0076	0.911	0.9522	6.914	0.030	6.5e-4	1.485	0.01102
PBFM	0.034	0.0071	0.810	0.9592	14.75	0.0094	1.2e-4	0.737	0.00519
RecFM (1-step)	**0.031**	**0.0064**	**0.959**	**0.9791**	1.588	0.0034	4.2e-5	**1.090**	0.00476
RecFM (2-step)	0.032	0.0068	0.932	0.9672	3.128	**0.0027**	**2.7e-5**	1.440	**0.00457**

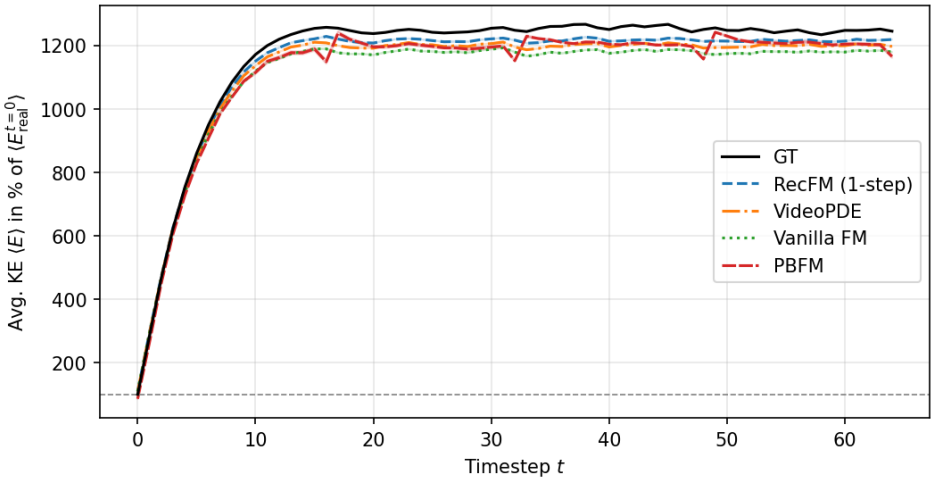


Figure 9: **Average kinetic energy over time.** $\langle E(t) \rangle$ normalized by $\langle E_{\text{real}}^{t=0} \rangle$. 100% corresponds to the initial ground-truth energy.

G STATISTICAL SIGNIFICANCE

We additionally report the standard deviation of all metrics (CRPS, MSE, and SSR) for RecFM, VideoPDE, Vanilla FM, and Shortcut Model (*i.e.*, all methods using the HV-DiT backbone, as shown in Tables 6 and 7.

Table 6: Quantitative forecasting results (mean ± std) on SST and Navier–Stokes datasets. Lower CRPS and MSE are better, while SSR closer to 1 indicates better calibration.

Method	**SST**			**Navier–Stokes**		
	CRPS ↓	MSE ↓	SSR → 1	CRPS ↓	MSE ↓	SSR → 1
VideoPDE	$0.216_{\pm 0.005}$	$0.162_{\pm 0.002}$	$0.746_{\pm 0.007}$	$0.033_{\pm 0.002}$	$0.0068_{\pm 0.0003}$	$0.205_{\pm 0.013}$
Vanilla FM	$0.260_{\pm 0.007}$	$0.232_{\pm 0.002}$	$0.914_{\pm 0.006}$	$0.036_{\pm 0.001}$	$0.0076_{\pm 0.0001}$	$0.911_{\pm 0.008}$
Shortcut	$0.249_{\pm 0.003}$	$0.297_{\pm 0.002}$	$0.798_{\pm 0.008}$	$0.034_{\pm 0.002}$	$0.0079_{\pm 0.0002}$	$0.717_{\pm 0.007}$
RecFM (1-step)	$0.217_{\pm 0.003}$	$0.162_{\pm 0.001}$	$0.984_{\pm 0.006}$	$0.031_{\pm 0.001}$	$0.0064_{\pm 0.0001}$	$0.959_{\pm 0.003}$
RecFM (2-step)	$0.216_{\pm 0.003}$	$0.161_{\pm 0.001}$	$1.004_{\pm 0.007}$	$0.032_{\pm 0.001}$	$0.0068_{\pm 0.0001}$	$0.932_{\pm 0.004}$

Table 7: Quantitative forecasting results (mean ± std) on the Helmholtz Staircase dataset. Lower CRPS and MSE are better, while SSR closer to 1 indicates better calibration.

Method	CRPS ↓	MSE ↓	SSR → 1
VideoPDE	$0.026_{\pm 0.001}$	5.6e-4$_{\pm 1e\text{-}5}$	$4.334_{\pm 0.071}$
Vanilla FM	$0.030_{\pm 0.001}$	6.5e-4$_{\pm 1e\text{-}5}$	$1.485_{\pm 0.012}$
Shortcut	$0.0144_{\pm 0.0006}$	1.6e-4$_{\pm 1e\text{-}5}$	$0.467_{\pm 0.063}$
RecFM (1-step)	$0.0034_{\pm 0.0001}$	4.2e-5$_{\pm 1e\text{-}6}$	$1.090_{\pm 0.010}$
RecFM (2-step)	$0.0027_{\pm 0.0001}$	2.7e-5$_{\pm 1e\text{-}6}$	$1.440_{\pm 0.012}$

H SHORTCUT MODELS VS. RECFM

Although both Shortcut Models (Frans et al., 2024) and RecFM enforce self-consistency to enable few-step generation, they impose structurally distinct constraints on the learned velocity field. Shortcut Models parameterize the network by a step-size d and enforce a *compositional* consistency condition: a single step of size $2d$ must produce the same result as two consecutive steps of size d,

$$x_t + 2d \cdot v_\theta(x_t, t, 2d) = \big(x_t + d \cdot v_\theta(x_t, t, d)\big) + d \cdot v_\theta\big(x_t + d \cdot v_\theta(x_t, t, d),\ t + d,\ d\big). \tag{21}$$

Crucially, the right-hand side evaluates v_θ at the *step-forward* state $x_t + d \cdot v_\theta(x_t, t, d)$, coupling the constraint to the spatial geometry of the trajectory. In contrast, RecFM conditions the network on a continuous scale parameter α and enforces a *pointwise scaling* relation at the same spatial location x_t:

$$\mathcal{L}_{\text{cons}} = \big\| v_\theta(x_t,\, \tau,\, \alpha) - \alpha\, v_\theta(x_t,\, t,\, 1) \big\|_2^2, \qquad \tau = t/\alpha. \tag{22}$$

No choice of the hyperparameters (α, λ) can reduce Equation 22 to Equation 21, because the former is evaluated at a single point while the latter intrinsically depends on a forward Euler update.

In our time convention, sampling moves from t toward 0, and the Shortcut condition reads

$$x_t - 2d\, s_\theta(x_t, t, 2d) = x' - d\, s_\theta(x', t - d, d), \qquad x' = x_t - d\, s_\theta(x_t, t, d), \tag{23}$$

where $s_\theta(x, t, d)$ is the step-size-conditioned output. At the optimum, $s^\star(x, t, d)$ is the *average* velocity of the exact flow over $[t - d, t]$, i.e., $x - d\, s^\star(x, t, d) = \psi_{t-d}(x)$. Taylor expanding the exact flow,

$$s^\star(x, t, d) = v^\star(x, t) - \tfrac{d}{2}\, a(x, t) + O(d^2). \tag{24}$$

Shortcut models therefore do not remove curvature; they absorb it into the dependence on d, so that a single large step lands on the exact ODE endpoint even when the flow is curved. A one-step Shortcut sample from noise thus approximates a sample from p_0.

RecFM behaves differently. Its leading-order consistency condition is satisfied identically by the rescaling $v(x, \tau, \alpha) = \alpha\, v(x, \alpha\tau, 1)$ for any primary field (Remark B.3), so it constrains neither curvature nor the flow map. At the optimum, the $\alpha = 1$ field is the instantaneous FM velocity $v^\star(x, t) = \mathbb{E}[x_1 - x_0 \mid x_t = x]$, and a one-step Euler sample returns $\mathbb{E}[x_0 \mid c]$. The two methods thus target different objects in the one-step regime: Shortcut targets a sample, while RecFM (like vanilla FM) targets the conditional mean.

Role of the scaled velocity. The scaled velocity is never used at inference; generation always proceeds with $\alpha = 1$. A state x_t lies both on the primary trajectory (at time t) and on secondary trajectories toward the partial target $x_\alpha = (1 - \alpha)x_0 + \alpha x_1$ (at time $\tau = t/\alpha$). At the population optimum, the secondary predictions are an exact rescaling of the primary ones (Proposition B.1), so they carry no additional information about $v^\star$.

Performance Comparison on Physics Dynamics. In Table 1, we compare the Shortcut Model with RecFM on the physics dynamics datasets for 1-step generation. RecFM achieves better performance, while the Shortcut Model underperforms despite hyperparameters being carefully tuned to the best of our knowledge. We note that the Shortcut Model is primarily designed for static image generation, and extending it to dynamic settings presents additional challenges.

I MeanFlow on Physics Dynamics

We also reimplement MeanFlow (Geng et al., 2025), which was originally developed for image generation, on our physics dynamics datasets using the same backbone as RecFM. We find that directly adapting MeanFlow to this setting leads to substantially less stable optimization than in its original image-generation setting, requiring considerable implementation effort and hyperparameter tuning. Despite these challenges, we obtain an one-step model on the Navier-Stokes dataset and report its results in Table 8. Under the same backbone and evaluation setting, RecFM consistently outperforms MeanFlow across all reported metrics.

Table 8: 1-step quantitative forecasting results of MeanFlow and RecFM on the Navier-Stokes dataset.

Method	**Navier-Stokes**		
	CRPS	**MSE**	**SSR**
MeanFlow (Geng et al., 2025)	0.042	0.0093	0.843
RecFM	0.031	0.0064	0.959

I.1 Trajectory Curvature

To test whether RecFM learns straighter trajectories than vanilla FM, we measure curvature of the learned $\alpha = 1$ field directly. For each test sample, we solve the sampling ODE with a 256-step Euler reference from the same noise draw for both models, on the first prediction window only. We report the straightness $S = \int_0^1 \|D - v(x_t, t)\|^2 dt \,/\, \|D\|^2$ with $D = x_1 - \hat{x}_0$ (Liu et al., 2022), the mean and 95th-percentile acceleration $\|a\|/\|D\|$ along the trajectory, and the gap between K-step and reference samples, the quantity bounded by Lemma 2.1. Our results in Table 9 show that RecFM consistently learns straighter trajectories.

Table 9: Curvature of the learned $\alpha = 1$ field (mean $\pm$ s.e.m.; paired noise). Lower is straighter.

Dataset	Method	S	$\overline{\|a\|}$	$\|a\|_{95}$	gap_1	gap_2
SST	Vanilla FM	$0.0135_{\pm 0.0018}$	$1.255_{\pm 0.051}$	$3.317_{\pm 0.143}$	$0.246_{\pm 0.026}$	$0.209_{\pm 0.023}$
	RecFM	$0.0124_{\pm 0.0011}$	$1.140_{\pm 0.037}$	$2.662_{\pm 0.260}$	$0.224_{\pm 0.015}$	$0.186_{\pm 0.014}$
Navier-Stokes	Vanilla FM	$0.0010_{\pm 0.0003}$	$1.258_{\pm 0.033}$	$0.617_{\pm 0.009}$	$0.024_{\pm 0.002}$	$0.014_{\pm 0.002}$
	RecFM	$0.0006_{\pm 0.0001}$	$1.135_{\pm 0.042}$	$0.609_{\pm 0.008}$	$0.023_{\pm 0.002}$	$0.013_{\pm 0.001}$
Helmholtz	Vanilla FM	$0.0006_{\pm 0.0001}$	$1.196_{\pm 0.141}$	$0.333_{\pm 0.007}$	$0.006_{\pm 0.001}$	$0.004_{\pm 0.001}$
	RecFM	$0.0002_{\pm 0.0001}$	$1.173_{\pm 0.138}$	$0.267_{\pm 0.003}$	$0.004_{\pm 0.001}$	$0.003_{\pm 0.001}$



Figure 10: **Selected samples from our** 256×256 **resolution RecFM-XL model.**

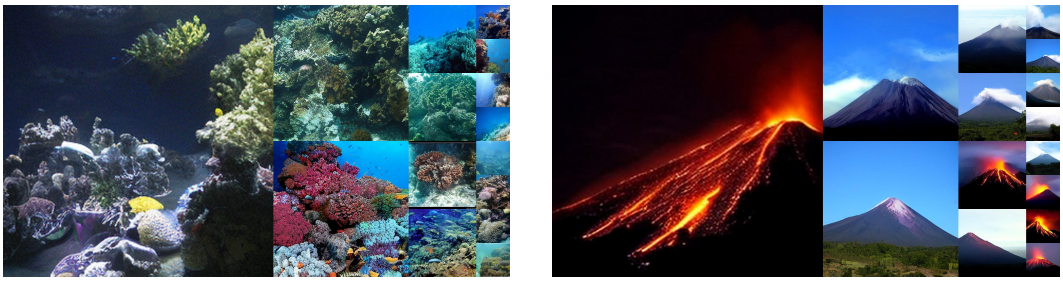
J Recursive Flow Matching for Image Generation

Table 10 presents the image generation performance of our proposed RecFM-XL model compared to other generative baselines. We report the Fréchet Inception Distance (FID) and evaluate all models with classifier-free guidance (CFG = 1.5) on the ImageNet-1k dataset (Deng et al., 2009). RecFM achieves an FID < 3 within 16 sampling steps. Our results show that RecFM is competitive in image generation as a multi-step flow matching method, while requiring fewer training epochs and inference steps compared with DiT (Peebles & Xie, 2023) and SiT (Ma et al., 2024). Notably, RecFM performs better with 16 inference steps than with 128, which we attribute to its training objective that emphasizes few-step generation, limiting improvements from additional steps.

Table 10: Comparison of generative models under different sampling regimes.

Model	FID ↓	Sampling Steps	Param Count	Epochs Trained
DiT-XL (Peebles & Xie, 2023)	2.27	500	675M	640
SiT-XL (Ma et al., 2024)	2.06	250	675M	640
ADM-G (Dhariwal & Nichol, 2021)	4.59	250	–	426
LDM-4-G (Rombach et al., 2022)	3.6	500	400M	106
Rectified Flow (XL) (Liu et al., 2022)	4.04	100	675M	-
Shortcut Model (XL) (Frans et al., 2024)	3.8	128	676M	250
MeanFlow (XL) (Geng et al., 2025)	6.16	16	676M	240
MeanFlow (XL) (Geng et al., 2025)	5.65	8	676M	240
MeanFlow (XL) (Geng et al., 2025)	3.43	1	676M	240
RecFM-XL	2.53	128	675M	160
RecFM-XL	2.49	16	675M	160
RecFM-XL	3.22	8	675M	160

We further include some visualizations of RecFM-XL model with CFG in Figures 10, 11, 12.



(a) Coral reef (973)

(b) Volcano (980)

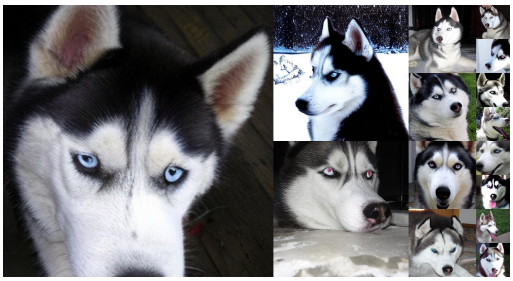
Figure 11: **Uncurated** 256×256 **RecFM-XL samples.** Each panel shows samples from a different ImageNet class.



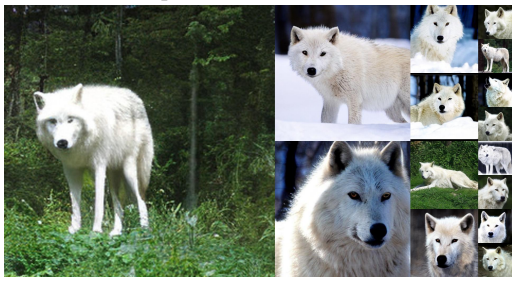
(a) Macaw (88)



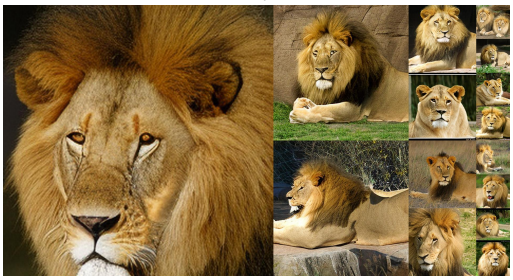
(b) Sulphur-crested cockatoo (89)



(c) Husky (250)



(d) Arctic wolf (270)



(e) Lion (291)



(f) Otter (360)



(g) Red panda (387)



(h) Panda (388)



(i) Balloon (417)



(j) Cliff drop-off (972)

Figure 12: **Uncurated** 256 × 256 **RecFM-XL samples (Continued).** Each panel shows samples from a different ImageNet class.