LEARNING INVARIANT MEASURES WITH PHYSICS SUPERVISION

Anonymous authors
Paper under double-blind review

ABSTRACT

Many fluid and plasma applications require samples of statistically stationary turbulence to estimate mean observables, fluctuations, and spatial correlations. However, obtaining these samples typically involves long-time evolution, either during sampling or when constructing training datasets for generative models. We introduce the Invariant Measure Generator (IMG), which learns a direct sampler of long-time stationary statistics using supervision from short physical evolutions. Its self-consistent conditional flow-matching objective uses evolved model samples as targets, providing tractable supervision for high-dimensional fields without requiring stationary training samples. Our analysis connects the self-consistent objective to distributional invariance. Experiments on Lorenz–63, the Kuramoto–Sivashinsky equation, and the Hasegawa–Wakatani (HW) system demonstrate accurate stationary statistics, including spectra and spatial increments in two-dimensional turbulence. In limited-data settings, the same objective improves data-trained samplers. Using an existing frozen neural operator to construct its targets improves probability density function (PDF), spectral, and increment statistics on HW without additional stationary training samples or numerical-solver calls during sampler training.

1 INTRODUCTION

Many scientific simulations seek the statistics of a saturated state rather than the details of its transient evolution. For example, statistically steady flow around an airfoil determines mean lift and drag (Jones et al., 2008), while saturated plasma turbulence determines heat and particle transport in magnetic-confinement fusion (Horton, 1999; Dimits et al., 2000). These tasks require representative states from a stationary distribution, or *invariant measure*, whose statistics remain unchanged under the governing dynamics.

Direct numerical simulation (DNS) and autoregressive neural solvers obtain such states by evolving initial conditions through a transient and collecting samples afterward (Foias et al., 2001; Li et al., 2021; 2022; Jiang et al., 2023; Schiff et al., 2024). Generative models can sample directly after training, but typically learn from datasets of saturated states produced by long simulations (Lienen et al., 2024; Lino et al., 2025; Galletti et al., 2026). Therefore, these approaches require long-time evolution either during sampling or when preparing training data. This raises a central question: can short-time physical evolution train a direct sampler of long-time stationary statistics without using stationary samples as training targets?

The invariance condition offers a route to this supervision, since evolving an invariant distribution under the system dynamics leaves it unchanged. Turning this condition into a scalable learning objective is challenging because directly estimating discrepancies between high-dimensional distributions can be expensive (Lipman et al., 2023; Fishman et al., 2025). Existing physics-supervised approaches learn stationary distributions through density-based or weak formulations (Tang et al., 2022; Xu et al., 2020; Cai et al., 2026). Density-based methods require probability densities and their derivatives, while weak formulations avoid these evaluations but still have mainly been demonstrated in lower-dimensional settings. The challenge is to construct a tractable distributional training signal for turbulent fields with more than 10^5 state coordinates.

We introduce the Invariant Measure Generator (IMG), which learns stationary-state samplers from short-time physical supervision. At each update, IMG draws states from its current model, advances

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

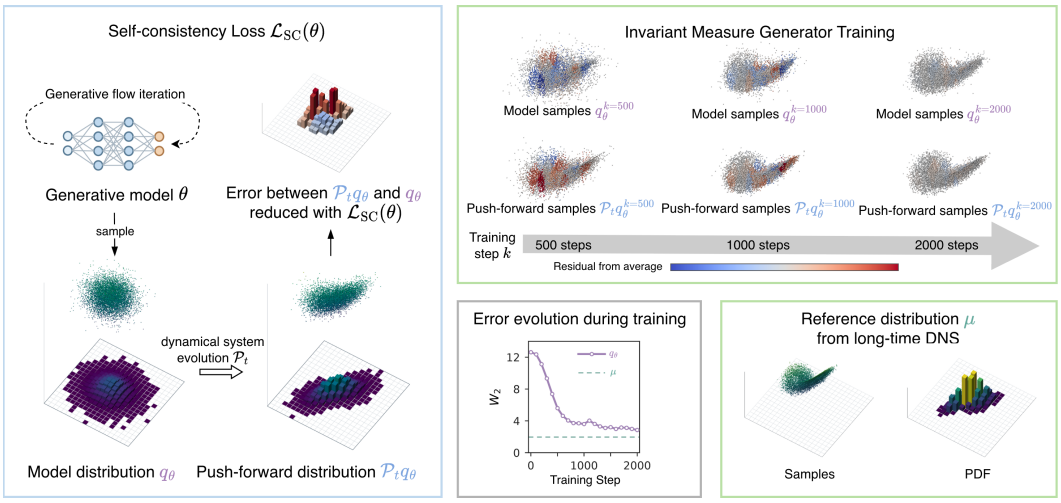


Figure 1: IMG learns stationary-state samplers by encouraging consistency between q_θ and its short-time push-forward $\mathcal{P}_{\delta t} q_\theta$. Generated states are advanced by the physical dynamics and used as frozen targets for conditional flow matching. The right panels illustrate training on Lorenz–63 and show the long-time DNS reference.

them over randomly sampled short physical times, and uses the evolved states as frozen targets for conditional flow matching. This supplies supervision without a precomputed dataset of saturated states or direct estimation of a high-dimensional distributional distance during training. The trained sampler can generate states without a physical burn-in stage.

We evaluate this approach on Lorenz–63 (L63), the Kuramoto–Sivashinsky (KS) equation, and the Hasegawa–Wakatani (HW) system, including two-dimensional turbulent fields with $131{,}072$ state coordinates after spatial discretization. We assess both agreement with DNS statistics and statistical changes under subsequent physical evolution. When a small stationary dataset is available, IMG also supplements conventional data-based flow matching. An existing neural operator can provide the short-time targets. Our HW results show improved probability density function (PDF), spectral, and increment statistics at the same stationary-data budget, without numerical-solver calls during the sampler's training.

We make three contributions. **(1)** We introduce a direct stationary-state sampler trained through short physical evolutions of its own samples, without a target-measure training dataset. **(2)** We develop a tractable self-consistent flow-matching objective and derive its connection to invariance for randomized physical durations. **(3)** We validate the method on chaotic and high-dimensional turbulent systems and show that a limited-data extension uses frozen neural operators to improve statistical accuracy without additional solver calls during sampler training.

2 Related Work

Neural operator methods typically learn short-time physical state evolution and require repeated model rollouts, such as the Fourier neural operator (FNO) (Kovachki et al., 2023; Li et al., 2021) and MeshGraphNet (Pfaff et al., 2021). These models may accumulate errors during long rollouts, which can distort long-time statistics (Li et al., 2022). Li et al. (2022) use a tailored loss to regularize long-term statistical behavior. Jiang et al. (2023) match distributions of summary statistics between model rollouts and training trajectories using a Sinkhorn divergence, an entropy-regularized optimal-transport loss. Dynamics Stable Learning by Invariant Measure (DySLIM) (Schiff et al., 2024) regularizes the maximum mean discrepancy (MMD) between state distributions before and after evolution. These methods learn evolution maps. Stationary samples are obtained by rolling out the learned dynamics. IMG instead trains a direct state sampler using short-time evolution as supervision.

Another stream of work directly operates in the state probability distribution space. TurbDiff (Lienen et al., 2024) learns a diffusion model from simulated saturated turbulence states. Lino et al. (2025) extend this approach with a graph neural network-based diffusion model. GyroFlow (Galletti et al., 2026) uses rectified flow matching in an autoencoder's latent space to generate saturated gyrokinetic states. These works aim at learning the invariant measure implicitly defined by training data from long-time system evolution in numerical simulations, rather than imposing the invariant measure conditions directly.

To learn the invariant measure directly without the need for samples from that distribution, Xu et al. (2020) construct training signals from the fixed point of the Fokker-Planck equation. This method parameterizes the probability density directly, so the training and sampling are less scalable in high-dimensional spaces. Tang et al. (2022) introduce adaptive deep density approximation using normalizing flows. The weak generative sampler avoids explicit density evaluation through a weak Fokker–Planck formulation (Cai et al., 2026). Li et al. (2023) learn time-dependent probability flows for mass-conserving partial differential equations (PDEs). At inference, their velocity-field parameterization integrates the learned flow over the physical time interval of interest, while their time-dependent-map parameterization evaluates the map directly. The weak generative sampler and this probability-flow approach have been demonstrated in 100 and 60 dimensions, respectively. IMG uses short physical evolutions to construct targets for self-consistent conditional flow matching, learning direct invariant-measure samplers for turbulent PDE fields with over 10^5 coordinates.

3 Background

3.1 Invariant Measures of Dynamical Systems

We consider a dynamical system, described by an ordinary differential equation (ODE) or a discretized partial differential equation (PDE): $\frac{\mathrm{d}u(t)}{\mathrm{d}t} = F(u(t))$, $u(t) \in \mathcal{X} \subseteq \mathbb{R}^d$, where $u(t)$ represents the system state and F specifies its dynamics. Let Φ^t denote the physical flow map, so that $u(t) = \Phi^t(u(0))$. If the initial state follows a distribution q, its distribution after time t is $\mathcal{P}_t q := (\Phi^t)_\sharp q$, where $\sharp$ denotes the pushforward of a probability measure. A probability measure μ is an invariant measure if

$$\mathcal{P}_t \mu = \mu \qquad \text{for all } t \geq 0. \tag{1}$$

On the invariant measure, states drawn from μ retain the same distribution under time evolution, although individual trajectories may continue to evolve. Equivalently, for every bounded continuous observable $a : \mathcal{X} \to \mathbb{R}$, $\mathbb{E}_{u\sim\mu}\big[a(\Phi^t(u))\big] = \mathbb{E}_{u\sim\mu}\big[a(u)\big]$. An invariant measure therefore describes a statistically stationary state distribution. For an ergodic invariant measure, expectations can also be estimated through time averages along typical trajectories, providing a basis for trajectory-based evaluation (Young, 1995).

3.2 Flow Matching Models

We use a flow-based generative model (Chen et al., 2018) to represent a state distribution q_θ on $\mathcal{X}$ that approximates the invariant measure. The parameters θ define a time-dependent velocity field $v_\theta : [0,1] \times \mathcal{X} \to \mathbb{R}^d$ and the ODE $\frac{\mathrm{d}u_\tau}{\mathrm{d}\tau} = v_\theta(\tau, u_\tau)$, $\tau \in [0,1]$, which transports samples u_0 from a simple base distribution π to samples $u_1 = \Psi_\theta(u_0)$. The induced model distribution is $q_\theta = (\Psi_\theta)_\sharp \pi$[1].

Conditional flow matching (CFM) trains v_θ using samples from a target distribution q (Lipman et al., 2023). Draw $u_0 \sim \pi$, $u_1 \sim q$, and the generative time $\tau \sim \mathcal{U}(0,1)$ independently, and define the linear interpolation $u_\tau = (1-\tau)u_0 + \tau u_1$. Lipman et al. (2023) give the conditional flow matching objective

$$\mathcal{L}_{\mathrm{CFM}}(\theta) = \mathbb{E}\left[\|v_\theta(\tau, u_\tau) - (u_1 - u_0)\|_2^2\right], \tag{2}$$

whose population minimizer matches the marginal velocity associated with the interpolation path from π to q. This framework has been scaled to high-dimensional problems (Galletti et al., 2026).

[1]Note that the generative time τ is distinct from the physical time t.

4 METHOD

Our goal is to learn a direct sampler q_θ of statistically stationary states from short physical evolutions. To construct its training objective, we start from the invariance residual of the model distribution q_θ, defined for any $t \geq 0$ as

$$R_t(q_\theta) := W_2(q_\theta, \mathcal{P}_t q_\theta)\,, \tag{3}$$

where W_2 is the Wasserstein-2 distance, $\mathcal{P}_t$ is the push-forward operator that evolves the state distribution. The invariance condition in Eq. 1 requires this residual to vanish for every t. Estimating and optimizing this distributional discrepancy can be costly in high dimensions. We instead construct a sample-based velocity-matching objective.

4.1 SELF-CONSISTENCY LOSS

We propose IMG with a tractable *self-consistency (SC) loss* $\mathcal{L}_{\mathrm{SC}}$ whose connection to the invariance residual follows from velocity matching.

The key idea is to compare two flows from the same prior, one ending at the model distribution and the other at its physical push-forward. Under suitable regularity assumptions, a small velocity mismatch keeps their endpoint distributions close. Following the flow matching (FM) formulation in Eq. 2, we consider a probability path p_τ from the prior π to the model distribution q_θ at generative time τ. $v_\theta(\tau, \cdot)$ is the velocity field that generates p_τ. Let $p_\tau^{\delta t}$ be a probability path from π to the evolved distribution $\mathcal{P}_{\delta t} q_\theta$, with marginal velocity $w_\tau^{\delta t}$. Condition on a physical duration δt, draw $\tau \sim \mathcal{U}(0,1)$ and then $u \sim p_\tau^{\delta t}$. Define the marginal velocity-matching loss and its residual bound by

$$\begin{aligned} \mathcal{L}_{\mathrm{FM}}^{\delta t}(\theta) &:= \mathbb{E}\left[\left\| v_\theta(\tau, u) - w_\tau^{\delta t}(u) \right\|_2^2\right], \\ R_{\delta t}(q_\theta)^2 &\leq C\, \mathcal{L}_{\mathrm{FM}}^{\delta t}(\theta). \end{aligned} \tag{4}$$

Under the regularity assumptions of Theorem 1, C depends on the Lipschitz bound of the model velocity. At $t = \delta t$, reducing the FM loss lowers an upper bound on the squared residual in Eq. 3, motivating optimization of the tractable CFM surrogate introduced below. Randomizing δt aims to avoid a distribution that returns to itself after one fixed duration by probing an interval of short evolutions. Under the assumptions of Theorem 1, averaging this mismatch bounds the invariance residual at every finite physical time via the semigroup property.

CFM makes velocity matching tractable using short-time evolutions of model samples as targets (Lipman et al., 2023). At the start of each update, we set $\bar{\theta} = \mathrm{stopgrad}(\theta)$ and use this frozen copy to generate targets. We optimize only θ and refresh the targets at the next update. For this fixed-target objective, $p_\tau^{\delta t}$ ends at $\mathcal{P}_{\delta t} q_{\bar{\theta}}$, and $w_\tau^{\delta t}$ denotes its marginal velocity. We write the corresponding loss as $\mathcal{L}_{\mathrm{FM}}^{\delta t}(\theta; \bar{\theta})$. At the start of the update, its value agrees with Eq. 4, while differentiation holds the target path fixed. Draw $\bar{u}_1 \sim q_{\bar{\theta}}$, $u_0 \sim \pi$, and the generative time $\tau \sim \mathcal{U}(0,1)$ independently. Independently draw the physical duration $\delta t \sim \mathcal{U}(0, \Delta T)$, where $\Delta T > 0$ is the maximum advancement time. Set $u_1 = \Phi^{\delta t}(\bar{u}_1)$ and $u_\tau = (1-\tau)u_0 + \tau u_1$. The self-consistency loss averages over these draws,

$$\mathcal{L}_{\mathrm{SC}}(\theta; \bar{\theta}) = \mathbb{E}\left[\left\| v_\theta(\tau, u_\tau) - (u_1 - u_0) \right\|_2^2\right], \tag{5}$$

where the generated targets and interpolation states are detached from the gradient computation. The theorem below relates this tractable objective to invariance for random physical durations with the target parameters fixed.

Algorithm 1 Training with the self-consistency loss

Require: Initial θ, prior π, maximum physical duration $\Delta T > 0$, batch size B, updates N, optimizer

for $k = 1, \ldots, N$ **do**
  $\bar{\theta} \leftarrow \mathrm{stopgrad}(\theta)$
  for $i = 1, \ldots, B$ **do**
    Draw independently:
    $\bar{u}_0^{(i)}, u_0^{(i)} \overset{\mathrm{iid}}{\sim} \pi$,
    $\tau_i \sim \mathcal{U}(0,1)$,
    $\delta t_i \sim \mathcal{U}(0, \Delta T)$
    $\bar{u}_1^{(i)} \leftarrow \Psi_{\bar{\theta}}(\bar{u}_0^{(i)})$
    $u_1^{(i)} \leftarrow \mathrm{stopgrad}\Big(\Phi^{\delta t_i}(\bar{u}_1^{(i)})\Big)$
    $u_{\tau_i}^{(i)} \leftarrow (1-\tau_i) u_0^{(i)} + \tau_i u_1^{(i)}$
  end for
  Compute the loss:
  $\widehat{\mathcal{L}}_{\mathrm{SC}} \leftarrow \frac{1}{B} \sum_{i=1}^{B} \left\| v_\theta(\tau_i, u_{\tau_i}^{(i)}) - (u_1^{(i)} - u_0^{(i)}) \right\|_2^2$
  Update θ using the optimizer and $\nabla_\theta \widehat{\mathcal{L}}_{\mathrm{SC}}$
end for
return θ

Theorem 1 (Fixed-target gradient connection to invariance)**.** *Fix $\bar{\theta}$ and $\Delta T > 0$. Assume that the physical and generative flows, including those generated by $w^{\delta t}$, are well defined with finite second moments. Assume also that the physical flow has an exponential Lipschitz bound and that $v_\theta(\tau, \cdot)$ is uniformly Lipschitz in state, with bounds independent of θ. Let the expectations below be finite and differentiable under the integral. For the outer expectation below, draw $\delta t \sim \mathcal{U}(0, \Delta T)$. Each conditional loss averages over τ and u as in Eq. 4. The marginal path velocity is $w_\tau^{\delta t}(u) = \mathbb{E}[u_1 - u_0 \mid u_\tau = u, \delta t]$ (Lipman et al., 2023), using the endpoint draws defined above. For every $t > 0$, there is a finite coefficient $C_t > 0$, depending only on t, ΔT, and these Lipschitz bounds. Define the residual-bound loss*

$$\mathcal{L}^t_{\mathrm{bound}}(\theta; \bar{\theta}) := C_t \, \mathbb{E}\left[\mathcal{L}^{\delta t}_{\mathrm{FM}}(\theta; \bar{\theta})\right]. \tag{6}$$

Then

$$W_2^2(q_\theta, \mathcal{P}_t q_\theta) \le \mathcal{L}^t_{\mathrm{bound}}(\theta; \bar{\theta}), \tag{7}$$

$$\nabla_\theta \mathcal{L}^t_{\mathrm{bound}}(\theta; \bar{\theta}) = C_t \, \nabla_\theta \mathcal{L}_{\mathrm{SC}}(\theta; \bar{\theta}). \tag{8}$$

Randomized short evolutions thus control residuals at every finite physical time, although C_t can grow with t. For exact advancement, zero marginal velocity mismatch recovers invariance. Appendix A gives the proof, and Appendix A.1 includes advancement error. Algorithm 1 summarizes training. The maximum durations used in the experiments are specified in Appendix C.1. The schematic is shown in Figure 1.

Physical measure selection. Self-consistency permits multiple invariant measures, raising the question of which one IMG learns. We seek a physical measure whose averages describe long-time statistics for a positive-volume set of initial states. Our experiments produce non-collapsed distributions without added target jitter. Initialization and model parameterization may influence selection, and neural approximation errors may act as implicit perturbations. A possible theoretical connection is stochastic stability. For smooth systems with a transitive uniformly hyperbolic attractor, the Sinai–Ruelle–Bowen (SRB) measure describes physical statistics (Young, 2002). If it is unique and stochastically stable, stationary measures under suitable small random perturbations approach it as the perturbations vanish (Cowieson & Young, 2005). How initialization, parameterization, and approximation errors select a measure in IMG remains unresolved.

4.2 Self-consistency loss in few-data regime

When a small dataset $X = \{u^{(n)}\}_{n=1}^{N_X}$ from the invariant measure is available, we can combine data-based training with self-consistency. By drawing $u_1 \sim \mathrm{Unif}(X)$, the data-driven conditional flow matching objective is defined as in Eq. 2. We supplement this objective with the self-consistency loss in Eq. 5. Suppressing the fixed target parameters in the notation gives

$$\mathcal{L}_{\mathrm{mixed}}(\theta) = \mathcal{L}_{\mathrm{CFM}}(\theta) + \lambda \mathcal{L}_{\mathrm{SC}}(\theta), \qquad \lambda \ge 0. \tag{9}$$

The data term fits the available stationary samples, while the self-consistency term supplies additional supervision from the dynamics using model-generated states. The weight λ controls this contribution, with $\lambda = 0$ recovering standard data CFM training.

IMG-neural uses a frozen neural operator (NO) to form targets $u_1 = \widehat{\Phi}^{\delta t}(\bar{u}_1)$, with the same stop-gradient convention. Stepwise neural operators predict over short physical intervals. We interpolate linearly only when the requested time lies between consecutive prediction times, using the operator output directly at those times (Appendix C.1). This local approximation introduces temporal interpolation error in addition to operator prediction error. Appendix A.1 extends the physical-flow bound to include both errors on generated states. With an existing frozen operator, sampler training requires neural evaluations but no additional stationary samples or numerical-solver calls. Operator pretraining remains a separate cost. Section 5.2 compares solver and neural targets.

5 Results

In this section, we ask whether short-time physical supervision can learn samplers that reproduce long-time DNS statistics, whether generated distributions retain their statistics under physical evolution, and whether the same supervision improves learning from limited data.

Experimental setting. We use Lorenz–63 as a low-dimensional benchmark and study saturated turbulent states in the one-dimensional Kuramoto–Sivashinsky (KS) equation and the two-dimensional Hasegawa–Wakatani (HW) system. Both PDE systems use periodic domains, with states given by the scalar field u for KS and the vorticity–density pair (ζ, n) for HW. The HW electrostatic potential ϕ is recovered from vorticity through the Poisson equation. Our goal is to sample their statistically stationary state distributions.

To construct DNS references for both systems, we evolve independently initialized fields, discard the initial transient, and collect snapshots after saturation. Figure 2 illustrates this process for KS. Individual fields continue to evolve, while the ensemble discrepancy becomes small after burn-in. We retain KS snapshots over $t \in [100, 199]$ for evaluation. Equations, parameters, and numerical settings are given in Appendix B.1.

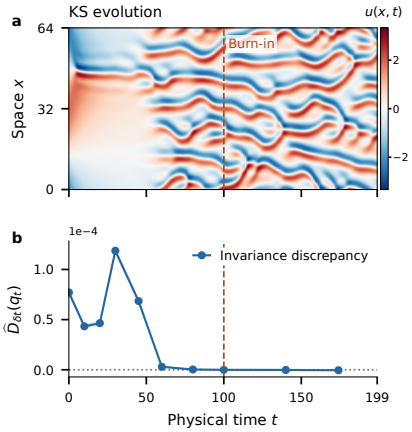


Figure 2: DNS sampling of saturated KS states. (a) A trajectory with the burn-in cutoff at $t = 100$ (dashed line). (b) Ensemble discrepancy before and after physical evolution, $D_{\delta t}(q_t) = \mathrm{MMD}_k^2(q_t, \mathcal{P}_{\delta t} q_t)/\delta t^2$, with $\delta t = 25$.

Physics-supervised training. IMG uses short-time evolution targets. Baselines include the weak generative sampler (WGS) (Cai et al., 2026) and, for L63, adaptive deep density approximation (ADDA) (Tang et al., 2022). Both use stationarity objectives without stationary training samples. Physics-supervised neural operators learn evolution maps from synthetic initial fields using equation residuals or short numerical-solver targets, as specified in Appendix C.1. The operator baselines use FNO or U-Net, an encoder–decoder with skip connections, and Gaussian random field (GRF) inputs. We also compare PI-DeepONets (Wang et al., 2021).

Data-assisted training. The data-assisted group includes DySLIM (Schiff et al., 2024), which adds invariant-measure regularization to trajectory-supervised dynamics learning. Data CFM is a generative baseline trained directly on the available saturated snapshots, following the data-based approach of turbulent-field generative models (Lienen et al., 2024; Galletti et al., 2026; Lino et al., 2025). IMG-augmented CFM uses the same architecture and stationary training data. The appendix details each baseline, its adaptation to our systems, and all data used for training or operator pretraining.

Evaluation. Within each comparison, all methods use the same DNS reference. For KS and HW, marginal PDF total variation (TV) measures discrepancies in field-value distributions, while spectrum errors assess how fluctuation power is distributed across spatial scales. For HW, spatial increment PDFs (Chevillard et al., 2006) and flatness (Ma et al., 2021) further probe multiscale fluctuations and intermittency. We also report coarse-field sliced Wasserstein-2 (SW_2) for KS and full three-dimensional empirical Wasserstein-2 for L63. HW sliced-Wasserstein results appear in Appendix E.1.

KS and HW use four and five evaluation repeats, respectively. Each KS repeat evaluates 4,096 generated fields. Each HW repeat compares 64 model fields with 64 DNS fields, with both ensembles varying across repeats. Data-free refers to the construction of training targets. Implementation and data protocols are given in Appendix C. Evaluation details and metric definitions appear in Appendices B.4 and B.2.

5.1 Learning without stationary training samples

We compare statistical accuracy across Lorenz–63, KS, and HW. The main table evaluates whether equation-based supervision can recover the DNS statistics as the state dimension increases. We report physics-supervised methods separately from trajectory- and snapshot-supervised baselines.

Figure 3 contrasts the sampling procedures and resulting HW structures. Autoregressive FNO and U-Net collect states from long physical-time rollouts after burn-in, whereas IMG maps independent latent draws to states through its generative flow. Appendix E.6 provides further field comparisons.

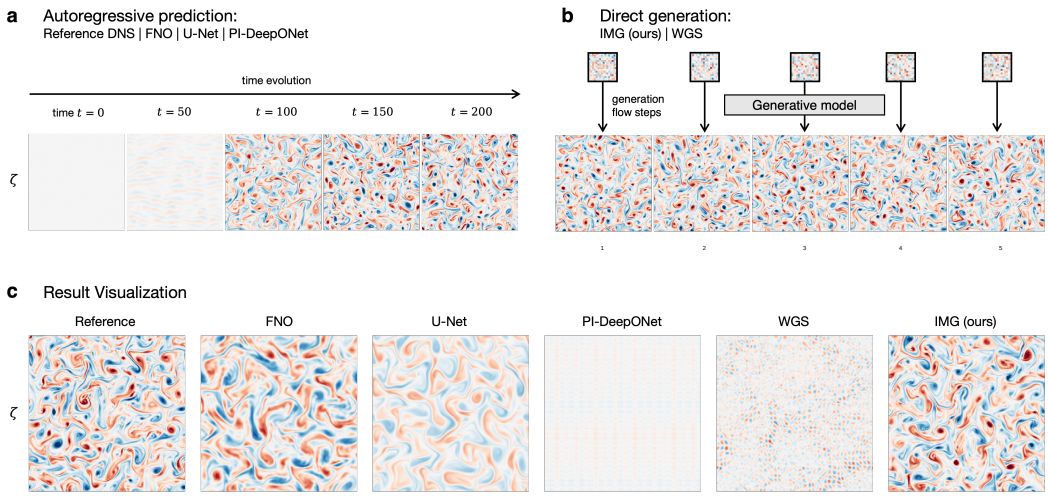


Figure 3: HW vorticity fields ζ. (a) Illustrative transient evolution. The evaluation protocol is given in Appendix C. (b) IMG samples through generative-flow integration. The five samples use independent latent draws at fixed trained weights. (c) DNS and physics-supervised samples illustrate spatial structure and fluctuation amplitudes.

Table 1: Physics-supervised KS/HW results against DNS. Each configuration uses one training run. Cells show the mean above and $\pm$ standard deviation below over four KS or five HW sampling repeats. Bold marks the lowest displayed mean. HW PDF-TV averages ζ, n, ϕ. Rounding follows Appendix B.5.

	KS Equation			HW Equations			
Method	Coarse SW_2 $\downarrow$	Spectrum rel. L_1 $\downarrow$	PDF TV $\downarrow$	Spectrum rel. L_1 $\downarrow$	PDF TV $\downarrow$	Increment PDF-TV $\downarrow$	Flatness rel. L_1 $\downarrow$
FNO	0.585 $\pm$0.001	1.647 $\pm$0.009	0.171 $\pm$0.003	0.48 $\pm$0.01	0.087 $\pm$0.002	0.074 $\pm$0.001	0.149 $\pm$0.002
U-Net	0.28 $\pm$0.06	0.9 $\pm$0.1	0.15 $\pm$0.03	0.50 $\pm$0.01	0.152 $\pm$0.004	0.2271 $\pm$0.0008	0.176 $\pm$0.002
PI-DeepONet	1.18 $\pm$0.01	1.05 $\pm$0.04	0.59 $\pm$0.05	1.2 $\pm$0.2	0.55 $\pm$0.02	0.68 $\pm$0.02	0.27 $\pm$0.01
WGS	0.349 $\pm$0.001	0.915 $\pm$0.002	0.192 $\pm$0.001	1.22 $\pm$0.01	0.257 $\pm$0.004	0.289 $\pm$0.001	0.210 $\pm$0.002
IMG (ours)	**0.050** $\pm$0.003	**0.038** $\pm$0.003	**0.0176** $\pm$0.0006	**0.063** $\pm$0.005	**0.012** $\pm$0.002	**0.0094** $\pm$0.0003	**0.027** $\pm$0.002

Table 1 shows that IMG attains the lowest reported mean error in every KS and HW column among the compared physics-supervised methods. The gains span amplitudes, spectral power, and spatial increments, indicating recovery of both pointwise fluctuations and spatial organization. The **particle-pool ablation** (Appendix E.9) shows that fresh IMG sampling improves spectra, marginal PDFs, and increment PDFs over persistent particles at matched physical advancement. Baseline implementation and hyperparameter searches are reported in Appendix C.4.

Statistical stability under physical evolution. We advance generated HW ensembles to test whether their statistics persist (Appendix E.7). Generalized enstrophy U measures the strength of coupled density–vorticity fluctuations, while the particle flux Γ_n measures transport across the magnetic field. Data-free IMG's short-time PDF change is close to DNS self-evolution, but its spectrum and generalized enstrophy drift over longer horizons. Its mean particle flux rises from its initial value, whereas the finite DNS control decreases. These changes indicate residual dynamical imbalance despite accurate static statistics.

A supplementary comparison of generation time and marginal PDF-TV is provided in Appendix E.3. Appendix D reports the full HW training cost.

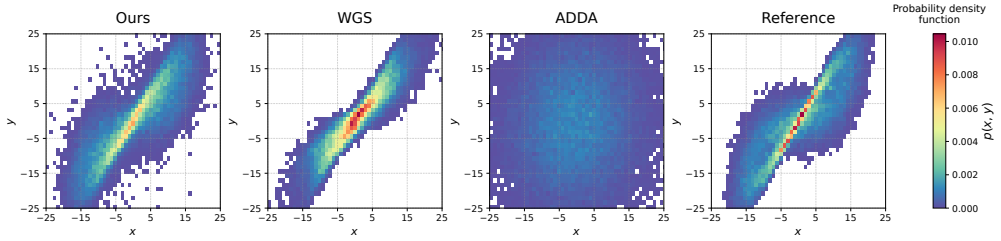


Figure 4: Lorenz–63 joint densities projected onto the (x, y) plane for IMG, WGS, ADDA, and DNS, using 32,768 samples per panel. All panels share histogram bins and a color scale. Densities are normalized over the full histogram domain before displaying $[-25, 25]^2$. The shared physical parameters are given in Appendix B.1. Table 2 evaluates the full three-dimensional distribution using 1,024 samples.

For Lorenz–63, all methods and DNS share the same stochastic dynamics, with parameters given in Appendix B.1. IMG has the lowest three-dimensional Wasserstein-2 distance in this exploratory comparison (Table 2). Figure 4 shows that IMG captures the elongated, positively correlated (x, y) structure, although its density is more diffuse than the sharp DNS ridge. WGS concentrates more narrowly along the diagonal, while ADDA spreads probability broadly across the displayed region.

Table 2: Lorenz–63 three-dimensional W_2 with one 2,000-update run per method.

Method	$W_2 \downarrow$
Ours	2.864
WGS	4.088
ADDA	12.463

5.2 Learning with limited data from the invariant measure

We add IMG supervision to Data CFM using the same limited stationary training data as the other data-assisted methods. Appendix C.2 details the few-data settings. We compare solver-based IMG with IMG-neural, which advances generated states over short physical times using a frozen, data-trained U-Net. The teachers use the baseline U-Net architectures and the same stationary data as data CFM. IMG-neural requires no additional stationary samples or numerical-solver calls during sampler training. Table 3 reports the resulting statistical accuracy.

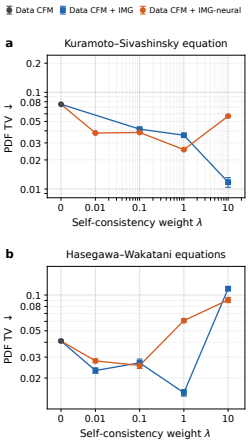


Figure 5: Self-consistency weight ablation on (a) KS and (b) HW. PDF-TV is compared with data CFM (gray). Bars show sampling variability at fixed weights.

For HW, solver supervision achieves the lowest mean error in all four metrics (Table 3). Neural supervision improves spectra, marginal PDFs, and increment PDFs over Data CFM, with little change in aggregate flatness. Both variants reduce excess power in low-wavenumber vorticity and potential modes relative to Data CFM (Appendix E.8). Solver supervision also improves small-scale density intermittency, but vorticity-increment tails remain underrepresented. Under physical evolution, solver-assisted CFM shows less spectral and PDF relaxation than Data CFM and less long-time generalized-enstrophy drift (Appendix E.7). Its short-time enstrophy drift and long-time particle-flux drift magnitude are nevertheless larger than Data CFM's. Thus, solver supervision improves agreement with DNS in several static and evolution-based statistical diagnostics, although some statistics still drift under physical evolution.

For KS, both solver supervision at $\lambda = 10$ and neural supervision at $\lambda = 1$ improve all three metrics over Data CFM, with solver supervision giving the lowest mean errors in Table 3. The joint gains in spatial power, field amplitudes, and the coarse-field distribution suggest useful dynamical constraints beyond fitting the limited snapshots. Neural supervision also improves on direct U-Net rollout. Extended FNO rollouts instead degrade statistical accuracy (Appendix E.2), illustrating the difficulty of preserving stationary statistics through repeated propagation.

Table 3: Few-data results against DNS. Each configuration uses one training run. Means and standard deviations are over four KS or five HW sampling repeats. Bold marks the lowest displayed mean. HW PDF-TV averages ζ, n, ϕ. Rounding follows Appendix B.5.

Method	KS Equation			HW Equations			
	Coarse SW_2 ↓	Spectrum rel. L_1 ↓	PDF TV ↓	Spectrum rel. L_1 ↓	PDF TV ↓	Increment PDF-TV ↓	Flatness rel. L_1 ↓
FNO	0.579 ±0.009	1.10 ±0.02	0.146 ±0.002	0.43 ±0.01	0.072 ±0.003	0.0887 ±0.0006	0.0971 ±0.0008
U-Net	0.10 ±0.01	0.15 ±0.02	0.031 ±0.002	0.355 ±0.004	0.064 ±0.002	0.0442 ±0.0006	0.171 ±0.003
DySLIM	0.100 ±0.006	0.11 ±0.02	0.029 ±0.001	0.097 ±0.008	0.024 ±0.001	0.0175 ±0.0006	0.064 ±0.003
Data CFM	0.1018 ±0.0009	0.161 ±0.004	0.0753 ±0.0004	0.19 ±0.01	0.041 ±0.001	0.0298 ±0.0004	0.084 ±0.002
Data CFM + IMG (ours)	**0.050** ±0.002	**0.050** ±0.004	**0.012** ±0.001	**0.090** ±0.006	**0.015** ±0.001	**0.0164** ±0.0008	**0.040** ±0.002
Data CFM + IMG-neural (ours)	0.063 ±0.002	0.102 ±0.003	0.0256 ±0.0003	0.13 ±0.01	0.026 ±0.002	0.0210 ±0.0005	0.084 ±0.004

Ablation studies. Figure 5 varies the self-consistency weight with other settings fixed, using data-only CFM as the zero-weight control. Appendix E.4 reports all metrics. Solver supervision improves KS throughout the tested range, whereas neural supervision favors an intermediate weight, consistent with a growing influence of target approximation error. For HW, both variants deteriorate at the largest weight, suggesting that excessive self-consistency can weaken the fit to stationary data. Across three independent training seeds per weight, KS neural supervision improves all three metrics on average over Data CFM, supporting gains beyond a single initialization (Appendix E.5, Table 13).

6 Conclusion

IMG learns direct samplers of long-time stationary statistics from short physical evolutions of its own generated states. Its self-consistent flow-matching objective requires no precomputed dataset of saturated states for training. Experiments on Lorenz–63, KS, and HW demonstrate statistical agreement with DNS, including spectra and spatial increments in high-dimensional turbulence, with residual relaxation under physical evolution. In the few-data setting, a frozen neural operator supplies additional supervision that improves PDF, spectral, and increment accuracy on HW without adding stationary samples or numerical-solver calls during sampler training. Together, these results establish a way to convert short-time dynamics into stationary-state sampling capability. Once trained, the sampler generates states without a physical burn-in stage.

7 Limitations and Future Work

Self-consistency does not automatically select the physically relevant invariant measure when several coexist. This also motivates exploring whether different initializations, architectures, or perturbations can help IMG find measures beyond those reached by a single DNS trajectory. The theory treats frozen targets, leaving convergence as targets change open. Neural prediction and interpolation between physical time steps introduce additional errors (Appendix A.1).

Currently, training requires repeated generation and physical advancement. Our few-data results show that frozen neural operators can provide useful supervision without solver calls during sampler training, although generation, operator pretraining, and neural evaluations still incur costs. Pretrained physical foundation models could lower the cost of supervision for systems with expensive solvers, provided they remain accurate on generated states. For kinetic plasma systems, latent-space learning could further reduce computational demands if it preserves stationary statistics.

AI USE STATEMENT

In this work, we used generative AI tools for the following tasks with required disclosure: help develop theoretical models or conceptual frameworks; formulate mathematical claims; propose or refine hypotheses; design or provide feedback on research methodology or experiments; implement methods; interpret results; provide critical ingredients for proving mathematical claims; assist in the writing of proofs; and assist with translation. We have not used generative AI tools for the following tasks with required disclosure: generate synthetic data sets; and clean and reformat datasets. Supporting qualitative and thematic data analysis does not apply to this work.

Additionally, we used generative AI tools for the following tasks with recommended disclosure: create or modify scientific figures or images; suggest experimental parameters; create or edit software code; creation of artifacts; draft parts of a research paper; summarize or analyze existing literature; discover research topics or identify gaps; brainstorming; sourcing/searching for information; and identify relevant literature. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies computational methods for sampling chaotic and turbulent dynamical systems using numerical simulations. The study involves no human participants, personal data, or animal experiments. We do not anticipate direct societal harms from the reported methods or experiments.

REPRODUCIBILITY STATEMENT

Algorithm 1 specifies the training procedure, and Appendix A gives the theoretical assumptions and proof. Appendix B defines the dynamical systems, numerical solvers, datasets, metrics, and evaluation repeats. Appendix C reports the architectures, optimization schedules, configurations, and hyperparameter searches. Appendix E.7 describes evolution diagnostics and step-size checks.

REFERENCES

Zhiqiang Cai, Yu Cao, Yuanfei Huang, and Xiang Zhou. Weak generative sampler to efficiently sample invariant distribution of stochastic differential equation. *SIAM J. Sci. Comput.*, 48(4): C708–C735, July 2026. ISSN 1064-8275. doi: 10.1137/24M1665271. URL https://doi.org/10.1137/24M1665271.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/69386f6bb1dfed68692a24c8686939b9-Paper.pdf.

L. Chevillard, B. Castaing, E. Lévêque, and A. Arneodo. Unified multifractal description of velocity increments statistics in turbulence: Intermittency and skewness. *Physica D: Nonlinear Phenomena*, 218(1):77–82, 2006. ISSN 0167-2789. doi: https://doi.org/10.1016/j.physd.2006.04.011.

William Cowieson and Lai-Sang Young. Srb measures as zero-noise limits. *Ergodic Theory and Dynamical Systems*, 25(4):1115–1138, 2005. doi: 10.1017/S0143385704000604.

A. M. Dimits, G. Bateman, M. A. Beer, B. I. Cohen, W. Dorland, G. W. Hammett, C. Kim, J. E. Kinsey, M. Kotschenreuther, A. H. Kritz, L. L. Lao, J. Mandrekas, W. M. Nevins, S. E. Parker, A. J. Redd, D. E. Shumaker, R. Sydora, and J. Weiland. Comparisons and physics basis of tokamak transport models and turbulence simulations. *Physics of Plasmas*, 7(3):969–983, 03 2000. ISSN 1070-664X. doi: 10.1063/1.873896. URL https://doi.org/10.1063/1.873896.

Nic Fishman, Gokul Gowri, Peng Yin, Jonathan Gootenberg, and Omar Abudayyeh. Generative distribution embeddings: Lifting autoencoders to the space of distributions for multiscale representation learning. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz,

M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 24644–24685. Curran Associates, Inc., 2025. doi: 10.52202/085713-0830. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/233b892889d0a18f9da39226ee5c9dc5-Paper-Conference.pdf.

C. Foias, O. Manley, R. Rosa, and R. Temam. *Navier-Stokes Equations and Turbulence*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2001.

Gianluca Galletti, Gerald Gutenbrunner, William Hornsby, Lorenzo Zanisi, Naomi Carey, Stanislas Pamela, Johannes Brandstetter, and Fabian Paischer. A shortcut to statistically steady-state turbulence with flow matching, 2026. URL https://arxiv.org/abs/2607.13022.

Akira Hasegawa and Masahiro Wakatani. Plasma edge turbulence. *Phys. Rev. Lett.*, 50:682–686, Feb 1983. doi: 10.1103/PhysRevLett.50.682. URL https://link.aps.org/doi/10.1103/PhysRevLett.50.682.

W. Horton. Drift waves and transport. *Rev. Mod. Phys.*, 71:735–778, Apr 1999. doi: 10.1103/RevModPhys.71.735. URL https://link.aps.org/doi/10.1103/RevModPhys.71.735.

Ruoxi Jiang, Peter Y. Lu, Elena Orlova, and Rebecca Willett. Training neural operators to preserve invariant measures of chaotic attractors. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 27645–27669. Curran Associates, Inc., 2023. doi: 10.52202/075280-1202. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/57d7e7e1593ad1ab6818c258fa5654ce-Paper-Conference.pdf.

L. E. Jones, R. D. Sandberg, and N. D. Sandham. Direct numerical simulations of forced and unforced separation bubbles on an airfoil at incidence. *Journal of Fluid Mechanics*, 602:175–207, 2008. doi: 10.1017/S0022112008000864.

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.

Lingxiao Li, Samuel Hurault, and Justin Solomon. Self-consistent velocity matching of probability flows. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=C6fvJ2RfsL.

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

Zongyi Li, Miguel Liu-Schiaffini, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Learning chaotic dynamics in dissipative systems. *Advances in Neural Information Processing Systems*, 35:16768–16781, 2022.

Marten Lienen, David Lüdke, Jan Hansen-Palmus, and Stephan Günnemann. From Zero to Turbulence: Generative Modeling for 3D Flow Simulation. In *The Twelfth International Conference on Learning Representations*, 2024.

Mario Lino, Tobias Pfaff, and Nils Thuerey. Learning distributions of complex fluid simulations with diffusion graph networks. In *International Conference on Learning Representations*, volume 2025, pp. 77089–77119, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Tian Ma, Bernhard Ott, Jochen Fröhlich, and Andrew D. Bragg. Scale-dependent anisotropy, energy transfer and intermittency in bubble-laden turbulent flows. *Journal of Fluid Mechanics*, 927:A16, 2021. doi: 10.1017/jfm.2021.760.

Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter Battaglia. Learning mesh-based simulation with graph networks. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=roNqYL0_XP.

Yair Schiff, Zhong Yi Wan, Jeffrey B. Parker, Stephan Hoyer, Volodymyr Kuleshov, Fei Sha, and Leonardo Zepeda-Núñez. DySLIM: Dynamics Stable Learning by Invariant Measure for Chaotic Systems. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 43649–43684. PMLR, July 2024. URL https://proceedings.mlr.press/v235/schiff24b.html.

Kejun Tang, Xiaoliang Wan, and Qifeng Liao. Adaptive deep density approximation for fokker-planck equations. *Journal of Computational Physics*, 457:111080, 2022. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2022.111080. URL https://www.sciencedirect.com/science/article/pii/S0021999122001425.

Sifan Wang, Hanwen Wang, and Paris Perdikaris. Learning the solution operator of parametric partial differential equations with physics-informed DeepONets. *Science Advances*, 7(40):eabi8605, 2021. doi: 10.1126/sciadv.abi8605.

Yong Xu, Hao Zhang, Yongge Li, Kuang Zhou, Qi Liu, and Jürgen Kurths. Solving fokker-planck equation using deep learning. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 30(1): 013133, 01 2020. ISSN 1054-1500. doi: 10.1063/1.5132840. URL https://doi.org/10.1063/1.5132840.

Lai-Sang Young. Ergodic theory of attractors. In S. D. Chatterji (ed.), *Proceedings of the International Congress of Mathematicians*, pp. 1230–1237, Basel, 1995. Birkhäuser Basel. ISBN 978-3-0348-9078-6.

Lai-Sang Young. What Are SRB Measures, and Which Dynamical Systems Have Them? *Journal of Statistical Physics*, 108(5):733–754, September 2002. ISSN 1572-9613. doi: 10.1023/A:1019762724717. URL https://doi.org/10.1023/A:1019762724717.

A Proof of Theorem 1

Proof. The proof has two parts. First, velocity matching bounds the average Wasserstein error over short physical rollouts. Second, this average error bounds the invariance residual at every physical time. We then relate the velocity-matching objective to the gradient of the self-consistency loss.

Fix the target parameters $\bar{\theta}$ and the maximum rollout duration $\Delta T > 0$. Let $L \geq 0$ be a Lipschitz bound for $v_\theta(\tau, \cdot)$, uniform in τ and θ, and assume $\mathrm{Lip}(\Phi^s) \leq e^{\kappa s}$ for $s \geq 0$, with $\kappa \geq 0$. All flows and expectations below are assumed well defined, as in the theorem.

Part 1: Velocity matching controls the average short-time Wasserstein error. Condition on a sampled physical duration δt and draw independently $u_0 \sim \pi$ and $\bar{u}_1 \sim q_{\bar{\theta}}$, and set

$$u_1 = \Phi^{\delta t}(\bar{u}_1), \qquad u_\tau = (1-\tau)u_0 + \tau u_1.$$

Let $p_\tau^{\delta t}$ be the law of u_τ. The marginal velocity

$$w_\tau^{\delta t}(u) = \mathbb{E}[u_1 - u_0 \mid u_\tau = u, \delta t]$$

generates this probability path from π to $\mathcal{P}_{\delta t} q_{\bar{\theta}}$. To compare its endpoint with q_θ, couple the two generative flows through the same initial state:

$$\frac{\mathrm{d}\widehat{u}_\tau}{\mathrm{d}\tau} = v_\theta(\tau, \widehat{u}_\tau), \qquad \frac{\mathrm{d}\widetilde{u}_\tau}{\mathrm{d}\tau} = w_\tau^{\delta t}(\widetilde{u}_\tau), \qquad \widehat{u}_0 = \widetilde{u}_0 \sim \pi. \tag{10}$$

Then $\widehat{u}_1 \sim q_\theta$, $\widetilde{u}_1 \sim \mathcal{P}_{\delta t} q_{\bar{\theta}}$, and $\widetilde{u}_\tau \sim p_\tau^{\delta t}$. The Lipschitz bound on v_θ gives

$$\frac{\mathrm{d}}{\mathrm{d}\tau} \left\|\widehat{u}_\tau - \widetilde{u}_\tau\right\|_2 \le L \left\|\widehat{u}_\tau - \widetilde{u}_\tau\right\|_2 + \left\|v_\theta(\tau, \widetilde{u}_\tau) - w_\tau^{\delta t}(\widetilde{u}_\tau)\right\|_2$$

for almost every τ. Since the initial difference is zero, Grönwall's inequality yields

$$\left\|\widehat{u}_1 - \widetilde{u}_1\right\|_2 \le \int_0^1 e^{L(1-\tau)} \left\|v_\theta(\tau, \widetilde{u}_\tau) - w_\tau^{\delta t}(\widetilde{u}_\tau)\right\|_2 \mathrm{d}\tau.$$

For fixed target parameters, the conditional velocity-matching loss corresponding to Eq. 4 is

$$\mathcal{L}_{\mathrm{FM}}^{\delta t}(\theta; \bar{\theta}) := \int_0^1 \mathbb{E}_{u \sim p_\tau^{\delta t}} \left\|v_\theta(\tau, u) - w_\tau^{\delta t}(u)\right\|_2^2 \mathrm{d}\tau. \tag{11}$$

With $C_{\mathrm{FM}} := \int_0^1 e^{2L(1-\tau)} \mathrm{d}\tau$, the endpoint coupling and Cauchy–Schwarz give

$$\boxed{\begin{aligned} W_2^2(q_\theta, \mathcal{P}_{\delta t} q_{\bar{\theta}}) &\le \mathbb{E} \left\|\widehat{u}_1 - \widetilde{u}_1\right\|_2^2 \\ &\le C_{\mathrm{FM}}\, \mathcal{L}_{\mathrm{FM}}^{\delta t}(\theta; \bar{\theta}). \end{aligned}} \tag{12}$$

To average this bound over randomized rollout durations, define

$$\mathcal{L}_{\mathrm{FM}}(\theta; \bar{\theta}) := \mathbb{E}_{\delta t \sim \mathcal{U}(0, \Delta T)}[\mathcal{L}_{\mathrm{FM}}^{\delta t}(\theta; \bar{\theta})].$$

Then

$$\mathcal{A}(\theta; \bar{\theta}) := \frac{1}{\Delta T} \int_0^{\Delta T} W_2^2(q_\theta, \mathcal{P}_s q_{\bar{\theta}}) \mathrm{d}s \le C_{\mathrm{FM}}\, \mathcal{L}_{\mathrm{FM}}(\theta; \bar{\theta}). \tag{13}$$

Here $\mathcal{A}$ is the mean short-time distributional error and $\mathcal{L}_{\mathrm{FM}}$ the mean marginal loss.

Part 2: Randomized short rollouts control invariance at every physical time. We now show that the average error $\mathcal{A}(\theta; \bar{\theta})$ controls $R_t(q_\theta) := W_2(q_\theta, \mathcal{P}_t q_\theta)$ for every $t > 0$. The physical Lipschitz bound implies, for any distributions μ, ν with finite second moments,

$$W_2(\mathcal{P}_a \mu, \mathcal{P}_a \nu) \le e^{\kappa a} W_2(\mu, \nu). \tag{14}$$

For $0 < a \le \Delta T/2$ and $0 \le s \le \Delta T/2$, insert the intermediate distribution $\mathcal{P}_{s+a} q_{\bar{\theta}}$. The triangle inequality, the semigroup property, and Eq. (14) give

$$\begin{aligned} R_a(q_\theta) &\le W_2(q_\theta, \mathcal{P}_{s+a} q_{\bar{\theta}}) + W_2(\mathcal{P}_a \mathcal{P}_s q_{\bar{\theta}}, \mathcal{P}_a q_\theta) \\ &\le W_2(q_\theta, \mathcal{P}_{s+a} q_{\bar{\theta}}) + e^{\kappa a} W_2(q_\theta, \mathcal{P}_s q_{\bar{\theta}}). \end{aligned} \tag{15}$$

Both intervals $[0, \Delta T/2]$ and $[a, a + \Delta T/2]$ lie within $[0, \Delta T]$. Taking the L^2 norm over $s \sim \mathcal{U}(0, \Delta T/2)$ and applying Minkowski's inequality therefore yields

$$\begin{aligned} R_a(q_\theta) &\le \left[\frac{2}{\Delta T} \int_0^{\Delta T/2} W_2^2(q_\theta, \mathcal{P}_{s+a} q_{\bar{\theta}}) \mathrm{d}s\right]^{1/2} \\ &\quad + e^{\kappa a} \left[\frac{2}{\Delta T} \int_0^{\Delta T/2} W_2^2(q_\theta, \mathcal{P}_s q_{\bar{\theta}}) \mathrm{d}s\right]^{1/2} \\ &\le \sqrt{2}(1 + e^{\kappa a})\, \mathcal{A}(\theta; \bar{\theta})^{1/2}. \end{aligned} \tag{16}$$

Randomizing the rollout duration allows the averaged error to control both the original and shifted time intervals in Eq. (15).

For an arbitrary $t > 0$, choose $n_t := \lceil 2t/\Delta T \rceil$ and $a_t := t/n_t$, so $0 < a_t \le \Delta T/2$. Applying the triangle inequality to successive physical evolutions of duration a_t gives

$$\begin{aligned} R_t(q_\theta) &\le \sum_{j=0}^{n_t - 1} W_2(\mathcal{P}_{j a_t} q_\theta, \mathcal{P}_{j a_t} \mathcal{P}_{a_t} q_\theta) \\ &\le \left(\sum_{j=0}^{n_t - 1} e^{\kappa j a_t}\right) R_{a_t}(q_\theta). \end{aligned} \tag{17}$$

Combining Eqs. (16) and (17) gives the bound from short rollouts to arbitrary physical times:

$$\boxed{W_2^2(q_\theta, \mathcal{P}_t q_\theta) \le C_{\mathrm{phys}}(t, \Delta T)\, \frac{1}{\Delta T} \int_0^{\Delta T} W_2^2(q_\theta, \mathcal{P}_s q_{\bar\theta})\, \mathrm{d}s,} \tag{18}$$

where

$$C_{\mathrm{phys}}(t, \Delta T) := 2(1 + e^{\kappa a_t})^2 \left(\sum_{j=0}^{n_t - 1} e^{\kappa j a_t} \right)^2 .$$

Together with Eq. (13), this proves $R_t(q_\theta)^2 \le C_t \mathcal{L}_{\mathrm{FM}}(\theta; \bar\theta)$, where $C_t := C_{\mathrm{FM}} C_{\mathrm{phys}}(t, \Delta T)$ is finite for every finite $t > 0$ and independent of θ and $\bar\theta$. At $t = 0$, the residual is zero.

To connect this bound to the training loss, condition on $(\delta t, \tau, u_\tau)$. Since $w_\tau^{\delta t}(u_\tau)$ is the conditional mean of $u_1 - u_0$, the squared-error decomposition gives

$$\mathcal{L}_{\mathrm{SC}}(\theta; \bar\theta) = \mathcal{L}_{\mathrm{FM}}(\theta; \bar\theta) + \mathbb{E} \left\| u_1 - u_0 - w_\tau^{\delta t}(u_\tau) \right\|_2^2 . \tag{19}$$

With $\bar\theta$ fixed, the last term is independent of θ. Hence the self-consistent CFM loss and the marginal velocity-matching loss have the same expected gradient:

$$\boxed{\nabla_\theta \mathcal{L}_{\mathrm{SC}}(\theta; \bar\theta) = \nabla_\theta \mathcal{L}_{\mathrm{FM}}(\theta; \bar\theta).} \tag{20}$$

Consequently, the residual-bound loss $\mathcal{L}_{\mathrm{bound}}^t := C_t \mathcal{L}_{\mathrm{FM}}$ satisfies $\nabla_\theta \mathcal{L}_{\mathrm{bound}}^t = C_t \nabla_\theta \mathcal{L}_{\mathrm{SC}}$. The derivatives hold the generated targets and interpolation states fixed, as in training. □

□

A.1 Approximate physical advancement

Theorem 1 uses the exact physical evolution. For a numerical solver or a frozen neural operator, let $\widehat{\mathcal{P}}_s$ be the push-forward induced by the approximate map $\widehat{\Phi}^s$. This family need not satisfy the semigroup property. Fix $\bar\theta$, draw $s \sim \mathcal{U}(0, \Delta T)$ and $u \sim q_{\bar\theta}$ independently, and define its mean-square target error by

$$\varepsilon_{\mathrm{adv}}^2(\bar\theta) := \mathbb{E} \left[\| \widehat{\Phi}^s(u) - \Phi^s(u) \|_2^2 \right]. \tag{21}$$

The state-map formulation here applies to deterministic dynamics, including KS and HW. Assume this expectation is finite and that the flow-matching regularity assumptions also hold for the approximate target paths. Let $\widehat{\mathcal{L}}_{\mathrm{FM}}$ and $\widehat{\mathcal{L}}_{\mathrm{SC}}$ denote the corresponding losses with $u_1 = \widehat{\Phi}^s(u)$. Coupling exact and approximate targets through the same u, then applying the triangle inequality in Wasserstein distance and in L^2 over s, gives

$$\mathcal{A}(\theta; \bar\theta)^{1/2} \le \left(C_{\mathrm{FM}} \widehat{\mathcal{L}}_{\mathrm{FM}}(\theta; \bar\theta) \right)^{1/2} + \varepsilon_{\mathrm{adv}}(\bar\theta). \tag{22}$$

Equation (18), which uses only the exact physical semigroup, therefore yields

$$R_t(q_\theta)^2 \le 2 C_t \widehat{\mathcal{L}}_{\mathrm{FM}}(\theta; \bar\theta) + 2 C_{\mathrm{phys}}(t, \Delta T) \varepsilon_{\mathrm{adv}}^2(\bar\theta). \tag{23}$$

With the operator and $\bar\theta$ fixed, the additional term is independent of θ, and the gradient of this upper bound is $2 C_t \nabla_\theta \widehat{\mathcal{L}}_{\mathrm{SC}}$ by the same conditional-mean decomposition as Eq. (19). Thus, vanishing velocity mismatch alone does not imply exact physical invariance when advancement error remains. This error is measured on generated states and can change when the target parameters are refreshed.

Local physical-time interpolation. For a deterministic trajectory over one operator interval h_{NO} between consecutive neural predictions, take $s = \alpha h_{\mathrm{NO}}$ with $0 \le \alpha \le 1$. Suppose $r \mapsto \Phi^r(u)$ is twice differentiable and $M(u) := \sup_{0 \le r \le h_{\mathrm{NO}}} \| \partial_r^2 \Phi^r(u) \|_2$ is finite. The linear-interpolation remainder and the triangle inequality imply

$$\begin{aligned} \| \widehat{\Phi}^s(u) - \Phi^s(u) \|_2 \le{}& \alpha \| \widehat{\Phi}^{h_{\mathrm{NO}}}(u) - \Phi^{h_{\mathrm{NO}}}(u) \|_2 \\ &+ \tfrac{1}{2} \alpha (1 - \alpha) h_{\mathrm{NO}}^2 M(u). \end{aligned} \tag{24}$$

The first term is the operator endpoint error. The second is the physical-time interpolation error, bounded by $h_{\mathrm{NO}}^2 M(u)/8$ and zero at the endpoints. Finite second moments of the endpoint error and $M(u)$ make this a bound on Eq. (21) after averaging. For longer rollouts, interpolation remains local to the two neighboring prediction times, with prediction errors at both endpoints. Short operator intervals reduce the interpolation contribution under this smoothness assumption, but do not remove operator error or guarantee accuracy on generated states. We do not assume a semigroup property for the interpolant.

B Experiment Setup

B.1 Dynamical systems

We consider Lorenz–63, the Kuramoto–Sivashinsky (KS) equation, and the Hasegawa–Wakatani (HW) equations. Throughout, t denotes physical time and $u(t)$ denotes the complete system state. For the PDE systems, $u(t)$ contains the field values on the spatial grid. The numerical integration step size is denoted by h, distinct from the sampled physical rollout duration δt and its upper limit ΔT in the self-consistency loss.

Lorenz–63 system. The Lorenz–63 system provides a low-dimensional setting in which the geometry of the learned invariant measure can be inspected directly. Its state is $u(t) = (x(t), y(t), z(t))^\top \in \mathbb{R}^3$, where x, y, and z are the three state coordinates. The dynamics are

$$\begin{aligned} \mathrm{d}x &= \sigma(y - x)\,\mathrm{d}t + \sqrt{2D}\,\mathrm{d}W_t^x, \\ \mathrm{d}y &= [x(\rho - z) - y]\,\mathrm{d}t + \sqrt{2D}\,\mathrm{d}W_t^y, \\ \mathrm{d}z &= (xy - \beta z)\,\mathrm{d}t + \sqrt{2D}\,\mathrm{d}W_t^z, \end{aligned} \tag{25}$$

Here W_t^x, W_t^y, W_t^z are independent standard Brownian motions. The parameters are $\sigma = 10.75$, $\rho = 35$, $\beta = 8/3$, and $D = 0.1$. IMG, WGS, ADDA, and the DNS reference use this same stochastic system. At this setting, typical trajectories move irregularly between the two lobes of the attractor. Recovering the invariant measure therefore requires capturing both the geometry of the lobes and their relative probability mass. Numerical evolution uses a time step of $h = 0.01$. Initial coordinates are drawn independently from $x(0), y(0) \sim \mathcal{U}(-15, 15)$ and $z(0) \sim \mathcal{U}(10, 40)$, where $\mathcal{U}(a, b)$ denotes the uniform distribution on $[a, b]$. We apply an initial warm-up of 1,000 integration steps before subsequent reference sampling. The system has no spatial boundary conditions.

Kuramoto–Sivashinsky equation. The KS equation tests invariant-measure learning for spatially extended chaotic dynamics. It combines a long-wavelength instability, short-wavelength dissipation, and nonlinear coupling between spatial scales. The scalar field $u(x, t)$ satisfies

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + \frac{\partial^2 u}{\partial x^2} + \frac{\partial^4 u}{\partial x^4} = 0, \qquad x \in [0, L_{\mathrm{KS}}), \tag{26}$$

where x is the spatial coordinate and $L_{\mathrm{KS}} = 64$ is the domain length. We impose periodic boundary conditions, $u(x + L_{\mathrm{KS}}, t) = u(x, t)$. The second-derivative term drives the instability, the fourth-derivative term damps small-scale fluctuations, and nonlinear advection couples the unstable and dissipative scales. Their interaction produces sustained spatiotemporal chaos.

We discretize the domain with $N_x = 256$ uniformly spaced grid points, so the discretized state is $u(t) \in \mathbb{R}^{N_x}$. Spatial derivatives are evaluated using a Fourier pseudospectral method, and time integration uses fourth-order exponential time-differencing Runge–Kutta (ETDRK4) with $h = 5 \times 10^{-4}$. Initial fields are sampled from a periodic Gaussian random field, and the spatial mean of each realization is removed. The dynamics preserve this zero-mean constraint. DNS reference states are retained from the post-transient window $t \in [100, 199]$.

Hasegawa–Wakatani equations. In magnetic confinement fusion, turbulence can drive particle transport across magnetic field lines and reduce plasma confinement. The HW equations describe resistive drift-wave turbulence driven by a background density gradient near the plasma edge (Hasegawa & Wakatani, 1983). This turbulence produces correlated density and electric-potential

fluctuations that contribute to radial particle transport. The model tests whether a learned invariant measure captures the joint spatial statistics of turbulence.

We consider a two-dimensional plane perpendicular to the background magnetic field, with x and y denoting the local radial and poloidal coordinates. The model evolves the vorticity field $\zeta(x, y, t)$ and the density fluctuation field $n(x, y, t)$. Here n is the normalized fluctuation about the background density, rather than the total density. The complete state concatenates these two fields:

$$u(t) = \begin{pmatrix} \mathrm{vec}\,\zeta(t) \\ \mathrm{vec}\,n(t) \end{pmatrix} \in \mathbb{R}^{2N_x N_y}, \tag{27}$$

where N_x and N_y are the numbers of grid points along the two directions, and vec flattens a grid field into a vector. The implementation stores the same state as a two-channel field in the order $[\zeta, n]$. The electrostatic potential $\phi(x, y, t)$ is reconstructed from ζ and is not an additional state channel.

The dimensionless ordinary HW equations with fourth-order dissipation are

$$\begin{aligned} \frac{\partial \zeta}{\partial t} + \{\phi, \zeta\} &= \alpha(\phi - n) - \nu \nabla_\perp^4 \zeta, \\ \frac{\partial n}{\partial t} + \{\phi, n\} &= \alpha(\phi - n) - \kappa \frac{\partial \phi}{\partial y} - \nu \nabla_\perp^4 n. \end{aligned} \tag{28}$$

Here α is the adiabaticity parameter controlling the resistive density–potential coupling, κ specifies the background density-gradient drive, and ν is the hyperdiffusion coefficient. The perpendicular Laplacian is $\nabla_\perp^2 = \partial_x^2 + \partial_y^2$, with $\nabla_\perp^4 = (\nabla_\perp^2)^2$. For scalar fields f and g, the Poisson bracket is

$$\{f, g\} := \frac{\partial f}{\partial x}\frac{\partial g}{\partial y} - \frac{\partial f}{\partial y}\frac{\partial g}{\partial x}. \tag{29}$$

Consequently, $\{\phi, f\}$ represents advection of f by the normalized electric drift velocity $\boldsymbol{v}_E = (-\partial_y \phi, \partial_x \phi)$.

The potential is obtained by periodic Poisson inversion. For each nonzero spatial Fourier wavevector $\boldsymbol{k} = (k_x, k_y)$, we set

$$\widehat{\phi}(\boldsymbol{k}) = -\frac{\widehat{\zeta}(\boldsymbol{k})}{|\boldsymbol{k}|^2}, \qquad \widehat{\phi}(\boldsymbol{0}) = 0, \tag{30}$$

where a hat denotes a spatial Fourier coefficient and $|\boldsymbol{k}|^2 = k_x^2 + k_y^2$. This fixes the potential to have zero spatial mean. The inversion corresponds to $\nabla_\perp^2 \phi = \zeta - \langle \zeta \rangle_{\mathrm{space}}$, where $\langle \zeta \rangle_{\mathrm{space}}$ is the spatial mean of the vorticity field. For zero-mean vorticity, it reduces to $\zeta = \nabla_\perp^2 \phi$.

We use a doubly periodic square domain with side length $L_{\mathrm{HW}} = 2\pi/k_0$, where $k_0 = 0.15$ is the fundamental wavenumber. The grid has $N_x = N_y = 256$, giving the complete state dimension $d = 2N_x N_y = 131{,}072$. The parameters are $\alpha = 1$, $\kappa = 1$, and $\nu = 10^{-4}$. Poisson brackets are discretized with the Arakawa scheme, and fourth-order dissipation uses repeated finite-difference Laplacians. Potential inversion and the density-gradient drive are evaluated in Fourier space. Time integration uses fourth-order Runge–Kutta (RK4) with $h = 0.0125$. Initial vorticity and density fields are independent Gaussian white-noise perturbations with standard deviation 10^{-9} at each grid point.

The connection between field statistics and transport can be expressed through the spatially averaged radial particle flux,

$$\Gamma_n(u) = -\frac{1}{L_{\mathrm{HW}}^2} \int_0^{L_{\mathrm{HW}}} \int_0^{L_{\mathrm{HW}}} n(x, y)\, \frac{\partial \phi(x, y)}{\partial y}\, \mathrm{d}x\, \mathrm{d}y. \tag{31}$$

Here Γ_n is an observable of the complete state u, since ϕ is determined by ζ. Its dependence on both density and potential shows why their joint spatial statistics matter for transport. The invariant measure μ determines the stationary mean flux $\mathbb{E}_{u\sim\mu}[\Gamma_n(u)]$.

B.2 Evaluation metrics

We evaluate the model distribution q_θ against DNS samples approximating the invariant measure μ. We report the empirical Wasserstein-2 distance for Lorenz–63. For KS, we report the sliced Wasserstein-2 distance, spectrum error, and marginal PDF error. For HW, we also report two spatial increment errors. All errors are smaller for better agreement with DNS.

Samples and field notation. Let $\{u_i^{\mathrm{gen}}\}_{i=1}^{N_{\mathrm{gen}}}$ and $\{u_j^{\mathrm{DNS}}\}_{j=1}^{N_{\mathrm{DNS}}}$ denote the generated and DNS states, respectively. Here N_{gen} and N_{DNS} denote sample counts, and superscripts identify the source.

For Lorenz–63, each state u is a vector in $\mathbb{R}^3$. For KS, u is a scalar field on a periodic grid. For the HW equation, $u = (\zeta, n)$ contains the vorticity field ζ and density fluctuation field n. The electrostatic potential ϕ is recovered from $\nabla^2\phi = \zeta - \langle\zeta\rangle_{\mathrm{space}}$ with zero spatial mean, where $\langle\zeta\rangle_{\mathrm{space}}$ denotes the spatial average of each vorticity field.

We use c to index a scalar field read from a state. Specifically, $f_u(u;\mathbf{x}) = u(\mathbf{x})$ for KS, and $f_\zeta(u;\mathbf{x}) = \zeta(\mathbf{x})$, $f_n(u;\mathbf{x}) = n(\mathbf{x})$, and $f_\phi(u;\mathbf{x}) = \phi(\mathbf{x})$ for the HW equation. Here $\mathbf{x}$ denotes a spatial grid location. For HW marginal PDF comparisons, the vorticity readout is spatially centered.

Let $\mathcal{G}$ be the set of spatial grid locations and $|\mathcal{G}|$ its size. For a scalar quantity $h(u,\mathbf{x})$, define its empirical sample-and-space average by

$$\langle h\rangle_s := \frac{1}{N_s|\mathcal{G}|}\sum_{i=1}^{N_s}\sum_{\mathbf{x}\in\mathcal{G}} h(u_i^s,\mathbf{x}), \qquad s\in\{\mathrm{gen},\mathrm{DNS}\}. \tag{32}$$

The source index s selects either the generated or DNS sample set.

Wasserstein-2 distance for Lorenz–63. For two sets of $N = 1{,}024$ states, we compute

$$\widehat{W}_2 = \left[\min_{\eta\in\mathfrak{S}_N}\frac{1}{N}\sum_{i=1}^{N}\left\|u_i^{\mathrm{gen}} - u_{\eta(i)}^{\mathrm{DNS}}\right\|_2^2\right]^{1/2}, \tag{33}$$

where $\mathfrak{S}_N$ is the set of permutations of $\{1,\dots,N\}$, η specifies a matching between the two sample sets, and $\|\cdot\|_2$ is the Euclidean norm. The optimal matching is computed in physical state coordinates, so the metric compares the joint three-dimensional distributions.

Coarse-field sliced Wasserstein-2 distance. For KS and the HW equation, we compare distributions of coarse-grained fields. Let $g:\mathcal{X}\to\mathbb{R}^D$ be the feature map used for this comparison, where D is the feature dimension. For KS, g averages each non-overlapping block of four grid points and flattens the result, giving $D = 64$. For the HW equation, g spatially centers vorticity, applies non-overlapping 4×4 average pooling to vorticity and density, and concatenates the flattened fields, giving $D = 8192$. These features retain their physical amplitudes.

We draw $M = 128$ random unit vectors $\xi_m\in\mathbb{R}^D$, indexed by $m = 1,\dots,M$. For each direction, define the scalar projections

$$z_{i,m}^s = \xi_m^\top g(u_i^s), \qquad s\in\{\mathrm{gen},\mathrm{DNS}\}. \tag{34}$$

The larger sample set is subsampled without replacement so that both sets contain $N = \min(N_{\mathrm{gen}}, N_{\mathrm{DNS}})$ states. Let $z_{(i),m}^s$ denote the ith smallest projection within source s along direction ξ_m. We report

$$\widehat{\mathrm{SW}}_2 = \left[\frac{1}{MN}\sum_{m=1}^{M}\sum_{i=1}^{N}\left(z_{(i),m}^{\mathrm{gen}} - z_{(i),m}^{\mathrm{DNS}}\right)^2\right]^{1/2}. \tag{35}$$

Each observation is a whole coarse-grained field, rather than an individual grid value.

Spectrum relative L_1 error. Let $S_c^s(k)$ denote the ensemble-averaged power spectrum of scalar field c from source s. For KS, k indexes the nonzero wavenumbers of the one-sided Fourier spectrum of the spatially centered field. For the HW equation, k indexes radial Fourier bins. We compute the relative error for each field as

$$E_{\mathrm{spec},c} = \frac{\sum_k \left|S_c^{\mathrm{gen}}(k) - S_c^{\mathrm{DNS}}(k)\right|}{\sum_k S_c^{\mathrm{DNS}}(k)}. \tag{36}$$

The spectra are averaged over states before computing the error. The reported metric is

$$E_{\mathrm{spec}} = \frac{1}{|\mathcal{C}_{\mathrm{spec}}|}\sum_{c\in\mathcal{C}_{\mathrm{spec}}} E_{\mathrm{spec},c}, \tag{37}$$

where $\mathcal{C}_{\mathrm{spec}} = \{u\}$ for KS and $\mathcal{C}_{\mathrm{spec}} = \{\zeta, n, \phi\}$ for the HW equation. The notation $|\mathcal{C}_{\mathrm{spec}}|$ denotes the number of fields in this set. Spectra are compared in linear units, retaining errors in both amplitude and shape.

Marginal PDF total variation. For each field c, we pool its values over sampled states and spatial locations. Let $\mathcal{B}_c$ denote a shared histogram partition, fixed from the DNS reference, and let $B \in \mathcal{B}_c$ denote one histogram bin. Its probability mass for source s is

$$p_c^s(B) = \langle \mathbf{1}\{f_c(u;\mathbf{x}) \in B\} \rangle_s \,, \tag{38}$$

where $\mathbf{1}\{\cdot\}$ is the indicator function. The partition includes out-of-range values, so their probability mass is not discarded. The total variation error is

$$E_{\mathrm{PDF}} = \frac{1}{|\mathcal{C}_{\mathrm{PDF}}|} \sum_{c \in \mathcal{C}_{\mathrm{PDF}}} \frac{1}{2} \sum_{B \in \mathcal{B}_c} \left| p_c^{\mathrm{gen}}(B) - p_c^{\mathrm{DNS}}(B) \right| , \tag{39}$$

where $\mathcal{C}_{\mathrm{PDF}} = \{u\}$ for KS and $\mathcal{C}_{\mathrm{PDF}} = \{\zeta, n, \phi\}$ for the HW equation. This metric compares field-value distributions without retaining their spatial arrangement.

Spatial increments for the HW equation. For increment statistics, the field index is $c \in \mathcal{C}_{\mathrm{inc}} := \{\zeta, n\}$. The direction index is $\alpha \in \{x, y\}$, and $\mathbf{e}_\alpha$ is the corresponding spatial unit vector. For a physical separation $r > 0$, define

$$\delta_{r,\alpha} f_c(u;\mathbf{x}) := f_c(u;\mathbf{x} + r\mathbf{e}_\alpha) - f_c(u;\mathbf{x}). \tag{40}$$

Shifts wrap around the periodic domain. For a square domain of side length L_{box} discretized with 256 grid points along each direction, the grid spacing is $\Delta x = L_{\mathrm{box}}/256$. We use the separation set

$$\mathcal{R} = \{1, 2, 4, 8, 16, 32, 64\}\Delta x. \tag{41}$$

Thus, c, α, and r specify the field, direction, and spatial separation of an increment, respectively.

Increment PDF total variation. Increment PDFs characterize spatial fluctuations across scales (Chevillard et al., 2006). For each (c, α, r), define the DNS root-mean-square increment

$$\sigma_{c,\alpha,r}^{\mathrm{DNS}} := \left\langle (\delta_{r,\alpha} f_c)^2 \right\rangle_{\mathrm{DNS}}^{1/2} . \tag{42}$$

Both generated and DNS increments are divided by this same reference scale. Let $\mathcal{B}_{\mathrm{inc}}$ be the histogram partition consisting of 240 uniform bins on $[-12, 12]$ and two additional bins covering values below and above this interval. For $B \in \mathcal{B}_{\mathrm{inc}}$, define

$$p_{c,\alpha,r}^s(B) := \left\langle \mathbf{1}\left\{ \frac{\delta_{r,\alpha} f_c}{\sigma_{c,\alpha,r}^{\mathrm{DNS}}} \in B \right\} \right\rangle_s . \tag{43}$$

We report

$$E_{\mathrm{inc}} = \frac{1}{28} \sum_{c \in \mathcal{C}_{\mathrm{inc}}} \sum_{\alpha \in \{x,y\}} \sum_{r \in \mathcal{R}} \frac{1}{2} \sum_{B \in \mathcal{B}_{\mathrm{inc}}} \left| p_{c,\alpha,r}^{\mathrm{gen}}(B) - p_{c,\alpha,r}^{\mathrm{DNS}}(B) \right| . \tag{44}$$

The factor $28 = 2 \times 2 \times 7$ averages equally over the two fields, two directions, and seven separations.

Flatness relative L_1 error. For source s, define the increment flatness by

$$\mathrm{Flat}_{c,\alpha}^s(r) := \frac{\left\langle (\delta_{r,\alpha} f_c)^4 \right\rangle_s}{\left\langle (\delta_{r,\alpha} f_c)^2 \right\rangle_s^2}. \tag{45}$$

The second and fourth moments are averaged over states and spatial locations before their ratio is formed. Flatness removes the overall amplitude scale and emphasizes large increments relative to typical fluctuations. Its scale dependence provides a diagnostic of spatial intermittency (Ma et al., 2021).

The reported relative error is

$$E_{\mathrm{flat}} = \frac{1}{4} \sum_{c \in \mathcal{C}_{\mathrm{inc}}} \sum_{\alpha \in \{x,y\}} \frac{\sum_{r \in \mathcal{R}} \left| \mathrm{Flat}_{c,\alpha}^{\mathrm{gen}}(r) - \mathrm{Flat}_{c,\alpha}^{\mathrm{DNS}}(r) \right|}{\sum_{r \in \mathcal{R}} \mathrm{Flat}_{c,\alpha}^{\mathrm{DNS}}(r)}. \tag{46}$$

For each field and direction, we sum the absolute differences over separations and divide by the sum of DNS flatness values. We then average these four relative errors.

B.3 Ablation of self-consistency loss weight λ

We study the effect of self-consistency weighting in hybrid data-CFM training on HW and KS, using the shared stationary windows from ten HW trajectories and 15 KS snapshots. For each system, we compare numerical-solver and frozen neural-operator push-forward against a shared data-only baseline. For both HW and KS, we compare self-consistency weights $\lambda \in \{0.01, 0.1, 1, 10\}$. The KS sweep across three training seeds covers every weight for both forms of supervision (Appendix E.5). Unless otherwise stated, ablations use the default settings and vary only the self-consistency weight. Evaluation follows Subsection B.4.

B.4 Evaluation and reporting protocol

For each configuration in the main tables, all metrics use the same fixed model weights. For KS and HW, the DNS reference trajectories are excluded from training. The main tables use test trajectories independent of those used for model selection. Checkpoint selection and ablation controls are specified in Appendix C.3. Within each evaluation repeat, all methods share the DNS reference and metric definitions. For KS, we use four evaluation repeats, each containing 4,096 generated samples. For HW, each of five evaluation repeats compares 64 generated fields with 64 DNS fields, one from each of 64 trajectories at $t = 3600$. Both ensembles vary with the sampling seed. The HW sliced Wasserstein-2 calculation uses all 64 fields in each ensemble and 128 projection directions. Each metric is computed separately for each repeat, and we report its mean and sample standard deviation across repeats. The HW standard deviations include variation from both generated and DNS reference samples at fixed trained weights. They describe evaluation variability rather than variation across training runs. The additional KS sweep in Appendix E.5 instead reports variation across three training seeds after averaging the four sampling repeats for each model. For L63, the displayed results use one trained model per method and full three-dimensional empirical W_2 on 1,024 samples from each model and DNS. The L63 reference is excluded from training.

B.5 Decimal display

In Tables 1 and 3, standard deviations are rounded to one significant digit, and means are rounded to the same decimal place. Small standard deviations retain the digits needed to avoid rounding nonzero values to zero. Rounding is applied only for display.

C Implementation Details and Hyperparameters

This section describes the implementations, main training settings, and hyperparameter searches for Lorenz–63 (L63), KS, and HW. Following the main text, δt denotes the sampled physical push-forward duration and ΔT its upper limit. IMG draws $\delta t \sim \mathcal{U}(0, \Delta T)$ in the reported L63, KS, and HW experiments, as in the theoretical formulation. The symbol $\tau \in [0, 1]$ is reserved for generative flow time. We distinguish these quantities from the physical prediction interval Δt_{NO} of a neural operator, the numerical PDE integration step h, and the number K of integration steps along the generative flow. The latter evolves over a dimensionless interval $[0, 1]$. A midpoint sampler uses $2K$ neural function evaluations (NFE), whereas Euler uses K. The training rollout length H counts consecutive neural-operator predictions.

C.1 Method implementations

IMG and data-trained conditional flow matching. IMG uses a time-conditioned velocity network and a standard Gaussian source. At each update, the current generator produces fresh samples. These samples are mapped to physical coordinates and advanced by the known dynamics over durations drawn independently from $\mathcal{U}(0, \Delta T)$, using the upper limits specified below. The advanced states are detached and used as endpoints for conditional flow matching. For independent source and target samples z and y, we sample $\tau \sim \mathcal{U}(0, 1)$, form $x_\tau = (1 - \tau)z + \tau y$, and minimize

$$\mathcal{L}_{\mathrm{CFM}} = \mathbb{E}\big[d^{-1}\|v_\theta(\tau, x_\tau) - (y - z)\|_2^2\big]. \tag{47}$$

Here d is the number of state coordinates. Squared-error losses are averaged over state coordinates and minibatch samples. The factor $1/d$ rescales the objective. The theoretical analysis uses

ideal flows. The numerical solvers and state constraints used in the experiments are specified in Appendix B.1 and below. Data CFM uses the available saturated training states as y. Data CFM + IMG combines this data objective with targets advanced by the numerical solver. Data CFM + IMG-neural uses a frozen, separately trained neural operator for the advancement. Neither the generation of a self-consistency target nor its subsequent advancement is backpropagated through. All these students are initialized from scratch.

All KS IMG variants use $\Delta T = 1$. HW uses $\Delta T = 1$ for pure IMG and $\Delta T = 0.6$ for both data-assisted solver supervision and IMG-neural. These values are upper limits, not fixed advancement times. With nominal solver steps of 0.0005 for KS and 0.0125 for HW, a target requires at most 2,000 KS steps or 80 and 48 HW steps, respectively. The actual number depends on the sampled duration. Neural targets use the same sampled duration as the corresponding solver-supervised variant.

Neural targets at intermediate physical times. Let h denote the physical interval between consecutive neural predictions. For a requested duration δt, set $m = \lfloor \delta t/h \rfloor$ and $\alpha = (\delta t - mh)/h$. Starting from $u^{(0)} = u$, take full steps $u^{(j+1)} = \widehat{\Phi}^h(u^{(j)})$ for $j < m$. If $\alpha = 0$, use $u^{(m)}$ directly. Otherwise, predict one further endpoint and interpolate only in the final interval,

$$\widehat{\Phi}^{\delta t}(u) = (1-\alpha)u^{(m)} + \alpha\widehat{\Phi}^h(u^{(m)}). \tag{48}$$

HW samples $\delta t \sim \mathcal{U}(0, 0.6)$ and uses neural steps of $h = 0.1$, while the U-Net baseline retains its 0.6 prediction interval. KS retains $h = \Delta T = 1$. The interpolant approximates physical evolution. Its error combines operator prediction error with local temporal interpolation error, as quantified in Appendix A.1. Appendix E.10 measures the effect of shortening the interpolation interval on saturated HW fields. This differs from the CFM interpolation in generative time τ, which joins an independent prior sample to the target. The entire target is held fixed when computing the sampler gradient.

FNO and U-Net neural operators. Both architectures predict a time-conditioned residual update to the input state. FNO uses Fourier spectral layers and pointwise channel mixing. U-Net uses an encoder–decoder with periodic convolutions and skip connections. KS physics training minimizes

$$\mathcal{L}_{\mathrm{PI}} = \mathbb{E}d^{-1}\|\partial_t\widehat{u}_\theta(u_0, t) - F(\widehat{u}_\theta(u_0, t))\|_2^2 + \mathbb{E}d^{-1}\|\widehat{u}_\theta(u_0, 0) - u_0\|_2^2, \tag{49}$$

where F is the spatially discretized PDE right-hand side. Time derivatives are obtained by automatic differentiation. Each initial field has four uniform time collocation points in $[0, \Delta t_{\mathrm{NO}}]$. Half the inputs are fresh Gaussian random fields (GRFs), and half come from a pool advanced only by the current neural operator. Pool states are refreshed at maximum physical age 200 or independently with probability $\Delta t_{\mathrm{NO}}/200$. No solver-generated target enters this KS physics loss.

The reported HW U-Net uses a differential PDE residual, a hard initial-condition parameterization, and fresh GRF inputs mixed with model-generated replay, with maximum physical age 300. Its four-step training rollout is differentiated end to end. The reported HW FNO uses GRF inputs and a different physical target: for an input state u, RK4 produces $\Phi^{\Delta t_{\mathrm{NO}}}(u)$, and the network minimizes normalized endpoint mean squared error (MSE) divided by Δt_{NO}^2. This is an online average-velocity objective. The RK4 target is detached. The model visits a sequence of detached predicted states. When its length exceeds four, two endpoints and two interior states are supervised. Thus the two HW physics-supervised rows have different target constructions.

For data supervision, KS operators fit exact pairs of stored states with their actual time separation, without interpolated labels. HW operators minimize equally weighted errors along a four-step free autoregressive rollout, retaining gradients through all four steps. Only inputs are perturbed, in normalized coordinates. Pairs and rollouts stay within each trajectory and training window.

Physics-informed DeepONet. The branch network encodes the initial state and the trunk network encodes space and query time. Their rank-r inner product gives the output field. The KS model uses a direct query over $t \in [0, 200]$, with fresh GRF initial states and the residual and soft initial-condition loss in Eq. 49. Its trunk inputs are $(\sin(2\pi x/L), \cos(2\pi x/L), t/200)$, and its output is divided by $\sqrt{r}$ and projected to spatial mean zero. The HW model is autoregressive (AR) with a hard initial condition and local interval $[0, \Delta t_{\mathrm{NO}}]$. Half of its local time queries are uniform and half are at the endpoint. Inputs combine fresh white noise with detached neural rollout states. The maximum prefix length follows the fixed schedule specified below.

DySLIM. We add a conditional maximum mean discrepancy (MMD) term to the supervised rollout loss, $\mathcal{L} = \mathcal{L}_{\mathrm{MSE}} + \lambda_{\mathrm{MMD}}\mathcal{L}_{\mathrm{cMMD}}$, following the conditional regularization construction of Schiff et al. (2024). The implementation uses the kernel $k(x, y) = \sum_{b\in\mathcal{B}} b^2/(b^2 + \|x-y\|_2^2)$ on full flattened standardized fields, without dividing distances by field dimension, a V-statistic including diagonal and paired cross terms, and no unconditional MMD term. The MSE and MMD branches use separate minibatches. The MMD branch receives no input noise. The KS result uses the same U-Net backbone as its supervised baseline. The HW result uses a periodic residual convolutional network with dilated convolutions.

Weak generative sampler. WGS penalizes weak stationarity using Gaussian tests ϕ_j:

$$\mathcal{L}_{\mathrm{WGS}} = \frac{1}{J}\sum_{j=1}^{J}\left(\frac{1}{B}\sum_{i=1}^{B}\mathcal{A}_D\phi_j(G_\theta(z_i))\right)^2 + \mathcal{L}_{\mathrm{boundary}}, \qquad \mathcal{A}_D\phi = F\cdot\nabla\phi + D\Delta\phi. \quad (50)$$

The particle average is taken before squaring. Gaussian tests have the form $\phi_{c,\ell}(x) = \exp(-\|x - c\|_2^2/(2\ell^2))$. For KS/HW, $\ell^2 = d_{\mathrm{active}}\kappa^2$, with $d_{\mathrm{active}} = 256 - 1$ for KS and $2\cdot 256^2 - 2$ for the zero-mean HW generator. The three width-group losses are averaged. For L63, $\ell = b$ in physical coordinates. KS and HW use $D = 0$, the discretized PDE right-hand side, dimension-normalized test widths, and detached test centers sampled from generated states and perturbed by Gaussian noise. Their boundary term is $w_b\mathbb{E}[\mathrm{sigmoid}(a_b(\|x\|_2^2/d - r_b^2))]$, which penalizes excessive root-mean-square (RMS) field amplitude. Its parameters are listed below. KS generates through a two-step Euler integration of a shared U-Net velocity field. HW uses a multiscale Haar convolutional real-valued non-volume-preserving (RealNVP) generator built from affine coupling layers.

For L63, we adapt the WGS authors' RealNVP implementation, including its Gaussian tests and coordinatewise boundary penalty.[2] For L63, $D = 0.1$ is the physical diffusion shared by IMG, WGS, ADDA, and the DNS reference in Eq. (25). It is distinct from test-center perturbation. Unlike the KS/HW implementation, L63 retains gradients through test centers and works directly in physical coordinates. Every ten updates it draws 10,000 latent samples, recomputes their model outputs at each update, and divides shuffled center indices into ten minibatches of 1,000. Center perturbations are fresh at every update.

Adaptive deep density approximation on L63. Our ADDA baseline uses the RealNVP comparison implementation accompanying WGS. It is not a reproduction of the original KRnet architecture of Tang et al. (2022), based on the Knothe–Rosenblatt rearrangement. It minimizes the squared density residual

$$R_\theta(x) = -\nabla\cdot(F(x)p_\theta(x)) + D\Delta p_\theta(x), \qquad \mathcal{L}_{\mathrm{ADDA}} = \lambda_{\mathrm{PDE}}\mathbb{E}_{x\sim q_k}[R_\theta(x)^2] + \mathcal{L}_{\mathrm{boundary}}. \quad (51)$$

Here q_k is an adaptive collocation pool. Although derivatives are computed through $\log p_\theta$, the multiplicative density p_θ is retained in the residual. The first 60,000-point pool is uniform on $[-30, 30]\times[-40, 40]\times[-15, 65]$. Subsequent pools are detached model samples. A pool is refreshed every 400 updates. Residual minibatches contain 2,000 points, shuffled without replacement. A separate set of 60,000 Gaussian latent samples is fixed within each stage for the boundary term, with model outputs recomputed each update. The reported configuration has $D = 0.1$ and $\lambda_{\mathrm{PDE}} = 10^6$. The latter is an explicitly tuned residual scaling, not an author-reported default. IMG, WGS, and DNS use these same stochastic L63 dynamics.

C.2 Training and model configurations

Optimization. Table 4 lists the training budgets and learning rates. The main comparisons use one common training seed. The KS solver- and neural-supervision sweeps, including their shared data-only baseline, each use three training seeds (Appendix E.5). Unless noted otherwise, we use Adam (adaptive moment estimation) with $(\beta_1, \beta_2) = (0.9, 0.99)$ and $\epsilon = 10^{-8}$. Weight decay is zero, and the gradient norm is clipped at 1. After warm-up, the learning rate follows a cosine schedule to 10% of its peak. Models use 32-bit floating-point precision, except L63 WGS and ADDA, which use 64-bit precision.

[2]Implementation source: `https://github.com/Czqstrong/WGS_code`, revision `a87e0b30b1f18dbba13f571f9f775797a8fb466e`.

Table 4: Training settings. Updates specify the full training budget. Paired batch sizes refer to separate loss branches. LR denotes learning rate.

System	Method	Batch size	Peak LR	Warm-up	Updates
L63	IMG	256	10^{-4}	500	2,000
L63	WGS	10,000	10^{-3}	0	2,000
L63	ADDA	2,000	10^{-4}	0	2,000
KS	IMG / Data CFM	50	$2 \cdot 10^{-4}$	1,000	20,000
KS	Data CFM + IMG	$50/50$	$2 \cdot 10^{-4}$	1,000	20,000
KS	FNO / U-Net / PI-DeepONet	32	10^{-4}	1,000	20,000
KS	DySLIM	$32/32$	10^{-4}	1,000	20,000
KS	WGS	64	10^{-4}	1,000	20,000
HW	IMG	50	$2 \cdot 10^{-4}$	1,000	20,000
HW	Data CFM	50	$2 \cdot 10^{-4}$	1,000	6,000
HW	Data CFM + IMG	$50/50$	$2 \cdot 10^{-4}$	1,000	6,000
HW	FNO / U-Net / AR PI-DeepONet	4	10^{-4}	1,000	20,000
HW	DySLIM	$4/4$	10^{-4}	1,000	20,000
HW	WGS	256	10^{-4}	0	20,000

Both solver and neural variants of Data CFM + IMG use the mixed-loss settings in the table. L63 WGS uses 1,000 test centers per update. L63 WGS and ADDA use Adam $(0.9, 0.999)$ with a constant learning rate and no gradient clipping. HW WGS uses microbatches of 32 and exponential learning-rate decay to 10^{-5}. HW AR PI-DeepONet accumulates four single-example gradients.

L63 models. All methods and the DNS reference share $(\sigma, \rho, \beta) = (10.75, 35, 8/3)$ and diffusion $D = 0.1$. Physical evolutions use $h = 0.01$. IMG draws physical durations from $\mathcal{U}(0, \Delta T)$ with $\Delta T = 1$ and uses 20 midpoint steps for generative sampling. Its velocity network is a width-64 residual multilayer perceptron with three blocks. Each block has internal width 256. The network uses a 64-dimensional sinusoidal time embedding, sigmoid linear unit activations, and layer normalization. The fixed normalizer has mean $(0, 0, 25)$ and standard deviation $(12, 12, 10)$. No exponential moving average, dropout, or replay is used.

WGS and ADDA use 12 affine coupling layers. Their scale and translation networks each have three linear layers of hidden width 32, with leaky rectified linear units of negative slope 0.01. The scale output uses $\tanh$. Linear weights are zero at initialization, while biases retain their initialized values. The source is $\mathcal{N}(0, 400 I_3)$, and both models operate in physical coordinates. For samples $X \sim p_\theta$, their shared boundary penalty is

$$\mathcal{L}_{\text{boundary}} = 5 \sum_{j=1}^{3} \mathbb{E}\left[\text{sigmoid}\big(5[(X_j - c_j)^2 - r_j^2]\big)\right], \tag{52}$$

where $c = (0, 0, 25)$ and $r = (30, 40, 40)$. WGS uses test width $b_s = 5(3/5)^{10\lceil s/10 \rceil / 2000}$, center-noise standard deviation 5, and diffusion coefficient $D = 0.1$. ADDA uses $D = 0.1$ and residual weight $\lambda_{\text{PDE}} = 10^6$. Its collocation pool is refreshed every 400 updates.

KS and HW architectures. The KS U-Nets and HW IMG/CFM generators use base width 64 and channel multipliers $(1, 2, 4, 4)$. Each level has two residual blocks. The networks use bottleneck attention, a 64-dimensional time embedding, and group normalization with 32 groups. Dropout is disabled. HW operator U-Nets use base width 32 and channel multipliers $(1, 2, 4, 8)$, with two blocks per level. All convolutions respect periodic boundaries.

FNO uses width 64, four spectral layers, and 32 retained modes. HW FNO tapers modes 16–32 with a cosine filter. KS enforces a real zero-frequency coefficient and Hermitian symmetry. The data-trained KS FNO is residual, uses time-conditioning scale 1, and advances one physical time unit per call.

KS PI-DeepONet uses four tanh layers of width 128 and rank 128. HW AR PI-DeepONet uses four tanh layers of width 256 and rank 128. Its spatial Fourier frequencies are $\{1, 2, 4, 8, 16, 32, 64\}$. HW DySLIM uses 96 channels and four residual processors. Each processor has convolutional dilations

$(1, 2, 4, 8, 4, 2, 1)$. HW WGS has eight Haar scales with $(2, 2, 2, 2, 2, 2, 4, 4)$ affine couplings. Its conditioners have width 64 and two residual blocks.

Table 5: Main KS/HW settings. Physics-supervised and data-assisted variants use the losses described in Section C.1.

Parameter	KS	HW
IMG maximum duration	$\Delta T = 1$	$\Delta T = 1$
Data-assisted solver IMG maximum duration	$\Delta T = 1$	$\Delta T = 0.6$
IMG-neural maximum duration	$\Delta T = 1$	$\Delta T = 0.6$
Few-data solver SC weight	$\lambda = 10$	$\lambda = 1$
Neural SC weight	$\lambda = 1$	$\lambda = 0.1$
Neural teacher	Data-trained U-Net	U-Net, baseline architecture and other hyperparameters
CFM training sampler	Midpoint, $K = 20$	Midpoint, $K = 20$
CFM evaluation sampler	Midpoint, $K = 20$	Midpoint, $K = 10$
Baseline operator interval	$\Delta t_{\mathrm{NO}} = 1$	$\Delta t_{\mathrm{NO}} = 0.6$
Neural target advancement step	$h = 1$	$h = 0.1$
Data-trained operator rollout	Stored state pairs	$H = 4$
DySLIM weight	$\lambda_{\mathrm{MMD}} = 10^{-4}$	$\lambda_{\mathrm{MMD}} = 100$
WGS generator	U-Net Euler flow, $K = 2$	RealNVP, 20 couplings
Stationary training data	15 snapshots	10 trajectories, $t \in [150, 160]$, spacing 0.1

Training data. The KS pool contains 15 snapshots from one trajectory at $\{100, 101, 110, 120, 121, 130, 140, 141, 150, 160, 161, 170, 180, 181, 190\}$. Data CFM, FNO, U-Net, DySLIM, and the frozen neural teacher share this stationary training pool. Data CFM fits the snapshots directly. The data-trained operators use stored pairs with their actual time separations. The KS neural teacher is the same data-trained U-Net used as a baseline, frozen during sampler training.

The data-trained KS FNO uses pairwise MSE on this same stationary training pool. No additional solver labels are generated. Training starts from random initialization and uses neither input noise nor an MMD penalty.

HW data-assisted methods share stationary windows from ten trajectories over $t \in [150, 160]$, stored at physical-time spacing 0.1. Data CFM samples fields from this pool. Neural operators, including the frozen teacher, sample input states and the corresponding targets from the same trajectories at their configured prediction intervals. Four-step training sequences remain within these windows. Thus 0.1-interval targets are available directly from the stored trajectories, while longer-interval configurations use more widely separated fields from the same pool. The data-trained HW operators use input-noise standard deviation 0.01 in normalized coordinates. The HW IMG-neural teacher uses the architecture and other hyperparameters of the data-trained U-Net baseline in Table 3, with neural advancement interval $h = 0.1$. It is frozen during sampler training, and the maximum total advancement duration remains $\Delta T = 0.6$.

HW spatial means. Write $Q_\zeta(\zeta, n) = (\zeta - \langle\zeta\rangle, n)$, where brackets denote the spatial mean. Pure HW IMG applies Q_ζ to the source and intermediate generative states, the generated physical states, and the evolved targets. Its numerical target map is $Q_\zeta \Phi^{\delta t}$ on generated states. There is no per-sample projection of the density mean. The RK4 solver evolves both means, while the Poisson solve sets the potential zero mode to zero. WGS projects both channels. Evaluation centers vorticity for every method and DNS, as described in Appendix B.2.

Rollout schedules and kernels. HW FNO increases its detached rollout length through $H = 1, 2, 5, 10, 20, 50, 100$. These stages begin at updates $1, 1001, 2001, 4001, 6001, 10001, 14001$. Each numerical target uses 48 RK4 steps of size 0.0125. HW AR PI-DeepONet uses white-noise standard deviation 0.1 and anchor fraction 0.5. Its detached prefix cap grows from zero at update 1,000 to 100 at update 10,000. KS solver targets use ETDRK4 with step size 0.0005.

The DySLIM kernel widths are $\{0.2, 0.5, 0.9, 1.3, 2, 5, 9, 13, 20, 50, 90, 130\}$ for KS and $\{2, 5, 9, 13, 20, 50, 90, 120\}$ for HW. Each list defines one simultaneous kernel mixture. Table 6 gives the WGS test-function settings.

Table 6: Fixed WGS settings for KS and HW. All arrows span the complete 20,000-update schedule.

Setting	KS	HW
Test-width groups	$(2, 1, 0.5) \to (1, 0.5, 0.25)$, exponential	1.5 fixed; $1.5 \to 0.75$ exponential; $\mathcal{U}(0.7, 0.98)$ per center
Center-noise standard deviation	$0.5 \to 0.1$, exponential	$0.7 \to 0.14$, exponential
Tests per group / total	16 / 48	24 / 72
Boundary weight / RMS radius	10 / 2	10 / 2
Boundary sharpness	10	6
Center gradients	Detached	Detached
Normalization	Identity	Identity
Spatial mean projection	KS field	Both ζ and n

Inference. The generative samplers use the step counts in Table 5. HW WGS uses one RealNVP pass. KS autoregressive operators advance to $T = 100$ in unit-time steps, while KS PI-DeepONet queries $T = 100$ directly. HW autoregressive models make 3,600 calls to reach $T = 2160$. They retain 64 fields after $T = 400.2$ using fixed random indices. This amounts to 56.25 operator calls per retained field. For the generation-cost comparison, we vary midpoint steps over $K \in \{1, 2, 5, 10, 20, 40\}$ at fixed trained weights.

C.3 Checkpoint selection and ablation controls

Model selection uses validation data independent of the final test data. The validation and test DNS references come from independent trajectories. All hyperparameters chosen through a search are determined using the validation data. Test data are not used to select checkpoints or hyperparameters. The data-free designation refers to IMG training targets, which require no precomputed saturated-state dataset. DNS validation data are still used for model selection. Table 7 summarizes the training, validation, and test data.

KS. For each saved checkpoint, we average each validation metric over four evaluation repeats. We select the checkpoint minimizing the geometric mean

$$J_{\mathrm{KS}} = \left[\frac{\overline{\widehat{\mathrm{SW}}_2}}{\sigma_{\mathrm{ref}}} \, \overline{E_{\mathrm{spec}}} \, \overline{E_{\mathrm{PDF}}} \right]^{1/3}.$$

Bars denote averages over the four repeats. The reference scale σ_{ref} is the standard deviation of the validation reference field values. The errors E_{spec} and E_{PDF} denote spectral relative L_1 error and marginal PDF-TV. Ties favor the earlier checkpoint.

HW. For both generators and neural operators, we select the checkpoint minimizing the largest validation error among the spectral relative L_1 and marginal PDF-TV errors for density n and vorticity ζ. HW IMG-neural follows this same rule with maximum duration $\Delta T = 0.6$ and neural advancement step $h = 0.1$. The frozen U-Net teacher uses the baseline architecture and other hyperparameters, with the same validation selection rule.

L63. We use the terminal model after 2,000 updates for each method.

Ablation controls. Within each ablation comparison, we vary only the factor under study. All other hyperparameters, the checkpoint-selection rule, and the evaluation protocol remain fixed. In particular, weight ablations apply the same selection protocol to every weight, including the data-only baseline. All reported metrics for a configuration use the same selected model.

C.4 Main hyperparameter searches

For IMG, we varied the maximum physical advancement time ΔT and the self-consistency weight. The operator searches varied the prediction interval and regularization. For HW, the data stor-

Table 7: Data used for training, validation, and final testing. Validation and test references come from separate trajectories. Training sample details are given in Appendix C.2, selection rules in Appendix C.3, and evaluation metrics in Appendix B.2.

Use	System / method	Trajectories and physical times	Role and statistics
Training	IMG, all systems	Model-generated states advanced over $\delta t \sim \mathcal{U}(0, \Delta T)$. No precomputed saturated-state trajectories.	Advanced states supply CFM targets. Maximum durations follow Table 5 and the L63 settings.
Training	KS, data-assisted	One trajectory supplies 15 saturated snapshots over $t \in [100, 190]$. Exact times are listed in Appendix C.2.	CFM and the data-trained operators share these snapshots. Operators use stored transitions. The neural teacher is the frozen U-Net baseline.
Training	HW, data-assisted	Ten trajectories over $t \in [150, 160]$, stored at spacing 0.1.	CFM samples fields from this shared stationary pool. Operators and the frozen teacher sample trajectory segments at their configured prediction intervals.
Validation	KS	Validation DNS trajectories, separate from training and final-test trajectories.	Four repeats per checkpoint. Minimize the geometric mean of reference-scale-normalized SW_2, spectral relative L_1, and PDF-TV. Ties favor the earlier checkpoint.
Validation	HW	Validation DNS trajectories, separate from training and final-test trajectories.	Generators and neural operators minimize the largest spectral relative L_1 or PDF-TV error across n and ζ.
Validation	L63	Validation DNS trajectories for hyperparameter comparisons, separate from final-test trajectories.	Compare configurations at their 2,000-update endpoints. The reported model for each method is its terminal checkpoint.
Final test	KS	Independent test DNS trajectories, retaining states at $t \in [100, 199]$.	SW_2, spectral relative L_1, and PDF-TV. Four repeats, each with 4,096 generated fields.
Final test	HW	64 DNS trajectories per repeat, each contributing one field at $t = 3600$.	Five repeats, each comparing 64 model and 64 DNS fields. Spectrum, PDF, increment, and flatness errors. SW_2 appears in Appendix E.1.
Final test	L63	Independent test DNS reference, with 1,024 reference points per comparison.	Full three-dimensional empirical W_2 against 1,024 model samples, using the terminal model.

age spacing remains 0.1 throughout these searches. The additional U-Net interval ablation uses $h \in \{0.1, 0.3, 0.6\}$ (Appendix E.10). WGS searches covered generator architecture, test width, and center noise. Tables 8–10 list the search ranges. Cartesian products apply only within the indicated grids.

Table 8: Main searches for physics-supervised KS/HW methods.

System	Method / parameter	Values or architectures
KS	IMG maximum duration ΔT	$\{0.1, 0.3, 0.6, 1\}$
HW	IMG maximum duration ΔT	$\{0.3, 0.6, 1, 1.2, 1.5\}$
KS	FNO / U-Net interval	$\Delta t_{\mathrm{NO}} \in \{0.2, 1, 2\}$
HW	FNO / U-Net interval	$\Delta t_{\mathrm{NO}} \in \{0.2, 0.6, 2\}$
HW	AR PI-DeepONet	Intervals $\{0.1, 0.6\}\times$ ranks $\{128, 256\}$
KS	WGS architecture	Direct U-Net, two 12-layer RealNVP variants, and U-Net Euler flows with $K \in \{2, 5\}$
HW	WGS architecture	Direct U-Net and convolutional RealNVP with $\{10, 20, 40\}$ couplings

Table 9: Main searches for data-assisted KS/HW methods. ConvNet denotes a convolutional network. Training data for the samplers, operators, and frozen teachers are specified in Appendix C.2.

System	Method / parameter	Values
KS	Solver SC weight	$\lambda \in \{0, 0.01, 0.1, 1, 10\}$
HW	Solver SC weight	$\lambda \in \{0, 0.01, 0.1, 1, 10\}$
HW	IMG-neural advancement	Maximum duration $\Delta T = 0.6$, neural step $h = 0.1$
KS	IMG-neural teacher	Four frozen teachers spanning FNO, U-Net, and regularized ConvNet
KS	FNO / U-Net interval	$\Delta t_{\mathrm{NO}} \in \{1, 10, 20\}$
HW	FNO / U-Net interval	$\Delta t_{\mathrm{NO}} \in \{0.1, 0.6, 1.2, 2.4\}$
KS	DySLIM weight	$\lambda_{\mathrm{MMD}} \in \{0, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$
HW	DySLIM weight / bandwidth	$\{0.01, 0.1, 1, 10, 100\} \times \{0.5, 1, 2\}$, plus a zero-MMD control

Additional KS operator searches varied FNO width over $\{32, 64\}$ and compared ConvNet and U-Net backbones. The ConvNet search also varied input noise and kernel bandwidth. These choices were not combined into a full Cartesian grid. The default-setting weight ablations appear in Appendix E.4.

The exploratory L63 WGS width grid fixes $D = 0$. The diffusion grid fixes width endpoints $(5, 3)$ and center-noise standard deviation 5. The reported comparison uses the common stochastic setting $D = 0.1$ for all methods and DNS. ADDA uses two sequential grids, with a finer logarithmic residual-weight search in the second. Together they cover 48 distinct configurations across 52 training runs. One refinement configuration failed the push-forward evaluation.

D Computational cost

IMG replaces the preparation of a stationary training dataset with repeated generation and short-time advancement of model samples. This changes where computation is spent. For the HW configuration considered here, the largest training expense is generative sampling, while the short physical evolution has a cost comparable to a single network training forward–backward pass.

The complete HW run uses 20,000 updates with batch size 50 and takes 52.71 graphics processing unit (GPU) hours on one H800, including online evaluation and logging. Each update constructs targets by generating fields with 20 midpoint steps and advancing them over sampled durations $\delta t \sim \mathcal{U}(0, \Delta T)$, with $\Delta T = 1$ and nominal RK4 step size 0.0125. Midpoint generation requires 40 network evaluations per field, giving 4.0×10^7 sample-level network evaluations over training. Each target requires at most 80 RK4 steps, so 8.0×10^7 sample-level steps is an upper bound for training-target advancement, excluding initialization and diagnostics. The actual count depends on the sampled durations. Each RK4 step uses four right-hand-side evaluations. The cost of constructing these online teacher targets is already included in the training time.

Table 10: Main L63 baseline grids. WGS has 36 zero-diffusion and 15 diffusion configurations, with three overlaps. The two ADDA grids have 16 and 36 combinations, with four overlaps. Each grid is a separate Cartesian product. Every configuration uses one shared training seed and 2,000 updates.

Study	Parameter	Values
WGS, $D = 0$	Width endpoints (b_0, b_T)	$\{(1,1),(3,3),(5,3),(10,5)\}$
	Center-noise standard deviation	$\{0, 2, 5\}$
	Learning rate	$\{2 \cdot 10^{-4}, 5 \cdot 10^{-4}, 10^{-3}\}$
WGS diffusion	Diffusion D	$\{0, 0.1, 1, 5, 20\}$
	Learning rate	$\{2 \cdot 10^{-4}, 5 \cdot 10^{-4}, 10^{-3}\}$
ADDA initial	Diffusion D	$\{0, 0.1, 1, 20\}$
	Learning rate	$\{10^{-4}, 10^{-3}\}$
	Residual weight λ_{PDE}	$\{1, 10^9\}$
ADDA refinement	Diffusion D	$\{0, 0.1\}$
	Learning rate	$\{10^{-4}, 3 \cdot 10^{-4}, 10^{-3}\}$
	Residual weight λ_{PDE}	$\{10^5, 10^6, 10^7, 10^8, 10^9, 10^{10}\}$

Table 11: HW training cost per update on one A100-SXM4-80GB GPU. Entries are median synchronized wall-clock times over updates 3–12 of a 12-update profile with the same training configuration. The full update includes target processing and optimizer work. These measurements are separate from the complete H800 training time.

Computation	Time (s)
Generative sampling	13.265
Physical push-forward	1.785
Loss forward	0.333
Loss backward	1.071
Full update	16.474

Table 11 separates the main update components. The physical push-forward takes 1.785 s, compared with 1.404 s for the loss forward and backward passes together, and accounts for approximately 11% of the update time. The physical interval is short, and target construction is detached from the gradient computation. Generative sampling accounts for approximately 81% of the measured update time. A few-step generative model could therefore reduce the dominant expense in online target construction as well as the cost of final sampling. This is a direction for further acceleration, whose statistical accuracy and any additional training or distillation cost must be assessed together.

IMG-neural provides a complementary reduction by replacing numerical advancement with a frozen neural operator $\widehat{\Phi}^{\delta t}$. Its online target advancement requires zero numerical-solver steps, and direct sampling from the trained generator likewise requires no physical solver. For HW, the frozen U-Net uses the baseline architecture and other hyperparameters, with the advancement interval specified in Appendix C.2. Neural-operator inference remains part of the computation. Operator training and training-data generation remain upfront costs when accounting for the full pipeline. Thus few-step generation and neural advancement address different components of the training expense.

For comparison, our HW DNS timing includes evolution from independent Gaussian white noise of standard deviation 10^{-9} in each channel. On the same A100 used for profiling, reaching the prescribed burn-in cutoff $t = 400$ requires 32,000 RK4 steps and 211.98 s. Continuing to $t = 800$ and retaining one field every 10 time units yields 41 correlated fields at a total cost of 64,000 RK4 steps and 422.63 s, including storage. The cutoff specifies the timing protocol rather than certifying stationarity. An end-to-end comparison must account for training, operator pretraining where applicable, and temporal correlations in DNS samples. These measurements characterize computational cost and its main reduction opportunities; they do not establish an equal-quality or equal-effective-sample speedup over DNS.

E Additional Experiment Results

E.1 HW sliced-Wasserstein results and reference calibration

Table 12 retains the HW coarse-field sliced-W_2 results corresponding to Tables 1 and 3. Each comparison uses 64 generated fields and 64 DNS reference fields. Both ensembles vary across the five evaluation repeats with the sampling seed. The reported standard deviations therefore include sampling variability from both ensembles, with model weights held fixed.

Table 12: Supplementary HW coarse-field sliced-W_2 against DNS. Values are means $\pm$ sample standard deviations over five evaluation repeats. The models and evaluation protocol are those of the main tables. Lower is better.

Method	Coarse sliced-W_2
Physics-supervised methods	
FNO	1.19 ± 0.05
U-Net	1.24 ± 0.03
PI-DeepONet	2.446 ± 0.008
WGS	1.70 ± 0.02
IMG (ours)	1.13 ± 0.02
Few-data methods	
FNO	1.13 ± 0.03
U-Net	1.13 ± 0.03
DySLIM	1.11 ± 0.03
Data CFM	1.12 ± 0.02
Data CFM + IMG (ours)	1.13 ± 0.02
Data CFM + IMG-neural (ours)	1.12 ± 0.02

DNS reference calibration. For this calibration, we compare one 64-field DNS reference ensemble with 64 additional DNS fields from disjoint trajectories at the same physical time $t = 3600$. We use the same preprocessing and the same 128 projection directions as in the model evaluation. The resulting reference–reference distance is 1.1065. With the projection directions held fixed, randomly repartitioning these 128 fields into two disjoint groups of 64 gives 1.089 ± 0.030 over 100 partitions, where the spread is the sample standard deviation. These partitions reuse the same finite DNS pool, rather than constituting 100 independently generated datasets. The additional DNS trajectories are not reserved for testing, so this comparison calibrates the reference discrepancy rather than providing a new held-out model test.

The reference discrepancy is comparable to the 1.11–1.13 scores of the few-data models. At the same sample size of 64 fields per ensemble, it provides an empirical finite-sample noise scale, not a universal lower bound or a correction to subtract from model scores. The small differences between those scores do not establish a reliable ranking or equivalence of stationary distributions.

Interpretation. This diagnostic has limited discrimination among the stronger HW models under our evaluation protocol. FNO and solver-assisted CFM receive nearly identical scores despite their markedly different spectral accuracy. The diagnostic still separates larger discrepancies, such as those of PI-DeepONet and WGS.

The feature map and sample size constrain what the reported value tests. Pooling removes fine-scale information, and 128 scalar projections summarize an 8,192-dimensional field representation using 64 observations per ensemble. These observations may highlight the difficulty of estimating distributional discrepancies from limited samples in high-dimensional state spaces. The concatenated vorticity and density features retain their physical amplitudes without channel standardization. Their contributions are therefore not balanced by a common dimensionless scale. A similar aggregate distance need not imply similar spectra, spatial increments, or higher-order statistics. We retain this diagnostic as supplementary evidence and use those complementary observables for the main HW comparison.

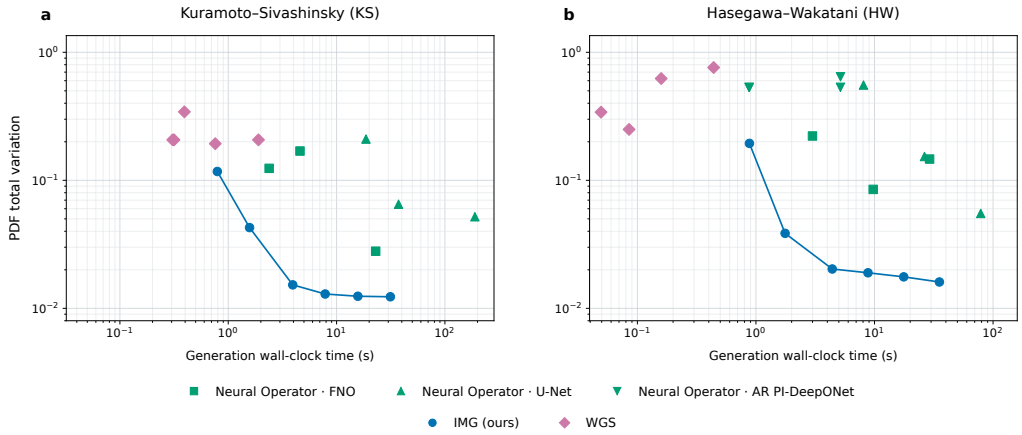


Figure 6: Generation time versus marginal PDF total variation for the evaluated configurations on (a) KS and (b) HW. Both axes use logarithmic scales. Lower values are preferable. Symbols distinguish methods, and the blue line connects the displayed IMG operating points as a visual guide. The comparison concerns generation time and marginal PDF-TV. Only generation is timed.

E.2 KS FNO UNDER EXTENDED ROLLOUT

The KS FNO entry in Table 3 uses 4,096 independent initial fields advanced to $t = 100$, at 100 neural evaluations per sample. We also assess the same trained model by retaining 4,096 consecutive states from $t = 100$ to 4195 at unit intervals, with four independent runs. This longer rollout requires 4,195 neural evaluations per run. Its mean sliced-W_2 and relative spectral error increase to approximately 112 and 2.3×10^4, respectively, with PDF-TV 0.87 ± 0.07. Thus the finite-horizon ensemble evaluation does not establish accurate long-time sampling by repeated FNO rollout. One possible mechanism is that propagation errors move the trajectory away from the states represented in its small training set, after which repeated application amplifies the mismatch.

E.3 GENERATION TIME AND MARGINAL STATISTICAL ACCURACY

Figure 6 complements the statistical comparisons with measured generation wall-clock time for KS and HW. Within each system, the comparison uses a common output sample count. Autoregressive generation includes the burn-in needed before retaining samples. The horizontal axis concerns generation after training, rather than the cost of training, hyperparameter search, or operator pretraining. Absolute timings depend on the implementation and execution setting.

Among the displayed configurations, IMG reaches lower marginal PDF-TV as its generation time increases, with smaller improvements at the larger measured times. Several displayed neural-operator configurations have both higher PDF-TV and longer generation times than an IMG operating point. Some WGS configurations generate samples faster than the fastest displayed IMG configuration, but have higher PDF-TV. Thus, this comparison does not identify IMG as the fastest sampler. Any Pareto interpretation is restricted to the displayed operating points and these two axes. It does not establish an optimum over untested configurations or superiority in spectrum, joint-distribution, or invariance diagnostics. The plot shows point estimates without uncertainty bars and is therefore a descriptive comparison, not a test of statistically significant differences. Equal field counts need not give equal effective sample sizes for direct and trajectory-based samplers.

E.4 THE EFFECT OF THE SELF-CONSISTENCY LOSS WEIGHT λ

Figures 7 and 8 extend the PDF-TV comparison in Figure 5 to the other statistical metrics. The ablations use the default settings and vary only the self-consistency weight. All metrics for a configuration use the same trained model. In these two weight-ablation figures, error bars show sampling variability at fixed trained weights. Appendix E.8 resolves the spectra and increment statistics for the default HW configurations.

HW. The improvement from self-consistency extends beyond the point-value PDF, but depends on both its weight and the advancement model. With solver push-forward, $\lambda = 1$ gives the lowest displayed spectral, PDF-TV, increment-PDF-TV, and flatness errors among the tested weights. The agreement across these diagnostics indicates improvement in both the distribution of fluctuation power and the spatial differences between nearby points. Neural-operator push-forward improves spectral and point-value PDF accuracy at $\lambda \in \{0.01, 0.1\}$, but its spectral and point-value PDF errors increase at $\lambda = 1$, even though increment and flatness errors remain below the data-only values. At $\lambda = 10$, both advancement choices have larger errors than the data-only baseline in these four diagnostics. In contrast, pooled-field sliced-W_2 changes little across most of the sweep. Thus, this coarse-field distance is less sensitive to the changes exposed by spectra and spatial increments. The deterioration at the largest weight also occurs with solver targets, which points beyond neural approximation error. A possible explanation is that emphasizing self-consistency too strongly weakens the fit to the sampled stationary distribution. Approximate self-consistency alone need not favor the same statistics as that data.

KS. Increasing the solver-based weight from 0.1 to 10 reduces all three displayed errors. The $\lambda = 10$ configuration is reported in Table 3. Improvement in both the spectrum and amplitude PDF suggests that the extra supervision constrains spatial power without sacrificing the distribution of field values. Neural-operator supervision gives its lowest displayed PDF-TV at $\lambda = 1$, the neural configuration reported in Table 3. At $\lambda = 10$, its PDF-TV increases relative to $\lambda = 1$, but all three errors remain below the data-only baseline. Unlike solver supervision, stronger neural supervision therefore introduces a tradeoff between constraints on spatial power and fidelity to the amplitude distribution. Increasing the weight may amplify frozen-propagator and interpolation errors. The sweep does not separate these errors from optimization effects.

E.5 KS VARIATION ACROSS TRAINING SEEDS

We train the sampler independently three times for both solver and neural supervision at each weight $\lambda \in \{0.01, 0.1, 1, 10\}$. The shared data-only baseline at $\lambda = 0$ also uses three independent trainings. All runs use the same 15 saturated snapshots, and all neural-supervision runs share one frozen U-Net teacher. For each trained model, we first average four sampling evaluations of 4,096 fields. Table 13 and Figures 9–10 report the mean and sample standard deviation across the three trainings. The error bars therefore measure variation after retraining, rather than sampling variability at fixed model weights.

Both forms of supervision improve the mean of every metric over Data CFM at all four nonzero weights. The gains across training seeds indicate that the benefit is not confined to one sampler initialization. Solver supervision at $\lambda = 10$ gives the lowest mean errors in all three metrics, with little variation in coarse-field sliced-W_2 across trainings. This agreement suggests that strong solver supervision can improve both spatial power and field-amplitude statistics consistently in this setting. The dependence on weight is not monotonic at smaller weights.

For neural supervision, $\lambda = 10$ gives the lowest mean spectral and coarse-field errors, whereas $\lambda = 1$ gives the lowest mean PDF-TV. The larger PDF-TV variation at $\lambda = 10$ indicates that amplitude statistics are more sensitive to initialization under strong neural supervision. One possible explanation is that stronger weighting amplifies teacher and interpolation errors while weakening the fit to the observed snapshots. The sweep does not separate these effects from optimization variability.

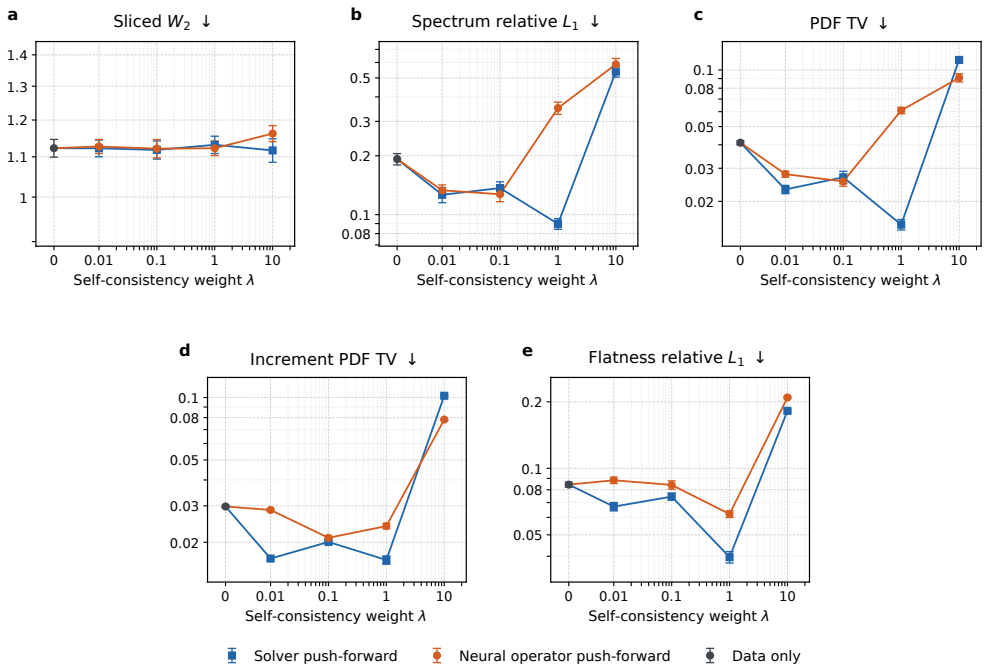


Figure 7: Full-metric HW loss-weight ablation. Panels show (a) pooled-field sliced-W_2, (b) relative spectral L_1 error, (c) point-value PDF TV, (d) increment PDF TV, and (e) relative flatness L_1 error against DNS. Lower is better. Blue squares and orange circles use solver and frozen neural-operator push-forward, respectively. The gray point is their shared data-only baseline at $\lambda = 0$. Points and bars are means and sample standard deviations over five evaluation repeats. All panels use the same trained model per configuration.

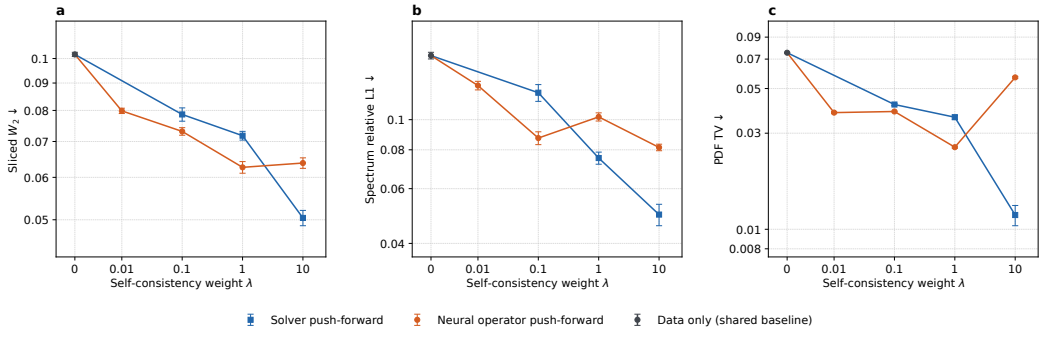


Figure 8: Full-metric KS loss-weight ablation using default settings. Panels show (a) sliced-W_2, (b) relative spectral L_1 error, and (c) point-value PDF TV against DNS. Lower is better. Colors and markers distinguish solver push-forward, frozen neural-operator push-forward, and the shared data-only baseline. Points and bars denote means and sample standard deviations over four sampling repeats at fixed trained weights. The solver configuration at $\lambda = 0.01$ is absent from this fixed-model comparison, so the connecting line does not represent a measurement there. It is included in the sweep across three training seeds in Appendix E.5.

Table 13: KS weight sweep with three independent training seeds per configuration. Entries are mean ± sample standard deviation across trainings, after averaging four sampling evaluations within each training. The data-only baseline is shared by both supervision variants. Bold marks the lowest mean in each metric.

Supervision	λ	Coarse $SW_2 \downarrow$	Spectrum rel. $L_1 \downarrow$	PDF-TV $\downarrow$
Data only	0	0.1039 ± 0.0066	0.1653 ± 0.0112	0.0738 ± 0.0015
Neural	0.01	0.0752 ± 0.0044	0.1135 ± 0.0150	0.0380 ± 0.0047
Neural	0.1	0.0688 ± 0.0049	0.1279 ± 0.0422	0.0383 ± 0.0005
Neural	1	0.0666 ± 0.0034	0.1136 ± 0.0173	0.0281 ± 0.0065
Neural	10	0.0634 ± 0.0005	0.0904 ± 0.0104	0.0368 ± 0.0178
Solver	0.01	0.0720 ± 0.0056	0.1198 ± 0.0308	0.0382 ± 0.0020
Solver	0.1	0.0783 ± 0.0006	0.1257 ± 0.0262	0.0412 ± 0.0042
Solver	1	0.0703 ± 0.0028	0.1002 ± 0.0216	0.0357 ± 0.0049
Solver	10	$\mathbf{0.05043} \pm 0.00024$	$\mathbf{0.0405} \pm 0.0079$	$\mathbf{0.0110} \pm 0.0022$

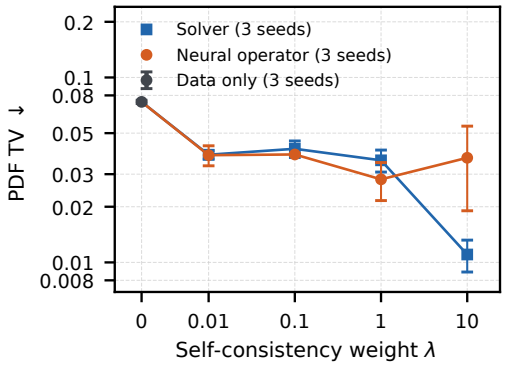


Figure 9: KS PDF-TV across training seeds as a function of self-consistency weight. All solver, neural-supervision, and data-only points show means and sample standard deviations across three training seeds, after averaging four sampling repeats within each run.

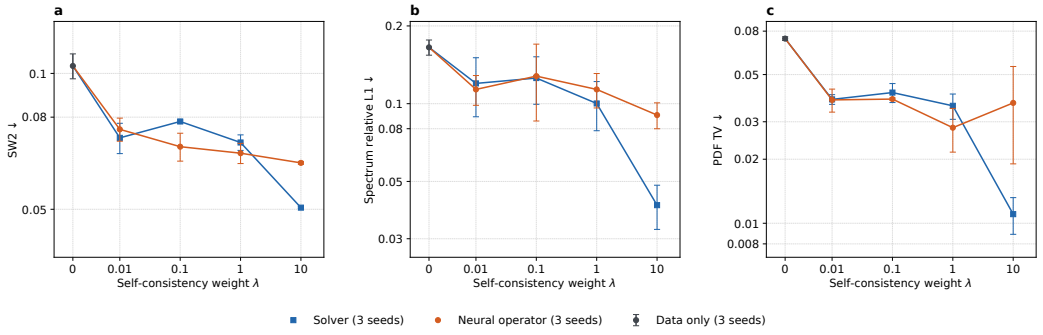


Figure 10: KS training-seed sweep for (a) sliced-W_2, (b) relative spectral L_1 error, and (c) PDF-TV. All points aggregate three training seeds with the same averaging convention as Figure 9. Bars show variation across training seeds.

E.6 QUALITATIVE COMPARISON OF HW FIELDS

The field comparisons complement the distributional metrics by displaying spatial organization and fluctuation amplitudes. Each row shows five sample realizations from one source, drawn from the evaluation ensembles described in Subsection B.4. The columns are sample realizations, not independent training runs. They do not pair a generated field with a DNS target. For each variable, all methods share a color scale normalized by its maximum absolute value over the DNS reference. Weak-amplitude outputs therefore retain low contrast without per-field rescaling.

IMG and DNS. Figure 11 compares IMG with DNS in both density and vorticity. The generated density fields exhibit alternating positive and negative structures, while the vorticity fields show finer filamentary structure. The comparison displays these features at a common reference amplitude scale for each variable.

Physics-supervised comparison. Figure 12 shows density fields from DNS, FNO, U-Net, PI-DeepONet, WGS, and IMG, in that order. FNO, U-Net, and IMG produce spatially organized fluctuations, whereas the displayed PI-DeepONet and WGS fields have weaker contrast at the shared DNS scale. These samples complement the statistical errors in Table 1.

Few-data comparison. Figure 13 compares density fields from the data-only CFM baseline, CFM with solver-based self-consistency, CFM with neural-operator-based self-consistency, the data-trained FNO and U-Net, and DySLIM against DNS. The panels illustrate differences in spatial texture and amplitude among the methods. Quantitative comparisons are reported in Table 3 and the loss-weight ablations. The resolved statistical curves in Appendix E.8 connect these field textures to spectra and spatial increments.

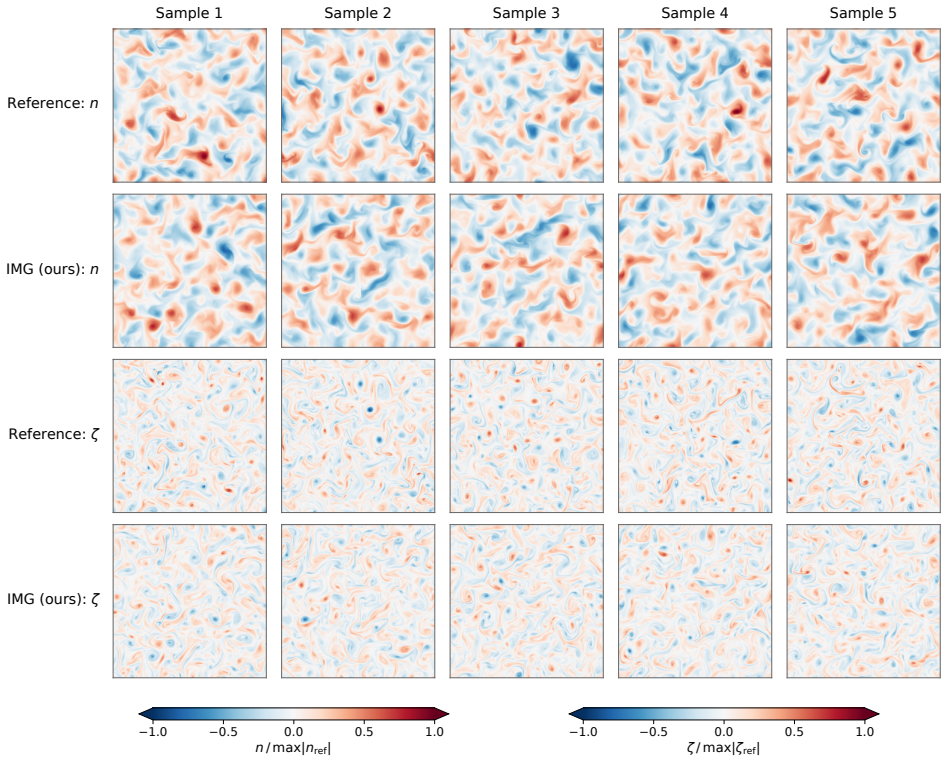


Figure 11: HW density and vorticity fields from IMG (ours) and DNS. From top to bottom, the rows show DNS density n, IMG density n, DNS vorticity ζ, and IMG vorticity ζ. Columns show five sample realizations. Density and vorticity correspond to the same state within each source, while IMG and DNS columns are not paired targets. The two color bars show $n/\max|n_{\mathrm{ref}}|$ and $\zeta/\max|\zeta_{\mathrm{ref}}|$, respectively, using a shared DNS normalization for each variable.

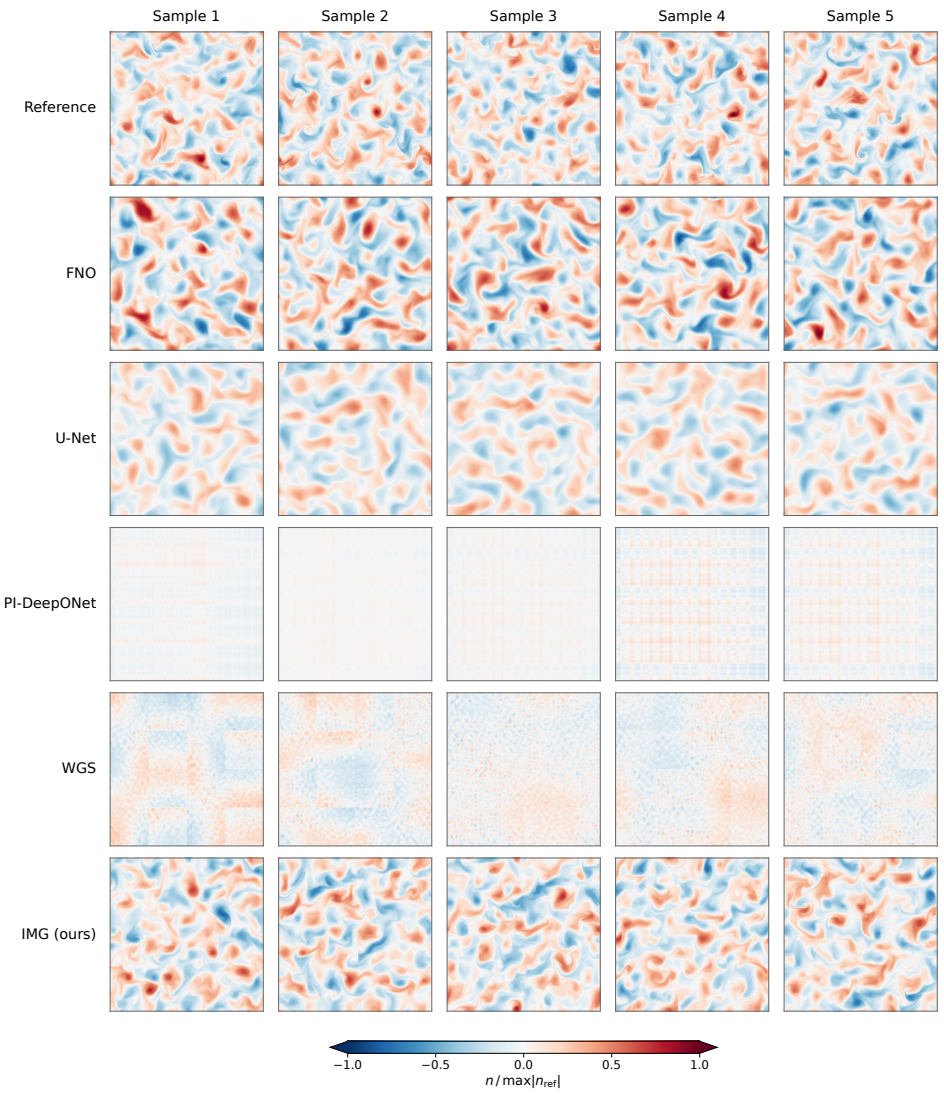


Figure 12: HW density fields for IMG and the physics-supervised baseline comparison. From top to bottom, rows show DNS, FNO, U-Net, PI-DeepONet, WGS, and IMG (ours). Each row contains five sample realizations. All panels show $n/\max|n_{\mathrm{ref}}|$ with the same color scale, normalized by the maximum absolute density in the DNS reference. Near-white panels indicate small amplitudes at this shared scale.



Figure 13: HW density fields in the few-data comparison. From top to bottom, rows show DNS, Data CFM, Data CFM + IMG (ours), Data CFM + IMG-neural (ours), FNO, U-Net, and DySLIM ($\lambda = 100$, bandwidth multiplier 1). Columns show five sample realizations from each source. All rows share the normalization $n/\max|n_{\mathrm{ref}}|$ and the same color scale. Color-bar extensions indicate values outside $[-1, 1]$.

E.7 STATISTICAL STABILITY UNDER PHYSICAL EVOLUTION

Evaluation protocol. We test whether the generated ensembles retain their statistics under the target Hasegawa–Wakatani (HW) dynamics, complementing the static comparison with DNS. The model and DNS ensembles each contain 64 fields, as in the main-table and weight-ablation evaluations. For each of three fixed trained models (IMG, Data CFM, and Data + SC CFM), we use five sampling replicates of 64 fields and integrate each field without burn-in or amplitude rescaling. The same trained models are used throughout this comparison. The five replicates vary sampling randomness while keeping model parameters fixed. We use the ordinary HW equations on a periodic 256×256 grid with $\alpha = \kappa = 1$, $\nu = 10^{-4}$, $k_0 = 0.15$, and the same finite-difference hyperdiffusion as the reference simulation. Integration uses float32 RK4 with $h = 0.0125$. Statistics are evaluated at $t \in \{0, 0.25, 0.5, 1, 1.5, 2, 5, 10, 37\}$, spanning times shorter and longer than the maximum IMG training duration $\Delta T = 1$ (the Data + SC CFM upper limit is $\Delta T = 0.6$). The longest horizon was fixed before advancing the generated fields using twice the larger estimated DNS integral autocorrelation time for energy and transport, rounded upward to a quarter time unit. As a control, we advance 64 saturated DNS fields from distinct trajectories at reference time $T = 3600$ with the same solver and output times. All requested trajectories remain finite.

Initial-to-evolved and cross-split comparisons. Here Data + SC CFM denotes Data CFM + IMG with numerical-solver targets. Let $X_0 = \{u_i\}_{i=1}^{64}$ and $X_t = \{\Phi^t(u_i)\}_{i=1}^{64}$, where the 64 model fields in each replicate are selected once and held fixed to match the DNS sample count. For each discrepancy D, we evaluate

$$D_{\mathrm{self}}(t) = D(X_0, X_t). \tag{53}$$

This is an ensemble comparison, not a pointwise trajectory error. Since $D_{\mathrm{self}}(0) = 0$, we also split the fixed ensemble into disjoint 32-field groups A_0 and B_0 and compute

$$D_{\mathrm{cross}}(t) = \tfrac{1}{2}\big[D(A_0, B_t) + D(B_0, A_t)\big]. \tag{54}$$

The same partition is used at every time. This comparison removes shared trajectory identities between the two sets being compared and has a nonzero finite-sample baseline at $t = 0$. We report the distance itself, without subtracting this baseline. Because it compares 32 against 32 fields, its absolute value is not a calibrated threshold for the 64-against-64 comparison. Both protocols measure evolution within each model or DNS ensemble, not distances between models and DNS.

We measure (i) sliced-W_2 on flattened 4×4 average-pooled fields with 128 fixed Gaussian unit projection directions, (ii) the relative L^1 difference of the ensemble-mean radial power spectra, averaged over ζ, n, ϕ, and (iii) total variation (TV) of the point-value PDFs, averaged over ζ, n. Spectral differences are normalized by the initial ensemble spectrum in each comparison. Vorticity is spatially centered for the PDF and sliced-W_2 diagnostics, while n is retained in its original units; spectra use fluctuation fields. PDF bins are fixed using DNS, with underflow and overflow probability retained. The metrics therefore probe specified observables and pooled features rather than every aspect of the full-resolution field distribution.

Physical observables and uncertainty. Using all 64 model fields per replicate and all 64 DNS fields, we evaluate

$$E = \tfrac{1}{2}\langle \widetilde{n}^2 - \phi\widetilde{\zeta}\rangle, \qquad U = \tfrac{1}{2}\langle (\widetilde{n} - \widetilde{\zeta})^2\rangle, \qquad \Gamma_n = -\langle n\, \partial_y \phi\rangle, \tag{55}$$

where angle brackets denote a spatial average, $\widetilde{f} = f - \langle f\rangle$, and $\nabla^2\phi = \widetilde{\zeta}$ with zero-mean ϕ. We report ensemble means and their signed changes $\Delta O(t) = \overline{O}(t) - \overline{O}(0)$. For the normalized drift plots and table, ΔE and ΔU are divided by their fixed DNS means, and $\Delta\Gamma_n$ by the fixed DNS sample standard deviation: $\overline{E}_{\mathrm{DNS}} = 4.71071$, $\overline{U}_{\mathrm{DNS}} = 12.12322$, and $s_{\Gamma,\mathrm{DNS}} = 0.05102$. Distribution curves and tables show the mean and sample standard deviation across the five model sampling replicates; DNS is one fixed ensemble. Scalar intervals are approximate pointwise 95% percentile-bootstrap intervals from 1,000 whole-trajectory resamples, preserving pairing across times. They do not quantify training-seed variability or provide a formal equivalence test. For nonlinear empirical distribution distances, these bootstrap ranges are not calibrated confidence intervals.

Results. Figures 14–16 and Tables 14–16 summarize the results. IMG's amplitude PDFs change little over the training-scale horizon, while its spectrum relaxes more than the DNS control. Over

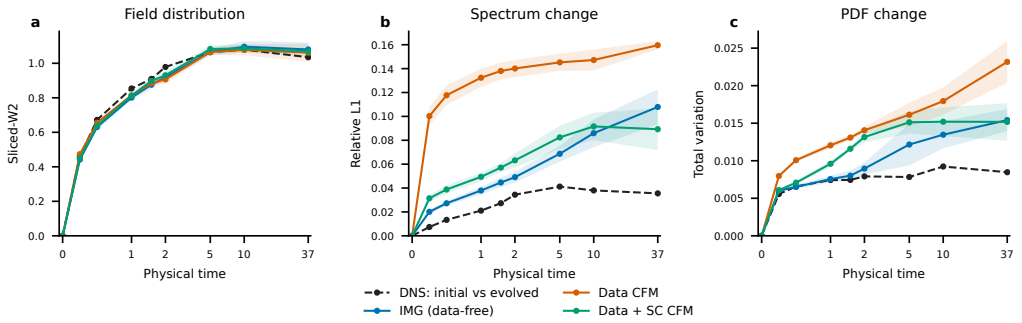


Figure 14: Initial-to-evolved ensemble comparison under the HW dynamics: pooled-field sliced-W_2 (a), relative spectral L^1 change (b), and point-value PDF TV over ζ, n (c). Model curves show means and sample-standard-deviation bands across five fixed-weight sampling replicates; the dashed black curve is DNS versus its own evolution. All curves start at zero by construction. Time axes are linear near zero and logarithmic beyond $t = 1$.

longer horizons, the spectral and PDF changes become more apparent even though the pooled-field sliced-W_2 remains comparable to DNS. The cross-split comparison also detects spectral relaxation, so the pattern is not specific to comparing an ensemble with its own evolved samples.

The scalar observables reveal different responses. IMG energy decreases slightly while generalized enstrophy and particle flux increase. This is consistent with a generated ensemble that approximately reproduces broad fluctuation levels but has not fully balanced the statistics governing their subsequent evolution. In the few-data comparison, solver supervision reduces spectral and PDF relaxation relative to data-only CFM. It also lowers generalized-enstrophy drift at the longest horizon, although the short-time drift increases. Mean transport drift is negative for solver-assisted CFM and positive for data-only CFM and data-free IMG. Its magnitude is larger for solver-assisted CFM at the longest horizon, so this observable does not share the improvement in spectral persistence.

Here $\Delta\Gamma_n$ is the change in mean flux, not the flux itself. Its sign therefore describes increasing or decreasing transport, not a reversal of transport direction. Stationarity requires preservation of the expected flux, rather than matching the drift sign of one finite DNS ensemble. At the longest horizon, the DNS drift interval includes zero. Matching field amplitudes and spectra alone leaves cross-field correlations, which enter the transport flux, only partially constrained.

Repeating all 1,024 initial fields with $h = 0.00625$ preserves the qualitative relaxation. This makes time discretization a less likely primary explanation and points to residual distributional error in the generated ensembles. Finite-sample DNS drift remains a comparison scale rather than an exact equilibrium threshold. Together, the diagnostics support approximate statistical persistence with relaxation that varies by observable.

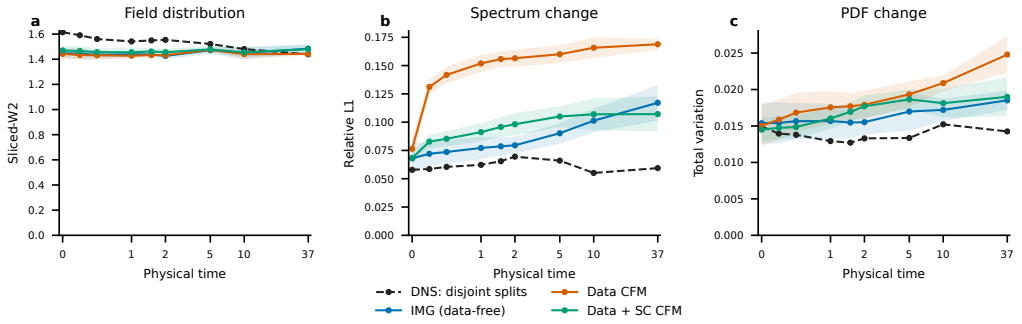


Figure 15: Cross-split ensemble distances, computed using Eq. (54) on fixed disjoint groups of 32 fields. Panels and model bands follow Fig. 14. The dashed black line uses disjoint DNS groups. The $t = 0$ value reflects differences between distinct finite samples; it has not been subtracted. This auxiliary comparison does not define an equivalence threshold for the larger ensembles in Fig. 14.

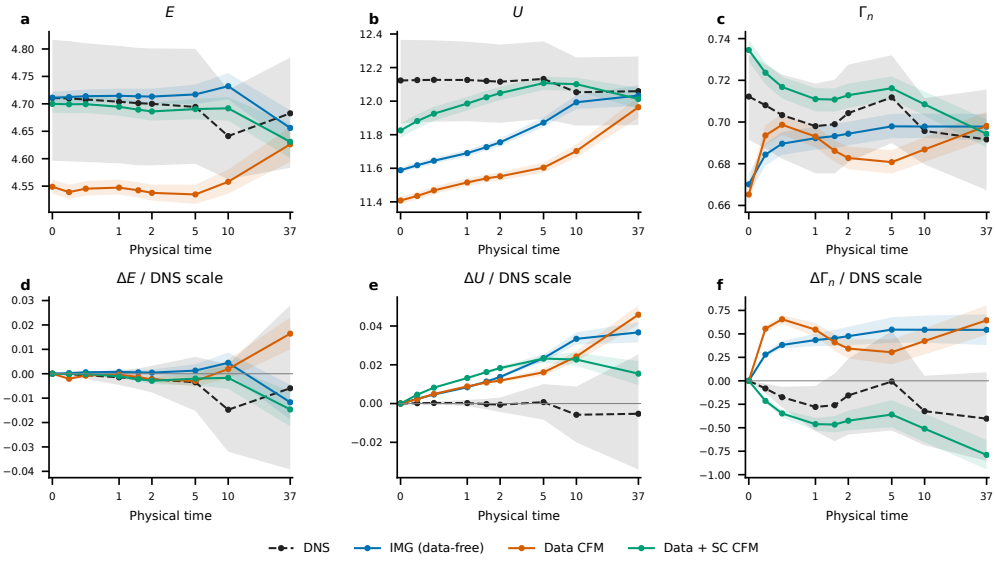


Figure 16: Physical statistics during HW evolution. Top: ensemble-mean energy E (a), generalized enstrophy U (b), and particle flux Γ_n (c). Bottom: changes from the initial ensemble means divided by the fixed DNS scales defined in the text (d–f). Model statistics are averaged over five sampling replicates. Shading denotes approximate pointwise 95% whole-trajectory-bootstrap intervals. The dashed black curves show DNS self-evolution.

Table 14: Selected initial-to-evolved distances from Fig. 14. PDF TV averages ζ, n. Model entries are mean (sample standard deviation) across five sampling replicates. DNS entries are point estimates. The zero-time self-distances are identically zero and omitted.

Ensemble	t	Coarse SW_2	Spectrum relative L^1	PDF TV
DNS	1	0.8541	0.0211	0.00743
DNS	37	1.0351	0.0355	0.00847
IMG	1	0.8011 (0.0113)	0.0379 (0.0033)	0.00756 (0.00046)
IMG	37	1.0802 (0.0414)	0.1079 (0.0147)	0.01541 (0.00147)
Data CFM	1	0.8142 (0.0072)	0.1324 (0.0077)	0.01206 (0.00045)
Data CFM	37	1.0616 (0.0510)	0.1596 (0.0034)	0.02317 (0.00277)
Data + SC CFM	1	0.8119 (0.0165)	0.0494 (0.0035)	0.00959 (0.00045)
Data + SC CFM	37	1.0693 (0.0137)	0.0893 (0.0175)	0.01518 (0.00251)

Table 15: Selected cross-split distances from Fig. 15, with the same uncertainty notation as Table 14. Each term compares 32 against 32 fields. PDF TV averages ζ, n. The initial values are included to distinguish finite-sample differences from subsequent changes.

Ensemble	t	Coarse SW_2	Spectrum relative L^1	PDF TV
DNS	0	1.6147	0.0579	0.01492
DNS	37	1.4387	0.0594	0.01425
IMG	0	1.4479 (0.0372)	0.0680 (0.0134)	0.01538 (0.00281)
IMG	37	1.4836 (0.0352)	0.1171 (0.0158)	0.01852 (0.00132)
Data CFM	0	1.4427 (0.0344)	0.0765 (0.0107)	0.01503 (0.00281)
Data CFM	37	1.4421 (0.0126)	0.1689 (0.0050)	0.02479 (0.00250)
Data + SC CFM	0	1.4700 (0.0342)	0.0682 (0.0052)	0.01449 (0.00165)
Data + SC CFM	37	1.4799 (0.0321)	0.1071 (0.0155)	0.01900 (0.00268)

Table 16: Normalized signed changes in the physical observables from Fig. 16. Entries give point estimates [approximate pointwise 95% trajectory-bootstrap intervals]. These are dimensionless changes divided by fixed DNS scales, not percentages of each model's initial mean.

Ensemble	t	$\Delta E/\overline{E}_{\mathrm{DNS}}$	$\Delta U/\overline{U}_{\mathrm{DNS}}$	$\Delta\Gamma_n/s_{\Gamma,\mathrm{DNS}}$
DNS	1	-0.0015 [-0.0046, $+0.0014$]	$+0.0002$ [-0.0014, $+0.0019$]	-0.2782 [-0.5348, -0.0540]
DNS	37	-0.0059 [-0.0392, $+0.0281$]	-0.0053 [-0.0340, $+0.0254$]	-0.4031 [-0.8538, $+0.0915$]
IMG	1	$+0.0008$ [$+0.0001$, $+0.0015$]	$+0.0084$ [$+0.0079$, $+0.0088$]	$+0.4341$ [$+0.3691$, $+0.5025$]
IMG	37	-0.0117 [-0.0188, -0.0046]	$+0.0367$ [$+0.0315$, $+0.0420$]	$+0.5429$ [$+0.3809$, $+0.7080$]
Data CFM	1	-0.0003 [-0.0010, $+0.0003$]	$+0.0089$ [$+0.0085$, $+0.0092$]	$+0.5456$ [$+0.4723$, $+0.6145$]
Data CFM	37	$+0.0164$ [$+0.0099$, $+0.0231$]	$+0.0458$ [$+0.0406$, $+0.0507$]	$+0.6444$ [$+0.4912$, $+0.8045$]
Data + SC CFM	1	-0.0011 [-0.0018, -0.0004]	$+0.0132$ [$+0.0127$, $+0.0136$]	-0.4617 [-0.5306, -0.3956]
Data + SC CFM	37	-0.0146 [-0.0216, -0.0072]	$+0.0154$ [$+0.0095$, $+0.0218$]	-0.7893 [-0.9443, -0.6301]

E.8 Resolved HW spectra and spatial increments

Figures 17–19 unpack the few-data comparison into variables, directions, and scales. They use saved single-evaluation curves from the same trained models as Table 3, with $\lambda = 1$ for solver supervision and $\lambda = 0.1$ for neural supervision. Spectra and increment diagnostics retain their respective archived DNS references. The main table reports averages over evaluation repeats.

Where the spectral power changes. Both forms of supervision reduce the excess power in the lowest vorticity and potential modes (Fig. 17). This matters particularly for ϕ, whose spectrum emphasizes large scales through the inverse Laplacian relating it to ζ. Solver supervision also brings density power closer to DNS around its peak and the neighboring decay range. The neural correction is less uniform across wavenumbers. All three models retain excess power in the far high-wavenumber tail, although supervision reduces it. This tail carries little total power and therefore contributes less to the linear-unit spectral error than discrepancies near the dominant modes. The aggregate improvement should thus be read as a better distribution of fluctuation power, with residual errors at the finest resolved scales.

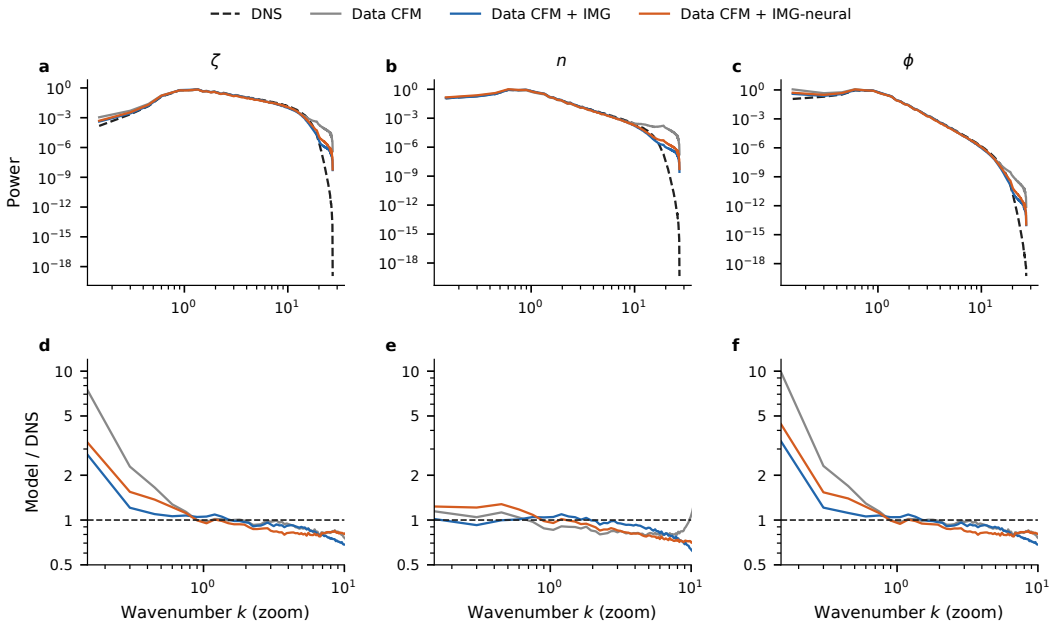


Figure 17: Resolved HW few-data spectra. Top panels show the radial power spectra of vorticity ζ, density n, and potential ϕ over the full saved wavenumber range. Bottom panels zoom into $k \leq 10$ and show model-to-DNS power ratios, with unity marked by the dashed line. Logarithmic axes expose both large-scale excess power and the small-amplitude high-wavenumber tail. Curves show a saved evaluation of each main-table model.

Typical increments and their tails. Figure 18 shows increments in both directions at separations of one, eight, and 64 grid cells, spanning the smallest, intermediate, and largest evaluated separations. The central parts of the distributions overlap much more closely than the tails. At one-cell separation, the vorticity tails remain underrepresented in both directions. Solver supervision brings the negative density-increment tail in x closer to DNS, while discrepancies remain in the positive tail. At 64-cell separation, the density curves nearly overlap. Thus, much of the unresolved structure lies in rare, sharp spatial changes, rather than in the typical large-separation density differences. A small increment-PDF TV can coexist with these tail discrepancies because rare bins contribute little total probability mass.

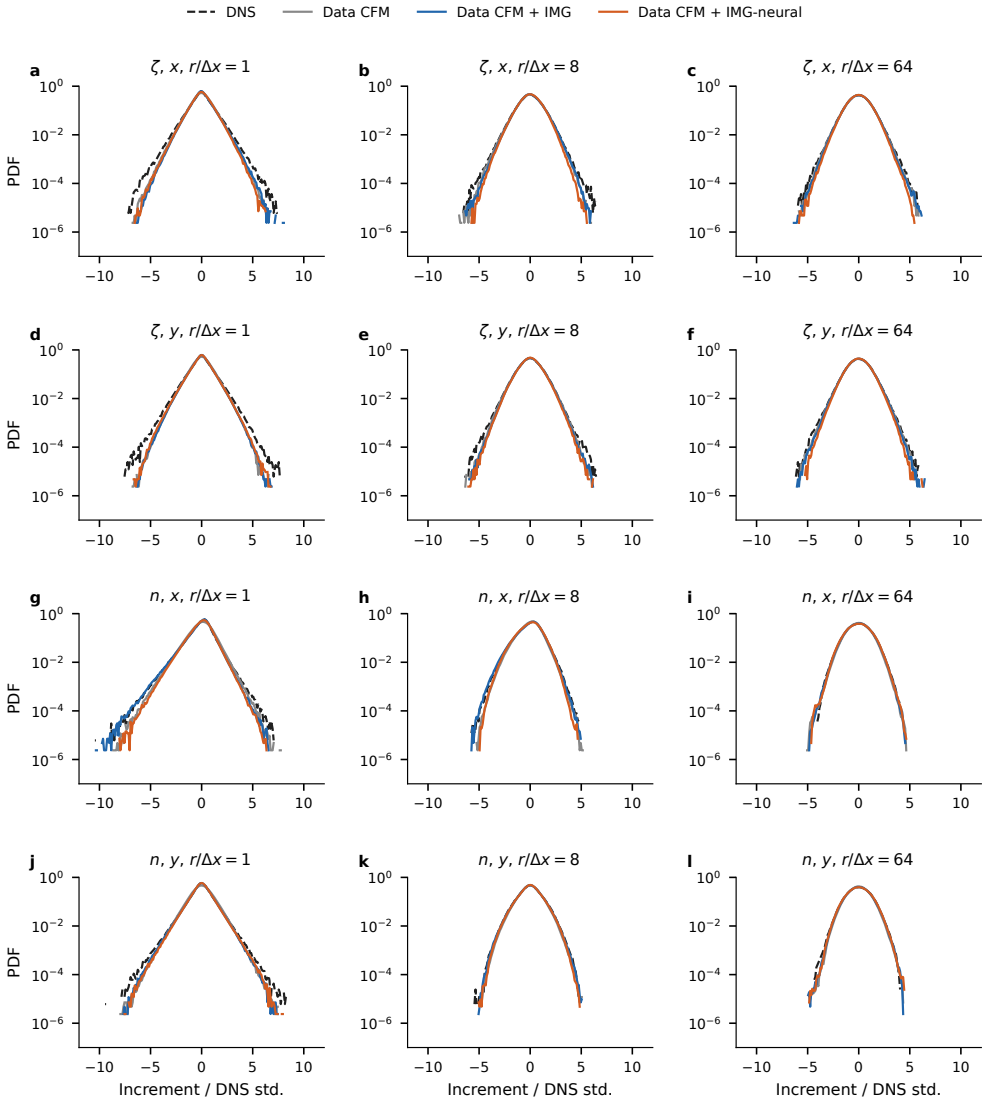


Figure 18: HW increment PDFs on logarithmic probability axes. Rows show ζ in x, ζ in y, n in x, and n in y. Columns show separations $r/\Delta x = 1, 8, 64$. Each panel uses the DNS increment standard deviation for all methods. Curves retain their saved histogram bins. Empty bins are omitted on the logarithmic axis. These are single-evaluation distributions from the main-table models, with the same method colors as Fig. 17.

Intermittency depends on variable and direction. Flatness weights large increments through their fourth moment and exposes features that the PDF-TV average obscures (Fig. 19). DNS density flatness is about six at the smallest separation and approaches three at large separations. Solver supervision recovers this small-scale rise especially well in x, where data-only CFM underestimates it. Neural supervision captures more of the rise in y than in x. Vorticity behaves differently. Both variants improve its one-cell flatness, but data-only CFM is closer to DNS at several intermediate and large separations. The neural variant underestimates vorticity flatness across the plotted range. These competing changes explain why its aggregate increment-PDF error improves without a corresponding reduction in aggregate flatness error. They identify partial recovery of intermittent spatial fluctuations, with remaining errors depending on both variable and scale.

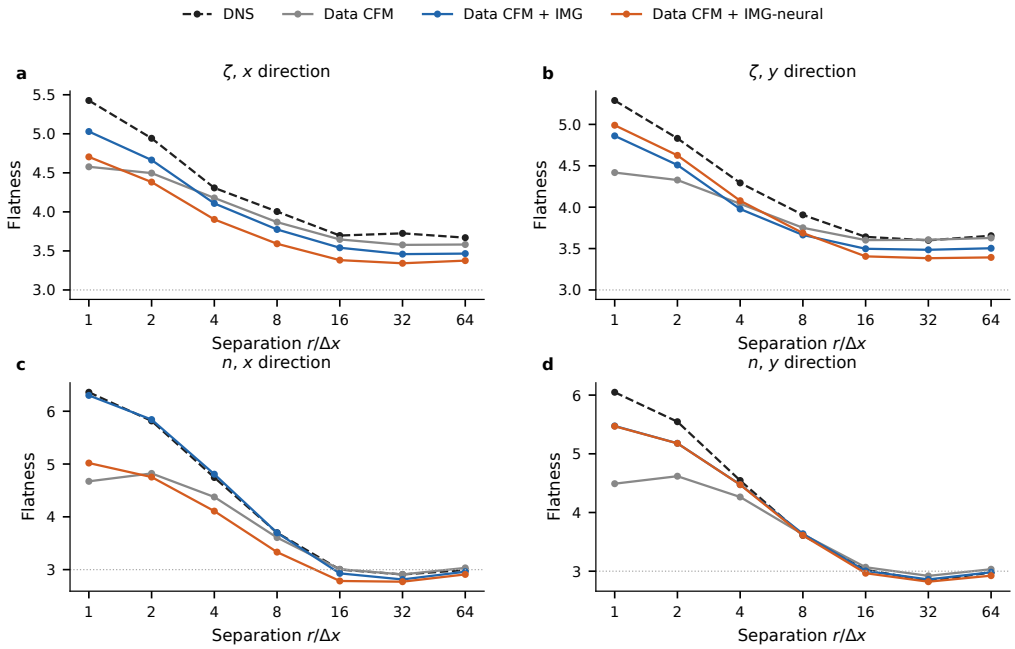


Figure 19: HW increment flatness versus separation for ζ and n in both coordinate directions. The dotted line marks Gaussian flatness, three. The seven separations span one to 64 grid cells. Curves show a saved evaluation of each main-table model. The scale dependence distinguishes the recovery of small-scale density intermittency from the remaining vorticity discrepancies.

E.9 HW PARTICLE-POOL ABLATION

We compare two ways to supply short-time physical supervision within our framework. IMG draws fresh states from its current generator before advancing them. The particle-pool variant continually advances persistent particles and trains CFM on their current states. Both use the same CFM architecture, cumulative physical-advancement budget, and remaining training and evaluation protocols. IMG additionally pays for generating fresh particles at each update. Table 17 uses the data-free IMG result from Table 1.

Particle-pool construction and training. The pool contains 50 particles, matching the data-free IMG batch size in Table 4. It is initialized by the first batch of samples from the flow model. At each update, we advance the current particles using the same physical-duration sampling as IMG and use the resulting 50 states as the CFM target batch. These states replace the previous pool, with no historical snapshots retained or replayed. Prior samples and generative interpolation times are drawn as in IMG. Thus, training uses only the current advanced batch.

Table 17: HW particle-pool ablation at matched physical-advancement budget. Cells show the mean above and ± sample standard deviation below over five evaluation repeats at fixed trained weights. IMG uses the same results as Table 1 and Appendix E.1. Bold marks the lowest displayed mean. All metrics are lower-is-better. Rounding follows Appendix B.5.

Method	Coarse SW_2 ↓	Spectrum rel. L_1 ↓	PDF TV ↓	Increment PDF-TV ↓	Flatness rel. L_1 ↓
Particle-pool CFM	**1.11** ±0.02	0.134 ±0.007	0.032 ±0.003	0.041 ±0.001	0.029 ±0.002
IMG (ours)	1.13 ±0.02	**0.063** ±0.005	**0.012** ±0.002	**0.0094** ±0.0003	**0.027** ±0.002

IMG improves spectral, marginal-PDF, and increment-PDF accuracy together. The difference therefore concerns both fluctuation amplitudes and their spatial organization, rather than only one aggregate distributional score. Flatness errors are similar, so the improvement in typical spatial increments does not establish a comparable gain in their intermittent extremes. The particle-pool sampler has a slightly lower coarse-field sliced-W_2. Both scores lie near the DNS reference discrepancy in Appendix E.1, where this diagnostic has limited resolution among otherwise distinct HW models.

At the same physical-advancement budget, the additional generator sampling thus improves several complementary statistics. Resampling keeps the physical targets coupled to the model's current output distribution, whereas persistent particles evolve from their previous states. This may help the supervision correct errors on states that the model currently produces. The ablation supports the benefit of this resampling step within the shared supervision framework.

The gain has a computational cost. Generative sampling dominates the profiled IMG update in Appendix D. The particle pool avoids this repeated generation while retaining short-time physical supervision, offering a less expensive target-construction option within the same framework. The comparison therefore shows an accuracy benefit from extra sampling at matched physical advancement, rather than an advantage at equal total GPU time. Reducing the cost of fresh generation could improve this tradeoff.

E.10 EFFECT OF PHYSICAL-TIME INTERPOLATION

We compare interpolated U-Net targets with numerical evolution to the same intermediate physical time. For an initial field u and $s = \alpha h$, the targets are

$$A = \Phi_{\mathrm{num}}^{\alpha h}(u), \qquad C = (1-\alpha)u + \alpha\widehat{\Phi}^{h}(u),$$

where $\widehat{\Phi}^{h}$ is the U-Net prediction. This comparison measures the combined effect of neural prediction and interpolation. Here h is the interval between interpolation endpoints, distinct from the sampled-duration upper limit ΔT. We compute errors at each intermediate time and then average over uniformly distributed times in $[0, h]$.

Table 18: HW interpolation errors using U-Net or numerical-solver endpoints, relative to numerical intermediate states. Second-moment changes are signed percentages. Spectrum and PDF errors are averaged over ζ, n, and ϕ, then over physical time within $[0, h]$. U-Net models use their final training weights.

Endpoint source	h	ζ moment (%)	n moment (%)	ϕ moment (%)	Spectrum rel. L_1	PDF-TV
U-Net	0.6	-13.86	-2.909	-1.425	0.06208	0.01516
U-Net	0.3	-6.384	-1.024	-0.6290	0.02681	0.006290
U-Net	0.1	-1.010	-0.1214	-0.07972	0.004060	0.001092
Numerical solver	0.1	-0.9872	-0.1268	-0.04597	0.003867	0.0009822

Shorter intervals substantially reduce the second-moment bias. At $h = 0.1$, the vorticity bias is about 1%, while the density and potential biases are below 0.13% in magnitude. These results indicate that shorter neural steps better preserve fluctuation amplitudes in this diagnostic. The spectrum and PDF errors decrease at each shorter interval, indicating improved preservation of spatial power and field-value distributions.

A separate control replaces the neural endpoint with its numerical counterpart, $B = (1-\alpha)u + \alpha\Phi_{\mathrm{num}}^{h}(u)$, to isolate interpolation error. It uses 64 saturated HW fields from 32 trajectories at $t = 3540$ and 3600. At $h = 0.1$, U-Net and numerical-endpoint interpolation yield similar errors in the reported statistics (Table 18).

For longer requested durations, IMG-neural takes full 0.1 steps and interpolates only over the final remainder (Appendix C.1). This keeps the interpolation local, while prediction errors can accumulate across full steps. The controls measure errors on saturated inputs. Appendix A.1 accounts for approximation errors on the evolving generator distribution.