Macrocanonical Generator Networks: Data-Efficient Neural Surrogates for Amortized Physics Simulation

Niall Jeffrey
Dept. of Physics &
King's Institute for Artificial Intelligence
King's College London
niall.jeffrey@kcl.ac.uk

Benjamin D. Wandelt
Dept. of Physics and Astronomy &
Dept. of Applied Mathematics and Statistics
Johns Hopkins University
wandelt@jhu.edu

Abstract

Many scientific simulations are computationally expensive, limiting their use in simulation-based inference, uncertainty quantification, and decision making. We introduce *macrocanonical generator networks*, a framework for learning fast, data-efficient neural surrogates for *amortized* simulation of stationary physical processes by matching multiscale statistics of a target process. Inspired by microcanonical maximum-entropy synthesis, which matches feature statistics per sample by inner-loop optimization, our approach instead trains a neural generator to satisfy the same constraints in expectation, preserving physically realistic sample-to-sample fluctuations. We prove that gradient-descent training induces preconditioned gradient-descent dynamics in sample space that inherit the symmetry-preservation properties of the microcanonical gradient descent of Bruna and Mallat, amortizing its per-sample inner-loop optimization into a single training run after which every sample is generated in one forward pass. We further show that residual generator parameterizations initialized near the identity, trained with a small learning rate, preserve entropy locally during early optimization and mitigate mode collapse. Experiments on cosmology and fluid simulations show that macrocanonical generators recover realistic samples from extremely small training sets, in some cases a single training example, while generating new realizations more than 10^6 times faster per sample than the reference simulator and more than 10^3 times faster than microcanonical gradient descent. The trained generators can be reused for downstream pretraining, including simulation-based inference, providing a data-efficient route to multi-fidelity scientific machine learning workflows.

1 Introduction

Scientific simulations are essential tools for understanding complex physical systems. However, high-fidelity simulators are often computationally expensive, sometimes requiring days to generate a single realization because each sample typically requires an iterative solve or stochastic rollout. This limits their use in tasks such as simulation-based inference [2, 7], uncertainty quantification, and decision making, where large numbers of simulations are required. Examples include cosmological simulation [33, 17], tsunami models [3], wind farm simulation [18], and nuclear fusion models [14].

Recent advances in deep generative modelling have demonstrated that neural networks can learn complex data distributions. Methods such as generative adversarial networks [9], diffusion models [13], and normalizing flows [31] have achieved remarkable success in image generation and related domains. Earlier work such as Generative Moment Matching Networks [21] trains generators by matching the empirical data distribution using the maximum mean discrepancy (MMD). However,

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

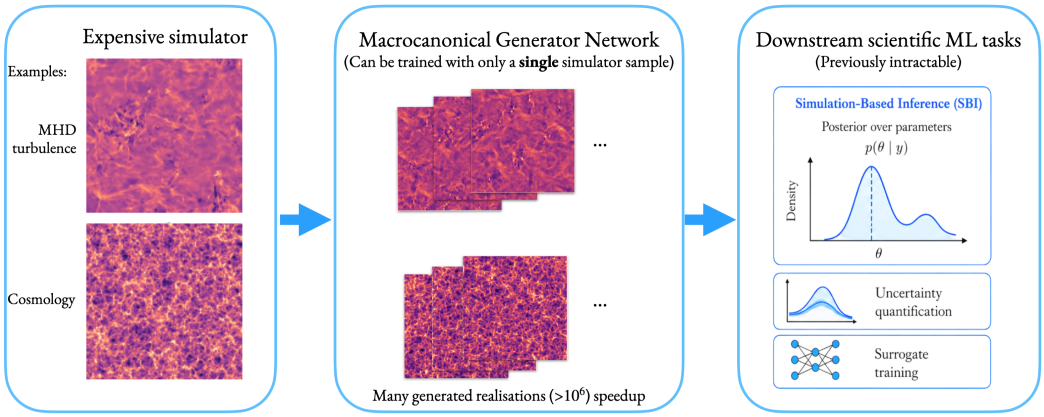

Figure 1: Macrocanonical generator networks (*centre*) turn a small set of expensive simulator outputs (*left*; turbulence and cosmology targets, Section 5) into large ensembles of new realizations matching the chosen statistical features (e.g. wavelet phase harmonics) of the targets, unlocking downstream scientific machine-learning tasks (*right*) that are intractable with the expensive simulator alone.

these methods typically rely on large training datasets and are not designed for settings in which only a small number of simulator realizations are available, a regime that is common in scientific computing.

Other moment-matching approaches, such as MMD-GAN [20], the method of simulated moments [25] and turbulence closures [28], as well as neural operators (which address the different task of predicting a simulator's output for given boundary conditions), likewise require many simulations or observations, whereas we match a fixed multiscale moment vector (features) that a single stationary realization can be sufficient to estimate.

Many physical processes of interest—turbulent flows, cosmological fields, multifractal processes—are *stationary*: their statistical properties are invariant under spatial translations. This stationarity is a powerful inductive bias. As shown by Bruna and Mallat [4], the translation symmetry of a stationary process means that a single realization of dimension d provides $O(d)$ translated copies of local structures, enabling accurate estimation of statistical moments from as few as a single example.

In such systems, knowledge about the process is naturally expressed through *statistical constraints*. Multiscale statistics derived from wavelet transforms, scattering transforms [1, 5], or wavelet phase harmonics [23, 35] can capture important non-Gaussian structure in complex signals. Throughout, "multiscale" refers to spatial scales rather than temporal, and all signals considered are equal-time snapshots. Classical texture synthesis methods exploit this structure by generating samples that match these statistics [29]. In these approaches, samples are synthesized individually by optimizing pixel values until the target statistics are satisfied. This procedure can be interpreted as a form of microcanonical maximum-entropy sampling [4, 35], where the probability distribution is conditioned on the empirical statistics of the observed signal. Related recent work extends moment-constrained generation with diffusion-based samplers that guide moments along stochastic trajectories toward maximum-entropy models [19]. These approaches remain per-sample iterative at generation time.

In this work we introduce *macrocanonical generator networks*, a neural framework that generalizes this idea. Instead of synthesizing each realization independently through optimization, we train a neural generator that produces samples whose statistics match those of the target process *in expectation*. The generator therefore learns a global transport map that produces samples satisfying the desired statistical constraints. Moreover, enforcing constraints in expectation rather than per sample is a better match to real physical processes, in which individual realizations fluctuate around their defining statistics (Section 2.2).

This shift from per-sample optimization to learning a global generator has important consequences. Once trained, the generator acts as a fast surrogate simulator capable of producing large numbers of realizations in one forward pass per sample, effectively amortizing the expense of iterative simulation and iterative moment-guided sampling. In this sense, our objective is complementary to trajectory-based approaches: rather than improving a per-sample iterative sampler, we learn a reusable transport map for high-throughput workflows. Because the training objective depends on statistical constraints

rather than individual examples, and because stationarity provides abundant constraints from few realizations, the method can operate effectively in extremely data-limited regimes. The resulting generators can be used to pretrain neural networks for downstream tasks, enabling new approaches to simulation-based inference [2, 7], multi-fidelity modelling [12], and the training of scientific foundation models [27, 24, 34].

Beyond the practical advantages, the macrocanonical formulation provides a conceptual bridge between classical statistical synthesis methods and modern deep generative modelling. In particular, we show that training such generators induces a preconditioned gradient descent dynamics in sample space, closely related to the optimization procedure used in microcanonical synthesis [4, 35]. Two design choices are key: (i) residual generator parameterizations initialized near the identity, and (ii) sufficiently low learning rates. Together, these help the generator preserve entropy locally during early optimization and mitigate collapse to degenerate outputs. Without these inductive biases, the generator can collapse toward a constant mapping $F(x) \approx y^*$, effectively giving the same output for every input, which is a situation of near-minimal entropy.

Table 1: Macrocanonical generator networks (ours) compared with prevailing deep generative methods and with classical microcanonical synthesis. Properties marked "exact, by construction" follow from architectural equivariance and the loss; properties marked "approximate (learned)" hold only insofar as the training set is large and diverse enough to make them statistically apparent.

	Macrocanonical (ours)	**GAN**	**Diffusion model**	**Microcanonical synthesis**
Training samples needed	≥ 1 realization	large dataset	large dataset	≥ 1 realization
Inference cost per sample	1 forward pass	1 forward pass	many ($\sim 10^{2\text{–}3}$) steps	inner-loop optimization
Symmetries of moment constraints	exact, by construction	approximate (learned)	approximate (learned)	exact, by construction
Per-realization fluctuations	yes (constraints in expectation)	yes	yes	no (Dirac on ϕ^*)
Training signal	feature-moment matching	adversarial	denoising score matching	per-sample feature matching
Distributional flexibility	limited to constrained moments	unrestricted	unrestricted	limited to constrained moments

The main contributions of this work are therefore:

(i) We introduce macrocanonical generator networks, a generative framework that learns neural transport maps producing samples whose feature statistics match those of a target process while replacing per-sample iterative simulation or sampling with amortized one-pass generation after training.

(ii) We establish a connection between neural generator training and microcanonical synthesis, showing that generator training induces preconditioned gradient descent dynamics.

(iii) We analyze the entropy behavior of residual generators and show that identity-initialized residual parameterizations with low learning rates help to preserve local volume and maintain diversity in generated samples, while generators without these inductive biases suffer from mode collapse.

(iv) We demonstrate that macrocanonical generators can learn effective surrogate simulators in extremely data-limited regimes, including cases with a single training example, leveraging the stationarity of the target process as a key inductive bias and making many-query analyses with surrogate simulators computationally practical.

(v) We show how such generators can be used for pretraining in simulation-based inference pipelines, enabling new multi-fidelity workflows and providing a potential foundation for training scientific foundation models.

Together, these results suggest that macrocanonical generator networks provide a new framework for constructing fast neural surrogate simulators of stationary physical processes, bridging ideas from statistical physics, classical synthesis methods, and modern deep generative modelling.

Table 1 compares macrocanonical generators with the most widely used deep generative model classes and classical microcanonical synthesis. The framework trades distributional generality for three properties that are important in the data-limited scientific regime: (i) training from a handful, or even a single, realization; (ii) by-construction preservation of every symmetry respected by the moment constraints, rather than approximate symmetry learned from data; and (iii) amortized one-pass generation, like a GAN and unlike a diffusion model.

2 Microcanonical and Macrocanonical Generation

We first formalize the distinction between microcanonical synthesis methods and the macrocanonical generator formulation introduced in this work.

2.1 Microcanonical synthesis

Let $\phi(x)$ denote a differentiable vector of feature statistics extracted from a signal x. Given a reference signal with statistics ϕ^*, microcanonical synthesis generates samples by minimizing

$$L(x) = \frac{1}{2}\|\phi(x) - \phi^*\|^2. \tag{1}$$

Starting from an initialization $x^{(0)}$ drawn from a stationary Gaussian distribution with periodic boundary conditions [4, 35], samples are obtained by gradient descent in signal space

$$x^{(t+1)} = x^{(t)} - \eta \nabla_x L(x^{(t)}). \tag{2}$$

Each synthesized sample is therefore produced by an independent optimization procedure that progressively adjusts the signal until the desired feature constraints are satisfied. When the feature statistics ϕ are shift-equivariant (as is the case for wavelet-based statistics), the gradient descent preserves the stationarity of the initial distribution [4]. The Gaussian initialization provides maximum entropy subject to a mean and variance constraint, and convergence to the microcanonical set is guaranteed when the step size lies below the inverse Lipschitz constant of the gradient [4]. The resulting distribution is not in general maximum entropy on the microcanonical set, but inherits symmetries of the initial measure including stationarity (Theorem 3.4 and Corollary 3.5 of Bruna and Mallat 4).

2.2 Macrocanonical generators

The terminology reflects a physical distinction. A microcanonical formulation requires each realization to satisfy $\phi = \phi^*$ exactly, eliminating fluctuations. This is generally unphysical, because real physical systems are generated by ensembles defined only through expectation values. In the macrocanonical formulation the ensemble fixes $\phi^* = \langle\phi\rangle$, so individual realizations fluctuate around the constraint, as observed for spectra and analogous features in quantum field configurations, cosmological fields, turbulence, and quantum many-body energy spectra.

Instead of optimizing each sample independently, we introduce a neural generator $y = F_\theta(x)$, where $x \sim p_0$ is drawn from a simple input distribution such as a stationary Gaussian field with periodic boundary conditions. The generator is trained so that feature constraints are satisfied in expectation:

$$\mathbb{E}_{x\sim p_0}[\phi(F_\theta(x))] \approx \phi^*. \tag{3}$$

This corresponds to a macrocanonical formulation [16, 6]: moment constraints are enforced on the ensemble of generated samples rather than on each individual sample. The trained network therefore learns a global transport map $F_\theta : x \mapsto y$ that produces samples satisfying the same statistical constraints that would otherwise be enforced by per-sample optimization. Crucially, once training is complete, generating a new sample requires only a single forward pass through the network, whereas microcanonical synthesis requires a full iterative optimization for every sample. The cost of learning the transport map is thus amortized over all subsequent samples.

Why few realizations still work: The claim that ϕ^* can be estimated from very few realizations, or even a single realization, is a consequence of stationarity together with the shift-invariant, spatially averaged form of the feature statistics [4]. Each wavelet phase harmonic coefficient is an average over translates, $\phi_k(x) = \frac{1}{d}\sum_u f_k(T_u x)$, of a local functional f_k, where T_u denotes translation by u on a periodic grid of d sites. For a stationary field the variance of this average is $\frac{1}{d}\sum_u \mathrm{Cov}\big(f_k(X), f_k(T_u X)\big)$, which is $O(1/d)$ if the spatial correlations of f_k decay sufficiently rapidly. A single realization then behaves like an average over $O(d)$ weakly correlated copies, and n independent realizations reduce the variance to $O(1/(nd))$, so field size and the number of realizations play the same statistical role [4].

2.3 Residual generator parameterizations

We consider generator architectures that are initialized near the identity map [11]. In particular we study residual parameterizations of the form

$$F_\theta(x) = x + G_\theta(x) \tag{4}$$

and

$$F_\theta(x) = G_\theta(x) + H_\theta(x) \odot x, \tag{5}$$

where G_θ and H_θ are convolutional networks. The choice of convolutional architecture is motivated by the analogy with microcanonical synthesis: in microcanonical gradient descent, shift-equivariant features ϕ ensure that the gradient transport preserves stationarity [4, 35]. Similarly, convolutional G_θ and H_θ make the generator F_θ translation-equivariant, relating the macrocanonical generator to this symmetry-preserving structure. These parameterizations are initialized so that

$$G_{\theta_0}(x) = 0, \quad H_{\theta_0}(x) = \mathbf{1}. \tag{6}$$

Under this initialization the generator initially acts as the identity mapping: $F_{\theta_0}(x) = x$. If the input x is drawn from a stationary distribution, the generated samples are therefore also stationary at initialization. During early stages of optimization with a low learning rate, the learned transport remains close to the identity, which plays an important role in preserving both the entropy and the stationarity of the generated distribution, as we analyze in the following sections.

3 Training Dynamics of Macrocanonical Generators

We now analyze how gradient-based training of the generator affects the generated samples. Let $y = F_\theta(x)$ for $x \sim p_0$, and define the macrocanonical training objective

$$L(\theta) = \frac{1}{2} \left\| \mathbb{E}_{x \sim p_0}[\phi(F_\theta(x))] - \phi^* \right\|^2. \tag{7}$$

To understand the effect of this update in sample space, we linearize the generator around the current parameters.

Theorem 1 (Generator training induces preconditioned gradient descent)**.** *Assume (i) the generator F_θ is twice continuously differentiable with respect to its parameters θ, (ii) the feature map ϕ is differentiable with respect to its input, (iii) the learning rate η is sufficiently small that the second-order remainder below is negligible relative to the first-order term, and (iv) the required expectations and derivatives exist and differentiation may be interchanged with expectation.*

Let $L(\theta)$ denote the macrocanonical objective defined above. Write $m(\theta) = \mathbb{E}_{x'' \sim p_0}[\phi(F_\theta(x''))]$ for the mean feature vector, so $L(\theta) = \frac{1}{2}\|m(\theta) - \phi^*\|^2$. For each input x', define a per-sample objective

$$\widetilde{L}_{x'}(y) \;=\; \big\langle \phi(y),\; m(\theta^{(t)}) - \phi^* \big\rangle. \tag{8}$$

Then, for a fixed input sample x, the corresponding change in the generated sample satisfies, to first order in the parameter update,

$$y^{(t+1)}(x) \approx y^{(t)}(x) - \eta \mathbb{E}_{x' \sim p_0} \left[K_{\theta^{(t)}}(x, x')\, \nabla_y \widetilde{L}_{x'} \right], \tag{9}$$

where $y^{(t)}(x) = F_{\theta^{(t)}}(x)$, the approximation error is $O(\|\Delta\theta\|^2) = O(\eta^2)$ by assumption (i),

$$K_{\theta^{(t)}}(x, x') = J_{\theta^{(t)}}(x) J_{\theta^{(t)}}(x')^\top, \qquad J_{\theta^{(t)}}(x) = \frac{\partial F_{\theta^{(t)}}(x)}{\partial \theta}. \tag{10}$$

$\nabla_y \widetilde{L}_{x'}$ denotes the gradient of the per-sample objective evaluated at the current output $y = F_{\theta^{(t)}}(x')$,

$$\nabla_y \widetilde{L}_{x'} = \left(\nabla_y \phi(F_{\theta^{(t)}}(x'))\right)^\top \left(m(\theta^{(t)}) - \phi^* \right). \tag{11}$$

Thus generator training induces a preconditioned gradient descent in sample space, governed by the kernel $K_{\theta^{(t)}}(x, x')$.

Kernel interpretation. The full sample-space update is coupled across inputs through the cross-kernel $K_\theta(x, x') = J_\theta(x) J_\theta(x')^\top$. The update can therefore be written

$$y^{(t+1)} = y^{(t)} - \eta K_\theta \nabla_y \widetilde{L}, \tag{12}$$

where K_θ acts as an integral operator on the gradient field

$$(K_\theta g)(x) = \mathbb{E}_{x' \sim p_0}[K_\theta(x, x') g(x')]. \tag{13}$$

Generator training therefore induces a kernel-preconditioned gradient descent in sample space. The kernel couples updates across inputs through the generator Jacobian, so that the evolution of each sample depends on the sample-space gradients evaluated at other inputs x'. We note that for a batch containing the single input x, the update of the output at that same input reduces to a pointwise preconditioned step with the self-kernel $K_\theta(x,x) = J_\theta(x)J_\theta(x)^\top$, analogous to the Gauss–Newton or neural tangent kernel preconditioner [15].

The theorem does not assume training on a single (noise) input x: the update $\Delta\theta = -\eta\nabla_\theta L(\theta^{(t)})$ uses the full macrocanonical objective, with the expectation over x' inside. The theorem gives the change in the output at arbitrary input x. Since all outputs share θ, one update changes every sample.

Proof sketch. The argument has two ingredients. First, by assumptions (i) and (iii) the parameter update $\Delta\theta = -\eta\,\nabla_\theta L(\theta^{(t)})$ is small, so we may linearize the generator around $\theta^{(t)}$,

$$F_{\theta^{(t)}+\Delta\theta}(x) \approx F_{\theta^{(t)}}(x) + J_{\theta^{(t)}}(x)\,\Delta\theta, \qquad J_{\theta^{(t)}}(x) = \frac{\partial F_{\theta^{(t)}}(x)}{\partial\theta}, \tag{14}$$

which converts the parameter update into a sample-space update $y^{(t+1)}(x) \approx y^{(t)}(x) - \eta\,J_{\theta^{(t)}}(x)\,\nabla_\theta L(\theta^{(t)})$. Second, the chain rule applied to $L(\theta) = \frac{1}{2}\|m(\theta) - \phi^*\|^2$ with $m(\theta) = \mathbb{E}_{x'\sim p_0}[\phi(F_\theta(x'))]$, together with the regularity assumptions (ii) and (iv), expresses

$$\nabla_\theta L(\theta^{(t)}) = \mathbb{E}_{x'\sim p_0}\big[J_{\theta^{(t)}}(x')^\top\,\nabla_y \widetilde{L}_{x'}\big], \tag{15}$$

with $\nabla_y \widetilde{L}_{x'}$ as defined in the theorem statement. Substituting this expectation into the linearized sample update and identifying $K_{\theta^{(t)}}(x,x') = J_{\theta^{(t)}}(x)\,J_{\theta^{(t)}}(x')^\top$ gives the stated kernel-preconditioned form. A complete derivation, with the chain-rule manipulations written out step by step, is given in Appendix A. □

Proposition 1 (Symmetry preservation under equivariant generators)**.** *Let $\mathcal{G}$ be a group acting on the sample space by measurable transformations. Let $X \sim p_0$ and define $Y = F_\theta(X)$. Assume that: (1) p_0 is $\mathcal{G}$-invariant, i.e. for all $g \in \mathcal{G}$, the random variable gX has the same distribution as X; (2) the generator is $\mathcal{G}$-equivariant, i.e. $F_\theta(gx) = gF_\theta(x)$ for all $g \in \mathcal{G}$.*

Then Y is $\mathcal{G}$-invariant: for all $g \in \mathcal{G}$, the random variables gY and Y have the same distribution.

Proof. Fix $g \in \mathcal{G}$. Since p_0 is $\mathcal{G}$-invariant: $gX \overset{d}{=} X$.

Since $gX \overset{d}{=} X$ and F_θ is measurable,

$$F_\theta(gX) \overset{d}{=} F_\theta(X) = Y. \tag{16}$$

By equivariance of the generator,

$$F_\theta(gX) = gF_\theta(X) = gY, \text{ therefore } gY \overset{d}{=} Y. \tag{17}$$

Since this holds for every $g \in \mathcal{G}$, the random variable Y is $\mathcal{G}$-invariant. □

Corollary 1 (Stationarity of macrocanonical samples)**.** *Let $\mathcal{G}$ be the group of periodic translations on the grid. If p_0 is stationary and F_θ is translation-equivariant (as enforced by a convolutional architecture with periodic boundary conditions), then $p_\theta = \mathcal{L}(F_\theta(X))$ is stationary.*

Proof. Take $\mathcal{G}$ to be the group of periodic translations on the grid. By assumption, p_0 is invariant under these translations, and F_θ is equivariant under the same action. Proposition 1 therefore implies that $F_\theta(X)$ is stationary. □

Corollary 2 (Symmetry inheritance and amortization of microcanonical synthesis)**.** *Assume the conditions of Theorem 1, let $\mathcal{G}$ be the group of periodic grid translations acting by permutation matrices g, and assume ϕ is shift-invariant ($\phi(gy) = \phi(y)$ for all $g \in \mathcal{G}$), F_θ is translation-equivariant ($F_\theta(gx) = gF_\theta(x)$; e.g., a convolutional network with periodic boundary conditions), and p_0 is stationary. Then, for every $g \in \mathcal{G}$ and all inputs x, x': (i) the per-sample gradient is equivariant, $\nabla_y \widetilde{L}_{gx'} = g\,\nabla_y \widetilde{L}_{x'}$; (ii) the kernel intertwines the translation action, $K_{\theta^{(t)}}(gx, gx') = g\,K_{\theta^{(t)}}(x,x')\,g^{-1}$; and (iii) consequently, the one-step update map $U^{(t)} : y^{(t)}(\cdot) \mapsto y^{(t+1)}(\cdot)$ of*

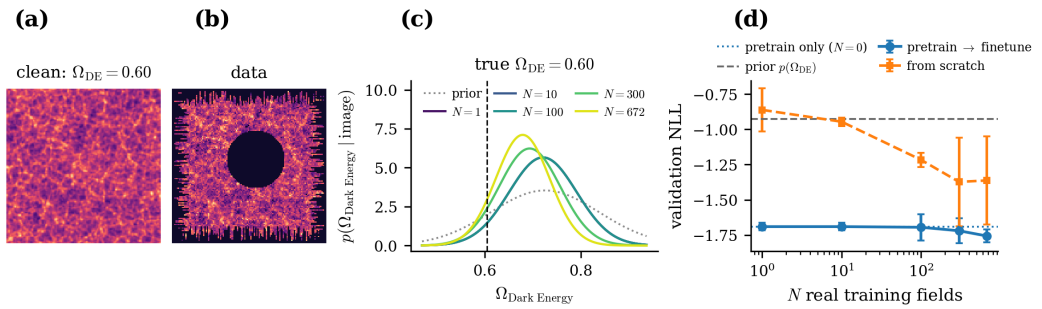

Figure 2: Simulation-based inference of the dark-energy fraction Ω_{DE} from 2-D density images (Section 5). **(a, b)** A validation example before and after the observational effects (i.e. masking). **(c)** Posterior after fine-tuning on N "real" simulations. **(d)** Validation NLL against N (inc. standard deviation with 3 seeds) using cosmologies in the Ω_{DE} training range. The surrogate-pretrained model improves on the prior before any "real" simulation is used and consistently outperforms the baseline.

Theorem 1 commutes with translations, $U^{(t)}(g\,\cdot) = g\,U^{(t)}(\cdot)$, and the generated distribution of $y^{(t)}$ is stationary at every training step t.

Stationarity of the generated distribution in fact holds exactly, for every parameter value, by Corollary 1; the content of (i)–(iii) is that the first-order training dynamics of Theorem 1 respect the same symmetry. Property (iii) is the same symmetry-preservation guarantee that Bruna and Mallat [4] establish for microcanonical pixel-space gradient descent (their Theorem 3.4), transferred to the training dynamics of the macrocanonical generator. After a single training run the generator amortizes the per-sample inner-loop optimization required by microcanonical synthesis [4, 35]: every subsequent sample is produced in one forward pass while inheriting the stationarity-preservation guarantee of property (iii).

Proof. See Appendix B.

Together, Theorem 1 and Corollary 2 provide the macrocanonical analogue of symmetry-preserving microcanonical dynamics. Theorem 1 describes the local sample-space geometry induced by training under the macrocanonical objective, in the regime of low learning rate where the generator remains well approximated by its local linearization. Proposition 1 and Corollary 1 show that equivariant generators preserve symmetries of the input distribution, including stationarity.

When $K_\theta(x, x)$ is close to the identity, the local induced dynamics resemble the pixel-space gradient descent used in microcanonical synthesis [4, 35].

This connection reveals the importance of the learning rate η: the linearization underlying Theorem 1 is valid when η is sufficiently small. This mirrors the requirement in microcanonical gradient descent that the step size be sufficiently small to ensure stable convergence. In the macrocanonical setting, a low learning rate typically ensures that each training step produces a small perturbation of the transport map, helping preserve entropy and maintain the intended stationarity (in practice, numerically) of the generated distribution.

4 Entropy and Volume Distortion

We now examine how the generator transformation affects the diversity and stationarity of generated samples. Let $x \sim p_0$ denote the input distribution and $y = F_\theta(x)$ the generated samples.

For a differentiable mapping F_θ, the input Jacobian $\mathcal{J}_\theta(x) = \partial F_\theta(x)/\partial x$ describes how local volumes are distorted by the transformation. We use $\mathcal{J}$ for the input Jacobian to distinguish it from the parameter Jacobian $J_\theta(x) = \partial F_\theta(x)/\partial\theta$ of Section 3. When F_θ is a diffeomorphism, the change of variables formula relates the differential entropy of the output to that of the input:

$$\mathcal{H}(Y) = \mathcal{H}(X) + \mathbb{E}_{x\sim p_0}\left[\log\left|\det \mathcal{J}_\theta(x)\right|\right], \tag{18}$$

where $\mathcal{H}(\cdot)$ denotes differential entropy. Even when F_θ is not globally invertible, the log absolute determinant of the Jacobian remains a useful measure of local volume change: regions where

$|\det \mathcal{J}_\theta(x)| \approx 1$ are approximately volume-preserving, while $|\det \mathcal{J}_\theta(x)| \to 0$ indicates that the mapping is collapsing a neighborhood to a lower-dimensional set.

For residual generators $F_\theta(x) = x + G_\theta(x)$, the input Jacobian takes the form

$$\mathcal{J}_\theta(x) = I + \frac{\partial G_\theta(x)}{\partial x}. \tag{19}$$

Because the generator is initialized with $G_{\theta_0}(x) = 0$, at initialization $\mathcal{J}_{\theta_0}(x) = I$ and therefore $\det \mathcal{J}_{\theta_0}(x) = 1$. The entropy of the generated distribution equals that of the input: $\mathcal{H}(Y) = \mathcal{H}(X)$. During early training with a low learning rate, the residual branch $G_\theta(x)$ can remain small, so the Jacobian tends to stay close to the identity and the mapping approximately preserves volume in input space.

This behavior helps maintain high entropy in the generated distribution and reduces the risk of mode collapse. In contrast, a generator without these inductive biases can rapidly learn a mapping that contracts input space, driving $|\det \mathcal{J}_\theta(x)| \to 0$ and producing a degenerate distribution. In the extreme case, training can drive the generator toward a near-constant mapping $F_\theta(x) \approx y^*$ for all x, in which the generator outputs essentially the same image for every input and the generated distribution has near-minimal entropy.

If the fixed output satisfies the constraints, $\phi(y^*) = \phi^*$ (i.e., y^* is a microcanonical solution), the constant mapping $x \mapsto y^*$ attains zero training loss and is therefore a global minimizer. Avoiding collapse is therefore a property of the training dynamics (identity initialization, residual parameterization and a small learning rate) rather than of the loss; empirically, the trained generator matches the across-sample variance of held-out simulations (Figure 4), whereas removing these inductive biases produces collapse (Figure 3).

A further consequence concerns stationarity. As noted in Section 2.3, the convolutional structure of G_θ and H_θ makes the generator translation-equivariant, mirroring the role of shift-equivariant features in microcanonical gradient descent where stationarity is preserved [4, 35]. Together with the residual parameterization and low learning rate, which keep the transport close to the identity, these design choices help preserve the stationarity of the generated field throughout training. Without these inductive biases, the generator can exhibit severe collapse, reducing entropy and sample diversity even when constrained moments are matched; in practical architectures this may yield apparent or effective stationarity violations in finite samples, even though exact equivariance would preserve stationarity in theory. This failure mode is particularly damaging because stationarity is the symmetry that enables estimation of statistics from a single realization in data-limited settings. Viewed through this lens, identity initialization and low learning rates act as an implicit regularization through the training dynamics, discouraging rapid collapse while feature constraints are being enforced.

To summarize, residual parameterizations initialized near the identity therefore provide two interrelated structural biases: (i) local volume preservation, maintaining entropy and avoiding mode collapse; and (ii) gradual departure from the identity, keeping training in the small-update regime of Theorem 1, in which the induced dynamics retain the per-sample transport structure of microcanonical synthesis. Combined with the convolutional architecture, which encodes translation equivariance, these biases help maintain both the diversity and the stationarity of the generated distribution. In practice, this avoids rapid mode collapse.

We find that both of these structural biases are needed. Removing either the residual shortcut or the small learning rate produces collapse (Figure 3), with detailed sweeps in Appendix G.

5 Experiments

We test macrocanonical generators on the axes that matter for scientific workflows: extreme data limits, faithful per-realization variability of summary statistics, wall-clock feasibility relative to reference simulators and iterative synthesis, and cold-start amortized inference when simulation budgets are inadequate. The two periodic stationary targets are a single 128×128 magnetohydrodynamic (MHD) turbulence snapshot ($N_{\text{train}} = 1$) and up to 100 fiducial Quijote cosmic-web slices (Appendix C); Figure 1 shows targets and syntheses. MHD turbulence produces stationary, strongly non-Gaussian fields with structure coupled across scales. The cosmic web is the filamentary matter distribution formed by gravitational collapse, whose physical structures depend on the cosmological parameters. In both cases, the simulators are often too expensive to generate the required training set sizes.

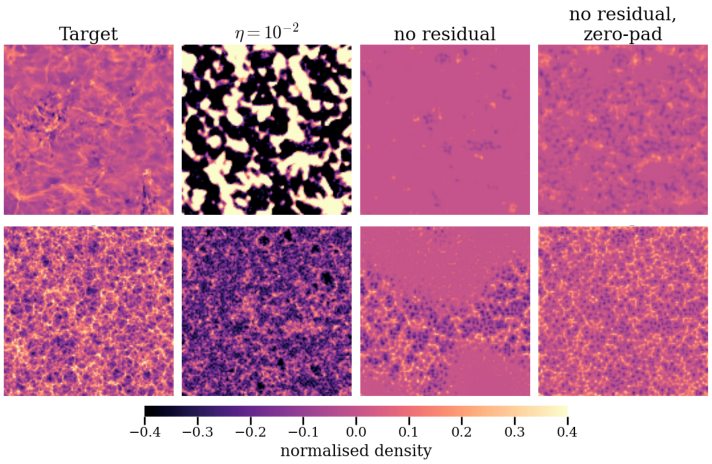

Figure 3: Inductive-bias ablation on the MHD (top) and cosmic-web (bottom) benchmarks; one target slice and one synthesis from each variant after 100 optimizer steps. The three failure modes correspond to removing the small step ($\eta = 10^{-2}$), the residual shortcut, and (additionally) the circular-padded convolutions; *zero-pad* means the U-Net is no longer translation-equivariant, so Corollary 1 no longer guarantees stationarity (producing boundary-correlated structure).

Setup. We instantiate (4)–(5) with a four-stage circular-padded U-Net, so F_θ is translation-equivariant on the torus (Corollary 1) and begins at the identity. Features ϕ are wavelet phase harmonics [23, 35]; ϕ^* is the empirical mean of ϕ over the available training fields; inputs $x \sim p_0$ are stationary Gaussian fields with power spectrum matched to the training pool, so second-order structure is correct before higher-order moments shape F_θ. We optimize with Adam at $\eta = 10^{-4}$ and global gradient-norm clipping. Architecture, WPH settings and optimizer schedules are detailed in Appendices E and H.

Effect of training-set size. Trained on 1, 10 or 100 fields from a separate suite of MHD turbulence simulations, scored against held-out runs (Appendix F), the generator produces new fields rather than copies even from a single training field, and its statistics approach those of held-out simulations within about ten fields, as expected from Section 2.2. On the cosmic-web benchmark, a single training field already reproduces the held-out sample-to-sample scatter.

Comparison with generative baselines. On the same data, WGAN-GP [10], flow matching [22] and a DDPM [13] memorize the training field when only one is available, even with translation augmentation [8], and close the gap only as many more fields are added (Appendix F). Microcanonical synthesis [4, 35] removes sample-to-sample fluctuations and is significantly slower per sample.

Per-realization feature distributions. Matching ϕ^* in expectation does not, by itself, guarantee that each realization exhibits realistic scatter in every summary channel. Yet simulation-based inference (SBI) fits conditional densities by exploiting precisely those fluctuations: posteriors become miscalibrated if every surrogate has near-identical statistics. Training on ϕ^* averaged over 100 Quijote slices without supervising per-realization variance, we draw 100 synthetic fields and compare each WPH coefficient against 100 held-out simulator slices. Coefficient-wise Kolmogorov–Smirnov tests give mean p-value $\langle p \rangle = 0.52$, consistent with draws from the same marginals (Appendix I, Figure 4), in line with Section 4. Microcanonical synthesis [4, 35] instead forces $\phi = \phi^*$ per realization, collapsing to a Dirac mass and removing real likelihood contributions from the posterior density.

Surrogate simulation speed. Cosmological inference ensembles routinely spend $\mathcal{O}(\text{hours})$–$\mathcal{O}(\text{days})$ of cluster time per nonlinear N-body realization [17], and Quijote-class simulations inherit the same bottleneck. After a single offline fit, our generator emits batches of fresh 128×128 fields at $\mathcal{O}(\text{ms})$ per sample on one NVIDIA A100—more than six orders of magnitude higher throughput than reference simulators at comparable fidelity, and more than 4×10^3 times faster than iterative microcanonical gradient-descent synthesis [4, 35] (which additionally destroys per-realization fluctuations,

as noted above). Once trained, the surrogate amortizes across arbitrarily many downstream queries—ensemble forecasting, uncertainty loops, iterative refinement—workflows otherwise inaccessible. Per-iteration training cost and exact wall-clock numbers are tabulated in Appendix D.

Pretraining for simulation-based inference. We infer $\Omega_{\mathrm{DE}} = 1 - \Omega_m$ from 128×128 matter-density projections in the Quijote Latin-hypercube suite. Ω_{DE} is the dark-energy fraction of the cosmic energy-density, which causes the accelerated expansion of the Universe, the effect of which is imprinted on the matter density structures (e.g. the morphology of filaments and halos). Posteriors on Ω_{DE} from such projected density images are a simplified version of the inference problem faced by galaxy surveys. Each cosmology requires a full N-body simulation (about 800 CPU core-hours per Quijote realization [32]), so conventional neural SBI hits an obstacle: even thirty cosmologies provide insufficient paired supervision for an image encoder without auxiliary simulators. We fit one macrocanonical generator per training cosmology from a single clean projection and synthesize a much larger labelled pool against which a small ResNet encoder coupled to a Gaussian neural density estimator can be pretrained, before fine-tuning on the available real simulations. Training the same architecture from scratch on identical simulation subsets remains worse on average with all 672 real fields than the pretrained model with none. Synthetic pretraining improves the validation NLL by 0.76 (from -0.93 to -1.69) before *any* genuine simulation is used for training (Figure 2). This shows that macrocanonical surrogates can unblock amortized scientific pipelines whose limiting factor is simulator throughput rather than model capacity. Appendix D gives further details.

Inductive-bias ablation. Finally we isolate the architectural biases advocated in Sections 2.3–4: identity residual initialization so Jacobian spectra begin at volume-preserving values, learning rates mild enough for Theorem 1's linearization, and circular convolutions enforcing translation equivariance so Corollary 1 applies on the torus. Removing each ingredient (training at $\eta = 10^{-2}$, deleting residual shortcuts, or swapping zero padding and circular padding) while freezing datasets and iteration budgets reproduces the predicted failure modes: collapsed transports, rapid entropy loss, or stationary-boundary artifacts (Figure 3).

6 Discussion and Limitations

We have introduced macrocanonical generator networks, an amortized neural surrogate for stationary physical processes that matches multiscale feature statistics in expectation, together with the entropy- and dynamics-based analysis (Sections 3–4) that identifies identity-initialized residual parameterizations and small learning rates as the inductive biases responsible for diversity-preserving training. On the cosmological dark-energy inference task of Section 5 (a real-world scientific task whose limiting factor is N-body simulator throughput) pretraining with the macrocanonical-generated surrogates makes simulation-based inference of dark energy from matter-density image data tractable in a regime where the available real-simulation budget alone is insufficient.

Limitation discussion:

- The method depends on the choice of feature statistics. If the chosen statistics do not capture the relevant structure of the target process, generated samples may fail to reproduce important properties.
- The framework is designed for stationary processes with periodic boundary conditions, where translation invariance provides the symmetry group that enables estimation from few realizations. For non-stationary processes the method cannot directly exploit this symmetry, and the data efficiency advantages may not apply. An interesting direction for future work is to pretrain generators on stationary processes (e.g., periodic fluid simulations) and then fine-tune on non-stationary simulations, which we have not yet tested. The periodic-boundary requirement can be relaxed, for example by appropriately masking or padding the region near the boundary (naive non-periodic application of the method instead produces the boundary artifacts in Figure 3).
- While the generator often appears empirically to produce samples consistent with maximum-entropy distributions under the imposed constraints, and we provide theoretical justification for this, we cannot currently provide a general theoretical guarantee of convergence to such a distribution (similarly, neither do [4, 35]), nor of avoiding collapse (Section 4).

Acknowledgments and Disclosure of Funding

NJ thanks Benjamin Joachimi and Lorne Whiteway for useful discussions. The authors acknowledge support from the Simons Collaboration on "Learning the Universe". NJ acknowledges support from the King's AI+ Academic Fellowship at King's College London.

References

[1] Erwan Allys, Théo Marchand, Jean-François Cardoso, Francisco Villaescusa-Navarro, Shirley Ho, and Stéphane Mallat. New interpretable statistics for large-scale structure analysis and generation. *Physical Review D*, 102(10):103506, 2020.

[2] Justin Alsing, Tom Charnock, Stephen Feeney, and Benjamin Wandelt. Fast likelihood-free cosmology with neural density estimators and active learning. *Monthly Notices of the Royal Astronomical Society*, 2019. doi: 10.1093/mnras/stz1960.

[3] Jörn Behrens and Frédéric Dias. New computational methods in tsunami science. *Philosophical Transactions of the Royal Society A*, 373(2053):20140382, 2015.

[4] Joan Bruna and Stéphane Mallat. Multiscale sparse microcanonical models. *Mathematical Statistics and Learning*, 1:257–315, 2018.

[5] Sihao Cheng and Brice Ménard. Weak lensing scattering transform: dark energy and neutrino mass sensitivity. *Monthly Notices of the Royal Astronomical Society*, 507(1):1012–1020, 2021.

[6] Thomas M Cover and Joy A Thomas. *Elements of Information Theory*. Wiley-Interscience, 2006.

[7] Kyle Cranmer, Johann Brehmer, and Gilles Louppe. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48):30055–30062, 2020.

[8] Alessandro Favero, Antonio Sclocchi, Francesco Cagnetta, Pascal Frossard, and Matthieu Wyart. How compositional generalization and creativity improve as diffusion models are trained. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025.

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

[10] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of Wasserstein GANs. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[12] Yuga Hikida, Ayush Bharti, Niall Jeffrey, and François-Xavier Briol. Multilevel neural simulation-based inference. *Advances in Neural Information Processing Systems*, 38, 2025.

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851, 2020.

[14] Mathias Hoppe, Ola Embreus, and Tünde Fülöp. DREAM: A fluid-kinetic framework for tokamak disruption runaway electron simulations. *Computer Physics Communications*, 268: 108098, 2021.

[15] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[16] Edwin T Jaynes. Information theory and statistical mechanics. *Physical Review*, 106(4):620, 1957.

[17] Niall Jeffrey et al. Dark energy survey year 3 results: likelihood-free, simulation-based wCDM inference with neural compression of weak-lensing map statistics. *Monthly Notices of the Royal Astronomical Society*, 536(2):1303–1322, 2025.

[18] Andrew Kirby, François-Xavier Briol, Thomas D Dunstan, and Takafumi Nishino. Data-driven modelling of turbine wake interactions and flow resistance in large wind farms. *Wind Energy*, 26(9):875–1011, 2023.

[19] Etienne Lempereur, Nathanael Cuvelle-Magar, Florentin Coeurdoux, Stephane Mallat, and Eric Vanden-Eijnden. MGD: Moment guided diffusion for maximum entropy generation. *arXiv preprint arXiv:2602.17211*, 2026.

[20] Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. MMD GAN: Towards deeper understanding of moment matching network. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[21] Yujia Li, Kevin Swersky, and Richard Zemel. Generative moment matching networks. In *International Conference on Machine Learning (ICML)*, 2015.

[22] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

[23] Stéphane Mallat, Sixin Zhang, and Gaspar Rochette. Phase harmonic correlations and convolutional neural networks. *Information and Inference: A Journal of the IMA*, 9(3):721–747, 2020.

[24] Michael McCabe, Payel Mukhopadhyay, Tanya Marwah, Bruno Regaldo-Saint Blancard, Francois Rozet, Cristiana Diaconu, Lucas Meyer, Kaze W. K. Wong, Hadi Sotoudeh, Alberto Bietti, et al. Walrus: A cross-domain foundation model for continuum dynamics. *arXiv preprint arXiv:2511.15684*, 2025.

[25] Daniel McFadden. A method of simulated moments for estimation of discrete response models without numerical integration. *Econometrica*, 57(5):995–1026, 1989.

[26] Ruben Ohana, Michael McCabe, Lucas Meyer, Rudy Morel, Fruzsina J. Agocs, Miguel Beneitez, Marsha Berger, Blakesley Burkhart, Keaton Burns, Stuart B. Dalziel, Drummond B. Fielding, Daniel Fortunato, Jared A. Goldberg, Keiya Hirashima, Yan-Fei Jiang, Rich R. Kerswell, Suryanarayana Maddu, Jonah Miller, Payel Mukhopadhyay, Stefan S. Nixon, Jeff Shen, Romain Watteaux, Bruno Régaldo-Saint Blancard, François Rozet, Liam H. Parker, Miles Cranmer, and Shirley Ho. The Well: a Large-Scale Collection of Diverse Physics Simulations for Machine Learning. *arXiv e-prints*, art. arXiv:2412.00568, November 2024. doi: 10.48550/arXiv.2412.00568.

[27] Liam Parker, Francois Lanusse, Jeff Shen, Ollie Liu, Tom Hehir, Leopoldo Sarra, Lucas Meyer, Micah Bowles, Sebastian Wagner-Carena, Helen Qu, Siavash Golkar, Alberto Bietti, et al. AION-1: Omnimodal foundation model for astronomical sciences. *arXiv preprint arXiv:2510.17960*, 2025.

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

[29] Javier Portilla and Eero P Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *International Journal of Computer Vision*, 40(1):49–70, 2000.

[30] B. Regaldo-Saint Blancard, F. Levrier, E. Allys, E. Bellomi, and F. Boulanger. Statistical description of dust polarized emission from the diffuse interstellar medium: A rwst approach. *Astronomy & Astrophysics*, 642:A217, October 2020. ISSN 1432-0746. doi: 10.1051/0004-6361/202038044. URL `http://dx.doi.org/10.1051/0004-6361/202038044`.

[31] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pages 1530–1538, 2015.

[32] Francisco Villaescusa-Navarro, ChangHoon Hahn, Elena Massara, Arka Banerjee, Andrij M. Delgado, Doogesh Kodi Ramanah, Tom Charnock, Elena Giusarma, Yin Li, Erwan Allys, et al. The Quijote simulations. *The Astrophysical Journal Supplement Series*, 250(1):2, 2020.

[33] Francisco Villaescusa-Navarro et al. The CAMELS project: Public data release. *The Astrophysical Journal Supplement Series*, 265(2):54, 2023.

[34] Florian Wiesner, Matthias Wessling, and Stephen Baek. Towards a physics foundation model. *arXiv preprint arXiv:2509.13805*, 2025.

[35] Sixin Zhang and Stéphane Mallat. Maximum entropy models from phase harmonic covariances. *Applied and Computational Harmonic Analysis*, 53:199–230, 2021.

A Proof of Theorem 1

We give the complete derivation of the kernel-preconditioned sample-space update stated in Theorem 1; a high-level argument is sketched in the main text.

Proof of Theorem 1. Let

$$m(\theta) = \mathbb{E}_{x' \sim p_0}\big[\phi(F_\theta(x'))\big], \qquad L(\theta) = \frac{1}{2}\|m(\theta) - \phi^*\|^2. \tag{20}$$

A gradient descent step gives

$$\theta^{(t+1)} = \theta^{(t)} - \eta \nabla_\theta L(\theta^{(t)}). \tag{21}$$

Because the learning rate is assumed to be sufficiently small, the parameter update is small, so for a fixed input x we may linearize the generator around $\theta^{(t)}$, with remainder $O(\|\Delta\theta\|^2)$ by assumption (i):

$$F_{\theta^{(t)}+\Delta\theta}(x) \approx F_{\theta^{(t)}}(x) + J_{\theta^{(t)}}(x)\Delta\theta, \tag{22}$$

where

$$J_{\theta^{(t)}}(x) = \frac{\partial F_{\theta^{(t)}}(x)}{\partial \theta}. \tag{23}$$

Writing $y^{(t)}(x) = F_{\theta^{(t)}}(x)$ and $\Delta\theta = -\eta \nabla_\theta L(\theta^{(t)})$, this gives

$$y^{(t+1)}(x) \approx y^{(t)}(x) - \eta J_{\theta^{(t)}}(x) \nabla_\theta L(\theta^{(t)}). \tag{24}$$

By the chain rule,

$$\nabla_\theta L(\theta) = \left(\frac{\partial m(\theta)}{\partial \theta}\right)^\top \big(m(\theta) - \phi^*\big). \tag{25}$$

Assuming differentiation may be interchanged with expectation,

$$\frac{\partial m(\theta)}{\partial \theta} = \mathbb{E}_{x' \sim p_0}\left[\frac{\partial \phi(F_\theta(x'))}{\partial \theta}\right]. \tag{26}$$

For each sample x', the chain rule gives

$$\frac{\partial \phi(F_\theta(x'))}{\partial \theta} = \nabla_y \phi(F_\theta(x'))\, J_\theta(x'). \tag{27}$$

Define

$$\nabla_y \widetilde{L}_{x'} = \big(\nabla_y \phi(F_\theta(x'))\big)^\top \big(m(\theta) - \phi^*\big). \tag{28}$$

Then

$$\nabla_\theta L(\theta) = \mathbb{E}_{x' \sim p_0}\left[J_\theta(x')^\top \nabla_y \widetilde{L}_{x'}\right]. \tag{29}$$

Substituting into the linearized sample update yields

$$y^{(t+1)}(x) \approx y^{(t)}(x) - \eta \mathbb{E}_{x' \sim p_0}\left[J_{\theta^{(t)}}(x) J_{\theta^{(t)}}(x')^\top \nabla_y \widetilde{L}_{x'}\right]. \tag{30}$$

Defining

$$K_{\theta^{(t)}}(x, x') = J_{\theta^{(t)}}(x) J_{\theta^{(t)}}(x')^\top \tag{31}$$

gives the stated result. □

B Proof of Corollary 2

Proof of Corollary 2. By assumption ϕ is shift-invariant, so for any translation g and input z, $\phi(gz) = \phi(z)$, and the residual $\phi(F_\theta(x')) - \phi^*$ is invariant under $x' \mapsto gx'$. Periodic translations act by permutation matrices, so $g^\top = g^{-1}$. Differentiating $\phi(gy) = \phi(y)$ in y and using translation equivariance of F_θ gives $\nabla_y \widetilde{L}_{gx'} = g\, \nabla_y \widetilde{L}_{x'}$. Translation equivariance of F_θ also gives $J_\theta(gx) = g\, J_\theta(x)$, hence $K_\theta(gx, gx') = g\, K_\theta(x, x')\, g^{-1}$. The integrand of the per-sample update of Theorem 1 therefore satisfies

$$K_\theta(gx, gx')\, \nabla_y \widetilde{L}_{gx'} = g\big(K_\theta(x, x')\, \nabla_y \widetilde{L}_{x'}\big). \tag{32}$$

Since p_0 is invariant under g, the change of variables $x' \mapsto gx'$ leaves the expectation unchanged, so the full update map $y^{(t)}(x) \mapsto y^{(t+1)}(x)$ is shift-equivariant. Stationarity of the generated distribution of $y^{(t)}$ at every training step t then follows from Proposition 1 applied to $F_{\theta^{(t)}}$, which is translation-equivariant for every $\theta^{(t)}$. □

C Quijote N-body simulations and slice preprocessing

The fixed-cosmology cosmic-web tests use 2-D slices from the single-cosmology fiducial Quijote pool described below: the cosmic-web training-set-size comparison in Section 5, the per-realization feature distributions, and the cosmic-web ablation. These slices are extracted from the Quijote suite of N-body simulations [32], a publicly released ensemble of $\mathcal{O}(10^4)$ dark-matter-only simulations spanning a wide range of cosmological parameters and designed to support large-scale-structure cosmology and machine-learning applications. The simulation-based inference experiment uses the Quijote Latin-hypercube subset, with its projection and split protocol in Appendix D.

Simulation setup. Each Quijote run evolves a set of dark-matter particles inside a periodic cubic box from a high-redshift initial condition under gravity. After the gravitational evolution, the particles are deposited onto a regular three-dimensional density grid using a mass-assignment scheme. For these fixed-cosmology tests we use the *fiducial* cosmology, in which the cosmological parameters are held fixed at values consistent with current cosmic-microwave-background and large-scale-structure constraints; this single-cosmology subset provides a large pool of statistically independent realizations of the matter density field, all governed by the same underlying physics.

Slice extraction. From each three-dimensional density snapshot we take a 2-D slice through the volume at $z = 0$, producing a square map of the projected matter density. We take 100 such slices in total, drawn from the fiducial Quijote pool, and treat them as independent realizations of the same stationary 2-D random field with periodic boundary conditions inherited from the simulation box.

Preprocessing. The raw surface-density values span several orders of magnitude due to the highly non-Gaussian, filamentary structure of the cosmic web. To compress this dynamic range we apply $x \mapsto \log_{10}(x + 1)$ to the surface density of each slice. We then downsample to a 128×128 pixel grid by spatial mean-pooling. The pixel values are subsequently normalized to zero mean and unit variance using the empirical mean and standard deviation computed over the entire pool of slices, and finally rescaled by a constant factor of $1/8$ so that the resulting fields lie comfortably inside the dynamic range of the WPH operator. The resulting slices are stationary, periodic on the simulation grid, and exhibit the characteristic non-Gaussian filamentary structure that motivates the use of higher-order summary statistics such as wavelet phase harmonic coefficients.

D Throughput and simulation-based inference protocol

Reference N-body cost. A single N-body simulation of the kind used in modern weak-lensing cosmology, for example the Gower Street suite produced for recent Dark Energy Survey analyses [17], typically requires $\mathcal{O}(\text{hours to days})$ of compute per realization, even when distributed across hundreds of cores. The Quijote suite required more than 35 million CPU core-hours for its $44{,}100$ simulations, about 800 core-hours per realization [32], so even 30 Latin-hypercube cosmologies constitute an aggressive pool for amortized inference.

Macrocanonical inference throughput. Once trained, the macrocanonical generator produces a batch of 200 new 128×128 realizations in one forward pass that completes in well under a second on

a single NVIDIA A100 GPU—a few milliseconds per field and more than six orders of magnitude higher throughput than a single Gower-Street-class N-body realization.

Training wall-clock. Training uses two A100 GPUs at batch size 200, costing roughly one minute per 25 iterations. The simulation-based inference configurations use 500 iterations ($\sim$20 minutes); the higher-fidelity runs for Section 5 use 5000 iterations (a few hours).

Microcanonical synthesis baseline. Microcanonical gradient-descent synthesis [4, 35] solves a fresh inner loop per sample and takes approximately 20 seconds per 128×128 field on the same A100—more than 4×10^3 slower per sample than the amortized generator—while pinning $\phi = \phi^*$ exactly (Section 5), so it cannot supply the fluctuations downstream inference requires.

Amortization. One-off generator training is amortized across all later queries, making simulation-based inference, uncertainty quantification, and sensitivity analysis feasible where repeated N-body runs are not.

SBI data splits and corruption. We draw 32 non-overlapping 128×128 projections from each of the first 30 Quijote Latin-hypercube simulations, split by cosmology into 24 training cosmologies (768 training fields) and 6 validation cosmologies (192 validation fields). The same fixed instrument-footprint corruption (heteroskedastic noise with per-pixel standard-deviation map, jagged border, centred-circle mask) is applied to every real and synthetic image. We use a small ResNet embedding f_ψ and a single-component Gaussian estimator $p(\Omega_{\rm DE} \mid \text{image}) = \mathcal{N}(\mu(f_\psi), \sigma(f_\psi)^2)$, with Adam at 5×10^{-5} for fine-tuning and 5×10^{-4} for from-scratch training, plus patience-based early stopping.

Pretraining and fine-tuning. We train one macrocanonical generator per training cosmology from a single 128×128 WPH target, draw 1000 synthetic fields per cosmology, yielding 24,000 synthetic training fields compared with 768 real training tiles (24×32 non-overlapping projections). We pretrain the encoder and Gaussian neural density estimator on synthetics, then fine-tune on real subsets of size $N \in \{1, 10, 100, 300, 672\}$ (96 of the 768 training tiles, four projections per training cosmology, are held out for early stopping of all models). An identical architecture trained from scratch on the same real subsets is the baseline.

Likelihood results. Validation NLL is averaged over the four validation cosmologies whose $\Omega_{\rm DE}$ lies inside the training range $[0.525, 0.887]$ (the other two require extrapolation) and over three seeds. Training from scratch stays at the empirical prior (-0.93) up to N=10 and reaches -1.4 ± 0.3 at $N \geq 300$; pretraining alone gives -1.69 ± 0.03 before any genuine simulation is used for training, and fine-tuning reaches -1.76 at N=672 (Figure 2).

Synthetic ensemble cost. After $\sim$20 minutes of generator training per cosmology on two A100s, each network emits $\sim$200 fields per second. The resulting 24,000-tile synthetic training corpus would correspond to 750 additional Quijote realizations, roughly 6×10^5 CPU core-hours, for comparably dense cosmology–image supervision.

E Experimental setup details

This appendix gives the architecture, initialization, and optimizer details summarized in Section 5.

Code availability. A reference implementation of the macrocanonical generator, with training and sampling scripts for both the cosmic web and MHD simulation of Section 5, is available at `https://github.com/NiallJeffrey/MacrocanonicalGeneratorNetwork`.

Generator architectures. We instantiate the two residual parameterizations introduced in Section 2.2. The *additive* form of eq. (4) uses a single convolutional U-Net for G_θ. The *multiplicative* form of eq. (5) uses two independent U-Nets of identical architecture for G_θ and H_θ. Both networks share the same backbone: a four-stage encoder–decoder with channel widths $64 \to 1024$, 3×3 convolutions, batch normalization, and ReLU activations. Because every target process in our experiments is periodic, all convolutions use circular padding and all bilinear upsamples are wrapped in a circular

pad/interpolate/crop block, so the network is exactly translation-equivariant on the periodic grid (Corollary 1). Convolutional weights are initialized from $\mathcal{N}(0, 10^{-4})$ so that $G_{\theta_0}(x) \approx 0$; for the multiplicative parameterization, an additive constant is appended to the forward map of H_θ so that $H_{\theta_0}(x) \approx \mathbf{1}$. Both parameterizations therefore begin training at the identity, $F_{\theta_0}(x) = x$, as required by the analysis of Sections 3 and 4. Activation checkpointing is used in the U-Net body to reduce peak memory at the cost of an additional forward pass during the backward, allowing batch size 200 at 128×128 resolution on a single GPU.

Feature statistics. We use wavelet phase harmonic (WPH) coefficients [23, 35] as the feature map ϕ, computed under periodic boundary conditions. The hyperparameters $(J, L, \Delta\ell, \Delta n)$ and the phase-harmonic exponent list are chosen per dataset; the per-benchmark values used in this work are summarized in Appendix H, Table 6. The training target ϕ^* is the empirical mean of the WPH coefficients over the available training realizations; in the single-realization regime this reduces to the WPH coefficients of that one signal.

MHD benchmark data. The single 128×128 MHD turbulence snapshot of Section 5 is derived from a 512×512 map of simulated dust emission from the magnetohydrodynamic simulation of the diffuse interstellar medium used by Regaldo-Saint Blancard et al. [30]. The map is stored as a complex-valued array; we take its modulus, apply a $\log_{10}$ transform, mean-pool to 128×128, subtract a smooth large-scale component (a Gaussian filter of width 8 pixels with periodic wrapping), standardize to zero mean and unit variance, and rescale by $1/16$.

Input distribution. p_0 is a stationary Gaussian random field with periodic boundary conditions, generated by FFT from the radially averaged power spectrum estimated from the training data. This matches the second-order statistics of the target process while leaving all higher-order structure to be learned by F_θ.

Optimization. We minimize the macrocanonical loss with Adam at base learning rate 10^{-4} and global gradient-norm clipping at 1.0, optionally preceded by a short linear warmup and followed by cosine annealing over the configured number of iterations. The generator and the WPH operator are placed on separate GPUs to accommodate their respective memory footprints; cross-device autograd is handled transparently by PyTorch.

F MHD turbulence suite for the controlled comparisons

The training-set-size sweep and the baseline comparison of Section 5, and the sweeps of Appendix G, require many independent realizations of one stationary process: to train at $N_{\text{train}} = 10$ and 100, and to score every method against held-out fields that no method has seen. The single MHD snapshot of the main benchmark cannot supply these, so for these experiments we use the `MHD_64` dataset of The Well [26]: three-dimensional isothermal, compressible MHD turbulence without self-gravity, simulated at 256^3 with periodic boundaries and released at 64^3 after an ideal low-pass filter, with independent runs for each combination of sonic and Alfvénic Mach number. We use the single combination $\mathcal{M}_s = 2$, $\mathcal{M}_A = 0.7$, so that all fields are realizations of one stationary process, and take two-dimensional 64×64 slices of the density at fixed z. The low-pass filter can produce small non-positive density values, which we clamp to the smallest positive value before taking $\log_{10}$. Training fields are drawn from the first of the 8 runs in the released training file, subsampled every 8 slices in z and every 5 snapshots in time to reduce correlation; the 100 held-out evaluation fields are drawn in the same way from the remaining 7 runs. The held-out fields are therefore independent simulations, whereas the 10 or 100 training fields are correlated slices of a single run. Fields are standardized with the mean and standard deviation of the training fields only and divided by 8; the WPH configuration is $J = 4$, $L = 4$, $\Delta n = 1$, $\Delta\ell = 2$ with the default phase-harmonic exponents. Every method in Tables 2 and 3 is trained and scored on exactly these fields.

The macrocanonical generator uses the multiplicative parameterization (5) with the optimizer settings of Appendix E, trained for 1500 iterations at batch size 64 and without augmentation. The learned baselines are trained for 20,000 iterations at batch size 32: WGAN-GP [10] with five critic steps

per generator step and gradient-penalty weight 10; flow matching [22] and the DDPM [13] with the same 5.3M-parameter circular-padded U-Net, sampled with 100 Euler steps and 250 DDIM steps respectively. During DDPM sampling we clamp the predicted clean field to the range of the training data (its 0.1st to 99.9th percentiles, widened by 10%); without this safeguard the sampler diverges on these fields. Translation augmentation, where used, applies a random cyclic shift to the training field at every step. Microcanonical synthesis runs 400 steps of pixel-space gradient descent per sample.

Metrics. The WPH distance is the root-mean-square difference, over the normalized WPH coefficients, between the mean feature vector of the generated samples and that of the held-out fields. KS mean-p is the mean, over WPH coefficients, of the two-sample Kolmogorov–Smirnov p-value comparing the across-sample distribution of that coefficient for generated and held-out fields: it is near 0.5 when the per-realization distributions match and near 0 when the scatter is missing or shifted. The nearest-neighbour (NN) ratio is the root-mean-square distance from a sample to its nearest training field, minimized over cyclic translations, divided by the same quantity for held-out fields; values much less than 1 indicate memorization, including memorization of shifted copies. Numbers in parentheses are held-out fields scored against themselves, which is the level a faithful surrogate should reach.

Table 2: Effect of training-set size (64×64 MHD suite of Appendix F, scored against 100 held-out fields; parentheses: held-out fields scored against themselves).

N_{train}	WPH distance	KS mean-p	NN ratio
1	0.20 (0.053)	0.08 (0.54)	0.89 (1.00)
10	0.075 (0.056)	0.27 (0.47)	0.98 (0.96)
100	0.086 (0.037)	0.31 (0.50)	0.99 (1.01)

Table 3: Baseline comparison on the 64×64 MHD suite of Appendix F: all methods trained on the same fields and scored against 100 held-out fields. NN ratios much less than 1 mean memorization; held-out fields scored against themselves give 1.00, 0.96 and 1.01 at $N_{\text{train}} = 1$, 10 and 100. Sampling time is per field on one A100.

	NN ratio at N_{train}				
Method	1	10	100	ms/sample	Outcome at $N_{\text{train}} = 1$
Macrocanonical generator (ours)	0.89	0.98	0.99	0.9	new fields
Microcanonical synthesis	0.87	–	–	9200	statistics pinned (no fluctuations)
DDPM	0.32	0.58	0.86	107	memorizes training field
WGAN-GP	0.010	0.33	0.84	0.09	memorizes training field
WGAN-GP + translation aug.	0.24	–	–	0.10	memorizes shifts of it
Flow matching	0.054	0.24	0.66	36	memorizes training field
Flow matching + translation aug.	0.052	–	–	36	memorizes training field

The baselines improve as more independent fields become available, which is the expected signature of fitting the empirical data distribution: diffusion models in particular generalize only above a dataset-size threshold and reproduce their training data below it [8]. By $N_{\text{train}} = 100$, WGAN-GP matches the mean feature vector at least as well as our generator (WPH distance 0.058 against our 0.086; held-out 0.037), whereas the DDPM still has a WPH distance of 0.14 and a KS mean-p of 0.007 (held-out 0.50), with a per-coefficient scatter of only 0.28 of the held-out level (ours: 0.67). The advantage of the macrocanonical generator is therefore confined to the regime in which fields are scarce, which is the regime that motivates the method.

Single training field at the paper's resolution. The $N_{\text{train}} = 1$ row of Table 2 is limited by the 64×64 field size rather than by the training budget: increasing the budget from 1500 to 5000 iterations leaves the WPH distance (0.20 in both cases) and the NN ratio (0.89 against 0.87) unchanged. Repeating the single-field protocol on the 128×128 Quijote benchmark (one training slice, 100 generator samples scored against 100 held-out slices) gives a per-coefficient scatter ratio, the mean over WPH coefficients of the generated to the held-out across-sample standard deviation, of 1.14

(held-out slices scored against themselves: 1.05), an NN ratio of 1.00 and a pixel-marginal KS distance of 0.015 (0.014). This is the result referred to in Section 5: the widths of the per-coefficient distributions are reproduced from a single 128×128 field, although their positions are not all matched at this training-set size (WPH distance 0.052 against 0.021 for held-out slices, KS mean-p 0.15 against 0.56), as on the MHD suite.

G Learning-rate, initialization-scale and input-distribution sweeps

Two hyperparameters govern whether training remains in the near-identity regime analyzed in Sections 3 and 4: the learning rate η and the standard deviation σ of the initial residual weights. We sweep each in turn on the 64×64 MHD suite of Appendix F ($N_{\text{train}} = 100$, 600 iterations at batch size 32, multiplicative parameterization, all other settings as in Appendix E) and record three diagnostics on a fixed probe batch every 10 iterations. The sample diversity is the per-pixel standard deviation across the generated batch divided by that of the input batch, so a value near 1 means the generator has not contracted the input distribution. The per-step output change, $\|F_{\theta^{(t+1)}}(x)-F_{\theta^{(t)}}(x)\|/\|F_{\theta^{(t)}}(x)\|$, is the quantity that the linearization of Theorem 1 requires to be small. The input-Jacobian deviation, $\|\mathcal{J}_\theta(x)v - v\|/\|v\|$ for a random direction v computed by finite differences, measures how far the map is from the identity and tracks the volume-preservation argument of Section 4.

Learning rate. For $\eta \in \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$ the median per-step output change grows as 0.03, 0.085, 0.14 and 0.46 (Table 4). At the paper's $\eta = 10^{-4}$ the input-Jacobian deviation grows gradually from 0 to 0.63 over training, and the sample diversity never falls below 0.93 and ends at 0.98, the input level. At $\eta = 10^{-2}$ the first updates take the map far from the identity (input-Jacobian deviation above 300 within the first 20 iterations, diversity falling to 0.70); this is the learning rate at which Figure 3 shows collapse after 100 steps. On this small problem the run later recovers its diversity, but it ends with a larger constraint error than $\eta = 10^{-3}$. The two smallest learning rates have not reached the constraints after 600 iterations, which is why the paper's runs use 1500 to 5000 iterations at $\eta = 10^{-4}$.

Table 4: Learning-rate sweep (64×64 MHD, 600 iterations, all other settings fixed). The per-step change is the median relative output change on a fixed probe batch; the linearization of Theorem 1 requires it to be small. The input-Jacobian deviation and diversity are given as the extreme value over training and, in parentheses, the value at the end of training; the loss is the mean over the last 50 iterations.

η	Median per-step change	Input-Jacobian deviation, max (final)	Diversity, min (final)	Loss
10^{-5}	0.03	0.40 (0.39)	0.94 (1.02)	2.0
10^{-4} (paper)	0.085	0.65 (0.63)	0.93 (0.98)	1.2
10^{-3}	0.14	0.70 (0.69)	0.83 (0.99)	0.76
10^{-2}	0.46	322 (0.90)	0.70 (0.98)	0.83

Initialization scale. The paper draws the initial residual weights from $\mathcal{N}(0,\, \sigma^2)$ with $\sigma = 10^{-4}$; σ controls how far the generator starts from the identity. We sweep $\sigma \in \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 3\times10^{-2}, 10^{-1}\}$ at $\eta = 10^{-4}$ (Table 5). Measured on the training inputs, the initial departure from the identity $\|F(x) - x\|/\|x\|$ runs from below 10^{-11} to 4.5, so the sweep spans the range from the identity to a map far from it. Training reaches the constraints from all six starting points: the mean loss over the final 50 iterations ranges from 0.36 to 0.85, against a batch-to-batch scatter of 0.23–0.38 within each run, with the largest σ the worst, and the sample diversity at the end of training lies between 0.90 and 0.96 of the input's. What changes is the trained map itself: its input-Jacobian deviation is 0.6–0.9 for $\sigma \le 10^{-3}$ but 2.1, 4.9 and 11.1 at $\sigma = 10^{-2}$, 3×10^{-2} and 10^{-1}, so a larger initialization leaves the trained map further from a perturbation of the identity, and the diversity is lowest at the largest σ.

Relation to the ablation of Figure 3. The initialization sweep complements the ablation rather than repeating it. In the ablation the residual shortcut is removed altogether and the transport collapses; with the shortcut in place the input still reaches the output even from a poor initialization, which

Table 5: Initialization-scale sweep (64×64 MHD, $\eta = 10^{-4}$, 600 iterations, all other settings fixed). The loss is the mean over the last 50 iterations; the diversity and input-Jacobian deviation are measured at the end of training.

σ	$\|F(x) - x\|/\|x\|$ at initialization	Input-Jacobian deviation	Diversity	Loss
10^{-5}	$< 10^{-11}$	0.81	0.96	0.51
10^{-4} (paper)	5×10^{-5}	0.61	0.94	0.70
10^{-3}	0.046	0.86	0.96	0.44
10^{-2}	0.44	2.1	0.93	0.36
3×10^{-2}	1.3	4.9	0.91	0.49
10^{-1}	4.5	11.1	0.90	0.85

is why the initialization scale has a wide margin around the paper's value. The learning rate is the setting that decides whether training stays in the regime described by Theorem 1.

Input distribution. The paper's p_0 is a stationary Gaussian field with the power spectrum of the training pool (Appendix E). To test the sensitivity to this choice, we repeated the $N_{\text{train}} = 100$ training on the same MHD suite with an input that carries no power-spectrum information, white noise of the same variance, which is also stationary, so Corollary 1 still applies; each input was run twice, for 1000 and 1200 iterations. Training is stable with either input and the sample variability is unchanged: relative to held-out fields, the per-pixel diversity is 1.02 with the matched input and 1.06 with white noise (held-out halves against each other: 1.02), the mean pairwise distance between samples is 0.99 and 1.02 (1.02), and the per-coefficient scatter ratio is 0.66 and 0.67, scale by scale. The results degrade measurably, however, in the highest-order statistic we measure: the relative error in the number of peaks above 1, 2 and 3σ of the held-out pixel distribution is 1.84 and 3.56 with white noise, against 0.43 in both spectrum-matched runs. The remaining held-out statistics are less sensitive on this small problem: the power-spectrum error is smaller with white noise (0.12 against 0.18–0.19), as is the pixel-marginal KS distance (0.08–0.12 against 0.13–0.14), and the mean-feature error differs by less than the run-to-run variation (0.18 against 0.13 in one run, 0.13 against 0.17 in the other). Matching the second-order structure of the target lets the generator spend its capacity on the higher-order structure rather than on building the correlations itself, which is where the white-noise input falls short, and we expect the gap to widen for steeper spectra, larger fields or shorter training. We therefore recommend the spectrum-matched input used throughout the paper.

H Wavelet phase harmonic summary statistics

We use wavelet phase harmonic (WPH) coefficients [23, 35] as the feature map ϕ in every experiment, computed by the `pywph` library with periodic boundary conditions (`pbc = True`).

WPH coefficients are constructed from the wavelet transform $\{W_{j,\ell}f\}$ of the field f at J dyadic scales and L orientations, augmented with *phase harmonics* $W_{j,\ell}f \mapsto |W_{j,\ell}f|\, e^{ip \arg W_{j,\ell}f}$ for an integer-valued exponent set p, and form translation-invariant second-order cross-moments across nearby scales (separated by up to Δj octaves), orientations ($\Delta\ell$ rotations), and spatial translations (Δn pixel shifts); see [23] for the full vocabulary. This descriptor captures phase coupling that pure power-spectrum or wavelet-scattering moments do not represent, including filamentary structure, ridges, and small-scale alignment.

Per-benchmark configuration. For both stationary benchmarks of Section 5 we use the same $J = 5$-level Morlet wavelet bank with $L = 4$ orientations and the `pywph` default scale and orientation separations $\Delta j = J - 1 = 4$ and $\Delta\ell = 2$, computed at the native field resolution $M = 128$. The two benchmarks differ only in the spatial-shift order Δn used in the cross-moments and in the set of phase-harmonic exponents (Table 6).

For the **cosmological cosmic web (Quijote)** target we set $\Delta n = 0$ (cross-moments at the same position only) and pass an empty phase-harmonic exponent list (`p_list` $= \emptyset$) to `pywph`'s `load_model`, retaining the wavelet-scattering-like subset of the WPH family. This choice is sufficient to capture the second-order non-Gaussian phase structure of the Quijote slices while keeping the coefficient vector compact.

For the **MHD turbulent fluid simulation** target we set $\Delta n = 1$ (adding a nearest-neighbour spatial shift to each cross-moment) and use `pywph`'s default phase-harmonic exponent list, yielding a richer WPH descriptor than the cosmic-web setting; we found this expanded coefficient vector to be necessary in order to match ϕ^* in the single-snapshot regime.

In both cases we use the library's automatic per-coefficient normalization (`norm = 'auto'`), which rescales each coefficient by its empirical magnitude on the first call so that the contribution of every scale/orientation/exponent triple to the macrocanonical loss is comparable. The first invocation of the operator on the training data primes this normalization cache; all subsequent calls during training reuse it.

Table 6: WPH summary-statistic configuration per benchmark; symbols match [23].

Benchmark	J	L	Δj	$\Delta \ell$	Δn	p-list
Cosmological cosmic web (Quijote)	5	4	4	2	0	$\emptyset$
MHD turbulent fluid simulation	5	4	4	2	1	`pywph` default

I Per-coefficient WPH distribution test

We back up the claim of Section 5 with a visual and a quantitative comparison. After training the macrocanonical generator on 100 Quijote cosmic-web slices against the empirical mean WPH vector ϕ^* alone, so the generator never sees per-realization fluctuations during training, we draw 100 generator samples and compare the across-sample distribution of every WPH coefficient to that of 100 held-out Quijote slices the network has never seen. Figure 4 shows the comparison for four randomly chosen coefficients: the generator's per-realization distribution closely matches that of the held-out simulator pool in both mean and width, even though only the mean was supplied as a training target. A complementary quantitative test runs a two-sample Kolmogorov-Smirnov (KS) test independently for every WPH coefficient; averaged over the full coefficient vector this yields a mean p-value of $\langle p \rangle = 0.52$, and the empirical distribution of p-values across coefficients is consistent with the uniform distribution on $[0, 1]$ expected under the null that both samples are drawn from the same underlying distribution. This empirically confirms the entropy-preservation analysis of Section 4: scatter that is never directly supervised is nonetheless inherited from the input distribution and preserved by the residual identity-initialized generator. Microcanonical synthesis [4, 35], by construction, produces $\phi(x_i) = \phi^*$ for every sample and would therefore collapse each of the panels in Figure 4 to a Dirac mass at ϕ^*, eliminating the per-feature variance that downstream simulation-based inference (Section 5) needs in order to learn a non-degenerate posterior.

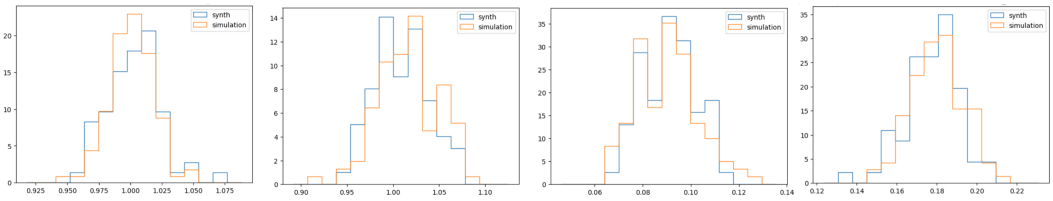

Figure 4: Across-sample distributions of four randomly chosen WPH coefficients from 100 macrocanonical generator outputs (*blue*) and 100 held-out Quijote simulator slices (*orange*). Microcanonical synthesis [4, 35] would instead collapse each panel to a Dirac mass at ϕ^*.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction state four claims that match the rest of the paper: (i) macrocanonical generators produce samples whose feature statistics match a target process in expectation while preserving sample-to-sample fluctuations (Section 2.2); (ii) gradient-descent training induces preconditioned gradient-descent dynamics in sample space that inherit the symmetry-preservation properties of microcanonical gradient descent (Theorem 1 and Corollary 2 in Section 3); (iii) residual identity-initialized parameterizations with small learning rates preserve entropy locally and mitigate mode collapse (Section 4 and the ablation of Section 5); and (iv) the resulting generators recover realistic samples from extremely small training sets and are orders of magnitude faster than both the reference simulator and microcanonical gradient descent (Sections 5–5). The scope is restricted to stationary, periodic targets, as stated in both the abstract and Section 5.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: A dedicated "Limitations" section (Section 6) discusses three limitations: (i) the dependence of generated samples on the choice of feature statistics; (ii) the restriction to stationary processes with periodic boundary conditions, where translation invariance provides the symmetry that enables estimation from few realizations, together with a proposed direction (pretraining on stationary processes followed by fine-tuning on non-stationary ones) for relaxing this assumption; and (iii) the absence of a general theoretical guarantee of convergence to a maximum-entropy distribution under the imposed constraints, or of avoiding collapse (Section 4).

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.

- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: All theoretical results are explicitly numbered and cross-referenced. Theorem 1 (Section 3) lists its four assumptions inline in the statement, with a proof sketch in the main text and a complete derivation in Appendix A. Proposition 1 states its two assumptions (group-invariance of p_0 and $\mathcal{G}$-equivariance of F_θ) and is proved in Section 3. Corollary 1 (stationarity) and Corollary 2 (symmetry inheritance from microcanonical synthesis) are derived from these results; Corollary 1 is proved in the main text and Corollary 2 in Appendix B.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Section 5 summarizes the experimental setup for both targets, and Appendices E and H provide full details: residual generator parameterizations (4)–(5) instantiated by a four-stage circular-padded U-Net (channel widths $64 \to 1024$, 3×3 convolutions, batch normalization, ReLU activations, circular padding throughout); convolutional weights initialized from $\mathcal{N}(0, 10^{-4})$ so that $F_{\theta_0}(x) = x$; Adam with base learning rate 10^{-4}, gradient-norm clipping at 1.0, optional linear warmup, and cosine annealing over the prescribed number of iterations; activation checkpointing to enable batch size 200 at 128×128 on a single GPU; a stationary Gaussian input distribution generated by FFT from the radially averaged power spectrum of the training data; and the per-benchmark wavelet phase harmonic configuration of Table 6 as passed to `pywph`. Appendix C documents the slice extraction and preprocessing pipeline used to build the cosmological dataset from the publicly released Quijote suite. Appendix F specifies the publicly released MHD suite, data splits, metrics and training settings of every method in the training-set-size sweep and baseline comparison, and Appendix G the settings of the learning-rate, initialization-scale and input-distribution sweeps.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

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

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

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

   Answer: [Yes]

   Justification: A reference implementation of the macrocanonical generator (the periodic residual U-Net, the WPH training loop, and training and sampling scripts for both benchmarks of Section 5) is released under the MIT license at `https://github.com/NiallJeffrey/MacrocanonicalGeneratorNetwork`. The simulation-based inference pipeline of Section 5 and the comparisons of Appendices F and G are not part of this release; together with the released generator code, the settings given in Appendices D, F and G (data splits, baselines, iteration counts, learning rates and metrics) are what is needed to reproduce them. The cosmological experiments use the publicly released Quijote N-body suite [32], accessible from `https://quijote-simulations.readthedocs.io` under its stated terms of use, and the controlled comparisons of Appendices F and G use the publicly released `MHD_64` dataset of The Well [26]; no new datasets are introduced. In addition, Section 5 and Appendices E, H, and C provide the full architecture, weight initialization, optimizer schedule, WPH configuration, and Quijote slice-extraction pipeline needed to reimplement the method.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).

- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Section 5 states the high-level training setup (residual U-Net generator initialized at the identity, wavelet phase harmonic feature map, stationary Gaussian input distribution, Adam at $\eta = 10^{-4}$ with gradient-norm clipping at 1.0). Appendix E gives the full architecture (channel widths, convolution kernel size, normalization, activations, circular padding, weight initialization, batch size, activation checkpointing, two-GPU placement of the generator and the WPH operator), and the optimizer schedule (linear warmup followed by cosine annealing). Appendix H documents the per-benchmark WPH hyperparameters $(J, L, \Delta j, \Delta \ell, \Delta n, \text{p-list})$ in Table 6, and Appendix C documents the data-preprocessing pipeline (log-density transform, mean-pool downsampling to 128×128, pool-level normalization fixed across all values of N_{train}). Appendix D gives the cosmology splits, early-stopping protocol and learning rates of the simulation-based inference experiment, and Appendices F and G give the data splits, iteration counts, batch sizes and samplers of the controlled comparisons and sweeps.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: The central distributional claim of the paper—that the generator reproduces the per-realization scatter of the target process and not merely its mean feature value—is supported by an explicit statistical test in Section 5: for each WPH coefficient we report a two-sample Kolmogorov–Smirnov test between 100 generator samples and 100 held-out Quijote slices, and the resulting p-values are consistent with the uniform distribution expected under the null (Figure 4). The simulation-based inference curves of Figure 2 show the standard deviation over three training seeds (Appendix D). The inductive-bias ablation of Section 5 reports qualitative samples for each variant on each benchmark. The comparisons and sweeps of Appendices F and G are single training runs scored against 100 held-out fields, with held-out fields scored against themselves reported alongside as the reference level. Per-coefficient error bars are not added to the WPH-distribution figure because the figure already shows the full empirical distribution for each coefficient, which is more informative than a summary error bar.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The main experiments were run on NVIDIA A100 40 GB GPUs. Each macrocanonical generator was typically trained using 2 GPUs in data-parallel mode. The architecture, hyperparameters, and per-experiment wall-clock training times are reported in Section 5 and Appendix D, and the throughput comparison against the reference simulator is reported in the surrogate-speed subsection. The 64×64 comparisons and sweeps of Appendices F and G are small (at most $20{,}000$ iterations per run) and were run on a single A100 or a laptop GPU; the per-sample timings in Table 3 were measured on one A100.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The authors have reviewed the NeurIPS Code of Ethics and confirm that the research conforms with it. The work develops methodology for neural surrogates of physical simulators and is evaluated exclusively on synthetic data and publicly available scientific simulations (e.g. the Quijote N-body suite). No human subjects, personally identifiable information, or sensitive data are involved at any stage of the research.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Macrocanonical generator networks are designed as fast surrogates for expensive scientific simulators. The intended positive impact is to reduce the computational cost (and therefore the energy cost) of many-query scientific workflows such as simulation-based inference, uncertainty quantification, and optimal experimental design, with applications in cosmology, fluid dynamics, climate-related modelling, and other domains where the underlying physical processes are stationary or approximately stationary. We are not aware of direct negative societal impacts: the surrogate replaces a numerical solver of a known physical process and does not produce content involving people, generate disinformation, or interact with high-risk decisions in deployment as proposed. A more general risk that applies to any neural surrogate of a physical simulator is over-reliance on a surrogate whose accuracy has not been validated for the task at hand, which is mitigated in our framework by quantitative comparison of generated samples to reference simulator statistics on every application.

    Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: This paper presents a methodology and does not release any pretrained generators or datasets. Even setting that aside, the method produces neural surrogates of stationary physical fields (e.g. cosmological density fields, idealised fluid simulations) trained on synthetic or publicly released scientific simulation data: such surrogates cannot be used to generate content depicting real people or to produce dual-use material, and we therefore consider that they do not pose a high risk of misuse that would require dedicated safeguards.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: The cosmological experiments use the publicly released Quijote N-body simulation suite [32, `https://quijote-simulations.readthedocs.io`], which is distributed for non-commercial academic use; the original Quijote paper is cited at the relevant points in the text and the dataset version used is recorded in the experimental setup. We use the dataset under its stated terms of use and credit the Quijote collaboration accordingly. The MHD turbulence snapshot of Section 5 is a simulated dust-emission map from the magnetohydrodynamic simulation analysed by Regaldo-Saint Blancard et al. [30], which is cited in Appendix E and in the released code. The MHD suite of Appendix F is the `MHD_64` dataset of The Well [26], distributed under CC-BY-4.0 according to its datasheet, which we cite at the point of use.

Guidelines:

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

13. **New assets**

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

Answer: [Yes]

Justification: The only new asset is the reference implementation of the macrocanonical generator at `https://github.com/NiallJeffrey/MacrocanonicalGeneratorNetwork`, released under the MIT license with a README, environment requirements, and training and sampling scripts for both benchmarks of Section 5. No new datasets or pretrained model checkpoints are released.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.

- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or any form of research with human subjects. All experiments are conducted on synthetic stationary fields and on publicly released N-body simulation data (Section 5, Appendix C).

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve human subjects, so no Institutional Review Board approval (or equivalent) was required.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

Justification: Large language models are not part of the core methodology of this work, which concerns neural surrogates of stationary physical processes trained against wavelet-based feature statistics. Any use of LLMs by the authors was limited to writing assistance (editing, formatting, and basic code suggestions) and does not impact the scientific content, methodology, or originality of the research, so declaration is not required under the NeurIPS LLM policy.

Guidelines:

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