# Benchmarking Stochastic Interpolants for Modeling Physical Systems

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

## Abstract

Generative models have recently emerged as powerful surrogates for physical systems, demonstrating increased accuracy, stability, and/or statistical fidelity. However, most approaches rely on iteratively denoising a Gaussian, a choice that may not be the most effective for autoregressive prediction tasks, where the current and future states are often closely related. We therefore benchmark generative models based on stochastic interpolants, which can learn maps between arbitrary densities, against standard Gaussian-denoising frameworks across a range of fluid and climate systems. Following prior work, we consider a transport map for dynamical systems that learns an interpolant between a point mass at the current state and a conditional distribution of future states. We find that no single model is universally best: whether a generative model is needed at all, and which one performs best, depends on the physics of the system, the sampling scheme, and the performance metric of interest. Within this landscape, stochastic interpolants can be a strong and competitive baseline. They benefit from the proximity of successive states and are highly flexible during inference, with the capability to tradeoff deterministic accuracy, spectral consistency, and probabilistic calibration. This flexibility makes them well-suited to tailoring a generative surrogate to a given system and objective, and motivates further exploration of transport maps designed specifically for physical emulation.

## 1 Introduction

The field of neural PDE surrogates has grown immensely since its initial foundations (Li et al., 2021; Lu et al., 2021), with a wide range of architectures now being applied across many scientific and engineering applications (Ohana et al., 2025). Within this landscape, generative models have emerged as a particularly promising class of methods for modeling physical systems, achieving state-of-the-art results in modeling phenomena such as turbulence (Molinaro et al., 2025) and weather (Price et al., 2024). These results are supported by many empirical findings, where diffusion models have demonstrated better accuracy and stability (Lippe et al., 2023), capable of resolving finer details than deterministic baselines (Oommen et al., 2025). In addition, diffusion models can be more effective at capturing the underlying statistics of physical systems, such as in turbulence (Lienen et al., 2023; Molinaro et al., 2025) or in weather forecasting and climate prediction (Price et al., 2024; Cachay et al., 2023). Overall, these capabilities are supported by studies that benchmark diffusion models in PDE systems (Kohl et al., 2024; Rozet et al., 2025), as well as by continued research that improves their accuracy or inference speed (Bastek et al., 2025; Shehata et al., 2025). Moreover, these models continue to benefit from larger advances in the generative modeling community, such as flow matching or improved samplers (Liu et al., 2022; Lu et al., 2025).

While promising, a key feature of prior generative models in PDE or climate modeling is the assumption of a Gaussian prior/source distribution. This is a logical choice for unconditional generation, where we have no prior knowledge about the source distribution and require it to be easily sampled. However, recent work has challenged this assumption for tasks where the source and target distributions are related, such as in image-to-image translation or super-resolution. Ordinarily, these tasks are framed as sampling from a Gaussian and evolving a reverse process conditioned on the source distribution, however, methods such as diffusion bridges or stochastic interpolants seek to directly learn a stochastic process between the source and

target distributions (Zhou et al., 2023; Albergo & Vanden-Eijnden, 2023). Directly evolving samples drawn from the source distribution (e.g., blurry images, masked images) to samples from the target distribution (e.g., sharp images, in-painted images) can require fewer sampling steps and produce higher quality samples (Albergo et al., 2024; Zheng et al., 2025). In these cases, it is believed that transporting Gaussian noise to a conditional distribution is more complex than directly mapping the source to the target distribution.

This observation motivates stochastic interpolants as a generative framework for physical systems. Predicting PDE or climate systems is usually framed as an autoregressive task, where future states are predicted from a current state. Because successive states are often tightly coupled, transporting Gaussian noise while conditioning on the current state may inefficient, whereas an interpolant can exploit this proximity by mapping directly between the current and future distributions. To test whether this and related approaches help in practice, we benchmark a range of generative models across several physical emulation tasks. Our experiments do not identify a single best model; rather, we find that the most effective approach depends on the underlying physics. For laminar or well-resolved systems a deterministic surrogate can be preferable, while for chaotic or multiscale systems generative models offer a clear benefit, and the relative ranking of generative frameworks can depend on the desired goals and metrics. Within this varied landscape, stochastic interpolants are can be a strong baseline: they leverage the proximity of successive states, and because the framework is highly flexible and subsumes common Gaussian-denoising models as special cases, it can be tuned to trade off deterministic accuracy, spectral consistency, and probabilistic calibration for a given task. This suggests that conventional generative frameworks are likely not optimized for physical systems and that there are opportunities to design transport maps tailored to these systems.

## 2 Related Work

**Generative Models** Modern generative models can be understood as the dynamical transport of one probability density into another. Diffusion models learn to reverse a forward noising process that gradually corrupts data into Gaussian noise, formulated either as denoising diffusion (Ho et al., 2020) or through the lens of score matching and stochastic differential equations (Song et al., 2021). Subsequent work improved the sampling process, introducing deterministic and accelerated samplers (Song et al., 2022) and a principled study of the design space governing noise schedules, preconditioning, and discretization (Karras et al., 2022). Flow matching offers another alternative, regressing directly onto the velocity field of a probability path to bridge exactly between two densities in finite time (Lipman et al., 2023), with rectified flows further straightening transport trajectories to enable few-step sampling (Liu et al., 2022). A common thread is that these methods transport a fixed Gaussian prior to a target density in order to sample from it.

A second line of work relaxes the requirement of a Gaussian prior, allowing transport between two arbitrary densities. Such formulations arise under several related frameworks, including Schrödinger Bridges (Bortoli et al., 2023), Diffusion Bridges (Su et al., 2023; Zhou et al., 2023), Optimal Transport (Tong et al., 2024), and Stochastic Interpolants (Albergo et al., 2023). The stochastic interpolant framework in particular unifies flow- and diffusion-based models by constructing an explicit bridge between any two densities in finite time and derives training objectives and sampling algorithms to both learn this interpolant and allow deterministic flows or stochastic samplers. Viewed through this lens, widely used methods such as DDPM and EDM can be seen as particular instantiations of a more general principle of dynamical transport between distributions. Together, these theoretical and practical foundations have greatly expanded the capabilities of generative models and enabled a wide range of applications. In particular, generative models have seen widespread adoption in predicting and modeling important physical phenomena.

**Deterministic Neural Surrogates** Deep learning surrogates for PDE and climate systems are a diverse and rapidly growing field, where surrogate models generally learn a deterministic map from the current system state to a future state. Many of these methods can be cast as neural operators, which learn mappings between infinite-dimensional function spaces and therefore generalize across discretizations (Li et al., 2021; Lu et al., 2021; Kovachki et al., 2023). Building on this paradigm, a wide range of architectures have been proposed, including convolutional and spectral models (Pathak et al., 2022; Raonic et al., 2023), transformer- and attention-based operators (Li et al., 2023; Bi et al., 2022; Li et al., 2024), the use of large-scale pretraining

and foundation models (Zhou & Farimani, 2024; McCabe et al., 2024; Zhou et al., 2024; Nguyen et al., 2023), and models that use physics-based priors or inductive biases (Verma et al., 2024; Zhou & Farimani, 2025).

For time-dependent systems, these surrogates are typically queried autoregressively, applying a learned single-step operator repeatedly to roll out a trajectory. While effective at minimizing pointwise error over short horizons, there are several well-studied failure modes. Models are generally trained on ground-truth inputs but use their own predictions at inference time; therefore, the input distribution gradually drifts from the training distribution and causes errors to compound over long rollouts (Lippe et al., 2023; List et al., 2025). This is compounded by the fact that models trained with mean-squared-error losses are subject to a spectral bias that prioritizes low-frequency modes, yielding predictions that regress toward the conditional mean and progressively lose high-frequency content. This issue is especially prevalent in chaotic and turbulent dynamics, resulting in model predictions that smooth over time or model errors in the high-frequency band that eventually transport to the lower frequencies (McCabe et al., 2023). Lastly, a deterministic surrogate produces only a single trajectory, therefore, it cannot represent the uncertainty inherent to chaotic systems or provide calibrated probabilistic forecasts. These limitations have motivated the use of generative models in certain physical systems, which instead learn to sample from a distribution over future states.

**Generative Surrogates for Physical Systems** Physical systems can be seen as probabilistic in many practical settings. A lack of spatial resolution, the need to ignore or approximate sub-grid effects, partial observation, or chaotic systems all contribute to underdetermined systems where the map from the current state to the future is multimodal and is poorly approximated by a point estimate. Generative surrogates address this directly by learning the full conditional distribution of future states given the current state, $p(u_{t+1} \mid u_t)$, and sampling from it rather than regressing onto its mean. Because individual samples are drawn from the learned data distribution, they retain sharp small-scale structure and physically consistent energy spectra instead of the over-smoothed fields that result from minimizing a mean-squared-error loss. This makes generative models compelling for turbulent and multiscale systems, where spectral and statistical fidelity are important, and a growing body of work has shown that they can match or exceed deterministic surrogates on both accuracy and distributional metrics (Oommen et al., 2025; Shu et al., 2023; Cachay et al., 2023). Generative models have also been shown to remain more stable than their deterministic counterparts during autoregressive rollout (Kohl et al., 2024; Pedersen et al., 2025; Liu et al., 2026); while the precise mechanism is unknown, it is hypothesized that the injection of noise at each timestep continuously perturbs the state and reintroduces high-frequency modes into the rollout that may otherwise decay over time.

Many prior works have applied diffusion models to predict PDE systems (Yang & Sommer, 2023; Huang et al., 2024; Shu et al., 2023; Cachay et al., 2023; Du et al., 2024; Serrano et al., 2024; Molinaro et al., 2025; Gao et al., 2024), typically by conditioning the generative process on the current state and sampling successive states autoregressively. Beyond direct prediction, these models have been adapted to the structure of physical problems in several ways, such as incorporating physics-informed losses to enforce governing equations (Bastek et al., 2025), operating on irregular meshes to handle complex geometries (Lino et al., 2025), and allowing text-conditioned generation of physical fields (Zhou et al., 2025). Progress in the broader field of generative models, such as latent diffusion (Rombach et al., 2022; Rozet et al., 2025), accelerated samplers (Shehata et al., 2025), or sampling guidance (Yao et al., 2026), have also improved performance in physics-based problems. Recent progress in flow matching has also been reflected in PDE emulation, where learning a velocity field along a straightened probability path can yield additional speed or accuracy benefits (Li et al., 2025b; Baldan et al., 2025; Utkarsh et al., 2025; Armegioiu et al., 2025; Shi et al., 2024).

Generative approaches are also prevalent and successful in weather forecasting, where the inherently probabilistic, chaotic nature of the atmosphere makes ensemble generation a natural fit and established diffusion and flow matching models as standard tools (Price et al., 2024; Zhuang & Duraisamy, 2025; Cachay et al., 2024; Couairon et al., 2024; Cachay et al., 2025). Indeed, current state-of-the-art weather forecasting models incorporate probabilistic or generative components (Alet et al., 2025; Lang et al., 2024) and are being deployed to complement numerical weather prediction in operational forecasts. There are also several practical tasks in weather forecasting that are ill-posed or inverse problems, where many states are consistent with the available information. In these tasks, such as downscaling or data assimilation, generative models have also shown strong performance (Mardani et al., 2024; Gao et al., 2023; Andry et al., 2025; Aich et al., 2024).

These prior works highlight the unique capabilities of generative models and identify applications and problems for which they are best suited. For chaotic or underdetermined systems, generative models can be a natural and effective choice. These systems exhibit forecast uncertainty that grows with lead time, and generative models can produce calibrated ensembles and uncertainty estimates that deterministic models cannot. Furthermore, for multiscale or turbulent systems, generative models can reproduce small-scale structure and energy spectra by sampling from the conditional distribution and avoiding the spectral collapse of mean-regression. Lastly, there are emergent phenomena such as increased autoregressive stability that are beneficial. However, the principal cost of these benefits is the increased inference time and training complexity. Generating a sample requires integrating a learned dynamical process over many steps, which can be orders of magnitude slower than a single deterministic forward pass. Furthermore, a wide variety of training losses and parameterizations have been proposed due to the complexity of learning a probabilistic map, as opposed to minimizing a more straightforward mean-squared regression objective. These practical limitations serve as the motivation for this work, where we seek to investigate if a principled choice of transport map, rather than a fixed Gaussian prior, can improve inference speed and overall performance.

**Stochastic Interpolants for Physical Systems** While diffusion and flow matching models based on Gaussian denoising are now well established and validated, interpolant-based approaches that map between arbitrary distributions remain comparatively rare in physical systems. Several works have applied them to super-resolution and data assimilation in PDEs and climate (Bischoff & Deck, 2023; Rout et al., 2025; Schiødt et al., 2025; Chen et al., 2025b), motivated by the resemblance of these problems to the denoising and inpainting tasks where stochastic interpolants have proven effective in computer vision. Most directly related to our work, Chen et al. (2024) apply interpolants to forecasting dynamical systems, learning a transport map from a point mass at the current state to the conditional distribution of future states and developing much of the supporting theory. Complementary theoretical advances have since refined the design of the interpolant itself, for example through scale-adaptive flows for multiscale scientific data (Chen & Vanden-Eijnden, 2026) and principled choices of interpolation schedule (Chen et al., 2026).

Whereas these prior works establish the theoretical foundations of generative models and stochastic interpolants for dynamical systems, our contribution is primarily empirical. Many theoretical frameworks for generative modeling now exist, but it remains unclear which are best suited to physical systems or what each one offers. Adding to this, physical emulation exhibits remarkable fragmentation: different systems exhibit different dynamics, variables, and spatial or temporal scales, and even within the same system, practitioners may prioritize very different objectives. In fluid dynamics, for instance, the Reynolds number can fundamentally reshape the relevant length and time scales. Even when the governing system is fixed, such as the Earth's climate, a weather forecaster may seek to minimize medium-range RMSE, whereas a climate researcher may focus on reducing long-term climatological biases. This diversity suggests that generative models may need to tailor their training or sampling to the demands of each problem, which in turn presents an opportunity to develop empirically grounded heuristics and best practices for these settings. We therefore conduct a comprehensive benchmark of stochastic interpolants for physical emulation, with the goal of understanding when they work and how these benefits occur. Our central finding is that principled choices for the transport map and sampling algorithm yield meaningful performance gains over standard generative frameworks from image or video modeling. Across fluid and climate benchmarks, our experiments demonstrate that stochastic interpolants can outperform standard choices and offer helpful inference-time tradeoffs. In doing so, we establish stochastic interpolants as a competitive baseline for physical emulation and offer broader insight into the behavior of generative surrogates in this setting.

## 3 Methods

### 3.1 Stochastic Interpolants

**Overview** We consider a class of generative models that admit arbitrary source and target distributions. These models have various names and instantiations, however, for simplicity, we consider stochastic interpolants since many of these models can be unified under this framework (Albergo et al., 2023; Zhang et al., 2025). Given samples $x_0$ from a source distribution $\rho_0$ and samples $x_1$ from a target distribution $\rho_1$, a

stochastic interpolant $x_t$ is defined as:

$$x_t = \alpha(t)x_0 + \beta(t)x_1 + \gamma(t)z, \qquad t \in [0,1] \tag{1}$$

The mixing coefficients $\alpha(t), \beta(t), \sigma(t)$ must satisfy the boundary conditions $\alpha(0) = \beta(1) = 1$, $\alpha(1) = \beta(0) = 0$, and $\gamma(0) = \gamma(1) = 0$ to recover the endpoints of the trajectory $x_0, x_1$. Furthermore $z$ is generally sampled from a standard Gaussian $z \sim N(0, I)$, and serves to mix the intermediate distributions $p_t(x)$. There are a few useful observations to make. Firstly, the stochastic interpolant produces samples $x_0 \sim \rho_0$ at $t = 0$ and $x_1 \sim \rho_1$ at $t = 1$ by construction. Furthermore, the interpolant $x_t$ maps between the densities $\rho_0, \rho_1$ exactly and in finite time. Lastly, this stochastic interpolant can be realized by an SDE or ODE, which can produce samples $x_t$ from intermediate distributions $p_t(x)$ at any time $t \in [0,1]$.

**Implementation** The overall framework of stochastic interpolants is designed to be broad, with the goal of unifying different generative modeling frameworks. However, for its practical deployment in physical systems, we must make specific choices of the interpolant $x_t$, as well as specify training and sampling algorithms. The key inductive bias that is leveraged is to consider a source distribution of a point mass centered at the current state $x_0 \sim \delta(u_t)$ and a target distribution of future states conditioned on the current state $x_1 \sim p(u_{t+1}|u_t)$, both supported on $\mathbb{R}^d$ for $d$-dimensional states. The hypothesis is that this reduces the transport cost compared to a Gaussian prior, which can simplify both the learning the transport map and sampling from a trained model. This choice of a Dirac source distribution induces the interpolant (Chen et al., 2024):

$$x_t = \alpha(t)x_0 + \beta(t)x_1 + \sigma(t)W_t, \qquad t \in [0,1] \tag{2}$$

where $W_t$ is a Wiener process and is realized with $W_t \stackrel{d}{=} \sqrt{t}z$ by Brownian increments generated from $z \sim N(0, \mathbf{I})$, and $\sigma(t)$ is a noise coefficient that vanishes at $t = 1$. We further specify linear choices of coefficients $\alpha(t) = 1 - t$, $\beta_t = t$, and $\sigma(t) = 1 - t$ which gives:

$$x_t = (1-t)x_0 + tx_1 + (1-t)W_t, \qquad t \in [0,1] \tag{3}$$

To realize the interpolant $x_t$, we parameterize a drift $b_\theta$, which is learned by minimizing the loss:

$$\mathcal{L}_b[b_\theta] = \frac{1}{N}\sum_{n=1}^{N} ||b_\theta(t, x_t^n, x_0^n) - R(t, x_1^n, x_0^n))||^2 \tag{4}$$

where

$$R(t, x_1^n, x_0^n) = \dot{\alpha}(t)x_0^n + \dot{\beta}(t)x_1^n + \dot{\sigma}(t)W_t \tag{5}$$

The timestep $t \in [0,1]$ is uniformly sampled and $x_t^n$ is a given data sample $n$ at time $t$ along the stochastic interpolant. Additionally, to map to a conditional distribution, the condition $x_0^n$ is provided as input to the estimated drift. Under linear choices of mixing coefficients, the training target reduces to:

$$\mathcal{L}_b[b_\theta] = \frac{1}{N}\sum_{n=1}^{N} |b_\theta(t, x_t^n, x_0^n) - (x_1^n - x_0^n - W_t)|^2 \tag{6}$$

which can be computed by sampling $x_0^n = u_t$, $x_1^n = u_{t+1}$ from a dataset (where $t$ denotes physical time) and $W_t \stackrel{d}{=} \sqrt{t}z$ from $z \sim N(0, I)$. Intuitively, $b_\theta$ aims to estimate the time derivative of the stochastic interpolant. This is useful during inference, where samples $x_0$ are integrated using conditional drift estimates $b_\theta$ according to the probability flow SDE, starting from $t = 0$ to $t = 1$:

$$dx_t^{SDE} = b_\theta(t, x_t, x_0)dt + \sigma(t)dW_t \tag{7}$$

At $t = 1$, this SDE produces samples from the conditional distribution $p(x_1|x_0)$. To realize this SDE, we use a numerical integrator, specifically the Euler-Maruyama method:

$$x_{t+1} = x_t + \Delta t * b_\theta(t, x_t, x_0) + (1-t)\sqrt{\Delta t} * z \tag{8}$$

where a linear choice of $\sigma(t)$ was substituted and timesteps $\Delta t$ are uniformly discretized on $[0,1]$. We can observe that at $t = 0$, the diffusion term $\sigma(t)dW_t$ is maximized to immediately spread the probability mass of the Dirac $x_0 \sim \delta(u_t)$. As $t \to 1$, the diffusion term vanishes to recover clean samples from the conditional distribution $x_1 \sim p(u_{t+1}|u_t)$.

**Variable Diffusion** Interestingly, this framework admits a family of SDEs with variable noise coefficients $\hat{\sigma}(t)$ by constructing a modified drift $\hat{b}_\theta(t, x_t, x_0)$. We refer to Chen et al. (2024) for details, but under regularity conditions, SDEs with variable diffusion can be sampled without retraining $b_\theta(t, x_t, x_0)$. For future states with more or less uncertainty, this can modify the stochasticity of the sampler at test time.

In our case, we allow the noise coefficient to be scaled by a constant $\hat{\sigma}(t) = \sigma(1-t)$ during sampling. While this requires constructing a correction to the drift $\hat{b}_\theta(t, x_t, x_0)$, we find that empirically, using the same drift without modification yields similar results and simplifies the sampling algorithm. One interesting case is when $\sigma(t) \to 0$ for all $t$. In this case, the SDE reduces to an ODE which generally allows for faster sampling by truncating the reverse process. However, this sampler degenerates the interpolant to predict the deterministic mean of the conditional distribution. For most generative models (e.g., in image or text generation) this is detrimental due to the multimodal nature of the target distribution, which would generate a blurry, average sample. However, certain physics applications have strongly unimodal target distributions, where future states are almost deterministic when conditioned on the present. This can happen if the temporal dynamics are predictable or the lead time between states is small. From this point of view, we can see that adjusting $\sigma(t)$ in Equation 7 forms a tradeoff: at small $\sigma(t)$ we produce a deterministic neural surrogate, and at large $\sigma(t)$ forward diffusion dominates the SDE. We will see that tailoring this to specific systems or desired criteria can result in speed or performance gains.

### 3.2 Generative Models

**Overview** In addition to the constructed stochastic interpolant, we also train a variety of commonly used diffusion and flow matching models. An overview of the relevant training and sampling procedures is given in Table 1. Denoising Diffusion Probabilistic Models (DDPMs) (Ho et al., 2020) define a fixed Markovian forward process that gradually corrupts a data sample $x_0$ into Gaussian noise over $T$ steps and trains a network to approximate the reverse transitions, typically by predicting the noise added at each step. While this yields high-quality outputs, it requires a large number of function evaluations (NFEs), which is a limitation for autoregressive physics surrogates. Denoising Diffusion Implicit Models (DDIMs) (Song et al., 2022) retain the DDPM training objective but uses a deterministic reverse process that can be sampled in fewer steps, which trades sample quality for inference speed. Elucidated Diffusion Models (EDMs) (Karras et al., 2022) reframe diffusion within a continuous-time framework that disentangles the the noise schedule, network preconditioning, loss weighting, and sampling; optimizing these results in achieving high fidelity at comparatively few NFEs. Truncated Sampling Models (TSMs) (Shehata et al., 2025) target the inference cost of DDPMs directly by truncating the low-noise regime of the reverse process. Sampling is carried out only down to an intermediate step after which the clean state is estimated via Tweedie's formula. For unimodal, conditional distributions in autoregressive prediction tasks, this enables single- and few-step sampling while preserving or even improving accuracy relative to full DDPM sampling. Lastly, flow matching (FM) (Liu et al., 2022; Lipman et al., 2023) seeks to learn a continuous-time velocity field that transports samples from a standard Gaussian to the target distribution. Rather than relying on stochastic denoising in diffusion, flow matching trains a network to directly regress a vector field and generates samples by integrating the resulting ODE from noise to data. When the probability path is constructed to linearly interpolate between the two distributions, the induced trajectories are nearly straight which enables accurate generation with comparatively few NFEs.

One remark is that the general framework of stochastic interpolants can encompass these standard generative models, however, for clarity we refer to the specific interpolant in Equation 2 as the SI model going forward.

**Implementation** DDPM, DDIM, and TSM models use some variant of the stochastic process:

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} z, \quad z \sim \mathcal{N}(0, \mathbf{I}) \tag{9}$$

which noises data $x_0$ over some time $t \in [1, T]$, where the noise schedule $\beta_t = 1 - \alpha_t$ is defined linearly with $\beta_{start} = 0.0001$ and $\beta_{end} = 0.02$. We found that scaling the noise schedule by $S = 400$ in $S * \beta_{start}$, $S * \beta_{end}$ to match the magnitude of the data was beneficial, following Shehata et al. (2025). In EDM, this stochastic process is reparameterized as:

$$x_t = x_0 + \sigma_t z \tag{10}$$

Table 1: Training and sampling for conditional diffusion, flow matching, and stochastic interpolant frameworks. Note that $x_t$ is defined differently for diffusion, flow matching, and stochastic interpolants. Samplers can also vary for EDM, TSM, FM, and SI.

| | **Training Objective** | **Sampling Procedure** |
|---|---|---|
| **DDPM** | $\mathcal{L} = \|\epsilon - \epsilon_\theta(x_t, c, t)\|^2$ | $x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\,\epsilon_\theta(x_t, c, t)\right) + \sigma_t z$ |
| **DDIM** | $\mathcal{L} = \|\epsilon - \epsilon_\theta(x_t, c, t)\|^2$ | $x_{t-1} = \sqrt{\alpha_{t-1}}\left(\frac{x_t - \sqrt{1-\alpha_t}\epsilon_\theta(x_t, c, t)}{\sqrt{\alpha_t}}\right) + \sqrt{1-\alpha_{t-1}-\sigma_t^2}\cdot\epsilon_\theta(x_t, c, t) + \sigma_t z$ |
| **EDM** | $\mathcal{L} = \|D_\theta(x+z, c, \sigma) - x\|^2$ | $x_{t+\Delta t} = x_t + \frac{\dot{\sigma}(t)}{\sigma(t)}\left(x - D_\theta(x, c, \sigma(t))\right)\Delta t$ (Euler) |
| **TSM** | $\mathcal{L} = \|\epsilon - \epsilon_\theta(x_t, c, t)\|^2$ | $x_0 = (x_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta(x_t, c, t))/\sqrt{\bar{\alpha}_t}$ (1-step) |
| **FM** | $\mathcal{L} = \|(x - z) - v_\theta(x_t, c, t)\|^2$ | $x_{t+\Delta t} = x_t + v_\theta(x_t, c, t)\Delta t,\ x_0 \sim \mathcal{N}(0, I)$ (Euler) |
| **SI** | $\mathcal{L} = \|((x_1 - x_0 - W_t) - b_\theta(x_t, c, t))\|^2$ | $x_{t+\Delta t} = x_t + b_\theta(x_t, c, t)\Delta t + \sigma(t)\sqrt{\Delta t}z,$ (Euler-Maruyama) |

for a time-varying noise variance $\sigma_t$. The design choices for EDM are already optimized, therefore, we did not find any substantial benefits by adjusting them. We adopt the commonly used linear schedule for flow matching, which simplifies many hyperparameter choices and results in the interpolant:

$$x_t = tx_0 + (1-t)z \tag{11}$$

where the source distribution is taken to be a Gaussian. This can be seen as a counterpart to the SI model, where the source distribution is a Gaussian rather than a Dirac distribution centered at the current state.

During sampling, each generative model proposes a different method, which is generally fixed. However, since the training and sampling for EDM is decoupled, this allows it to be sampled deterministically (Det.) or stochastically (Sto.) using their proposed stochastic sampler. We choose to use the Euler method, as well as the stochastic churn parameters from Shehata et al. (2025). The stochastic interpolant can also be flexibly sampled by adjusting $\hat{\sigma}(t) = \sigma(1-t)$ when integrating the SDE (Equation 7) during inference. We consider two limiting cases $\sigma = 0$, which reduces the stochastic interpolant to predict the deterministic conditional mean and $\sigma = 1$ to match the noise coefficient during training.

### 3.3 Models

Within each generative model, the score, velocity, or denoiser is parameterized by a neural network. To ensure a consistent comparison, we standardize our model architectures and training procedures.

**Autoencoders** We choose to use latent diffusion (Rombach et al., 2022) to reduce the training and inference cost; however, in PDE problems it can also stabilize rollouts and improve performance (Rozet et al., 2025; Li et al., 2025a). For PDE tasks, we use a Deep Compression Autoencoder (DCAE) (Chen et al., 2025a), which is based on 2D convolution and residual pooling/unpooling layers. Following Rozet et al. (2025), we apply a saturation function to latent vectors $z = \frac{z}{\sqrt{1+z^2/b^2}}$, where $b = 5$, to avoid arbitrary variance. For climate tasks, we find that KL regularization is more beneficial, potentially due to the need for long-term consistency in climate emulators. For PDE problems we use a compression ratio of 64× and for climate problems we use a compression ratio of 32×.

**Architectures** We use a diffusion transformer (DiT) (Peebles & Xie, 2023) as the backbone for all latent-space generative models. Within a given task, the architecture and model size is kept constant across all generative models. To condition on the diffusion or interpolant timestep, adaptive layer normalization is used. Additionally, the current state of the system is concatenated to the noisy, intermediate states to

enable conditional generation. Moreover, as a result of the compressed latent space, the backbone does not need patchification or sparse attention. Although the work focuses on generative models, we also include a deterministic emulator as a baseline to understand the difficulty of tasks. For PDE tasks, we consider FNO (Li et al., 2021) and for climate tasks, we consider SFNO (Bonev et al., 2023).

### 3.4 Datasets

**Kolmogorov Flow** Kolmogorov Flow (KF) is described by the 2D Navier-Stokes equations driven by unidirectional periodic forcing. In vorticity form, Kolmogorov Flow can be described by the PDE (Koehler et al., 2024):

$$\frac{\partial \omega}{\partial t} = -b\left(\begin{bmatrix}1\\-1\end{bmatrix} \odot \nabla(\Delta^{-1}\omega)\right) \cdot \nabla\omega + \nu\nabla \cdot \nabla\omega + \lambda\omega - k\cos(k\frac{2\pi}{L}y) \tag{12}$$

The leftmost term represents vorticity convection, controlled by the coefficient $b$. Diffusion is controlled by the viscosity $\nu$ and a drag term $\lambda\omega$ is introduced. Lastly, a sinusoidal forcing term is introduced, controlled by the magnitude $k$.

Although common, this is a fairly challenging task and most PDE surrogates are unstable when rolled out to the training horizon (Lippe et al., 2023; Zhou & Barati Farimani, 2025). Data is generated from APEBench (Koehler et al., 2024) at a resolution of $160 \times 160$ on a domain $(x, y) = [-10, 10]^2$. The simulation is saved at a resolution of $\Delta t = 0.2s$ for 100 timesteps, resulting in a rollout from $t = 0$ to $t = 20$ seconds. Initial conditions are uniformly sampled from a random truncated Fourier series with 5 modes, and the viscosity $\nu$ is set to $10^{-2}$ to simulate a Reynolds number of $\sim 10^2$.

**Rayleigh-Bénard Convection** Rayleigh-Bénard Convection (RBC) is a phenomenon that describes the mixing of horizontal layers of fluid driven by a temperature gradient. The equations are governed by a buoyancy and Navier-Stokes equation (Ohana et al., 2025):

$$\frac{\partial b}{\partial t} - \kappa\Delta b = -u \cdot \nabla b \tag{13}$$

$$\frac{\partial u}{\partial t} - \nu u + \nabla p - b\mathbf{e}_z = -u \cdot \nabla u \tag{14}$$

The thermal diffusivity $\kappa$ and viscosity $\nu$ are determined by the Rayleigh and Prandtl numbers:

$$\kappa = (\text{Rayleigh} \times \text{Prandtl})^{-\frac{1}{2}}, \quad \nu = \left(\frac{\text{Rayleigh}}{\text{Prandtl}}\right)^{-\frac{1}{2}} \tag{15}$$

Rayleigh and Prandtl numbers are varied throughout the training and validation data, where Rayleigh $\in \{1e6, 1e7, 1e8, 1e9, 1e10\}$ and Prandtl $\in \{0.1, 0.2, 0.5, 1, 2, 5, 10\}$. Furthermore, initial conditions for the buoyancy are generated by $b(t = 0) = (Ly-y)\times\delta b_0 + y(Ly-y)\times\epsilon$, where $\delta b_0$ is sampled from $\{0.2, 0.4, 0.6, 0.8, 1.0\}$ and $\epsilon$ is sampled from a Gaussian scaled to $10^{-3}$. This results in a linear buoyancy gradient in the vertical direction with a small perturbation. All other fields are initialized to zero.

Current PDE surrogates usually struggle since the system is highly chaotic and features a transition between laminar and turbulent regimes. Data is obtained from the Well (Ohana et al., 2025; Burns et al., 2020), which includes 2D simulations on a $512 \times 128$ grid with buoyancy, pressure, and velocity. In addition, we use 100 timesteps with an interval of $\Delta t = 0.5$.

**PlaSim** Global 3D atmospheric data are generated from an intermediate-complexity climate model (PlaSim) to evaluate emulators for weather forecasting and climate prediction (Lunkeit et al., 2021; Ragone et al., 2018; Lancelin et al., 2025). PlaSim solves the Navier-Stokes equation on a rotating sphere along with parameterizations for various atmospheric (e.g., moist convection, radiation) and land processes, while the sea surface temperature and sea ice cover are prescribed and vary with a yearly period. A full description of the climate variables and the pressure levels used in the climate simulation dataset is given in Table 2.

Prognostic atmospheric variables (temperature, humidity, zonal and meridional wind) are saved at a resolution of $(128 \times 64 \times 10)$ (latitude, longitude, model level) on an Gaussian horizontal grid. These are then

Table 2: Climate variables grouped into surface, atmospheric, and forcing variables. Additionally, 13 pressure levels are used.

| **Surface Variables (8)** | **Atmospheric Variables (5)** | **Forcing Variables (6)** | **Pressure Levels (13)** |
|---|---|---|---|
| lwe of water evaporation (`evap`) | specific humidity (`hus`) | land sea mask (`lsm`) | 50, 100, 150, 200, |
| surface runoff (`mrro`) | air temperature (`ta`) | surface geopotential (`sg`) | 250, 300, 400, 500, |
| lwe of soil moisture content (`mrso`) | eastward wind (`ua`) | surface roughness length (`z0`) | 600, 700, 850, 925, |
| log surface pressure (`pl`) | northward wind (`va`) | TOA Incident Radiation (`rsdt`) | 1000 |
| 12h acc. precipitation (`pr_12h`) | geopotential (`zg`) | sea ice cover (`sic`) | |
| 6h acc. precipitation (`pr_6h`) | | sea surface temperature (`sst`) | |
| air temperature 2m (`t2m`) | | | |
| surface temperature (`ts`) | | | |

vertically interpolated onto 13 equipressure levels, and the geopotential height is computed using the hydrostatic equation. 8 surface variables, including 2-meter temperature and accumulated precipitation, and 6 forcing variables are also saved. Models are trained on 100 years of data at 6-hour intervals and validated on a year of held-out data, except when evaluating climatological biases, which uses 10 years of data. This results in ∼144,000 training samples and ∼1,440 validation samples.

### 3.5 Metrics

**Deterministic Metrics** We follow standard metrics to evaluate the deterministic and statistical performance of surrogate models. For PDE tasks, we use Variance Scaled RMSE (VRMSE), and for weather forecasting, we use latitude-weighted RMSE (lRMSE). Given a spatio-temporal data sample $\mathbf{u} \in \mathbb{R}^{n_t \times n_x \times n_y}$ and a model rollout $\mathbf{u}_\theta \in \mathbb{R}^{n_t \times n_x \times n_y}$ the Variance-Scaled RMSE (VRMSE) is given by (Ohana et al., 2025):

$$VRMSE(\mathbf{u}, \mathbf{u}_\theta) = \frac{1}{n_t} \sum_{t=1}^{n_t} \frac{||\mathbf{u}_\theta(t) - \mathbf{u}(t)||_2}{||\mathbf{u}(t) - \bar{\mathbf{u}}(t)||_2 + \epsilon}, \quad ||\mathbf{u}||_2 = \sqrt{\frac{1}{n_x n_y} \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} |\mathbf{u}(i,j)|^2} \tag{16}$$

The term $\epsilon = 10^{-6}$ is added for numerical stability. This metric scales the error by the variance of the input sample, then averages over the prediction horizon $n_t$. Predicting the mean field will result in $VRMSE(u, \bar{u}) \approx 1$, which is a useful interpretation.

Latitude-weighted RMSE (lRMSE) is calculated at each lead time, for each variable and pressure level separately. For a forecasted variable $f_{tl} \in \mathbb{R}^{n_{lat} \times n_{lon}}$ and ground-truth $o_{tl} \in \mathbb{R}^{n_{lat} \times n_{lon}}$ at level $l$ and time $t$, lRMSE is given by Rasp et al. (2024):

$$lRMSE = \sqrt{\frac{1}{n_{lat} n_{lon}} \sum_{i=1}^{n_{lat}} \sum_{j=1}^{n_{lon}} w(i)(f_{tl}(i,j) - o_{tl}(i,j))^2}, \quad w(i) = \frac{\sin\theta_i^u - \sin\theta_i^l}{\frac{1}{I}\sum_{i=1}^{I}(\sin\theta_i^u - \sin\theta_i^l)} \tag{17}$$

For latitude weights $w(i)$, $i$ denotes the index of the discretized latitude, and $\theta_i^u$ and $\theta_i^l$ denote the upper and lower bounds of the cell at latitude $i$. Latitude weights are used to account for distortion at the poles in an equiangular grid, which would otherwise over-emphasize predicted values near the poles. Lastly, lRMSE is calculated for a given lead time (i.e., 10-day lRMSE) by initializing a forecast at each day in the validation set and averaging across all forecasts for that lead time.

Annual climatological biases are also calculated to evaluate the consistency of the climate emulator; these are calculated as the lRMSE between the time average of a 10-year ground truth from PlaSim and a 10-year emulation for each atmospheric and surface variable. In particular, the 10-year bias is calculated for each variable and pressure level by:

$$\text{Bias}_{\text{10-year}} = lRMSE(f_{avg}, o_{avg}), \quad f_{avg} = \frac{1}{n_t} \sum_{t=1}^{n_t} f_l(t) \tag{18}$$

where $n_t \approx 10 * 365 * 4$, since a forecast is made every 6 hours for each day in 10 years. Note that $f_l(t) \in \mathbb{R}^{n_{lat} \times n_{lon}}$, $f_{avg} \in \mathbb{R}^{n_{lat} \times n_{lon}}$, and lRMSE reduces over latitude and longitude.

**Statistical Metrics** For PDE tasks, the statistical consistency of fluid flows is evaluated by comparing the power spectrum of predicted and true rollouts with spectral RMSE (SRMSE). Exact predictions of turbulent flows over time is usually not possible for PDE surrogates, therefore spectral metrics can quantify the distribution and scale of predicted features rather than relying on point-wise accuracy.

Given a spatio-temporal data sample $\mathbf{u} \in \mathbb{R}^{n_t \times n_x \times n_y}$ and a model rollout $\mathbf{u}_\theta \in \mathbb{R}^{n_t \times n_x \times n_y}$ the DFT is used at each timestep to generate the power spectrum $p(t)$ and frequencies $k$. The power spectrum is partitioned based on its frequency into three evenly log-spaced bins (low, mid, high). Following Ohana et al. (2025), the SRMSE between the true and predicted spectra $(p, p_\theta)$ for each bin is then calculated as:

$$SRMSE(\mathbf{u}, \mathbf{u}_\theta) = \frac{1}{n_t} \sum_{t=1}^{n_t} \sqrt{1 - \left( \frac{p(t)}{p_\theta(t)} \right)^2} \tag{19}$$

For inputs with multiple channels, the SRMSE is calculated for each channel separately and then averaged.

**Probabilistic Metrics** For weather forecasting, the performance of probabilistic predictions is measured using the continuous ranked probability score (CRPS) and the spread-skill ratio (SSR). CRPS is minimized when samples from the generative model are drawn from the same distribution as the data (Zamo & Naveau, 2018). Similar to lRMSE, CRPS is calculated at each lead time, and for each variable and pressure level separately, however with the addition of $M$ different ensemble members. CRPS makes use of the latitude-weighted mean absolute error (lMAE):

$$lMAE = \frac{1}{n_{lat} n_{lon}} \sum_{i=1}^{n_{lat}} \sum_{j=1}^{n_{lon}} w(i) |f_{tl}(i,j) - o_{tl}(i,j)| \tag{20}$$

CRPS for a given lead time $t$, variable, and level $l$ is given by:

$$CRPS_{tl} = \frac{1}{M} \sum_{m=1}^{M} lMAE(f_{tl}^{(m)}, o_{tl}) - \frac{1}{2M(M-1)} \sum_{m=1}^{M} \sum_{n=1}^{N} lMAE(f_{tl}^{(m)}, f_{tl}^{(n)}) \tag{21}$$

The first term penalizes deviations of the individual ensemble members from the ground truth, and the second term encourages spread between ensemble members. To calculate the CRPS for a given lead time, ensemble forecasts are initialized every three days for the validation year, and CRPS values at each lead time are averaged for across each forecast. An ensemble size of 32 is used.

SSR values of 1 are considered optimal as the uncertainty of the forecast matches its error (Fortin et al., 2014). The spread is calculated for a given time, level, and variable as the square root of the ensemble variance:

$$\text{Spread}_{tl} = \sqrt{\sum_{i=1}^{n_{lat}} \sum_{j=1}^{n_{lon}} w(i) var_m(f_{tl}^{(m)}(i,j))} \tag{22}$$

where $var_m$ calculates the variance over $M$ ensemble members. The ensemble skill is given by the RMSE of the ensemble mean:

$$\bar{f}_{tl} = \frac{1}{M} \sum_{m=1}^{M} f_{tl}^{(m)}, \quad \text{Skill}_{tl} = lRMSE(\bar{f}_{tl}, o_{tl}) \tag{23}$$

where $f_{tl}^{(m)} \in \mathbb{R}^{n_{lat} \times n_{lon}}$ is an individual forecast for a variable at lead time $t$ and level $l$. The SSR is given by:

$$SSR_{tl} = \frac{\text{Spread}_{tl}}{\text{Skill}_{tl}} \tag{24}$$

Calculating SSR for a given lead time is done similarly to CRPS, where SSR values for a lead time are averaged across all ensemble forecasts made for the validation year. SSR values less than 1 indicate an under-dispersive forecast, where the ensemble fails to capture the full range of possible outcomes, while values over 1 indicate an over-dispersive forecast, where the ensemble is overly uncertain.

Table 3: Pointwise (VRMSE) and Spectral (SRMSE) errors of models on Kolmogorov Flow, sampled with different NFEs. The average error across three seeds is reported. L, M, and H denote the low, mid, and high-band SRMSE metrics.

| Model: | FNO | DDPM | DDIM | TSM | EDM$_{(\text{Det.})}$ | EDM$_{(\text{Sto.})}$ | FM | SI$_{(\sigma\,=\,0)}$ | SI$_{(\sigma\,=\,1)}$ | AE |
|---|---|---|---|---|---|---|---|---|---|---|
| NFEs: | 1 | 100 | 10 | 1 | 10 | 20 | 2 | 2 | 20 | 1 |
| VRMSE | 0.621 | 0.684 | 0.735 | 0.835 | 0.616 | 0.620 | 0.593 | 0.552 | 0.569 | 0.011 |
| SRMSE$_L$ | 0.064 | 0.078 | 0.335 | 0.124 | 0.063 | 0.068 | 0.073 | 0.056 | 0.064 | 0.011 |
| SRMSE$_M$ | 0.042 | 0.053 | 0.180 | 0.068 | 0.043 | 0.045 | 0.044 | 0.039 | 0.045 | 0.006 |
| SRMSE$_H$ | 0.380 | 0.795 | 0.801 | 0.850 | 0.792 | 0.793 | 0.792 | 0.791 | 0.795 | 0.791 |

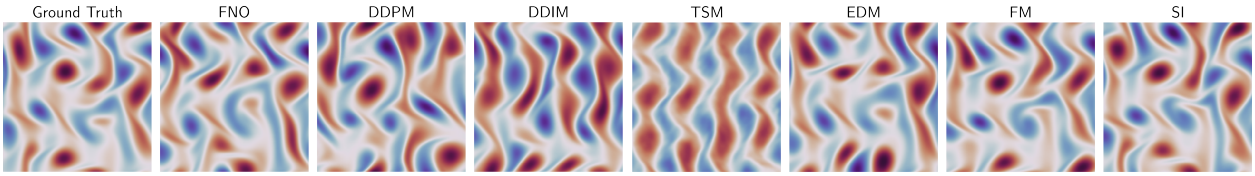


Figure 1: Model predictions of Kolmogorov Flow at $t = 50$. Deterministic versions of EDM and SI are plotted in both figures.

## 4 Results

In this section, we present results after training each generative model on the three physical systems. Within each task, the model architectures, the training compute and the size of the dataset are kept constant. Each system also exhibits different dynamics and metrics of interest, resulting in different insights for each set of experiments.

### 4.1 Kolmogorov Flow

We report the pointwise and spectral performance of the models in Table 3, where the lowest errors are shaded and the second-lowest errors are lightly shaded. All metrics are reported as the average of three seeds. The number of function evaluations (NFEs) needed for a single prediction is shown, and the autoencoder (AE) performance is given as a reference for the upper bound on performance. Models are grouped into related training frameworks and EDM and SI models are evaluated with a deterministic or stochastic sampler. The number of sampling steps was determined empirically to reduce error until diminishing returns were observed.

Interestingly, a primary takeaway from these experiments is that generative models are not always necessary. The best-performing model is the deterministic interpolant (SI, $\sigma = 0$), and even the FNO model outperforms or is on-par with many generative models. This suggests that the use of generative models as a physical surrogate depends on the system of interest. The Kolmogorov Flow dataset has a low Reynolds number ($\sim 10^2$) and does not exhibit multiscale features. Furthermore, the time and spatial discretization are sufficiently fine and the system is laminar, therefore the transition between states is likely deterministic. In this case, and in other PDE tasks like it, a generative model and its associated training and sampling complexity may not be the best fit compared to a deterministic neural surrogate.

Despite this, it is still useful to compare the performance of different generative frameworks. Contemporary models such as EDM or flow matching are generally seen as better default choices than DDPM, which is corroborated here. We can also observe that deterministic sampling tends to outperform stochastic samplers; which suggests that the additional NFEs of stochastic sampling is not necessary in these types of problems. However, fully truncating the sampling can be detrimental, and iterative sampling is still needed. Lastly, the choice of source distribution seems to also have an effect on performance. By shifting from flow matching ($x_0 \sim N(0, I)$) to a stochastic interpolant ($x_0 \sim \delta(u_t)$) we observe a modest gain in performance.

Table 4: Pointwise (VRMSE) and Spectral (SRMSE) errors of models on Rayleigh-Bénard Convection, sampled with different NFEs.

| Model: | FNO | DDPM | DDIM | TSM | EDM(Det.) | EDM(Sto.) | FM | SI$_{(\sigma=0)}$ | SI$_{(\sigma=1)}$ | AE |
|---|---|---|---|---|---|---|---|---|---|---|
| NFEs: | 1 | 100 | 10 | 10 | 10 | 50 | 5 | 5 | 50 | 1 |
| VRMSE | >10 | 0.765 | 0.675 | 8.961 | 0.681 | 0.699 | 0.733 | 0.665 | 0.726 | 0.027 |
| SRMSE$_L$ | 0.357 | 0.405 | 0.612 | 1.113 | 0.363 | 0.329 | 0.323 | 0.346 | 0.296 | 0.086 |
| SRMSE$_M$ | 1.739 | 0.242 | 0.644 | 0.883 | 0.321 | 0.261 | 0.243 | 0.601 | 0.184 | 0.061 |
| SRMSE$_H$ | 2.406 | 1.822 | 3.276 | 1.133 | 2.594 | 2.194 | 2.478 | 6.078 | 2.096 | 1.528 |

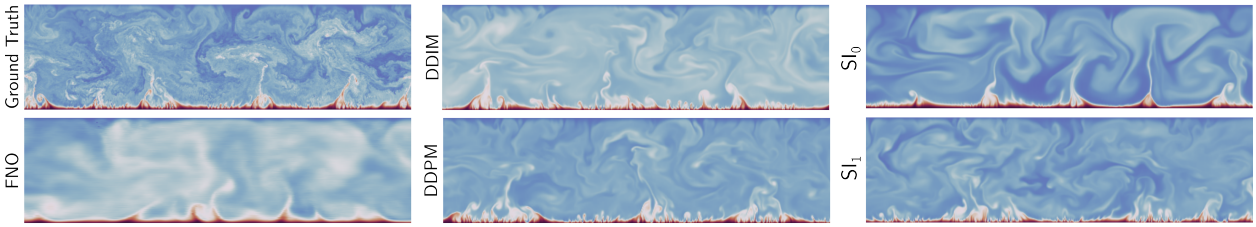


Figure 2: Model predictions of Rayleigh-Bénard Convection at $t = 48$. For conciseness, only a few models are shown. SI models are plotted with $\sigma = 0$ or $\sigma = 1$.

Given that the Kolmogorov Flow dataset is laminar and that the time discretization is sufficiently small, the conditional distribution $p(u_{t+1}|u_t)$ is likely unimodal and is better approximated by ODE samplers that regress towards the conditional mean. Indeed, flow matching has been shown to be successful in many previous works (Li et al., 2025b; Armegioiu et al., 2025), and a smoothly decaying spectrum also avoids the need for noise injection during sampling. Lastly, for all generative models, pointwise error is largely driven by autoregressive drift rather than the reconstruction error of the autoencoder. In the highest frequency band, the spectral error of the autoencoder also thresholds the spectral error of latent generative models.

We can look at model predictions in Figure 1 to examine some potential failure modes. In general, most models decorrelate point-wise after $\sim t = 50$ steps, although the distribution and scale of features remains largely consistent. The SI, FM, EDM, and FNO models all seem to perform well. For TSM and DDIM the predictions interestingly start to regress towards the time-mean of the flow, which resembles the sinusoidal forcing in the $y$-direction. This is reflected in the higher spectral error in the low band ($SRMSE_L$) for both models and suggests that directly truncating the reverse process can be detrimental.

**Rayleigh-Bénard Convection** Rayleigh-Bénard Convection offers a more complex system featuring a transition between laminar and turbulent states. Due to the large size of the dataset, only a single set of experiments was run and errors are reported in Table 4. Due to the transition to turbulence and resulting chaos, the performance of a deterministic surrogate such as FNO can decorrelate quickly, matching previous benchmarks from Ohana et al. (2025). This transition to turbulence and emergence of multiscale features also justifies the use of generative models, which show a distinct performance benefit.

For these generative models, Rayleigh-Bénard convection reveals an interesting trade-off where lower pointwise error usually comes at the cost of higher spectral error. After convective mixing, it is intractable for models to exactly predict turbulent states, especially over 100 autoregressive predictions. Therefore, minimizing pointwise error tends to push models toward overly smoothed predictions. This behavior can be mitigated by increasing the sampling length or introducing more stochasticity, since taking smaller, random steps adds perturbations to help recover the true spectrum. However, greater stochasticity causes predictions to deviate further from the exact state, even as the spectral characteristics remain consistent. This can be qualitatively seen in Figure 2, and suggests that designing generative models for physical emulation requires understanding the metrics of interest and task at hand, rather than using a generic framework.

Table 5: lRMSE of benchmarked models on weather forecasting. Errors are reported at different lead times ($\{1, 3, 5, 10\}$ days) and climate variables. DDPM uses 100 sampling steps, while EDM and DDIM use 10 steps. FM and SI use 5 sampling steps. Additionally, deterministic versions of EDM and SI are reported.

| Var: | **z500** $[m]$ | | | | **t2m** $[K]$ | | | | **t850** $[K]$ | | | | **u250** $[m/s]$ | | | | **pr_6h** $[mm]$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Days: | 1 | 3 | 5 | 10 | 1 | 3 | 5 | 10 | 1 | 3 | 5 | 10 | 1 | 3 | 5 | 10 | 1 | 3 | 5 | 10 |
| SFNO | 11 | 33.5 | 57.1 | 119 | 1.27 | 2.33 | 3.28 | 7.55 | 1.17 | 2.21 | 3.24 | 7.94 | 2.37 | 5.26 | 7.99 | 16.0 | 0.87 | 1.29 | 1.47 | 2.27 |
| DDPM | 8.5 | 18.2 | 30.9 | 61.9 | 0.93 | 1.38 | 1.88 | 3.03 | 0.95 | 1.40 | 1.98 | 3.27 | 1.85 | 3.32 | 4.88 | 8.44 | 0.85 | 1.14 | 1.33 | 1.57 |
| DDIM | 7.3 | 15.6 | 26.5 | 54.9 | 0.84 | 1.24 | 1.67 | 2.76 | 0.88 | 1.26 | 1.75 | 2.99 | 1.65 | 2.89 | 4.20 | 7.44 | 0.78 | 1.01 | 1.18 | 1.40 |
| EDM | 7.2 | 15.4 | 26.6 | 57.6 | 0.83 | 1.21 | 1.65 | 2.87 | 0.87 | 1.24 | 1.76 | 3.13 | 1.64 | 2.88 | 4.21 | 7.71 | 0.73 | 1.00 | 1.21 | 1.50 |
| FM | 7.1 | 15.2 | 26.5 | 57.7 | 0.79 | 1.15 | 1.60 | 2.81 | 0.84 | 1.19 | 1.72 | 3.07 | 1.60 | 2.83 | 4.18 | 7.82 | 0.72 | 0.98 | 1.19 | 1.49 |
| SI | 6.2 | 12.9 | 22.9 | 52.2 | 0.73 | 1.05 | 1.44 | 2.66 | 0.80 | 1.10 | 1.55 | 2.89 | 1.48 | 2.53 | 3.71 | 7.01 | 0.61 | 0.87 | 1.10 | 1.43 |

Deterministically sampled stochastic interpolants achieve the lowest VRMSE, yet exhibit large spectral errors. This is remedied by sampling the model stochastically, where the interpolant is able to achieve low spectral errors. Other generative models also exhibit similar tradeoffs: DDIM/DDPM and EDM either have better pointwise or spectral accuracy, depending on how they are sampled. This suggests that evaluating surrogates on turbulence modeling requires more than just pointwise mean-squared errors, and that the best generative model depends on the metric of interest.

SI-based generative models also perform well on this task, suggesting that a good choice of source distribution can lead to gains in performance. However, the primary benefit seems to be in the flexibility of the stochastic interpolant framework. By providing a theoretical framework for training and sampling transport between arbitrary distributions, it admits many possible transport maps, including common Gaussian denoising models. One of these is likely better than a predefined framework from another field or even another PDE task. As physics can change drastically between problem settings, it makes sense to tailor a generative model to each task—which is convenient in a stochastic interpolant framework.

**Weather Forecasting** Many commonly studied PDEs are deterministic; when fully observed, a future state should be known provided that a sufficiently small time interval is used. Although uncertainty and statistical metrics are useful for PDEs, climate systems benefit more directly from probabilistic modeling due to the need for well-calibrated forecasts as well as inherent uncertainty in weather data collection and numerical weather prediction (Palmer, 2019; Dueben et al., 2022; Bracco et al., 2025). This makes weather forecasting a good benchmark not only for evaluating the accuracy of generative models but also for understanding their ability to approximate underlying distributions and to capture uncertainty.

After training, models are evaluated on medium-range weather forecasting for up to 10 days. The latitude-weighted RMSE (lRMSE) of models is reported in Table 5. At this time horizon, we observe that deterministic samplers tend to do well. This corroborates previous work where flow matching has also been shown to outperform DDPMs Couairon et al. (2024). There are a few likely reasons for this. Firstly, medium-range lRMSE generally optimizes for deterministic performance, where a standard approach is intentionally smooth predictions through ensembling and reduce lRMSE Price et al. (2024). Additionally, at this resolution ($64 \times 128$), low-frequency variability tends to dominate global weather, and high-frequency responses may take longer than 10 days to propagate to low-frequency regimes of the spectrum. This task setting also favors the SI model where subsequent states separated by 6 hours still tend to be related, and therefore a Dirac source distribution can improve performance and sampling speed.

Although low lRMSE is associated with a good forecast, it has many shortcomings such as encouraging overly smoothed predictions. Therefore, minimizing lRMSE can produce forecasts that lack physical consistency or fidelity Bonavita (2024). Therefore, to better understand the performance of generative models, we also examine the CRPS and SSR of ensemble forecasts, which is a measure of probabilistic performance. Ensembles are initialized at every 3rd day in the validation year to make a 30-day forecast; the CRPS/SSR is calculated at each timestep and metrics are averaged across all initializations. The resulting CRPS and SSR plots are shown in Figure 3. DDPM is omitted due to its computational expense and its lower performance

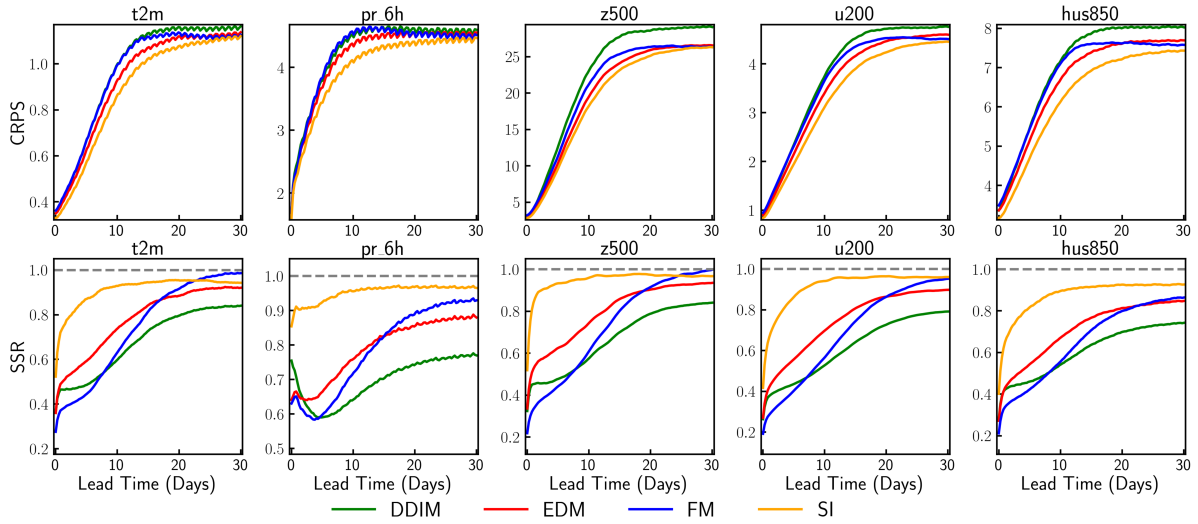


Figure 3: CRPS and SSR plots for generative models over 30 days. DDIM, EDM, and SI use 10 steps, while FM uses 5 steps.

on 10-day forecasts. To generate different ensemble members, stochastic interpolants are sampled with $\sigma = 1$, which is run with 10 steps.

Up to 30 days, stochastic interpolants show lower CRPS and better-calibrated SSR values. Up to a constant, the CRPS is equal to the squared L2 error between the true and predicted cumulative distribution functions (Zamo & Naveau, 2018), suggesting that stochastic interpolants have a lower distributional shift over time. While most generative models are under-dispersive at shorter lead times, stochastic interpolants tend to mitigate this and calibrate their uncertainty earlier. This is likely a result of a larger spread, which could be caused by the use of a stochastic sampler rather than a deterministic sampler in DDIM/EDM/FM. Lastly, the uncertainty of stochastic interpolants can be tuned by adjusting the noise coefficient during sampling, which could allow the model to produce ensembles with more or less variance.

**Climate Emulation** An advantage of using PlaSim instead of real-world data is the availability of a very large dataset of true samples from a long PlaSim simulation. This allows models to be evaluated for long-term climate consistency. After training to predict states at 6-hour intervals, generative models are queried to make predictions up to 10 years. For each variable and at each grid point and pressure level, predictions are averaged across all timesteps; the lRMSE between true and predicted averages is the 10-year climatological bias. Prior work has observed the lack of correlation between medium-range forecast error and climatological biases, stemming from error accumulation as the model trained for fast, weather dynamics is used for climatological timescales (Chattopadhyay & Hassanzadeh, 2023; Cachay et al., 2024; Watt-Meyer et al., 2025). Therefore, while 10-day forecasting errors converge and are reported after 30 epochs, models are trained for an additional 20 epochs in order for their climatological biases to also converge. We calculate the 10-year bias at each epoch and the model with the lowest bias is chosen and reported in Table 6.

In general, learning a consistent climate emulator with uniformly small biases globally from 6-hour prediction intervals is challenging. No model is the best over all variables, and differences between models and from epoch to epoch are large. Smoother fields such as temperature or geopotential are usually better modeled using deterministic samplers, while higher frequency fields such as precipitation or humidity are better modeled with stochastic samplers. Despite this, stochastic interpolants generally perform well, however its choice of source distribution provides less of a benefit at this timescale. Beyond a few weeks, predictions of future states are decorrelated from the true states, which undermines the source distribution, and the bias of the emulator relies on accurately predicting longer seasonal or yearly cycles rather than 6-hourly increments.

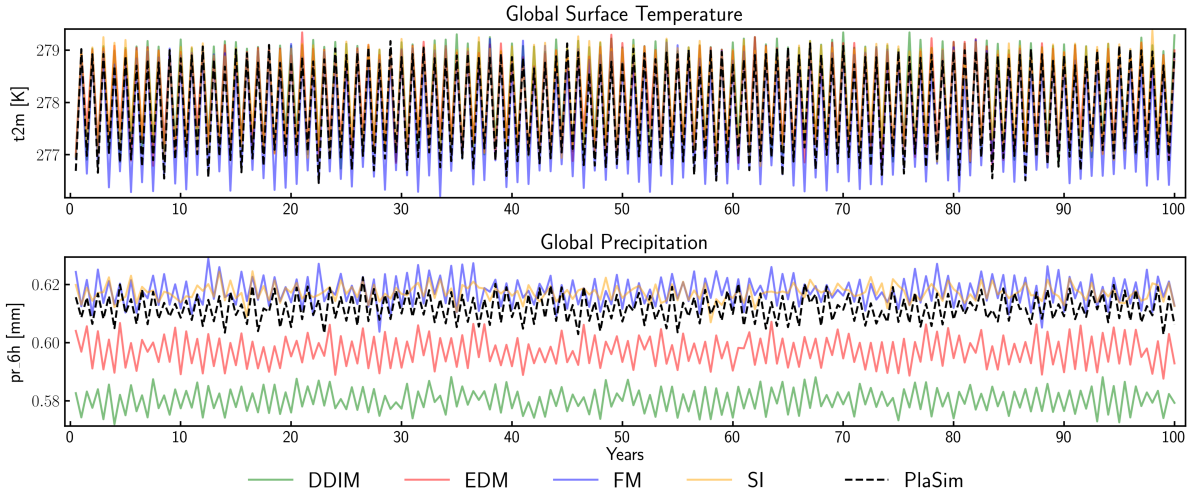


Figure 4: Global average surface temperature and precipitation, reported as an average ove 6 months for a 100 year period. The SI model is reported using $\sigma = 1$.

Lastly, we examine emulator responses for up to 100 years, where we track global temperature and precipitation to visualize trends in long-term model performance, shown in Figure 4. In general, most models respond well to the forcing of the seasonal cycle, leading to a globally averaged timeseries that is dominated by a periodic, annual timescale. Although models occasionally underestimate or overestimate global temperature, this effect is more consistent in more challenging fields such as precipitation. Furthermore, all models are stable despite being queried for nearly 150,000 autoregressive steps. However, at this time scale the benefits of an SI model are less clear, and in general, it is not clear how to make design choices to encourage long-term consistency while training on a 6-hour interval.

Table 6: 10-year Climatological Biases. DDIM, EDM, and SI ($\sigma = 1$) use 10 sampling steps, while FM and SI ($\sigma = 0$) use sampling 5 steps.

| Var: | z500 | t2m | t850 | u250 | pr_6h | hus850 |
|---|---|---|---|---|---|---|
| DDIM | 15.7 | 0.55 | 0.45 | 1.91 | 0.125 | 0.437 |
| EDM$_{(\text{Det.})}$ | 9.66 | 0.54 | 0.56 | 1.34 | 0.148 | 0.380 |
| FM | 8.06 | 0.35 | 0.31 | 1.30 | 0.081 | 0.268 |
| SI$_{(\sigma\,=\,0)}$ | 8.81 | 0.43 | 0.42 | 1.22 | 0.110 | 0.257 |
| SI$_{(\sigma\,=\,1)}$ | 10.7 | 0.51 | 0.48 | 1.03 | 0.069 | 0.187 |

## 5 Discussion

What is the best generative model for a given PDE task? Throughout this work, empirical evidence suggests that this is heavily dependent on the phenomena and physics of interest. In certain tasks, a generative model may not even be needed, such as when predicting laminar or well-resolved physics. Even when fixing the physical system, the most suitable generative model depends on the metric or task of interest. The relative benefit of different generative frameworks can change when evaluating different errors (pointwise vs. spectral), timescales (medium-range vs. climatological), or even variables (temperature vs. precipitation).

Despite this, stochastic interpolants have shown promise as a generative model for modeling physical systems. The key inductive bias that is leveraged is the assumption that successive states are related over a given time interval. When architectures and parameterizations are kept constant, empirical evidence suggests that this can be an effective way to train and sample a generative model. However, when the system becomes too chaotic or models are rolled out for extremely long horizons, assumptions of coupled source and target distributions become less helpful. No single generative model outperforms others on all evaluations (i.e., simple vs. complex systems, pointwise vs. spectral accuracy, etc.) which suggests that training and sampling frameworks should be informed by the problem at hand.

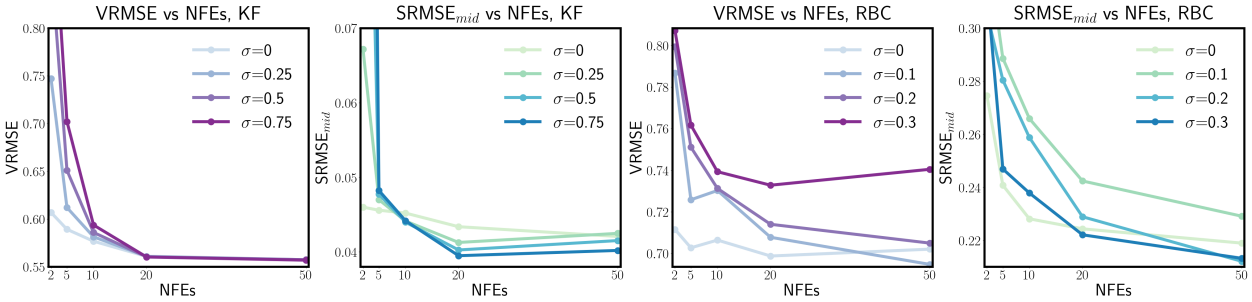


Figure 5: SI-EM models trained on either Kolmogorov Flow (KF) or Rayleigh-Bénard Convection (RBC), sampled with varying stochasticity. VRMSE and SRMSE$_{mid}$ are reported for different numbers of sampling steps (NFEs), and $\sigma$ scales the noise term $\sigma(1-t)$.

Stochastic interpolants are also remarkably flexible. Beyond choosing the source distribution and the mixing coefficients at training time, the stochasticity of the sampling process can also be varied. To better understand this, we vary the diffusion magnitude during sampling as well as the number of sampling steps in Kolmogorov Flow or Rayleigh-Bénard Convection, and plot the resulting VRMSE and mid-band SRMSE in Figure 5. The trends are consistent with prior intuition. SDEs are more sensitive to discretization error, and at low NFEs, stochasticity harms performance because samplers need more steps to converge. However, when given enough sampling steps, using more stochasticity during sampling usually converges in terms of deterministic accuracy and outperforms less stochastic samplers in terms of spectral accuracy. In more chaotic systems, where the latent space has more variance, excessive noise during sampling can cause predictions to diverge more quickly. However, stochasticity can still be important in recovering the correct spectrum, provided enough sampling steps are used to mix and sample correctly from the underlying modes.

Beyond interpolants, there are interesting trends in other generative models. EDM and flow matching are remarkably consistent in their improvement over DDPM, suggesting that reparameterizing vanilla diffusion with an ODE sampler offers benefits in accuracy and speed in physical surrogates. Conditional distributions in physical systems tend to be unimodal, as a single state is usually the most probable when observing a past state. Therefore, adding noise in the sampling process when solving the reverse SDE may not be necessary to mix different modes. However, uni-modality can be violated in highly chaotic systems, which can result in worse spectral performance when using an ODE sampler. Therefore, these results suggest that Gaussian denoising models can also tradeoff pointwise and spectral performance during sampling.

Finally, to better understand the benchmarked generative models, we provide a set of ablation studies in Appendix A. Consistent with prior evidence (Rozet et al., 2025) we show that the compression ratio of the autoencoder does not influence prediction error substantially. This is because the primary driver of rollout error is autoregressive error accumulation, rather than reconstruction error; however, excessive compression ratios will degrade the spectral fidelity of the predicted solution. We also verify that generative models are more effective in latent space than in pixel space. Previous works have also observed this, with the hypothesis that a compressed latent space has less variance for autoregressive error accumulation and a well-trained decoder may smooth out errors. Lastly, for stochastic interpolants we investigate the effect of varying the noise coefficient during training, rather than sampling. Adding noise to the interpolant encourages modes from the source and target distributions to mix and produces a smoother trajectory through probability space. This has the effect of regularizing intermediate interpolant states $x_t$ to improve rollout errors during sampling. However, too much noise can transport intermediate distributions too far from the source or target distributions and increase the difficulty of learning a drift or sampling the transport map.

## 6 Conclusion

**Limitations** There is ample room to continue investigating stochastic interpolants for physical emulation, as it is a highly general framework. Firstly, different source distributions beyond the Gaussian and point masses considered in this work may be beneficial. In fact, these distributions can be seen as the limits of a

continuum of source distributions defined as partially noised current states. For certain physics problems, there are also other prior distributions that may be relevant, such as the mean flow in Kolmogorov Flow or the climatology in climate emulation. Additionally, we did not thoroughly investigate many training parameters such as the mixing coefficients or loss function, which may benefit from preconditioning or input/output scaling (Karras et al., 2022). It is also likely that sampling the interpolant timestep $t$ uniformly is not ideal, as intermediate steps along the transport map are more difficult to learn. Lastly, higher-order samplers could also further improve the performance of stochastic interpolants.

**Takeaways** Modeling physical phenomena is challenging, as each system exhibits distinct dynamics, variables, and spatial or temporal scales. This variability suggests that the design of generative surrogates should consider the physical system and desired performance metrics. Our experiments support this idea: even with the same architecture and training budget, changes in the transport map and sampling can result in different deterministic and statistical performance. Despite this, stochastic interpolants can be a strong baseline for generative models, motivated by the proximity of subsequent states in autoregressive prediction. Furthermore, its flexibility can be a useful tool to tailor model performance to specific systems, or to investigate new samplers or transport maps for physical emulation, all under a unified framework.

## References

Michael Aich, Philipp Hess, Baoxiang Pan, Sebastian Bathiany, Yu Huang, and Niklas Boers. Conditional diffusion models for downscaling & bias correction of earth system model precipitation, 2024. URL https://arxiv.org/abs/2404.14416.

Michael S. Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants, 2023. URL https://arxiv.org/abs/2209.15571.

Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions, 11 2023. URL http://arxiv.org/abs/2303.08797.

Michael S. Albergo, Mark Goldstein, Nicholas M. Boffi, Rajesh Ranganath, and Eric Vanden-Eijnden. Stochastic interpolants with data-dependent couplings, 2024. URL https://arxiv.org/abs/2310.03725.

Ferran Alet, Ilan Price, Andrew El-Kadi, Dominic Masters, Stratis Markou, Tom R. Andersson, Jacklynn Stott, Remi Lam, Matthew Willson, Alvaro Sanchez-Gonzalez, and Peter Battaglia. Skillful joint probabilistic weather forecasting from marginals, 2025. URL https://arxiv.org/abs/2506.10772.

Gérôme Andry, François Rozet, Sacha Lewin, Omer Rochman, Victor Mangeleer, Matthias Pirlet, Elise Faulx, Marilaure Grégoire, and Gilles Louppe. Appa: Bending weather dynamics with latent diffusion models for global data assimilation, 2025. URL https://arxiv.org/abs/2504.18720.

Victor Armegioiu, Yannick Ramic, and Siddhartha Mishra. Rectified flows for fast multiscale fluid flow modeling, 2025. URL https://arxiv.org/abs/2506.03111.

Giacomo Baldan, Qiang Liu, Alberto Guardone, and Nils Thuerey. Flow matching meets pdes: A unified framework for physics-constrained generation, 2025. URL https://arxiv.org/abs/2506.08604.

Jan-Hendrik Bastek, WaiChing Sun, and Dennis M. Kochmann. Physics-informed diffusion models, 2025. URL https://arxiv.org/abs/2403.14404.

Kaifeng Bi, Lingxi Xie, Hengheng Zhang, Xin Chen, Xiaotao Gu, and Qi Tian. Pangu-weather: A 3d high-resolution model for fast and accurate global weather forecast, 2022. URL https://arxiv.org/abs/2211.02556.

Tobias Bischoff and Katherine Deck. Unpaired downscaling of fluid flows with diffusion bridges, 2023. URL https://arxiv.org/abs/2305.01822.

Massimo Bonavita. On some limitations of current machine learning weather prediction models. *Geophysical Research Letters*, 51(12):e2023GL107377, 2024. doi: https://doi.org/10.1029/2023GL107377. URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2023GL107377. e2023GL107377 2023GL107377.

Boris Bonev, Thorsten Kurth, Christian Hundt, Jaideep Pathak, Maximilian Baust, Karthik Kashinath, and Anima Anandkumar. Spherical fourier neural operators: Learning stable dynamics on the sphere, 2023. URL https://arxiv.org/abs/2306.03838.

Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling, 2023. URL https://arxiv.org/abs/2106.01357.

Annalisa Bracco, Julien Brajard, Henk A Dijkstra, Pedram Hassanzadeh, Christian Lessig, and Claire Monteleoni. Machine learning for the physics of climate. *Nature Reviews Physics*, 7(1):6–20, 2025.

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.

Salva Rühling Cachay, Bo Zhao, Hailey Joren, and Rose Yu. Dyffusion: A dynamics-informed diffusion model for spatiotemporal forecasting, 2023. URL https://arxiv.org/abs/2306.01984.

Salva Rühling Cachay, Brian Henn, Oliver Watt-Meyer, Christopher S. Bretherton, and Rose Yu. Probabilistic emulation of a global climate model with spherical dyffusion, 2024. URL https://arxiv.org/abs/2406.14798.

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, 2025. URL https://arxiv.org/abs/2506.20024.

Ashesh Chattopadhyay and Pedram Hassanzadeh. Long-term instabilities of deep learning-based digital twins of the climate system: The cause and a solution. *arXiv preprint arXiv:2304.07029*, 2023.

Junyu Chen, Han Cai, Junsong Chen, Enze Xie, Shang Yang, Haotian Tang, Muyang Li, Yao Lu, and Song Han. Deep compression autoencoder for efficient high-resolution diffusion models, 2025a. URL https://arxiv.org/abs/2410.10733.

Siyi Chen, Yixuan Jia, Qing Qu, He Sun, and Jeffrey A Fessler. Flowdas: A stochastic interpolant-based framework for data assimilation, 2025b. URL https://arxiv.org/abs/2501.16642.

Yifan Chen and Eric Vanden-Eijnden. Scale-adaptive generative flows for multiscale scientific data, 2026. URL https://arxiv.org/abs/2509.02971.

Yifan Chen, Mark Goldstein, Mengjian Hua, Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Probabilistic forecasting with stochastic interpolants and föllmer processes, 2024. URL https://arxiv.org/abs/2403.13724.

Yifan Chen, Eric Vanden-Eijnden, and Jiawei Xu. Lipschitz-guided design of interpolation schedules in generative models, 2026. URL https://arxiv.org/abs/2509.01629.

Guillaume Couairon, Renu Singh, Anastase Charantonis, Christian Lessig, and Claire Monteleoni. Archesweather & archesweathergen: a deterministic and generative model for efficient ml weather forecasting, 2024. URL https://arxiv.org/abs/2412.12971.

Pan Du, Meet Hemant Parikh, Xiantao Fan, Xin-Yang Liu, and Jian-Xun Wang. Confild: Conditional neural field latent diffusion model generating spatiotemporal turbulence, 2024. URL https://arxiv.org/abs/2403.05940.

Peter D. Dueben, Martin G. Schultz, Matthew Chantry, David John Gagne, David Matthew Hall, and Amy McGovern. Challenges and benchmark datasets for machine learning in the atmospheric sciences: Definition, status, and outlook. *Artificial Intelligence for the Earth Systems*, 1(3):e210002, 2022. doi: 10.1175/AIES-D-21-0002.1. URL https://journals.ametsoc.org/view/journals/aies/1/3/AIES-D-21-0002.1.xml.

V. Fortin, M. Abaza, F. Anctil, and R. Turcotte. Why should ensemble spread match the rmse of the ensemble mean? *Journal of Hydrometeorology*, 15(4):1708 – 1713, 2014. doi: 10.1175/JHM-D-14-0008.1. URL https://journals.ametsoc.org/view/journals/hydr/15/4/jhm-d-14-0008_1.xml.

Han Gao, Sebastian Kaltenbach, and Petros Koumoutsakos. Generative learning for forecasting the dynamics of high-dimensional complex systems. *Nature Communications*, 15(1):8904, Oct 2024. ISSN 2041-1723. doi: 10.1038/s41467-024-53165-w. URL https://doi.org/10.1038/s41467-024-53165-w.

Zhihan Gao, Xingjian Shi, Boran Han, Hao Wang, Xiaoyong Jin, Danielle Maddix, Yi Zhu, Mu Li, and Yuyang Wang. Prediff: Precipitation nowcasting with latent diffusion models, 2023. URL https://arxiv.org/abs/2307.10422.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. URL https://arxiv.org/abs/2006.11239.

Jiahe Huang, Guandao Yang, Zichen Wang, and Jeong Joon Park. Diffusionpde: Generative pde-solving under partial observation, 2024. URL https://arxiv.org/abs/2406.17763.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models, 2022. URL https://arxiv.org/abs/2206.00364.

Felix Koehler, Simon Niedermayr, Rüdiger Westermann, and Nils Thuerey. Apebench: A benchmark for autoregressive neural emulators of pdes, 2024. URL https://arxiv.org/abs/2411.00180.

Georg Kohl, Li-Wei Chen, and Nils Thuerey. Benchmarking autoregressive conditional diffusion models for turbulent flow simulation, 2024. URL https://arxiv.org/abs/2309.01745.

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. URL http://jmlr.org/papers/v24/21-1524.html.

Amaury Lancelin, Alex Wikner, Laurent Dubus, Clément Le Priol, Dorian S. Abbot, Freddy Bouchet, Pedram Hassanzadeh, and Jonathan Weare. Ai-boosted rare event sampling to characterize extreme weather, 2025. URL https://arxiv.org/abs/2510.27066.

Simon Lang, Mihai Alexe, Mariana C. A. Clare, Christopher Roberts, Rilwan Adewoyin, Zied Ben Bouallègue, Matthew Chantry, Jesper Dramsch, Peter D. Dueben, Sara Hahner, Pedro Maciel, Ana Prieto-Nemesio, Cathal O'Brien, Florian Pinault, Jan Polster, Baudouin Raoult, Steffen Tietsche, and Martin Leutbecher. Aifs-crps: Ensemble forecasting using a model trained with a loss function based on the continuous ranked probability score, 2024. URL https://arxiv.org/abs/2412.15832.

Zijie Li, Kazem Meidani, and Amir Barati Farimani. Transformer for partial differential equations' operator learning, 2023. URL https://arxiv.org/abs/2205.13671.

Zijie Li, Anthony Zhou, Saurabh Patil, and Amir Barati Farimani. Cafa: Global weather forecasting with factorized attention on sphere, 2024. URL https://arxiv.org/abs/2405.07395.

Zijie Li, Saurabh Patil, Francis Ogoke, Dule Shu, Wilson Zhen, Michael Schneier, Jr. John R. Buchanan, and Amir Barati Farimani. Latent neural pde solver: a reduced-order modelling framework for partial differential equations, 2025a. URL https://arxiv.org/abs/2402.17853.

Zijie Li, Anthony Zhou, and Amir Barati Farimani. Generative latent neural pde solver using flow matching, 2025b. URL https://arxiv.org/abs/2503.22600.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations, 2021. URL https://arxiv.org/abs/2010.08895.

Marten Lienen, David Lüdke, Jan Hansen-Palmus, and Stephan Günnemann. From zero to turbulence: Generative modeling for 3d flow simulation, 5 2023. URL http://arxiv.org/abs/2306.01776.

Mario Lino, Tobias Pfaff, and Nils Thuerey. Learning distributions of complex fluid simulations with diffusion graph networks, 2025. URL https://arxiv.org/abs/2504.02843.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023. URL https://arxiv.org/abs/2210.02747.

Phillip Lippe, Bastiaan S. Veeling, Paris Perdikaris, Richard E. Turner, and Johannes Brandstetter. Pde-refiner: Achieving accurate long rollouts with neural pde solvers, 2023. URL https://arxiv.org/abs/2308.05732.

Bjoern List, Li-Wei Chen, Kartik Bali, and Nils Thuerey. Differentiability in unrolled training of neural physics simulators on transient dynamics. *Computer Methods in Applied Mechanics and Engineering*, 433: 117441, 2025. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2024.117441. URL https://www.sciencedirect.com/science/article/pii/S0045782524006960.

Qi Liu, Laure Zanna, and Joan Bruna. Towards infinitely long neural simulations: Self-refining neural surrogate models for dynamical systems, 2026. URL https://arxiv.org/abs/2603.17750.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow, 2022. URL https://arxiv.org/abs/2209.03003.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *Machine Intelligence Research*, 22(4):730–751, June 2025. ISSN 2731-5398. doi: 10.1007/s11633-025-1562-4. URL http://dx.doi.org/10.1007/s11633-025-1562-4.

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, March 2021. ISSN 2522-5839. doi: 10.1038/s42256-021-00302-5. URL http://dx.doi.org/10.1038/s42256-021-00302-5.

Frank Lunkeit, Simon Blessing, Klaus Friedrich, Heiko Jansen, Edilbert Kirk, Ute Luksch, and Frank Sielmann. PlaSim: Planet Simulator. Astrophysics Source Code Library, record ascl:2107.019, July 2021.

Morteza Mardani, Noah Brenowitz, Yair Cohen, Jaideep Pathak, Chieh-Yu Chen, Cheng-Chin Liu, Arash Vahdat, Mohammad Amin Nabian, Tao Ge, Akshay Subramaniam, Karthik Kashinath, Jan Kautz, and Mike Pritchard. Residual corrective diffusion modeling for km-scale atmospheric downscaling, 2024. URL https://arxiv.org/abs/2309.15214.

Michael McCabe, Peter Harrington, Shashank Subramanian, and Jed Brown. Towards stability of autoregressive neural operators, 2023. URL https://arxiv.org/abs/2306.10619.

Michael McCabe, Bruno Régaldo-Saint Blancard, Liam Holden Parker, Ruben Ohana, Miles Cranmer, Alberto Bietti, Michael Eickenberg, Siavash Golkar, Geraud Krawezik, Francois Lanusse, Mariel Pettee, Tiberiu Tesileanu, Kyunghyun Cho, and Shirley Ho. Multiple physics pretraining for physical surrogate models, 2024. URL https://arxiv.org/abs/2310.02994.

Roberto Molinaro, Samuel Lanthaler, Bogdan Raonić, Tobias Rohner, Victor Armegioiu, Stephan Simonis, Dana Grund, Yannick Ramic, Zhong Yi Wan, Fei Sha, Siddhartha Mishra, and Leonardo Zepeda-Núñez. Generative ai for fast and accurate statistical computation of fluids, 2025. URL https://arxiv.org/abs/2409.18359.

Tung Nguyen, Johannes Brandstetter, Ashish Kapoor, Jayesh K. Gupta, and Aditya Grover. Climax: A foundation model for weather and climate, 2023. URL https://arxiv.org/abs/2301.10343.

Ruben Ohana, Michael McCabe, Lucas Meyer, Rudy Morel, Fruzsina J. Agocs, Miguel Beneitez, Marsha Berger, Blakesley Burkhart, Keaton Burns, Stuart B. Dalziel, Drummond B. Fielding, Daniel Fortunato, Jared A. Goldberg, Keiya Hirashima, Yan-Fei Jiang, Rich R. Kerswell, Suryanarayana Maddu, Jonah Miller, Payel Mukhopadhyay, Stefan S. Nixon, Jeff Shen, Romain Watteaux, Bruno Régaldo-Saint Blancard, François Rozet, Liam H. Parker, Miles Cranmer, and Shirley Ho. The well: a large-scale collection of diverse physics simulations for machine learning, 2025. URL https://arxiv.org/abs/2412.00568.

Vivek Oommen, Aniruddha Bora, Zhen Zhang, and George Em Karniadakis. Integrating neural operators with diffusion models improves spectral representation in turbulence modeling, 2025. URL https://arxiv.org/abs/2409.08477.

Tim N Palmer. Stochastic weather and climate models. *Nature Reviews Physics*, 1(7):463–471, 2019.

Jaideep Pathak, Shashank Subramanian, Peter Harrington, Sanjeev Raja, Ashesh Chattopadhyay, Morteza Mardani, Thorsten Kurth, David Hall, Zongyi Li, Kamyar Azizzadenesheli, et al. Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators. *arXiv preprint arXiv:2202.11214*, 2022.

Chris Pedersen, Laure Zanna, and Joan Bruna. Thermalizer: Stable autoregressive neural emulation of spatiotemporal chaos, 2025. URL https://arxiv.org/abs/2503.18731.

William Peebles and Saining Xie. Scalable diffusion models with transformers, 2023. URL https://arxiv.org/abs/2212.09748.

Ilan Price, Alvaro Sanchez-Gonzalez, Ferran Alet, Tom R. Andersson, Andrew El-Kadi, Dominic Masters, Timo Ewalds, Jacklynn Stott, Shakir Mohamed, Peter Battaglia, Remi Lam, and Matthew Willson. Gencast: Diffusion-based ensemble forecasting for medium-range weather, 2024. URL https://arxiv.org/abs/2312.15796.

Francesco Ragone, Jeroen Wouters, and Freddy Bouchet. Computation of extreme heat waves in climate models using a large deviation algorithm. *Proceedings of the National Academy of Sciences*, 115(1):24–29, 2018. doi: 10.1073/pnas.1712645115. URL https://www.pnas.org/doi/abs/10.1073/pnas.1712645115.

Bogdan Raonic, Roberto Molinaro, Tim De Ryck, Tobias Rohner, Francesca Bartolucci, Rima Alaifari, Siddhartha Mishra, and Emmanuel de Bezenac. Convolutional neural operators for robust and accurate learning of PDEs. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=MtekhXRP4h.

Stephan Rasp, Stephan Hoyer, Alexander Merose, Ian Langmore, Peter Battaglia, Tyler Russel, Alvaro Sanchez-Gonzalez, Vivian Yang, Rob Carver, Shreya Agrawal, Matthew Chantry, Zied Ben Bouallegue, Peter Dueben, Carla Bromberg, Jared Sisk, Luke Barrington, Aaron Bell, and Fei Sha. Weatherbench 2: A benchmark for the next generation of data-driven global weather models, 2024. URL https://arxiv.org/abs/2308.15560.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. URL https://arxiv.org/abs/2112.10752.

Siddharth Rout, Eldad Haber, and Stéphane Gaudreault. Probabilistic forecasting for dynamical systems with missing or imperfect data, 3 2025. URL http://arxiv.org/abs/2503.12273.

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, 2025. URL https://arxiv.org/abs/2507.02608.

Martin Schiødt, Nikolaj Takata Mücke, and Clara Marika Velte. Generative super-resolution of turbulent flows via stochastic interpolants, 2025. URL https://arxiv.org/abs/2508.13770.

Louis Serrano, Thomas X Wang, Etienne Le Naour, Jean-Noël Vittaut, and Patrick Gallinari. Aroma: Preserving spatial structure for latent pde modeling with local neural fields, 2024. URL https://arxiv.org/abs/2406.02176.

Youssef Shehata, Benjamin Holzschuh, and Nils Thuerey. Improved sampling of diffusion models in fluid dynamics with tweedie's formula. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=0FbzC7B9xI.

Yaozhong Shi, Angela F. Gao, Zachary E. Ross, and Kamyar Azizzadenesheli. Universal functional regression with neural operator flows, 2024. URL https://arxiv.org/abs/2404.02986.

Dule Shu, Zijie Li, and Amir Barati Farimani. A physics-informed diffusion model for high-fidelity flow field reconstruction. *Journal of Computational Physics*, 478, 4 2023. ISSN 10902716. doi: 10.1016/j.jcp.2023.111972.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2022. URL https://arxiv.org/abs/2010.02502.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2021. URL https://arxiv.org/abs/2011.13456.

Xuan Su, Jiaming Song, Chenlin Meng, and Stefano Ermon. Dual diffusion implicit bridges for image-to-image translation, 2023. URL https://arxiv.org/abs/2203.08382.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport, 2024. URL https://arxiv.org/abs/2302.00482.

Utkarsh Utkarsh, Pengfei Cai, Alan Edelman, Rafael Gomez-Bombarelli, and Christopher Vincent Rackauckas. Physics-constrained flow matching: Sampling generative models with hard constraints, 2025. URL https://arxiv.org/abs/2506.04171.

Yogesh Verma, Markus Heinonen, and Vikas Garg. Climode: Climate and weather forecasting with physics-informed neural odes, 2024. URL https://arxiv.org/abs/2404.10024.

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, May 2025. ISSN 2397-3722. doi: 10.1038/s41612-025-01090-0. URL https://doi.org/10.1038/s41612-025-01090-0.

Gefan Yang and Stefan Sommer. A denoising diffusion model for fluid field prediction, 2023. URL https://arxiv.org/abs/2301.11661.

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, 2026. URL https://arxiv.org/abs/2505.17004.

Michaël Zamo and Philippe Naveau. Estimation of the continuous ranked probability score with limited information and applications to ensemble weather forecasts. *Mathematical Geosciences*, 50(2):209–234, 2018. doi: 10.1007/s11004-017-9709-7. URL https://doi.org/10.1007/s11004-017-9709-7.

Shaorong Zhang, Yuanbin Cheng, and Greg Ver Steeg. Exploring the design space of diffusion bridge models, 2025. URL https://arxiv.org/abs/2410.21553.

Kaiwen Zheng, Guande He, Jianfei Chen, Fan Bao, and Jun Zhu. Diffusion bridge implicit models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=eghAocvqBk.

Anthony Zhou and Amir Barati Farimani. Predicting change, not states: An alternate framework for neural pde surrogates. *Computer Methods in Applied Mechanics and Engineering*, 441:117990, June 2025. ISSN 0045-7825. doi: 10.1016/j.cma.2025.117990. URL http://dx.doi.org/10.1016/j.cma.2025.117990.

Anthony Zhou and Amir Barati Farimani. Masked autoencoders are pde learners, 2024. URL https://arxiv.org/abs/2403.17728.

Anthony Zhou and Amir Barati Farimani. Hamiltonian neural pde solvers through functional approximation, 2025. URL https://arxiv.org/abs/2505.13275.

Anthony Zhou, Cooper Lorsung, AmirPouya Hemmasian, and Amir Barati Farimani. Strategies for pre-training neural operators, 2024. URL https://arxiv.org/abs/2406.08473.

Anthony Zhou, Zijie Li, Michael Schneier, John R Buchanan Jr, and Amir Barati Farimani. Text2pde: Latent diffusion models for accessible physics simulation, 2025. URL https://arxiv.org/abs/2410.01153.

Linqi Zhou, Aaron Lou, Samar Khanna, and Stefano Ermon. Denoising diffusion bridge models, 2023. URL https://arxiv.org/abs/2309.16948.

Yilin Zhuang and Karthik Duraisamy. Ladcast: A latent diffusion model for medium-range ensemble weather forecasting, 2025. URL https://arxiv.org/abs/2506.09193.

## A Ablation Studies

**Pixel/Latent Space Ablations** We train flow matching and stochastic interpolant models in either pixel or latent space to compare their performance. The Normalized RMSE (NRMSE), or Relative L2 error, is reported for the Kolmogorov Flow validation set in Table 7. Model sizes are kept roughly constant, and all models are trained for the same number of epochs. To account for the larger spatial input to the DiT ($160 \times 160$), a patch size of $8 \times 8$ was used to match the compression ratio of the autoencoder.

In general, we find that pixel space models have higher errors than latent space models, in addition to being more expensive to train and query. One hypothesis is that latent space models could be more stable, as the latent space has less variance and a well-trained decoder may smooth out errors. Indeed, we find that pixel space models accumulate error faster than latent space models.

Table 7: Comparison of pixel and latent space generative models on Kolmogorov Flow.

| Metric | $\text{FM}_{pixel}$ | $\text{FM}_{latent}$ | $\text{SI}_{pixel}$ | $\text{SI}_{latent}$ |
|---|---|---|---|---|
| NRMSE | 0.812 | 0.649 | 0.966 | 0.609 |

**Compression Ratio Ablations** We compare the effects of training autoencoders with different compression ratios $(64, 256)$ on the reconstruction error of the autoencoder and the rollout error of the generative model. The NRMSE for Rayleigh-Bénard convection is reported in Table 8. A larger compression ratio results in a smaller latent space, increasing the reconstruction error of the autoencoder. Despite having nearly double the reconstruction error, generative models with more aggressive compression only have a modest increase in rollout error, which is consistent with Rozet et al. (2025).

Table 8: Comparison of NRMSE for models using different compression ratios ($\div$) on Rayleigh-Bénard Convection. Reconstruction error is reported for the autoencoder (AE), while rollout error is reported for the generative models.

| $\div$ | AE | FM | SI |
|---|---|---|---|
| 64 | 0.0252 | 0.619 | 0.605 |
| 256 | 0.0457 | 0.649 | 0.622 |

**Noise Coefficients for Interpolants** The stochastic interpolant considered in this work is given by: $x(t) = \alpha(t)x_0 + \beta(t)x_1 + \sigma(t)W_t$. We define $\sigma(t) = \sigma(1-t)$ as the noise coefficient, where $\sigma$ controls the scale of the noise in the stochastic process. More noise can encourage mixing between $x_0$ and $x_1$, however, too much noise can make the stochastic process more difficult to learn or sample from. We can see this in Table 9, where SI models are trained to learn stochastic processes with different noise scales $\sigma$ and are sampled with an Euler sampler using 2 steps. In general, $\sigma$ can be tuned to find an optimal amount of noise for the stochastic process.

Table 9: Comparison of different $\sigma$ coefficients for SI models in Kolmogorov Flow. $\sigma$ scales the amount of noise in the stochastic process.

| $\sigma$: | 0.1 | 0.5 | 1 | 3 |
|---|---|---|---|---|
| NRMSE | 0.702 | 0.632 | 0.609 | 0.803 |