REFRAMING GENERATIVE MODELS FOR PHYSICAL SYSTEMS USING STOCHASTIC INTERPOLANTS

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. Most approaches rely on iteratively denoising a Gaussian, a choice that may not be the most effective for autoregressive prediction tasks in PDEs and dynamical systems such as climate. In this work, we benchmark generative models across diverse physical domains and tasks, and highlight the role of stochastic interpolants. By directly learning a stochastic process between current and future states, stochastic interpolants can leverage the proximity of successive physical distributions. This allows for generative models that can use fewer sampling steps and produce more accurate predictions than models relying on transporting Gaussian noise. Our experiments suggest that generative models need to balance deterministic accuracy, spectral consistency, and probabilistic calibration, and that stochastic interpolants can potentially fulfill these requirements by adjusting their sampling. This study establishes stochastic interpolants as a competitive baseline for physical emulation and gives insight into the abilities of different generative modeling frameworks.

1 INTRODUCTION

Generative models have recently become a promising class of models for physical systems. Empirically, 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 these prior works 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 both inefficient and more complex than directly mapping the source to the target distribution.

This observation makes stochastic interpolants well-suited as a generative model for physical systems. Specifically, predicting PDE or climate systems are usually framed as an autoregressive task, where future states are predicted based on a current state. While current and future states are often tightly coupled, generative models still predominantly follow the approach of transporting Gaus-

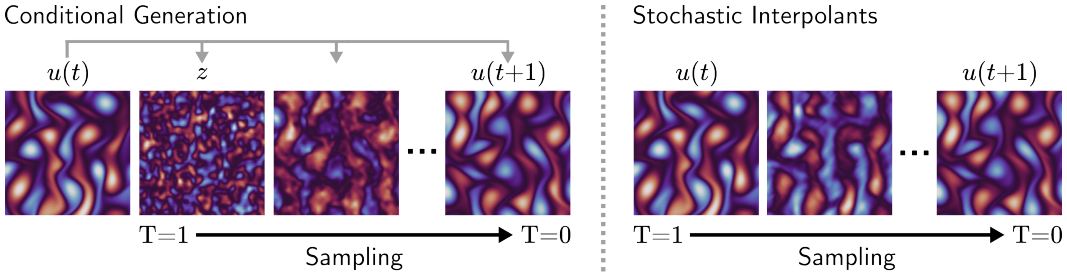

Figure 1: *Left:* Typical generative models rely on transporting Gaussian noise z, conditioned on a current state $u(t)$. *Right:* Learning a stochastic process to transport current to future states can be more efficient and accurate. The mixing of the source and target distributions can be controlled by the amount of added noise. T denotes time along a stochastic process, while t is the physical time.

sian noise while conditioning on the current state. This common framework is likely wasteful, and stochastic interpolants trained to map current to future states can result in faster or more accurate generative models for physical systems. We provide a conceptual comparison in Figure 1.

2 Stochastic Interpolants

Definition 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 ρ_0 and samples x_1 from a target distribution ρ_1, a stochastic interpolant is defined as a stochastic process x_t such that:

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

The interpolant I satisfies the boundary conditions $I(0, x_0, x_1) = x_0$ and $I(1, x_0, x_1) = x_1$. Furthermore, z is sampled from a standard Gaussian $\mathcal{N}(0, I)$, and the noise coefficient $\gamma(t)$ satisfies the conditions $\gamma(0) = \gamma(1) = 0$ and $\gamma(t) > 0$. 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 ρ_0, ρ_1 exactly and in finite time, which is not true for typical DDPMs in PDE and climate domains. Lastly, a stochastic interpolant can be realized by either an ODE or an SDE, which can produce samples x_t at any time $t \in [0, 1]$.

Implementation Stochastic interpolants have many instantiations, as well as different training and sampling procedures. We consider spatially linear interpolants, where x_0 is sampled from a current solution $u(t)$ and x_1 is sampled from a future solution $u(t + 1)$:

$$x_t = \alpha(t)x_0 + \beta(t)x_1 + \gamma(t)z \tag{2}$$

The coefficients $\alpha(t), \beta(t), \gamma(t)$ are chosen to satisfy boundary conditions. Since these coefficients are specified, we can learn the drift b of the stochastic interpolant with a network b_θ by minimizing the empirical loss on the dataset $\{x^1, x^2, \ldots, x^N\}$:

$$\mathcal{L}_b[b_\theta] = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{2}|b_\theta(t_i, x^i_{t_i})|^2 - b_\theta(t_i, x^i_{t_i})(\partial_t I(t_i, x^i_0, x^i_1) + \dot{\gamma}(t)z^i)\right) \tag{3}$$

where $t_i \in [0, 1]$ is uniformly sampled and $x^i_{t_i}$ is a given data sample i at time t_i along the stochastic interpolant. Intuitively, b_θ aims to estimate the time derivative of the stochastic interpolant. This is useful during inference, where samples x_0 are integrated using drift estimates b_θ according to the probability flow ODE or SDE:

$$dX^{ODE}_t = b_\theta(t, X_t)dt, \qquad dX^{SDE}_t = b_\theta(t, X_t)dt + \frac{\dot{\gamma}(t)}{\sqrt{t}}dW_t \tag{4}$$

where $W_t := \sqrt{t}z$ is a Wiener process on $t \in [0, 1]$. A given drift b_θ and noise coefficient $\gamma(t)$ also define a family of SDEs that describe the same stochastic process, allowing the diffusion term dW_t to be adjusted during sampling without retraining (Chen et al., 2024).

While the overall loss and sampling framework gives exact generative models, in practice, implementation choices can influence how effectively the drift is learned due to numerical and statistical errors. For example, the loss $\mathcal{L}_b$ can have high variance around the endpoints $t = 0$ and $t = 1$ if $\dot{\gamma}(t)$ is singular. Furthermore, choices for coefficients $\alpha(t), \beta(t), \gamma(t)$ affect the mixing of the source, target, and noise distributions. We report our implementation details in Appendix D.

3 METHODS

3.1 DATASETS

Kolmogorov Flow Kolmogorov Flow (KF) is described by the 2D Navier-Stokes equations driven by unidirectional periodic forcing. 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 160×160 on a domain $(x, y) = [-10, 10]^2$, with the vorticity being recorded. 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 sampled from a random truncated Fourier series with 5 modes, and the viscosity ν is set to 10^{-2} to simulate a Reynolds number of approximately 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. Current PDE surrogates usually struggle since the system is highly chaotic and features a transition between laminar and turbulent regimes. The system is described by its Rayleigh and Prandtl numbers, which govern the convection and diffusivity of the flow. Data is obtained from the Well (Ohana et al., 2025; Burns et al., 2020), which includes 2D simulations on a 512×128 grid with buoyancy, pressure, and velocity. In addition, we use 100 timesteps with an interval of $\Delta t = 0.5$ seconds and a variety of Rayleigh and Prandtl numbers are used for training and validation.

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., in prep.). 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. 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 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 $\sim$144,000 training samples and $\sim$1,440 validation samples. Additional detail on datasets can be found in Appendix C.

3.2 MODELS

Overview Although the work focuses on generative models, we 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). To benchmark generative models, we consider: denoising diffusion probabilistic models (DDPM) (Ho et al., 2020), denoising diffusion implicit models (DDIM) (Song et al., 2022), elucidated diffusion models (EDM) (Karras et al., 2022), truncated sampling models (Shehata et al., 2025) (TSM), flow matching (FM) (Lipman et al., 2023), and stochastic interpolants (SI). These frameworks can have many variations, therefore, formulas for the training objective and sampling procedures used are given in Table 11.

Beyond overall frameworks, generative models are also determined by hyperparameters such as the noise schedule. We use a linear schedule for DDPM, DDIM, and TSM models. The noise schedule for EDM is reproduced from Karras et al. (2022). For FM models, we use the rectified flow schedule where $x_t = (1 - t)x_0 + tz$. For SI models, we choose $\alpha(t) = 1 - t$, $\beta_t = t$, and $\gamma(t) = (1 - t)\sqrt{t}$. ODE samplers use the Euler method and SDE samplers use the Euler-Maruyama method.

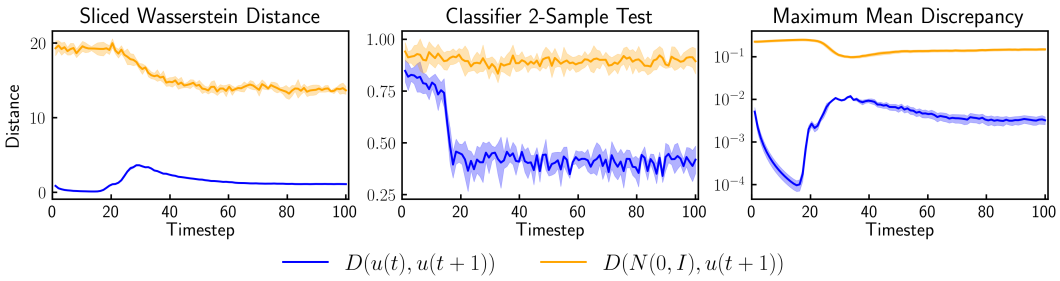

Figure 2: Distance heuristics for the Rayleigh-Bénard dataset. Distances between samples drawn from successive timesteps $D(u(t), u(t+1))$ and between Gaussian noise and future timesteps $D(N(0, I), u(t+1))$ are plotted for the buoyancy field. Each metric is averaged over a 5-fold cross validation, with the standard deviation shaded.

Autoencoders To effectively train generative models, we pretrain a latent space using an autoencoder (Rombach et al., 2022). This is effective in reducing computation, however, for PDE problems, this can additionally stabilize rollouts (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, rather than KL regularization, to avoid arbitrary variance. For climate tasks, we find that KL regularization is beneficial, potentially due to the need for long-term consistency in climate emulators. KF and RBC autoencoders use a compression ratio of $64\times$ and the PlaSim autoencoder uses a compression ratio of $32\times$.

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 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 estimate of the future state as conditioning. Moreover, as a result of the compressed latent space, the backbone does not need patchification or sparse attention. Additional details on the autoencoder and diffusion architectures can be found in Appendix D.

Metrics We follow standard metrics to evaluate the deterministic and statistical performance of models. For PDE tasks, we use Variance Scaled RMSE (VRMSE), and for weather forecasting, we use latitude-weighted RMSE (lRMSE). 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 3D variable.

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. Following Rozet et al. (2025), the power spectrum is calculated and partitioned into three evenly distributed frequency bands and reported as the RMSE of the relative power spectrum ($\sqrt{(1 - p/p_\theta)^2}$). For weather forecasting, the statistical 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. SSR values of 1 are considered optimal as the uncertainty of the forecast matches its error (Fortin et al., 2014). Additional information on the considered metrics is given in the Appendix D.2.

4 Results

Understanding Distances for Physical Distributions A major hypothesis of this work is that the distributions of current and future states are closer together than Gaussian noise and future states, which allows stochastic interpolants to be more efficient or accurate than conditional generation. Consider an example where initial states are uniformly sampled from a set of initial conditions. For

Model:	FNO	DDPM	DDIM	EDM	TSM	FM	SI	AE
NFEs:	1	100	10	10	1	2	2	1
VRMSE	0.621±0.008	0.684±0.022	0.735±0.007	0.616±0.013	0.835±0.044	0.593±0.029	0.552±0.005	0.011
SRMSE_{low}	0.064±0.001	0.078±0.005	0.335±0.073	0.063±0.002	0.124±0.029	0.073±0.011	0.056±0.004	0.011
SRMSE_{mid}	0.042±0.001	0.053±0.003	0.180±0.056	0.043±0.002	0.068±0.011	0.044±0.005	0.039±0.002	0.006
SRMSE_{high}	0.380±0.017	0.795±0.001	0.801±0.005	0.792±0.001	0.850±0.033	0.792±0.001	0.791±0.000	0.791

Table 1: Pointwise (VRMSE) and Spectral (SRMSE) errors of models on Kolmogorov Flow.

dissipative PDEs, this uniform distribution is transported over time to a stationary distribution as energy is lost. If the time interval is small, we may intuitively believe that subsequent distributions are close; however, we seek to visualize and loosely quantify this difference.

Calculating statistical distances between high-dimensional empirical distributions is usually intractable. Despite this, there are several heuristics that are used. For example, Fréchet Inception Distance (FID) (Heusel et al., 2018) computes distances based on activations of a neural network; while this has no mathematical basis, it is useful to estimate distances between image distributions. We consider more general heuristics such as the Sliced Wasserstein Distance (SW), Classifier 2-Sample Test (C2ST), and Maximum Mean Discrepancy (MMD) (Bischoff et al., 2024). For example, C2ST trains a classifier to discriminate samples drawn from two distributions; if the classifier is perfect (100% accuracy), then the distributions can be viewed as farther than if the classifier cannot identify samples (50% accuracy). Additional information on heuristics are given in Appendix D.3.

We adopt a perspective where each timestep $\mathbf{u}_t \in \mathbb{R}^{n_x \times n_y}$ of a dataset is sampled from a different underlying distribution $\mathbf{u}_t \sim \rho_t$. While ρ_t is unknown, we have access to an empirical distribution with n samples $\rho_t^n = \{\mathbf{u}_t^1, \dots \mathbf{u}_t^n\}$, where n is the dataset size. Heuristics are computed using these empirical distributions as well as n samples $\mathbf{z} \in \mathbb{R}^{n_x \times n_y}$ drawn from a Gaussian $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$. For the RBC dataset, we calculate each heuristic D over time for either $D(\rho_t^n, \rho_{t+1}^n)$ or $D(\mathcal{N}(0, \mathbf{I}), \rho_{t+1}^n)$. Results are plotted and shown in Figure 2.

In general, the heuristics suggest that distances between subsequent timesteps are closer than distances between future timesteps and Gaussian noise. For the SW and MMD metrics, Gaussian noise more closely resembles physical states after convective mixing (around $t = 20$), where there is a transition between laminar and turbulent regimes. This aligns with our understanding of turbulence as a multiscale and chaotic phenomena. Conversely, subsequent states seem to be farther during and after this transition, as turbulence can cause large changes even in small time intervals. Interestingly, classifiers struggle to take advantage of this; subsequent timesteps have $\sim 50\%$ classification accuracy in turbulent regimes, as consistent changes between states become harder to learn.

Kolmogorov Flow We report the pointwise and spectral performance of models in Table 1, where the lowest errors are shaded and the second-lowest errors are lightly shaded. Errors are averaged over three seeds and standard deviations are reported. The number of function evaluations (NFEs) needed for a single prediction is shown, and autoencoder (AE) performance is given as a reference.

FNO performs well in Kolmogorov Flow, likely due to the low Reynolds number ($\sim 10^2$) and the consistent spectrum over time due to the sinusoidal forcing. Truncated sampling based on Tweedie's formula does not seem to work well, which suggests iterative sampling is still necessary for generative models. DDIM/DDPM models also under-perform, with high VRMSE and SRMSE metrics, yet EDM performance suggests that this is an issue of parameterizing the forward/reverse process rather than diffusion itself. Further simplifying the stochastic processes to linear interpolation also provides benefits, as demonstrated by flow matching. Lastly, due to the low Reynolds number, subsequent states are highly related, which allows stochastic interpolants to be accurate and statistically consistent. It achieves this performance with only 2 sampling steps, alongside flow matching. For all generative models, pointwise error is largely driven by autoregressive drift rather than reconstruction error of the autoencoder. Furthermore, in the highest frequency band, the spectral error of the autoencoder thresholds the spectral error of latent generative models.

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

Model:	FNO	DDPM	DDIM	EDM	TSM	FM	SI-E	SI-EM	AE
NFEs:	1	100	10	10	10	5	5	50	1
VRMSE	>10	0.765	0.675	0.681	8.961	0.733	0.665	0.726	0.027
SRMSE_{low}	0.357	0.405	0.612	0.363	1.113	0.323	0.346	0.296	0.086
SRMSE_{mid}	1.739	0.242	0.644	0.321	0.883	0.243	0.601	0.184	0.061
SRMSE_{high}	2.406	1.822	3.276	2.594	1.133	2.478	6.078	2.096	1.528

Table 2: Pointwise (VRMSE) and Spectral (SRMSE) errors of models on Rayleigh-Bénard.

a single set of experiments was run and errors are reported in Table 2. After training, stochastic interpolants are deterministically or stochastically sampled, using either the Euler method (-E) or Euler-Maruyama method (-EM) to solve the reverse ODE/SDE. Adding noise in the reverse SDE necessitates a finer discretization, which results in using more sampling steps.

The performance of FNO matches previous benchmarks from Ohana et al. (2025), where it is unstable across the trajectory. For generative models, Rayleigh-Bénard convection reveals an interesting trade-off: 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 50-100 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; 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 8.

Deterministically sampled stochastic interpolants achieve the lowest VRMSE, yet exhibit large spectral errors. To remedy this, noise can be added when sampling the stochastic interpolant by solving the reverse SDE, although this uses more sampling steps. Other generative models fall somewhere along this spectrum. DDPM and flow matching have good spectral accuracy, while EDM and DDIM have better pointwise accuracy, potentially from sub-sampling the probability flow SDE. In general, this task is very challenging; pointwise errors are high and no model can resolve the highest frequency band, with SRMSE values above 1 being largely meaningless. Additional plots of VRMSE/SRMSE over time for KM and RBC can be found in Appendix A.

Weather Forecasting Many commonly used 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 3. At this time horizon, stochastic interpolants tend to do well. One hypothesis is that, while complex, global weather systems evolve at multiple timescales, including low-frequency variability, which increases the large-scale predictability compared to turbulence or other chaotic PDEs. In this scenario, subsequent states separated by 6 hours tend to still be related, which stochastic interpolants can leverage during sampling. Not only does this produce more accurate forecasts, using 5 sampling steps also pushes the frontiers of efficiency for generative weather models, where it is currently typical to use around 20-40 steps Price et al. (2024); Couairon et al. (2024); Zhuang & Duraisamy (2025).

Within this timescale, we evaluate the probabilistic performance of generative models by calculating the CRPS and SSR of ensemble forecasts. 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 performance on 10-day forecasts. To generate different ensemble members, stochastic interpolants are sampled with a stochastic sampler by using the Euler-Maruyama method (-EM), which is run with 10 steps.

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	32.9	55.8	119	1.27	2.31	3.19	7.27	1.17	2.21	3.18	7.72	2.37	5.26	7.96	15.6	.87	1.28	1.38	2.08
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	.85	1.14	1.33	1.57
DDIM	7.3	15.6	26.5	54.9	.84	1.23	1.67	2.76	0.88	1.26	1.75	2.99	1.65	2.89	4.20	7.44	.78	1.01	1.18	1.40
EDM	7.2	15.4	26.6	57.6	0.83	1.21	1.65	2.87	0.86	1.24	1.76	3.11	1.64	2.88	4.21	7.71	.73	1.00	1.21	1.47
FM	6.9	15.0	26.3	57.0	0.79	1.15	1.60	2.81	0.84	1.19	1.72	3.06	1.59	2.80	4.13	7.63	.71	0.98	1.19	1.48
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	.61	0.87	1.10	1.43

Table 3: 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 both use 5 sampling steps.

Up to 30 days, stochastic interpolants shows 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. Additionally, the uncertainty of stochastic interpolants can be tuned by adjusting the noise coefficient of the EM sampler, which allows 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 integration. 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 emulations up to 10 years. DDIM, EDM, and SI-EM are run with 10 steps, and FM and SI-E are run with 5 steps. 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 integrated to climate (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 fine-tuned for an additional 20 epochs for bias evaluations. During fine-tuning, biases are calculated at each epoch, and the results for each model are reported in Table 4 for the best epoch.

Var:	z500	t2m	t850	u250	pr_6h	hus850
DDIM	15.7	0.55	0.45	1.91	0.125	0.437
EDM	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-E	8.81	0.43	0.42	1.22	0.110	0.257
SI-EM	10.7	0.51	0.48	1.03	0.069	0.187

Table 4: 10-year Climatological Biases.

In general, learning a consistent, long-term climate emulator with uniformly small biases globally from 6-hour prediction intervals is challenging (Wikner et al., in prep.). 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 or linear samplers, while higher frequency fields such as wind speed, precipitation, or humidity are better modeled with stochastic samplers. Random perturbations added by the EM sampler may help to resolve high-frequency features but can lead to inconsistent trends in smoother fields. This can allow SI models to use different samplers based on the variable of interest, which can be done without re-training.

At longer time horizons of up to 100 years, 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 globally averaged timeseries that are 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. Despite this, interpolants have the ability to match or exceed other models in matching long-term trends, depending on how they are sampled. Future work can perhaps mitigate model biases or find better training strategies to ensure long-term consistency.

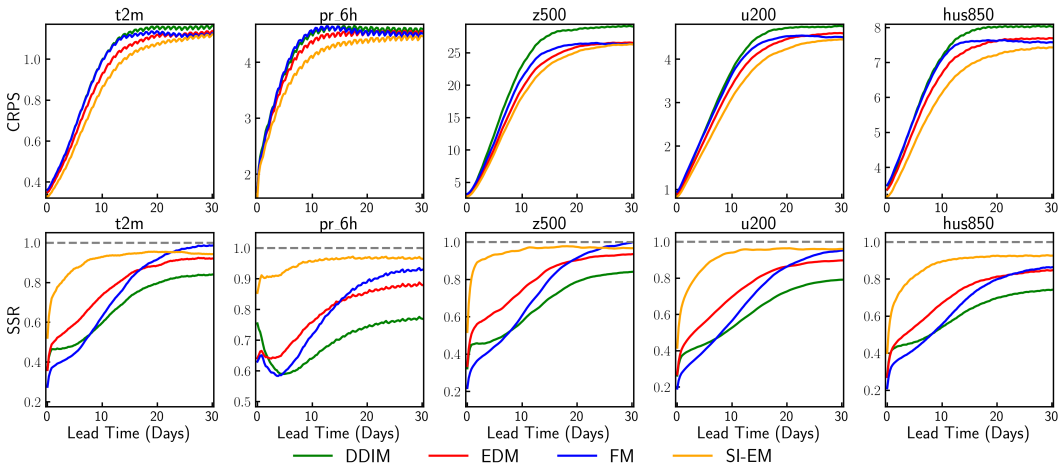

Figure 3: CRPS and SSR plots for the considered generative models over 30 days.

5 Discussion

Throughout the work, 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 is a remarkably effective way to train and sample a generative model. However, when the system becomes chaotic or models are rolled out for extremely long horizons, assumptions of coupled source and target distributions become less clear or less helpful.

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. 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, unimodality can be violated in highly chaotic systems, which can result in worse spectral performance when using an ODE sampler.

Finally, to better understand the benchmarked generative models, we provide a set of ablation studies in Appendix B. Consistent with prior evidence (Rozet et al., 2025) we show that the compression ratio of the autoencoder does not influence prediction error substantially. We also corroborate prior results (Li et al., 2025b) that probabilistic training is more effective than deterministic neural solvers in latent space. We additionally verify that generative models are more effective in latent space than in pixel space. Lastly, for stochastic interpolants we investigate the effect of the number of sampling steps and the amount of noise added. Adding noise to the interpolant encourages modes from the source and target distributions to mix, and produces a smoother trajectory through probability space. 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 stochastic process.

6 Related Works

Generative Models There are several foundational works on which generative models and stochastic interpolants are based. Diffusion models (Ho et al., 2020; Song et al., 2021), and its extensions to different samplers (Song et al., 2022; Karras et al., 2022) and flow matching (Lipman et al., 2023; Liu et al., 2022) propose a variety of methods to transport Gaussian noise to an arbitrary density, with the goal of sampling from a target distribution. Subsequent work has relaxed the requirement for a Gaussian prior distribution, allowing transport between arbitrary densities. These models fall under many frameworks, such as Schrödinger Bridges (Bortoli et al., 2023), Diffusion Bridges (Su et al., 2023; Zhou et al., 2023), Optimal Transport (Tong et al., 2024), or Stochastic Interpolants (Albergo et al., 2023).

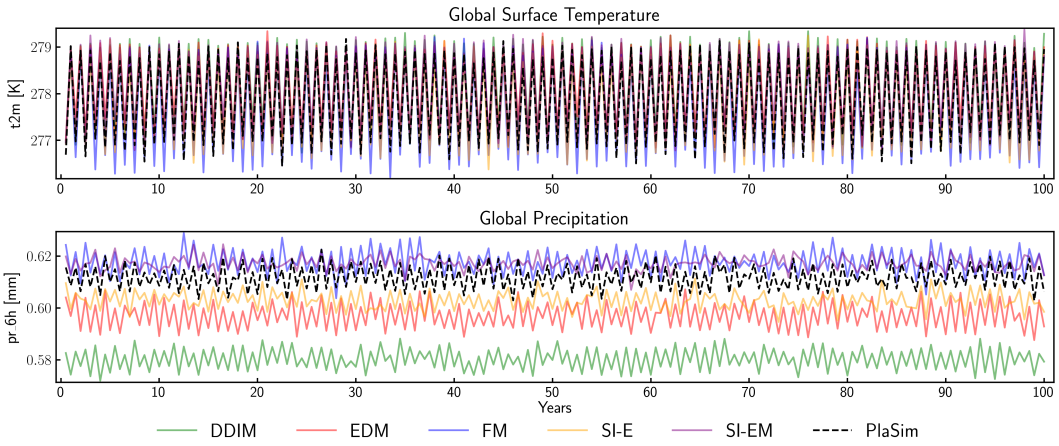

Figure 4: Global surface temperature and precipitation averaged for every 6 months over 100 years.

Applications to PDEs/Climate The use of deep learning to approximate PDE or climate systems is a diverse field, with approaches based on transformers (Pathak et al., 2022; Li et al., 2023; Bi et al., 2022; Li et al., 2024), large-scale pretraining (Zhou & Farimani, 2024; McCabe et al., 2024; Zhou et al., 2024; Nguyen et al., 2023), or physics-based priors (Verma et al., 2024; Zhou & Farimani, 2025). A subset of these works study the application of 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; Shysheya et al., 2024; Gao et al., 2024). Extensions of diffusion models have also been investigated to introduce physics-informed losses (Bastek et al., 2025), operate on meshes (Lino et al., 2025), or use text-conditioned generation (Zhou et al., 2025). Recent progress in flow matching has also been reflected in emulating PDEs, with works that report additional speed or accuracy benefits (Baldan et al., 2025; Utkarsh et al., 2025; Armegioiu et al., 2025; Shi et al., 2024). Similar applications are also prevalent in climate prediction, where diffusion and flow matching models are popular approaches (Zhuang & Duraisamy, 2025; Cachay et al., 2024; Couairon et al., 2024). Beyond prediction, climate downscaling and data assimilation are also relevant applications of diffusion models (Mardani et al., 2024; Gong et al., 2024; Gao et al., 2023; Andry et al., 2025; Aich et al., 2024; Tomasi et al., 2025; Brenowitz et al., 2025).

Interpolant or diffusion-bridge approaches in physical systems are less common. Previous works have addressed super-resolution and data assimilation in PDEs and climate (Bischoff & Deck, 2023; Schiødt et al., 2025; Chen et al., 2025b; Rout et al., 2025), motivated by their resemblance to denoising/inpainting tasks where stochastic interpolants have proven effective in computer vision. Chen et al. (2024) applies interpolants to forecast stochastic PDEs, and Mücke & Sanderse (2025) show promise that stochastic interpolants can perform well in modeling fluid problems. Expanding on these prior works, we present a more comprehensive benchmark of generative models across a diverse set of physical systems. We consider more challenging tasks, such as laminar-turbulent transitions and long-term climate emulation, and seek to understand when and how stochastic interpolants work. In doing so, we find stochastic interpolants are a strong baseline for modeling PDE and climate systems, while also providing insights into a variety of different generative models.

7 Conclusion

Modeling physical phenomena is challenging, as each system exhibits distinct dynamics, variables, and spatial or temporal scales. While generative models have shown promise for such tasks, not all are created equal. Even with the same architecture and training, changes in the loss objective and sampling can result in different deterministic and statistical performance. Despite this, stochastic interpolants can be a good baseline for generative models, motivated by the proximity of subsequent states in autoregressive prediction. We hope future work can continue to investigate this and advance the capabilities of generative models for physical systems, such as improving performance in turbulent systems or forecasting weather extremes (Sun et al., 2025; Wikner et al., in prep.).

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.

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.

Sebastian Bischoff, Alana Darcher, Michael Deistler, Richard Gao, Franziska Gerken, Manuel Gloeckler, Lisa Haxel, Jaivardhan Kapoor, Janne K Lappalainen, Jakob H Macke, Guy Moss, Matthijs Pals, Felix Pei, Rachel Rapp, A Erdem Sağtekin, Cornelius Schröder, Auguste Schulz, Zinovia Stefanidi, Shoji Toyota, Linda Ulmer, and Julius Vetter. A practical guide to sample-based statistical distances for evaluating generative models in science, 2024. URL https://arxiv.org/abs/2403.12636.

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

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.

Noah D Brenowitz, Tao Ge, Akshay Subramaniam, Peter Manshausen, Aayush Gupta, David M Hall, Morteza Mardani, Arash Vahdat, Karthik Kashinath, and Michael S Pritchard. Climate in a bottle: Towards a generative foundation model for the kilometer-scale global atmosphere. *arXiv preprint arXiv:2505.06474*, 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.

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, 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.

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.

Junchao Gong, Lei Bai, Peng Ye, Wanghan Xu, Na Liu, Jianhua Dai, Xiaokang Yang, and Wanli Ouyang. Cascast: Skillful high-resolution precipitation nowcasting via cascaded modelling, 2024. URL https://arxiv.org/abs/2402.04290.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2018. URL https://arxiv.org/abs/1706.08500.

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.

Amaury Lancelin, Alexander 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. in prep.

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.

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.

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, 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.

Nikolaj T. Mücke and Benjamin Sanderse. Physics-aware generative models for turbulent fluid flows through energy-consistent stochastic interpolants, 2025. URL https://arxiv.org/abs/2504.05852.

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.

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.

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.

Aliaksandra Shysheya, Cristiana Diaconu, Federico Bergamin, Paris Perdikaris, José Miguel Hernández-Lobato, Richard E. Turner, and Emile Mathieu. On conditional diffusion models for pde simulations, 2024. URL https://arxiv.org/abs/2410.16415.

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.

Y Qiang Sun, Pedram Hassanzadeh, Mohsen Zand, Ashesh Chattopadhyay, Jonathan Weare, and Dorian S Abbot. Can ai weather models predict out-of-distribution gray swan tropical cyclones? *Proceedings of the National Academy of Sciences*, 122(21):e2420914122, 2025.

E. Tomasi, G. Franch, and M. Cristoforetti. Can ai be enabled to perform dynamical downscaling? a latent diffusion model to mimic kilometer-scale cosmo5.0_clm9 simulations. *Geoscientific Model Development*, 18(6):2051–2078, 2025. doi: 10.5194/gmd-18-2051-2025. URL https://gmd.copernicus.org/articles/18/2051/2025/.

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.

Alexander Wikner, Amaury Lancelin, Troy Arcomano, Dhruvit Patel Karan Jakhar, Freddy Bouchet, and Pedram Hassanzadeh. Can ai climate emulators quantify the statistics of the rarest unseen weather extremes? in prep.

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

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. Neural functional: Learning function to scalar maps for neural pde surrogates, 2025. URL https://arxiv.org/abs/2505.13275.

Anthony Zhou, Cooper Lorsung, AmirPouya Hemmasian, and Amir Barati Farimani. Strategies for pretraining 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.

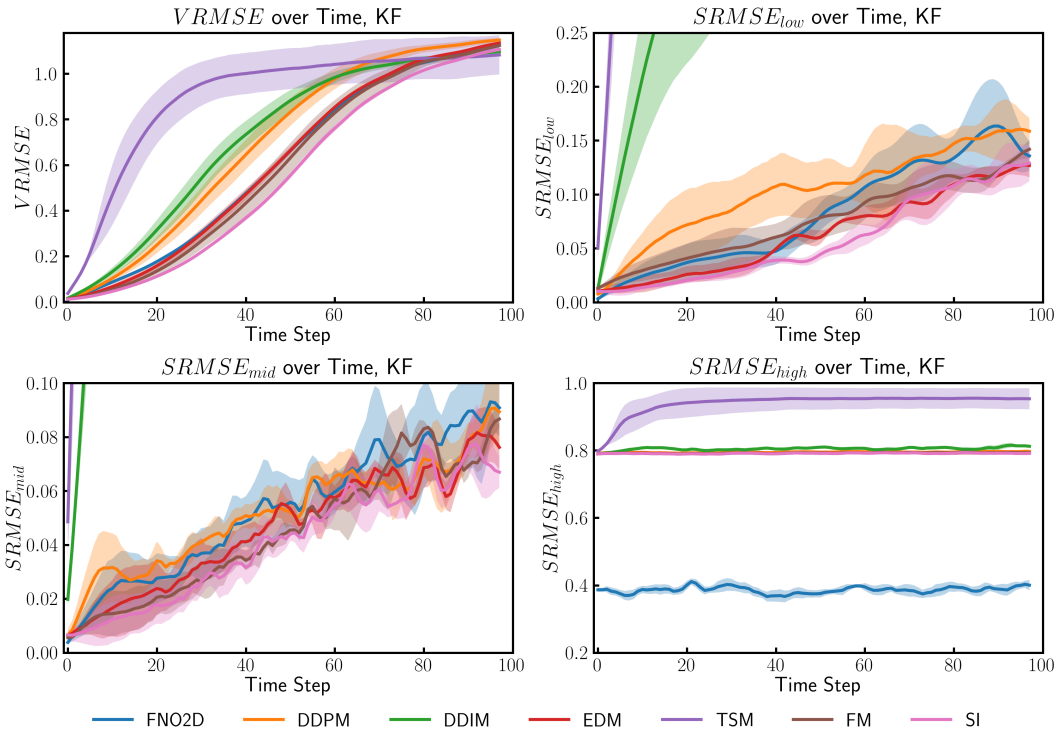

Figure 5: VRMSE/SRMSE for model predictions over time on the Kolmogorov Flow (KF) dataset. Each model is trained with three seeds, mean errors are plotted with standard deviations shaded.

A Supplementary Results

A.1 Error Plots

To visualize time-dependent trends in model errors, we plot the VRMSE and SRMSE over time for model predictions, averaged over all samples the validation set. Errors for Kolmogorov Flow are plotted in Figure 5 and errors for Rayleigh-Bénard Convection are plotted in Figure 6.

Error trends for Kolmogorov Flow tend to be more clear. Stochastic interpolants consistently have lower VRMSE across the prediction horizon, and all models accumulate error near the middle of the trajectory. When autoregressive drift sufficiently shifts the input distribution, VRMSE growth tends to taper off as predictions reach a steady-state, albeit with mostly incorrect predictions. At the low- and mid-frequency bands, stochastic interpolants also tend to have lower SRMSE across the trajectory. The mid-band SRMSE tends to oscillate more, as higher frequency features tend to form and dissipate more quickly than lower frequency features in Kolmogorov Flow. Lastly, at the high-frequency band, the performance of generative models is thresholded by the SRMSE of the autoencoder (~ 0.8), while FNO can have a lower SRMSE. In general, stochastic interpolants, flow matching, and EDM tend to perform well on this dataset, with good spectral and pointwise accuracy.

Examining time-dependent errors for RBC can also reveal insights. All models rapidly accumulate pointwise errors during turbulent mixing ($\sim t = 20$) and reach steady state during subsequent dissipation. In this regime, all models have have decorrelated from the true trajectory and based on our observations, pointwise accuracy tends to be achieved by smoother fields. In the low-frequency band, we see a spike in error as the fluid undergoes mixing. Despite this, most generative models can recover low-frequency features after mixing. Similar observations can also be made for the mid-frequency SRMSE, however, accurately capturing features after mixing becomes more difficult. At the high-frequency band, most models reach errors above 1 after $t = 20$ and stay there.

These plots shed more insight into tradeoffs between point-wise and spectral accuracy in chaotic systems. DDPM has low spectral error yet has high point-wise error; using a different sampler

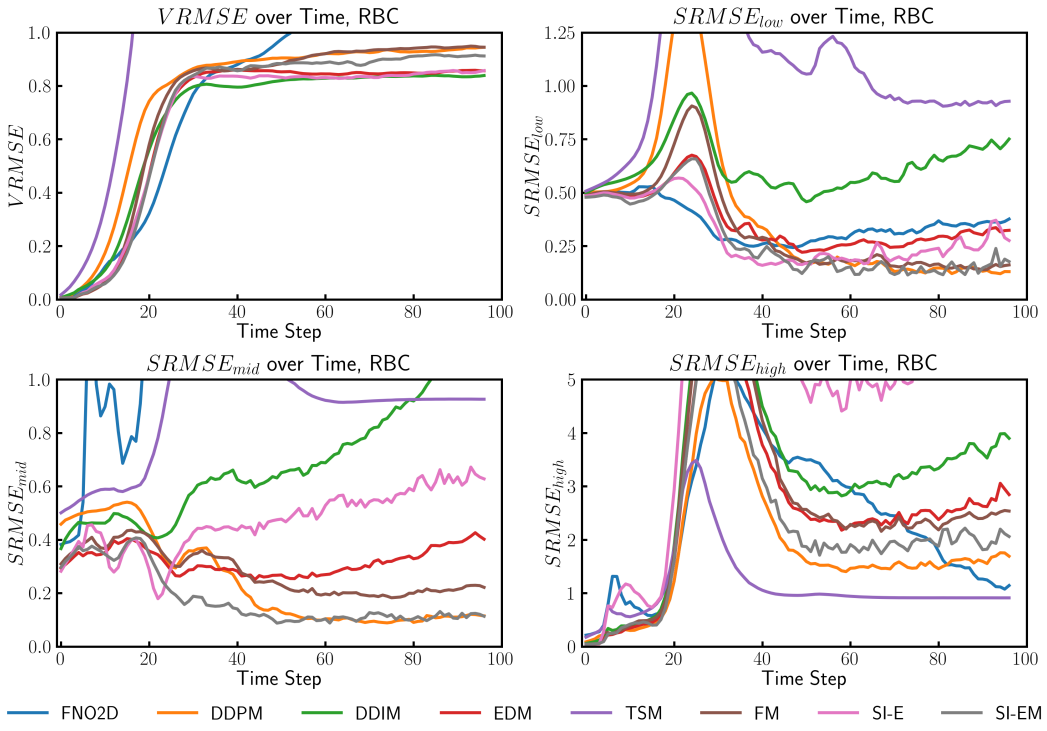

Figure 6: VRMSE/SRMSE for model predictions over time on the Rayleigh-Bénard Convection (RBC) dataset. In general, there is an increase in errors at $\sim t = 20$ as fluid layers mix.

with DDIM achieves low point-wise error but with high spectral errors. We also observe a similar phenomenon when training stochastic interpolants and using either an ODE or SDE based sampler. Perhaps it is still an open question as to what the desired behavior should be in turbulence modeling or if we can train models to accomplish both point-wise and spectral accuracy.

A.2 Visualizations

To qualitatively evaluate model performance, we provide a set of visualizations of model performance on Kolmogorov Flow (KF) and Rayleigh-Bénard Convection (RBC) in Figures 7 and 8. KF predictions are shown at $t = 50$, when model predictions begin to decorrelate. RBC predictions are shown at $t = 18$ and $t = 45$, which is before and during turbulent mixing. In the turbulent regime, the effects of SDE-based samplers is more clear. DDPM and SI-EM are able to capture more high-frequncy features, despite having larger pointwise errors. Interestingly, SI-E can roughly model the correct position and size of plumes, although the predictions are smoothed.

Additionally, medium-range forecasts using the PlaSim dataset are plotted in Figure 9 for different models, variables, and lead times. At this length scale and time horizon, differences between models are challenging to distinguish, although they exist. To visualize long-horizon consistency, we look at the zonally-averaged power spectrum of model predictions after 100 years, since this lead time is well beyond the limit for pointwise consistency. After being rolled out to 100 years, we plot the zonally-averaged power spectrum of different model predictions as well as the ground truth in Figure 10. At this timescale, the considered generative models seem to produce similar spectra, although their biases may be different (Figure 4). Although no model is the best, the spectrum of generative models remains largely accurate and remains stable over 100 years, which is a promising sign.

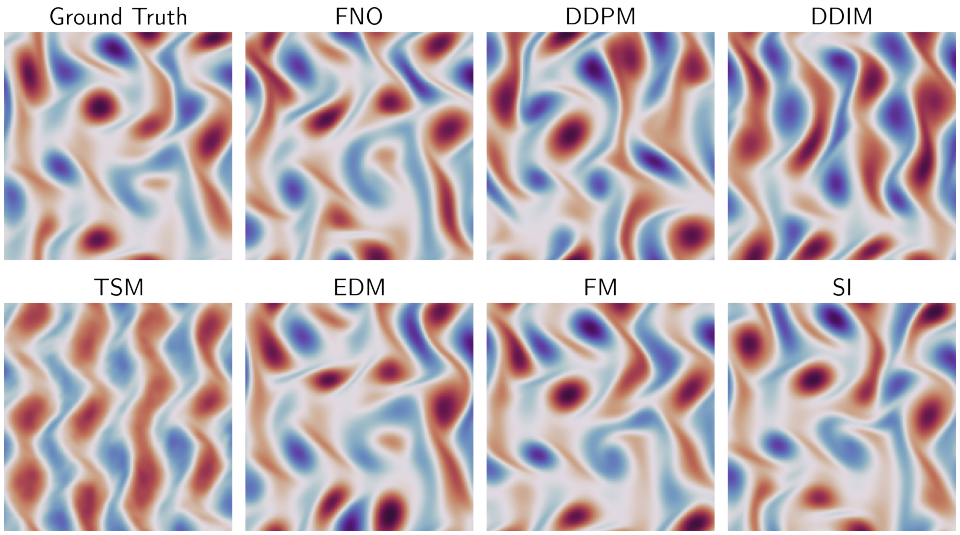

Figure 7: Model predictions of Kolmogorov Flow at $t = 50$, shown with the ground truth. Qualitatively, stochastic interpolants seem to capture most of the relevant features, although all models start to de-correlate at this timestep.

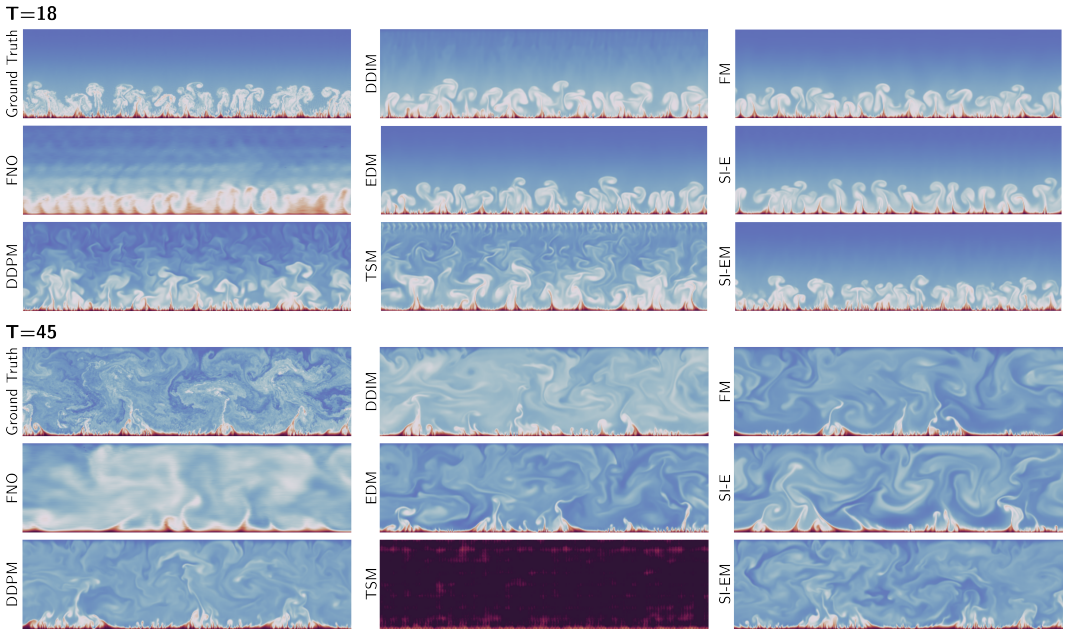

Figure 8: Model predictions of Rayleigh-Bénard Convection at $t = 18$ and $t = 45$, shown with the ground truth. In the laminar regime, most models can model initial mixing. After mixing, the effects of SDE-based samplers become more clear. DDIM/EDM are noticeably smoother than DDPM, likewise, FM/SI-E are smoother than SI-EM. Qualitatively, SI-EM seems to model the most detail, however SI-E seems to approximately capture the size and location of plumes.

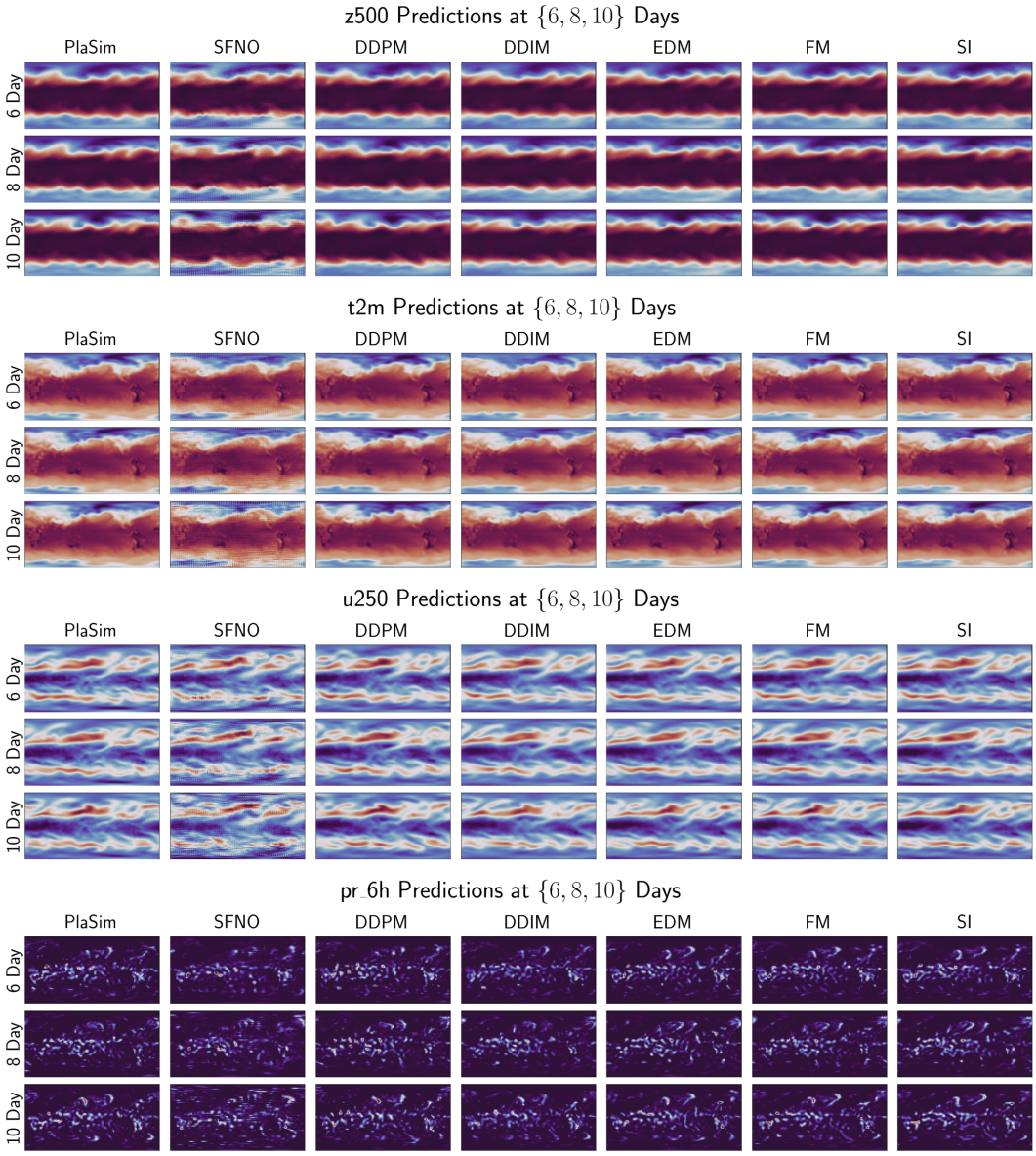

Figure 9: Model predictions of z500, t2m, u250, and pr_6h at lead times of $\{6, 8, 10\}$ days. At a coarse scale 128×64, latent generative models tend to work well for medium-range weather forecasting.

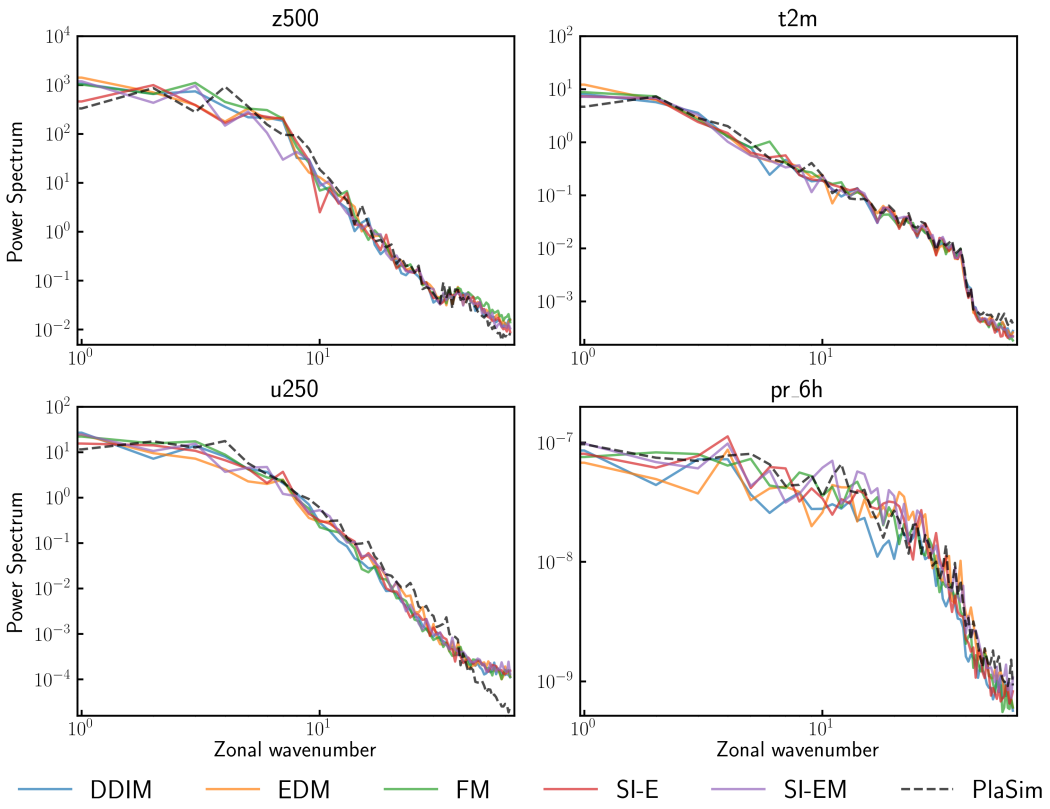

Figure 10: Zonally-averaged power spectra for different models and weather variables for predictions at a 100 year lead time. Despite having different biases, the considered generative models have consistent spectra and remain stable at long horizons.

B 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 5. 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 (160×160), a patch size of 8×8 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 autoregressive drift is the primary contributor to large errors in pixel space models.

Metric	FM_{pixel}	FM_{latent}	SI_{pixel}	SI_{latent}
NRMSE	0.812	0.649	0.966	0.609

Table 5: Comparison of pixel and latent space generative models on Kolmogorov Flow.

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 6. 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).

÷	AE	FM	SI
64	0.0252	0.619	0.605
256	0.0457	0.649	0.622

Table 6: Comparison of NRMSE for models using different compression ratios (÷) on Rayleigh-Bénard Convection. Reconstruction error is reported for the autoencoder (AE), while rollout error is reported for the generative models.

Deterministic/Probabilistic Latent Models We compare training a deterministic, latent neural solver (LNS) (Li et al., 2025a) against latent generative models in Table 7. LNS is trained to regress future latent states with an MSE loss, rather than a denoising loss. During inference, LNS makes a single future prediction, whereas generative models need to iteratively sample future states. Similar to prior works, we find that probabilistic training is more beneficial than deterministic models in latent space, although the benefit is not as large as previously reported (Rozet et al., 2025).

Model:	LNS	FM	SI
NFEs:	1	2	2
NRMSE	0.623	0.570	0.548

Table 7: Comparison of latent neraul solver (LNS) and generative models on Kolmogorov Flow.

Number of Sampling Steps We compare the effect of using different samplers and numbers of sampling steps for stochastic interpolant models in Table 8. After training on the Kolmogorov Flow dataset, models are either sampled with an Euler sampler or Euler-Maruyama sampler to solve the probability flow ODE or SDE. In general, we find that ODE-based samplers require fewer steps to obtain good performance. Furthermore, performance tends to increase consistently with more sampling steps, although these performance gains will saturate at some point.

Sampler:	Euler			Euler-Maruyama		
NFEs:	2	5	10	10	20	50
NRMSE	0.560	0.548	0.537	0.555	0.535	0.534

Table 8: Comparison of different samplers and number of sampling steps for SI models on Kolmogorov Flow.

Noise Coefficients for Interpolants The forward and reverse processes for stochastic interpolants are set by Equation 2, or $x(t) = \alpha(t)x_0 + \beta(t)x_1 + \gamma(t)z$. We define $\gamma(t) = \sigma(1-t)\sqrt{t}$ as the noise coefficient, where σ 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 σ and are sampled with an Euler sampler using 2 steps. In general, σ can be tuned to find an optimal amount of noise for the stochastic process.

σ:	0.1	0.5	1	3
NRMSE	0.702	0.632	0.609	0.803

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

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 accumulated precipitation (pr_12h)	geopotential (zg)	sea ice cover (sic)	
6h accumulated precipitation (pr_6h)		sea surface temperature (sst)	
air temperature 2m (t2m)			
surface temperature (ts)			

Table 10: Climate variables grouped into surface, atmospheric, and forcing variables. Additionally, the 13 pressure levels are reported.

C Dataset Information

Kolmogorov Flow In vorticity form, Kolmogorov Flow can be described by the PDE (Koehler et al., 2024):

$$\frac{\partial \omega}{\partial t} = -b\left(\begin{bmatrix}1\\-1\end{bmatrix} \odot \nabla(\Delta^{-1}\omega)\right)\cdot\nabla\omega + \nu\nabla\cdot\nabla\omega + \lambda\omega - k\cos(k\frac{2\pi}{L}y) \tag{5}$$

The leftmost term represents vorticity convection, controlled by the coefficient b. Diffusion is controlled by the viscosity ν and a drag term $\lambda\omega$ is introduced. Lastly, a sinusoidal forcing term is introduced, controlled by the magnitude k. Coefficients are kept constant throughout data generation. Initial conditions are uniformly sampled from a truncated Fourier series. While not straightforward to write in 2D/3D, in 1D the series is written as:

$$\omega_0 = \sum_{k=1}^{5} a_k \sin(k\frac{2\pi}{L}x + \phi_k) \tag{6}$$

where L is the length of the domain and terms $a_k \in [-1, 1]$ and $\phi_k \in [0, 2\pi]$ are uniformly sampled. This results in a uniform distribution for the initial states of the system, which does not dissipate over time due to the sinusoidal forcing.

Rayleigh-Bénard Convection The equations for Rayleigh-Bénard Convection are governed by a buoyancy and Navier-Stokes equation (Ohana et al., 2025):

$$\frac{\partial b}{\partial t} - \kappa\Delta b = -u\cdot\nabla b \tag{7}$$

$$\frac{\partial u}{\partial t} - \nu u + \nabla p - b\mathbf{e}_z = -u\cdot\nabla u \tag{8}$$

The thermal diffusivity κ and viscosity ν are determined by the Rayleigh and Prandtl numbers:

$$\kappa = (\text{Rayleigh}\times\text{Prandtl})^{-\frac{1}{2}}, \quad \nu = \left(\frac{\text{Rayleigh}}{\text{Prandtl}}\right)^{-\frac{1}{2}} \tag{9}$$

Rayleigh and Prandtl numbers are varied throughout the training and validation data. In particular, 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 δb_0 is sampled from $\{0.2, 0.4, 0.6, 0.8, 1.0\}$ and ϵ 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. Therefore, the initial conditions can be approximated by a categorical distribution, but as the system evolves, each sample quickly diverges and produces unique trajectories.

PlaSim Simulations are solved using PlaSim, which assumes a set of governing equations for planetary climate based on the conservation of mass, momentum, and energy. Additionally, many variable-specific equations are used. A full description of the climate variables and the pressure levels used in the climate simulation dataset is given in Table 10. There are 8 surface variables and 5 atmospheric variables at 13 pressure levels, resulting in 73 prognostic variables. Additionally, 6 constant or yearly constant forcing variables are included as extra inputs.

	Training Objective	Sampling Procedure
DDPM	$\mathcal{L} = \left\|\epsilon - \epsilon_\theta\left(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\,\epsilon,\ t\right)\right\|^2$	$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\,\epsilon_\theta(x_t, t)\right) + \sigma_t z$
DDIM	$\mathcal{L} = \left\|\epsilon - \epsilon_\theta\left(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\,\epsilon,\ t\right)\right\|^2$	$x_{t-1} = \sqrt{\alpha_{t-1}}\left(\frac{x_t - \sqrt{1-\alpha_t}\epsilon_\theta(x_t,t)}{\sqrt{\alpha_t}}\right) + \sqrt{1-\alpha_{t-1}-\sigma_t^2}\cdot\epsilon_\theta(x_t,t) + \sigma_t z$
EDM	$\mathcal{L} = \left\|D_\theta(x+z,\sigma) - x\right\|^2$	$x_{t+\Delta t} = x_t + \frac{\dot{\sigma}(t)}{\sigma(t)}\left(x - D_\theta(x,\sigma(t))\right)\Delta t$ (Euler)
TSM	$\mathcal{L} = \left\|\epsilon - \epsilon_\theta\left(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\,\epsilon,\ t\right)\right\|^2$	$x_0 = (x_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta(x_t,t))/\sqrt{\bar{\alpha}_t}$ (1-step)
FM	$\mathcal{L} = \left\|(x-z) - v_\theta((1-t)z + tx, t)\right\|^2$	$x_{t+\Delta t} = x_t + v_\theta(x_t,t)\Delta t,\ x_0 \sim \mathcal{N}(0, I)$ (Euler)
SI	$\mathcal{L} = \left\|(\dot{I}(x_0,x_1,t) + \dot{\gamma}(t)z) - b_\theta(I(x_0,x_1,t) + \gamma(t)z, t)\right\|^2$	$x_{t+\Delta t} = x_t + b_\theta(x_t,t)\Delta t,\ x_0 \sim \rho_0$ (Euler)

Table 11: Training and sampling for diffusion, flow matching, and stochastic interpolant frameworks.

D Additional Methods

D.1 Models

Generative Models DDPM, DDIM, TSM, and EDM models use some variant of the stochastic process:

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}z, \quad z \sim \mathcal{N}(0, \mathbf{I}) \tag{10}$$

which noises data x_0 over some time $t \in [1, T]$. While x_t approaches z as $t \to \infty$, this does not happen in finite time. This can cause inconsistencies when not using enough timesteps, and indeed, DDPM/DDIM models usually train with a fine discretization $T = [100, 1000]$ and require careful choice of noise schedule α_t. To remedy this, flow matching uses the following stochastic process:

$$x_t = tx_0 + (1-t)z, \quad z \sim \mathcal{N}(0, \mathbf{I}) \tag{11}$$

where $t \in [0, 1]$. This process is exact at the endpoints $t = \{0, 1\}$, where x_0 is considered to be fully noised and x_1 is considered to be denoised. Under a linear choice of $\alpha(t), \beta(t)$, stochastic interpolants use the same stochastic process if the source distribution x_0 is chosen to be a Gaussian. However, admitting arbitrary source and target distributions allows stochastic interpolants to use the stochastic process:

$$x_t = (1-t)x_0 + tx_1 + \sigma(1-t)\sqrt{t}z, \quad z \sim \mathcal{N}(0, \mathbf{I}) \tag{12}$$

where we make the appropriate choices for $\alpha(t), \beta(t), \gamma(t)$. In practice, to train generative models to learn and sample these stochastic processes, we use training objectives and sampling algorithms detailed in Table 11.

We make a few implementation choices to stabilize the training and sampling of stochastic interpolants. We find that antithetic sampling (Albergo et al., 2023) helps to reduce the variance of the training loss and improves model performance. When $\dot{\gamma}(t)$ is singular, the variance of the loss can be infinite at the endpoints as $t \to 0$ or $t \to 1$. Antithetic sampling combines loss functions for the two stochastic processes $x_t^+ = I(x_0, x_1, t) + \gamma(t)z$ and $x_t^- = I(x_0, x_1, t) - \gamma(t)z$ to jointly learn the drift for both x_t^+ and x_t^-. This results in a finite variance as $t \to 0$ or $t \to 1$. Furthermore, following Chen et al. (2024), the first sampling step for the EM sampler is analytically computed to avoid potential numerical singularities in the probability flow SDE:

$$x_{\Delta t} = x_0 + \Delta t b_\theta(x_0, 0) + \sqrt{\Delta t}\sigma(1-t)z \tag{13}$$

Model Architectures For a given task (KM, RBC, Climate), the architecture is kept constant across all autoencoders and diffusion backbones. Model sizes for KM are 21M for FNO, 20.5M for AE, and 57.9M for DiT backbones. Model sizes for RBC are 68.4M for FNO, 57.3M for AE, and 232M for DiT backbones. Model sizes for PlaSim are 218M for the SFNO, 89.5M for AE, and 313M for DiT backbones.

For autoencoders, we rely on convolution to process and downsample/upsample inputs. At each layer a standard Residual block processes inputs; at downsample/upsample layers PixelShuffle is

used. Latent vectors are constrained either with a saturation function $z = \frac{z}{\sqrt{1+z^2/b^2}}$, where $b = 5$, or a small KL regularization loss.

For the DiT backbone, the diffusion timestep t is passed into the model with adaptive layer norm (AdaLN) after sinusoidal embedding. For datasets with extra scalar information (Rayleigh/Prandtl number, day of year/hour of day), it is added to the timestep embedding after sinusoidal embedding. For additional fields (PlaSim forcing variables), they are embedded and passed into the model with cross attention. Furthermore, every generative model is conditional since it is provided information about the current state to sample a future state. To facilitate this, the noised state is concatenated along the channel dimension to the current state as input to the DiT.

D.2 METRICS

Variance-Scaled RMSE 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:

$$VRMSE(\mathbf{u}, \mathbf{u}_\theta) = \frac{1}{n_t} \sum_{t=1}^{n_t} \frac{||\mathbf{u}_\theta(t) - \mathbf{u}(t)||_2}{||\mathbf{u}(t) - \bar{\mathbf{u}}(t)||_2 + \epsilon}, \quad ||\mathbf{u}||_2 = \sqrt{\frac{1}{n_x n_y} \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} |\mathbf{u}(i,j)|^2} \tag{14}$$

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. The metric is more representative when nonnegative fields are present, as the more common Normalized RMSE (NRMSE) or Relative L2 Error tends to down-weight these channels. Additionally, predicting the mean field will result in $VRMSE(u, \bar{u}) \approx 1$, which is a useful interpretation.

Spectral RMSE 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. The SRMSE between the true and predicted spectra (p, p_θ) for each bin is then calculated as:

$$SRMSE(\mathbf{u}, \mathbf{u}_\theta) = \frac{1}{n_t} \sum_{t=1}^{n_t} \sqrt{1 - \frac{p(t)}{p_\theta(t)}} \tag{15}$$

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

Latitude-Weighted RMSE 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):

$$lRMSE = \sqrt{\frac{1}{n_{lat} n_{lon}} \sum_{i=1}^{n_{lat}} \sum_{j=1}^{n_{lon}} w(i)(f_{tl}(i,j) - o_{tl}(i,j))^2}, \quad w(i) = \frac{\sin\theta_i^u - \sin\theta_i^l}{\frac{1}{I}\sum_{i=1}^{I}(\sin\theta_i^u - \sin\theta_i^l)} \tag{16}$$

For latitude weights $w(i)$, i denotes the index of the discretized latitude, and θ_i^u and θ_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.

Climatological Biases Climatological biases involve averaging over a long rollout to evaluate the consistency of a climate emulator. In particular, the 10-year bias is calculated for each variable and pressure level by:

$$\text{Bias}_{\text{10-year}} = lRMSE(f_{avg}, o_{avg}), \quad f_{avg} = \frac{1}{n_t} \sum_{t=1}^{n_t} f_l(t) \tag{17}$$

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.

Continuous Ranked Probability Score 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):

$$lMAE = \frac{1}{n_{lat} n_{lon}} \sum_{i=1}^{n_{lat}} \sum_{j=1}^{n_{lon}} w(i) |f_{tl}(i,j) - o_{tl}(i,j)| \tag{18}$$

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

$$CRPS_{tl} = \frac{1}{M} \sum_{m=1}^{M} lMAE(f_{tl}^{(m)}, o_{tl}) - \frac{1}{2M(M-1)} \sum_{m=1}^{M} \sum_{n=1}^{N} lMAE(f_{tl}^{(m)}, f_{tl}^{(n)}) \tag{19}$$

Intuitively, 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.

Spread-Skill Ratio The spread-skill ratio (SSR) is the ratio of the ensemble spread to the ensemble skill. The spread is calculated for a given time, level, and variable as the square root of the ensemble variance:

$$\text{Spread}_{tl} = \sqrt{\sum_{i=1}^{n_{lat}} \sum_{j=1}^{n_{lon}} w(i) var_m(f_{tl}^{(m)}(i,j))} \tag{20}$$

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

$$\bar{f}_{tl} = \frac{1}{M} \sum_{m=1}^{M} f_{tl}^{(m)}, \quad \text{Skill}_{tl} = lRMSE(\bar{f}_{tl}, o_{tl}) \tag{21}$$

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:

$$SSR_{tl} = \frac{\text{Spread}_{tl}}{\text{Skill}_{tl}} \tag{22}$$

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.

D.3 Distance Heuristics

Background Considering each timestep of a PDE/climate trajectory as a distribution is not a common perspective, although it is often implicitly assumed when applying generative models to these emulation tasks. As such, we seek to build some intuition on this perspective through a set of visualizations, in Figures 11 and 12.

We use t-SNE to visualize samples from the Kolmogorov Flow or Rayleigh-Bénard Convection datasets. This is purely for visualization and intuition, no claims about distances or distributions can be made based on the plots. Interestingly, t-SNE can portray initial distributions based on what we expect, since we know the distributions that are used to sample randomized initial conditions. Over time, the initial distribution may be transported over time based on the PDE, which we visualize both for all samples and a single trajectory. At each timestep, we don't know this true distribution or if it even exists, however, we have access to some of its samples. Can we quantify a distance between two subsequent, empirical distributions? Since we can sample a Gaussian, can we quantify a distance between a Gaussian and an empirical distribution at a given timestep?

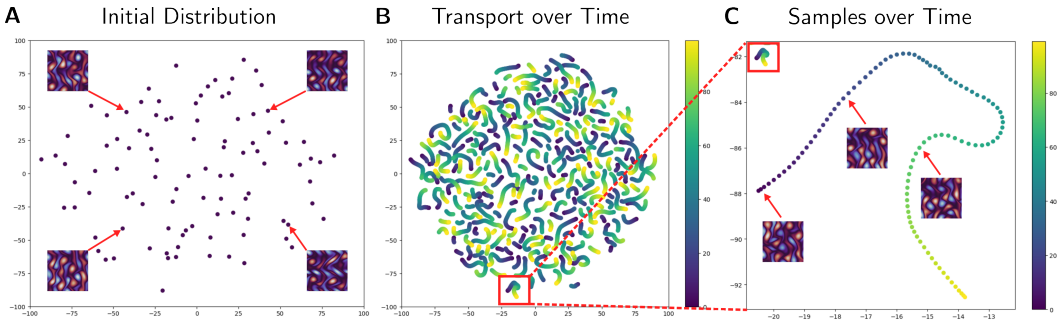

Figure 11: t-SNE visualizations for samples from Kolmogorov Flow. *Left, A:* We plot initial conditions $u(t = 0)$ from the entire dataset at after dimensionality reduction using t-SNE. As expected, the initial distribution is roughly uniform as Fourier coefficients used for initial conditions are uniformly sampled. *Middle, B:* Samples from the entire dataset are plotted, where each sample can be from a different initialization or timestep. Samples are colored by their timestep, where lighter colors are later timesteps. *Right, C:* A single trajectory is enlarged and visualized. There is some path that transports a single initial condition through time. As a whole, there may be a distribution at each timestep that is transported through time.

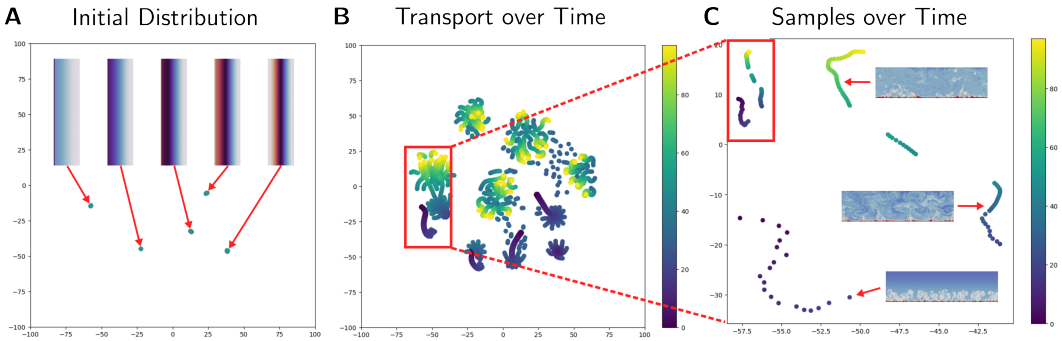

Figure 12: t-SNE visualizations for samples from Rayleigh-Bénard Convection. *Left, A:* We plot initial conditions $u(t = 0)$ from the entire dataset at after dimensionality reduction using t-SNE. As expected, the initial distribution is roughly categorical, as initial condition coefficients δb_0 are sampled from $\{0.2, 0.4, 0.6, 0.8, 1.0\}$. *Middle, B:* Samples from the entire dataset are plotted, where each sample can be from a different initialization or timestep. Samples are colored by their timestep, where lighter colors are later timesteps. Although samples are instantiated at similar initial conditions, chaotic mixing causes different instantiations to diverge. *Right, C:* A single trajectory is enlarged and visualized. There is some path that transports a single initial condition through time, although perhaps not easy to visualize. As a whole, there may be a distribution at each timestep that is transported through time.

Calculation Heuristics for calculating distances between distributions where we only have access to samples exist (Bischoff et al., 2024), although their quality may be impacted by many factors. One factor is the dimensionality of samples drawn from the considered distributions. High-dimensional distributions are more challenging to work with and heuristics are less accurate, therefore we first use a dimensionality reduction that preserves distances, based on the Johnson-Lindenstrauss Lemma.

Consider a set of n samples $\{x^1, x^2, \ldots, x^n\}$, $x^i \in \mathbb{R}^d$, where d can be very large. In our case, this can be a flattened sample $u(t)$. We define a reduced dimension $m < d$ and a random projection matrix $P \in \mathbb{R}^{m \times d}$, where each entry is sampled from a normal Gaussian $P_{ij} \sim \mathcal{N}(0, 1)$. P is additionally scaled to obtain $\hat{P} = \frac{1}{\sqrt{m}} P$. Consider a Euclidean distance on vectors $||x||_2 = \sqrt{\sum_{i=1}^{d} x_i^2}$. Given some error ϵ if $m = O(\frac{\log n}{\epsilon^2})$, then:

$$(1 - \epsilon)||x^i - x^j||_2^2 \leq ||\hat{P}x^i - \hat{P}x^j||_2^2 \leq (1 + \epsilon)||x^i - x^j||_2^2 \tag{23}$$

Fortunately, the choice of m does not depend on the original dimension d, which is beneficial if d is large. We can therefore leverage this to preserve pairwise Euclidean distances between x^i, x^j while reducing the dimensionality of the samples. This is assuming we have enough samples and choose modest error bound ϵ, which we set to $\epsilon = 0.2$. After projecting each sample to a lower dimension, we use implementations from Bischoff et al. (2024) to calculate the Sliced Wasserstein Distance, Classifier 2-Sample Test, and Maximum Mean Discrepancy between distributions, where each distribution is represented by n samples at a given timestep. Additionally, we perform 5-fold cross validation by taking 80% of the total samples at each timestep as the empirical distribution.