entropy

MDPI

Article

Unambiguous Models and Machine Learning Strategies for Anomalous Extreme Events in Turbulent Dynamical System

Di Qi

Department of Mathematics, Purdue University, 150 North University Street, West Lafayette, IN 47907, USA; qidi@purdue.edu

Abstract: Data-driven modeling methods are studied for turbulent dynamical systems with extreme events under an unambiguous model framework. New neural network architectures are proposed to effectively learn the key dynamical mechanisms including the multiscale coupling and strong instability, and gain robust skill for long-time prediction resistive to the accumulated model errors from the data-driven approximation. The machine learning model overcomes the inherent limitations in traditional long short-time memory networks by exploiting a conditional Gaussian structure informed of the essential physical dynamics. The model performance is demonstrated under a prototype model from idealized geophysical flow and passive tracers, which exhibits analytical solutions with representative statistical features. Many attractive properties are found in the trained model in recovering the hidden dynamics using a limited dataset and sparse observation time, showing uniformly high skill with persistent numerical stability in predicting both the trajectory and statistical solutions among different statistical regimes away from the training regime. The model framework is promising to be applied to a wider class of turbulent systems with complex structures.

Keywords: turbulent systems; machine learning; multiscale modeling; long short-term memory

Citation: Qi, D. Unambiguous Models and Machine Learning Strategies for Anomalous Extreme Events in Turbulent Dynamical System. *Entropy* **2024**, *26*, 522. https://doi.org/10.3390/e26060522

Academic Editor: Geert Verdoolaege

Received: 1 May 2024
Revised: 3 June 2024
Accepted: 14 June 2024
Published: 17 June 2024

1. Introduction

Extreme events and the related anomalous statistics are fascinating phenomena universally observed in a wide class of natural and engineering systems [1–5]. An active, contemporary topic with a grand challenge is understanding, predicting, and controlling such events using qualitative and quantitative models [6–10]. Dynamical systems with extreme events are often characterized by strong internal instabilities and the competing effects of coherent large-scale structures and multiple interacting small-scale processes [11–13]. The accurate quantification of such features requires solving complex nonlinear equations among different parameter regimes to draw a complete picture of the statistical solution profile. Direct strategies by explicitly resolving all the scales with many repeated evaluations become inefficient and often impractical due to the very high computational overload [14,15]. Effective modeling and parameterization methods are still needed to capture the key dynamical features with computational efficiency and robustness to the noise errors amplified by inherent instabilities.

Data-driven modeling using machine learning ideas [16–19] has become one appealing approach to learn the unresolved physical processes given sufficient data covering complete solution regimes. Such data-driven strategies have shown potential in recovering unresolved subscale dynamics which are difficult to derive via first principles [20–22], or suffer high computational cost in direct approaches [23–27]. The increasing amount of observational data further helps the development of various data-driven models to advance the understanding of the underlying physical mechanisms and thus to provide fast and accurate solvers [28–32]. In the case of learning model dynamics showing extreme events and anomalous statistics, however, the available data for training are often restricted with incomplete observations (such as a limited dynamical regime and sparse measurements)

and are polluted with various model errors (such as amplified noise and uncertainties from the model instability, as well as the imperfect model approximation). The challenge remains to find a universally applicable model framework with uniform prediction skill among different statistical regimes that are beyond the limited training dataset. Large model uncertainty and exponentially growing model error often lead to the breakdown of long-time prediction typical for complex turbulent systems without a complete physical understanding. An effective model of turbulent dynamical systems requires computational stability against noises and perturbations.

In this paper, we aim to investigate useful machine learning techniques to recover the unresolved components of complex physical systems with coupled multiscale processes. In the development of new machine learning strategies, it is useful to start with simplified prototype models, where the key dynamical structures of interests can be identified for a thorough understanding. We propose a group of unambiguous models for both the understanding of key dynamical structures in generating the representative extreme phenomena, and also the development of effective machine learning strategies based on explicit physical structures. The unambiguous models are drawn from geophysical turbulence [33–35], and accept analytically tractable conditional Gaussian structures incorporating essential processes of multiscale flow interaction and the turbulent transport of passive tracers. The system generates representative anisotropic flows with topographic blocking, qualitatively resembling those observed in the midlatitude ocean and atmosphere [36–38], and passive tracer fields exhibiting strong extreme events with skewed or fat-tailed probability distributions comparable to laboratory experiments [6,29,37,39]. In addition, the model fits into the general conditional Gaussian framework [32,40–42], which provides explicit analytic formulas by expressing the small-scale variables as a group of Gaussian processes depending on the realizations of the observed large-scale process. The explicit analytical formulas draw a detailed characterization of the trajectory solution structures as well as the anomalous statistics. Based on the explicit solutions of the detailed large–small-scale interaction mechanism, effective machine learning methods are developed, only replacing the unresolved small-scale processes by data-driven models. Such simple but comprehensive equations are shown to serve as a group of unambiguous test models for the central problems in the development of data-driven strategies with various datasets and model architectures.

Contributions of This Work

We introduce novel modeling strategies that are able to learn the hidden dynamical processes coupled with multiple temporal and spatial scales. The neural network model can be trained under a single set of data with sparse time measurements, and the trained model is capable of predicting different types of extreme events and distinct statistical features among various dynamical regimes where data are unavailable. By construction, the new neural network architectures provide an accurate higher-order approximation of the original dynamics informed of its essential physical structures. The coupled multiscale processes are modeled efficiently by exploiting the conditional Gaussian structure and decomposing the full model into smaller subsystems that are easier to be learned. In addition, the model stability for long-time iterations is effectively improved through training with a feasible loss function considering multistep outputs, including model errors.

In building the neural network model, the original long short-term memory (LSTM) network [43,44] is improved by adding detailed inner connections to track the correlated long-time history in the data from turbulent signals. The new neural networks are applied for multiple small scales for efficient modeling, and the outputs for different scales are combined in the explicit large-scale mean flow equation to introduce physics-based updates to the learning process. We improve the idea in [31,32] to introduce the loss function for the optimization procedure by combining the use of a new relative entropy distance [45,46] and the standard mean square error. The calibration of model approximation errors is made to focus on the dominant shapes of the turbulent signals instead of an exhausting fitting of the unnecessary pointwise turbulent errors. With the combined contributions of the new

designs of model architectures and loss functions, the proposed neural network model overcomes the inherent difficulties of early divergence and large training errors common among the traditional LSTM networks [17,47].

The neural network model is then tested on the proposed unambiguous model from geophysical turbulence with the large mean flow interacting with two small-scale modes. The model provides the simplest setup restricted to a two-mode interaction, while still maintaining a variety of dynamical regimes displaying different types of extreme events for testing the skill of the neural network. The neural network model focusing on the small-scale processes is trained in one statistical regime from the available partial data. Then, the trained model can be applied universally for various scenarios with distinct non-Gaussian statistics. By applying the model to different datasets in the prediction stage, we show the predictive capability in the trained model to recover the key dynamics from incomplete data and limited information. The model also allows a sparse dataset with longer time measurement steps, showing stable performance for long-time prediction. The model with unresolved processes can be further generalized to a wider class of complex turbulent systems [1,48,49] to construct computationally efficient, reduced models with nonlinear high-order feedback [26,27].

In the rest part of the paper, the unambiguous model framework with explicit solutions and representative statistical regimes is introduced in Section 2. The general machine learning strategy to learn the complex dynamical processes is constructed in Section 3. Then, the neural network model is combined with the explicit physical structures in the dynamical system in Section 4 to capture extreme events and the statistical features. Numerical tests are carried out in a two-mode topographic model in Section 5 as an illustration for the scope of skill of the strategy. A summary with discussions for future research directions is given in Section 6.

2. An Unambiguous Model Framework for the Investigation of Extreme Events

We first propose a group of prototype models with tractable mathematical structures to serve as a clean testbed for the investigation of the various distinctive phenomena found in natural systems. The models are constructed by including the key features in realistic turbulent systems, such as a wide variety of extreme events and anomalous statistics with fat-tailed or skewed probability density functions (PDFs). The prior information of the dynamical system is then exploited for guidelines to design new neural network architectures in the next sections.

2.1. General Formulation of the Unambiguous Mathematical Models

A general mathematical framework for a wide group of systems can be introduced in the following abstract form for the multiscale states $\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2) \in \mathbb{R}^{N_1+N_2}$

$$\begin{aligned} \mathrm{d}\mathbf{u}_1 =& [A_0(t,\mathbf{u}_1) + A_1(t,\mathbf{u}_1)\mathbf{u}_2]\mathrm{d}t + \Sigma_1(t,\mathbf{u}_1)\mathrm{d}\mathbf{W}_1, \\ \mathrm{d}\mathbf{u}_2 =& [B_0(t,\mathbf{u}_1) + B_1(t,\mathbf{u}_1)\mathbf{u}_2]\mathrm{d}t + \Sigma_2(t,\mathbf{u}_1)\mathrm{d}\mathbf{W}_2. \end{aligned} \tag{1}$$

Usually, $\mathbf{u}_1 \in \mathbb{R}^{N_1}$ can be viewed as the observed slow process and $\mathbf{u}_2 \in \mathbb{R}^{N_2}$ as the unobserved fast states in a much larger phase space $N_2 \gg N_1$. A conditional Gaussian framework [42,50] is developed based on the general formulation (1), where $\mathbf{u}_2$ can be expressed as a Gaussian process given the complete history of $\mathbf{u}_1(s)\,|_{s\le t}$

$$p(\mathbf{u}_2(t) \mid \mathbf{u}_1(\cdot)) = \mathcal{N}(\bar{\mathbf{u}}_2(t;\mathbf{u}_1(\cdot)), R_2(t;\mathbf{u}_1(\cdot))),$$

where $\bar{\mathbf{u}}_2, R_2$ are the conditional mean and covariance matrix depending on the realization of $\mathbf{u}_1$. This implies that the final probability distribution for the process $\mathbf{u}_2$ can be expressed as a mixture conditional on different realizations of the 'observables' $\mathbf{u}_1$. Still, the full

probability distribution of the system (1) could be highly non-Gaussian as a combination of all the probability realizations from the above conditional Gaussian structure

$$p(\mathbf{u}_2, t) \sim \int p(\mathbf{u}_2(t) \mid \mathbf{u}_1) d\mu(\mathbf{u}_1),$$

where $\mu(\mathbf{u}_1)$ is the probability measure for the entire observed process $\mathbf{u}_1(s), s \le t$. As a further comment, many realistic turbulent systems such as those found in climate forecast may not exactly follow the conditional Gaussian structure (1). Therefore, usually, additional approximations are needed with the introduction of imperfect model errors.

Topographic Barotropic Model with Large-Scale Mean Flow Interaction and Strong Small-Scale Feedbacks

We focus on a special group of the general model framework (1) with reference to geophysical turbulent flows. The topographic barotropic system [33,34] models the complex interactions of a large-scale mean flow U and small-scale vortical fluctuations q in quasi-geostrophic turbulence

$$\frac{\partial q}{\partial t} + \mathbf{v} \cdot \nabla q = \mathcal{D}(\Delta)\psi + F_q, \tag{2a}$$

$$\frac{dU}{dt} + \fint \frac{\partial h}{\partial x}\psi' = -d_U U + F_U. \tag{2b}$$

The topographic model is defined on the two-dimensional doubly periodic plane $\mathbf{x} \in D = [-\pi, \pi]^2$ for simplicity, with the potential vorticity q, stream function ψ, and flow velocity $\mathbf{v}$ defined as

$$q = \nabla^2\psi' + h + \beta y, \ \psi = -U(t)y + \psi', \ \mathbf{v} = \nabla^\perp\psi = \left(U - \partial_y\psi', \partial_x\psi'\right). \tag{3}$$

Above, the small-scale stream function ψ' is separated from other large-scale terms. There exists a multiscale coupling between the small-scale fluctuations (2a) and the large-scale uniform mean flow (2b) through the domain-averaged quantity $\fint \frac{\partial h}{\partial x}\psi' \equiv \frac{1}{|D|}\int_D \frac{\partial h}{\partial x}\psi'$ with $|D|$, the computational domain area. The model (2) combines several crucial features in geophysical turbulence [1,35]: the effects of topography (h), rotation (β), external forcing (F_q, F_U), frictions $\mathcal{D}(\Delta)$ for the fluctuation states (for example, Ekman drag r and higher-order dissipation $\nu\Delta^2$) and linear damping d_U acting on the large-scale mean flow U. It is easy to check that the model fits into the general framework (1) by setting $\mathbf{u}_1 = U$ as the large-scale mean flow and $\mathbf{u}_2 = q$ for all the small-scale processes.

Using the flow solution of (2), we can introduce an additional equation modeling the turbulent transport of passive tracers through the advection and diffusion of the passive tracer density field $T(\mathbf{x}, t)$

$$\frac{\partial T}{\partial t} + \mathbf{v} \cdot \nabla T = -d_T T + \kappa\Delta T, \tag{4}$$

where the advection flow $\mathbf{v} = \nabla^\perp\psi$ is provided by the velocity solution in (3), and the tracer field is subject to damping and diffusion effects due to parameters d_T and κ on the right-hand side. A wide variety of properties are found and analyzed under this tracer framework (4) for both theories and applications in turbulent transport and diffusion [6,29,51]. Again, the tracer Equation (4) can be also categorized into the general framework (1) with $\mathbf{u}_2 = T$ and no direct feedback to the flow state $\mathbf{u}_1 = \mathbf{v}$. Instabilities and uncertainties are introduced through the multiscale interactions [35,52] in the above flow system (2) as well as the passive tracer field (4). The prediction of both the large and small processes from the partially observed data and unknown dynamics forms a general challenge for the accuracy and stability.

Despite the simplicity, the stringent paradigm models for multiscale coupled flow (2) and passive tracer (4) fields exhibit complex statistics characterizing a number of crucial realistic phenomena, such as the atmospheric blocking, topographic instability, and nonlinear energy transfer through scales [1,33]. The tractable dynamical structures in the simplified models enable a series of detailed analyses for the mathematical understanding of various physical processes [24,35] and the construction of comprehensive computational strategies [29,34,53].

2.2. Analytical Solutions from the Conditional Gaussian Models

Still, the model (2a) couples all the small-scale fluctuation modes through the nonlinear advection term on the left-hand side. This will lead to very complex dynamical structures for the analysis. Here, in order to find more analytically tractable solutions, we propose further simplification to the original system so that we are able to focus on the most important mean–fluctuation interactions in determining the final flow structure.

Starting from the topographic barotropic model (2) and the corresponding passive tracer Equation (4) on the periodic domain D, we project the states of the flow velocity field $v = \partial_x \psi$ and the tracer density field T on the Fourier spectral space by

$$v(\mathbf{x}, t) = \sum_k \hat{v}_k(t) e^{ik\mathbf{l}\cdot\mathbf{x}}, \; T(\mathbf{x}, t) = \sum_k \hat{T}_k(t) e^{ik\mathbf{l}\cdot\mathbf{x}} + \alpha y. \tag{5}$$

Above, the *layered topography* [29] is applied along one characteristic wavenumber direction $\mathbf{l}$ with $|\mathbf{l}| = 1$ (for simplicity, we take $l_x = 1, l_y = 0$ in the following analytical results without loss of generality). In addition, a background mean gradient profile αy along the y direction is assumed for the tracer density field. The layered topography eliminates the nonlinear interactions between the fluctuation modes, thus enabling us to focus on the coupling effect between the large and small scales. Then, we aim to find the trajectory and statistical solutions of the following coupled flow system under the spectral representation.

$$\frac{\mathrm{d}\hat{v}_k}{\mathrm{d}t} - i\Big(k^{-1}\beta - kU\Big)\hat{v}_k + \hat{h}_k U = -d_k \hat{v}_k + \sigma_{v,k}\dot{W}_k, \tag{6a}$$

$$\frac{\mathrm{d}U}{\mathrm{d}t} - \sum_k h_k^* \hat{v}_k = -d_U U + \sigma_U \dot{W}_0. \tag{6b}$$

Above, through orthogonal projection (5), the linear terms are decoupled into each spectral mode $\hat{v}_k(t)$. The integration representing topographic stress on the left-hand side of (2b) becomes the summation over all the spectral modes from the inner product. In particular, the nonlinear coupling between small-scale modes in (2a) vanishes due to the assumed layered topography along one wavenumber direction $\mathbf{l}$. The additional unresolved effects are summarized in the white noise as $F_{q,k} = \sigma_{v,k}\dot{W}_k$ and $F_U = \sigma_U \dot{W}_0$. Accordingly, the associated passive tracer equation is given by

$$\frac{\mathrm{d}\hat{T}_k}{\mathrm{d}t} = (-\gamma_{T,k} - ikU(t))\hat{T}_k - \alpha \hat{v}_k, \tag{6c}$$

with $\gamma_{T,k} = d_T + \kappa k^2$ and white noise amplitudes $\sigma_{v,k}, \sigma_U$ defined for small and large scales. The shear flow modes $\hat{v}_k$ serve as a forcing on the passive tracer mode $\hat{T}_k$ induced by the mean gradient α. The detailed derivation of the equations and their properties are discussed in [6,29,34] with many applications.

In particular, we refer to the large-scale mean flow U as the 'observed state' that is measured at a time frequency Δt (note that this is usually much longer than the admissible integration time step in the direct numerical scheme); the small-scale velocity modes $\hat{v}_k$ with important feedback to the mean flow equation are treated as the unresolved states. In the following sections, the neural network model is designed to predict all the unresolved small-scale processes $\hat{v}_k, \hat{T}_k$ without pre-knowledge of the original dynamical model (6a).

And together with the neural network output, Equation (6b) can be used to update the solution of the large-scale U with different noise levels σ_U.

2.2.1. Explicit Solutions to the Topographic Model with Damping and Stochastic Forcing

The conditional Gaussian structure in the spectral flow and tracer model (6) enables to derive closed analytic formulas for discovering the typical properties in the topographic flow field and passive tracer solutions. It shows that the statistics in the velocity and tracer modes $\hat{v}_k, \hat{T}_k$ in (5) can be determined by the statistics in U. These analytic formulas help to provide an improved understanding of the rich physical phenomena observed in both the flow and tracer fields, under which the machine learning models can be constructed. Below, we list the main conclusions, where the detailed derivations can be found in Appendix A.

First, we can find the long-time steady-state solution when the initial state becomes irrelevant. Following the decoupled dynamics from the diagonal coefficients of each wavenumber (6a), the conditional trajectory solution of the steady-state small-scale mode (that is, for $t \gg 1$ large enough to 'forget' its initial information) can be written as

$$\hat{v}_k(t; U(\cdot)) = -\int_0^t e^{-\gamma_{v,k}(t-s)-ikU[s,t]}\Big[\hat{h}_k U(s)ds + \sigma_{v,k}dW_k(s)\Big], \tag{7}$$

with the coefficients $\gamma_{v,k} = d_k - ik^{-1}\beta$ and $U[s,t] \equiv \int_s^t U(\tau)d\tau$ depending on one realization of the zonal mean solution of $U(s)$ during the entire time period $0 < s \le t$. Similarly with the above formula for the shear flow solution $\hat{v}_k$, the corresponding solution for the passive tracer Equation (6c) can be solved based on the advection flow

$$\hat{T}_k(t; U(\cdot)) = \frac{\alpha}{\gamma_{R,k}}\int_0^t e^{-\gamma_{T,k}(t-s)-ikU[s,t]}\Big[e^{\gamma_{R,k}(t-s)} - 1\Big]\Big[\hat{h}_k U(s)ds + \sigma_{v,k}dW_k(s)\Big], \tag{8}$$

with the effective damping and dispersion relation $\gamma_{R,k} = \gamma_{T,k} - \gamma_{v,k}$. Note that the above state solutions at time t are dependent on the entire history of the large-scale mean flow U. The above explicit Formulas (7) and (8) for the trajectory solutions of flow and tracer modes imply that we can recover the small-scale flow and tracer trajectories based on the information from the large-scale mean flow information. This provides an instructive guideline for the solution structures with the small- and large-scale interaction mechanisms.

2.2.2. Statistical Solutions for the Mean and Variance

Next, we compute the steady-state statistical solutions of the flow and tracer modes based on the conditional trajectory Formulas (7) and (8). We assume that the large-scale mean flow U reaches a statistical steady state with dominant leading order moments. Therefore, the mean and variance of the small-scale modes can be written in terms of the equilibrium mean $\bar{U}$, variance r_U, and the autocorrelation function $\mathcal{R}_U$ of the large-scale flow.

Using the leading order expansion of the moments, the mean state for the shear flow modes can be computed by

$$\begin{aligned}\bar{v}_k \equiv \langle \hat{v}_k \rangle &= -\hat{h}_k \int_0^t e^{-\gamma_{v,k}(t-s)}\Big\langle e^{-ikU[s,t]}U(s)\Big\rangle ds \\ &= -\hat{h}_k \int_0^t e^{i\left(k^{-1}\beta - k\bar{U}\right)s} e^{-d_k s - k^2 r_U J_U(s)}(\bar{U} - ikr_U I_U(s))ds.\end{aligned} \tag{9}$$

Above, we denote $\bar{U} = \langle U \rangle$, $r_U = \Big\langle (U-\bar{U})^2 \Big\rangle$, and the autocorrelation function $\mathcal{R}_U(\tau) = r_U^{-1}\langle U(\tau)U(0)\rangle$ (with $\langle\cdot\rangle$ represents the statistical expectation at equilibrium).

The time-dependent operators I_U, J_U are computed from the integration of the autocorrelation function $\mathcal{R}_U$

$$I_U(t) = \int_0^t \mathcal{R}_U(\tau)d\tau, \quad J_U(t) = \int_0^t (t-\tau)\mathcal{R}_U(\tau)d\tau.$$

Only the first two moments of the stochastic process U are used in the above computation of the statistical expectation $\left\langle e^{-ikU[s,t]}U(s)\right\rangle$. Accordingly, the tracer mean state can be also derived based on the statistics of the mean flow U in a similar fashion using the trajectory solution

$$\begin{aligned}\bar{T}_k \equiv \langle \hat{T}_k \rangle &= -\frac{\alpha \hat{h}_k}{\gamma_{R,k}} \int_0^t e^{-\gamma_{T,k}(t-s)} \left[e^{\gamma_{R,k}(t-s)} - 1\right] \left\langle U(s) e^{-ikU[s,t]} \right\rangle ds \\ &= -\frac{\alpha \hat{h}_k}{\gamma_{R,k}} \int_0^t e^{-\gamma_{T,k}s - k^2 r_U J_U(s)} e^{-ik\bar{U}s} (e^{\gamma_{R,k}s} - 1)(\bar{U} - ikr_U I_U(s))ds,\end{aligned} \tag{10}$$

with the new dispersion relation $\gamma_{R,k} = \gamma_{T,k} - \gamma_{v,k}$. Thus, we see that the tracer and flow means $\bar{v}_k$ and $\bar{T}_k$ are closely linked. The second-order moments of the flow and tracer modes can be computed by multiplying the corresponding states on both sides of (7) and (8) depending on the statistics and time correlations of the zonal mean flow statistics of U. Then, the flow velocity variance r_{v_k} for each mode becomes

$$r_{v_k} \equiv \left\langle |\hat{v}_k|^2 \right\rangle = \frac{1}{2d_k}\left[|\sigma_{v,k}|^2 + 2\left|\hat{h}_k\right|^2 \mathfrak{Re} \int_0^\infty e^{-\left(d_k + ik^{-1}\beta\right)\tau} \left\langle U(0)U(\tau)e^{-ikU[0,\tau]} \right\rangle d\tau \right]. \tag{11}$$

The tracer variance $r_{T_k} = \langle |\hat{T}_k|^2 \rangle$ can also be found similarly. The above expression for the variance is linked to the triad correlation of the large-scale steady-state flow $\left\langle U(0)U(\tau)e^{-ikU[0,\tau]} \right\rangle$. We can compute the expectation in terms of the statistics in the mean flow, that is, $\bar{U}, r_U$ and $\mathcal{R}_U$. The explicit expression can be found in Appendix A.

From the above explicit formulas for the flow and tracer statistics, we observe the already complicated structures in the leading statistics from coupling between large and small scales as well as the flow and tracer interaction. In particular, in order to resolve the mean and variance of the small-scale flow and tracer modes, it shows that detailed higher-order moments are required from the large-scale mean state U. This often demands huge amounts of data and expensive computational cost to achieve a desirable accuracy. On the other hand, the above mean and variance Formulas (9)–(11) with the conditional Gaussian structures imply that the essential leading-order statistical information among all small-scale processes can be recovered from the statistical measurements in the large-scale mean flow. The informed statistical solutions can offer crucial guidance for the construction of combined physics and data-driven models in the following section. Therefore, machine learning strategies will be designed to find the unresolved small-scale dynamics directly from data, while explicit physics equations will be used for capturing extreme features in the large-scale mean state. In particular, the trained machine learning model can greatly reduce the computational cost of directly running the expensive full model, and provide an efficient alternative way to predict the small-scale processes without the further requirement of intense data.

2.3. Different Statistical Regimes of Flow and Tracer Fields in the Two-Mode Model

Before the construction of data-driven models to learn the turbulent dynamics, we first illustrate the typical dynamical and statistical structures found in the flow and tracer solutions using direct numerical simulations. The above analytical Formulas (7) and (8) show that the flow and tracer models reach various dynamical regimes relying on mean flow statistics in U, while from the mean flow Equation (6b), the solution of the zonal mean flow U in turn is determined by the combined feedback from the small-scale fluctuation modes.

Here, we choose a prototype *two-mode topographic model* under the simplest setting using only two Fourier modes, $k, 2k$. Accordingly, a two-mode topography can be adopted as a combination of the two scales

$$h = H_1(\cos kx + \sin kx) + H_2(\cos 2kx + \sin 2kx)$$

The two-mode system includes five coupling modes $(U, \hat{v}_{\pm 1}, \hat{v}_{\pm 2})$ for the flow equations and $(\hat{T}_{\pm 1}, \hat{T}_{\pm 2})$ for the passive tracer state. This simple two-mode formulation keeps the central interaction mechanism between the mean flow and fluctuation modes, which is still able to create a wide range of remarkably different statistical regimes representing different kinds of extreme events. Therefore, we use this model as a basic test model to display the various distinctive statistical regimes, and then as a standard testbed for the design of neural network architectures in the following sections.

To illustrate different model statistics, our strategy is to modify the major driving effect from the topographic stress $H = H_1 = 2H_2$. The other model parameters are fixed as $\beta = 2$, $d_U = d_k = 0.0125$, $\sigma_U = \sigma_k = \frac{1}{2\sqrt{2}}$ for the flow equations, and $d_T = 0.1$, $\kappa = 0.001$ for the tracer equation. These parameters are picked according to [34,42] to simulate realistic climate scenarios. The typical trajectory solutions from direct numerical simulations are shown in Figure 1 with different topographic forcing strengths H. We observe the distinct dynamical structures under this simple two-mode model setup. With a weak topographic stress $H = 1$, the mean flow displays a slow varying time scale with intermittent extreme values on the negative side. Strong extreme events are triggered in both the small-scale flow and tracer states as the large-scale U reaches regimes of positive values. In contrast with a strong topographic stress $H = 10$, the mean flow U develops a fast oscillating time scale on top of the slow transiting packages of extreme events. The time scale difference appears more obvious between the flow and tracer fields. The tracer modes show a much slower time scale in comparison with the flow modes, and strong skewness with multiscale structures.

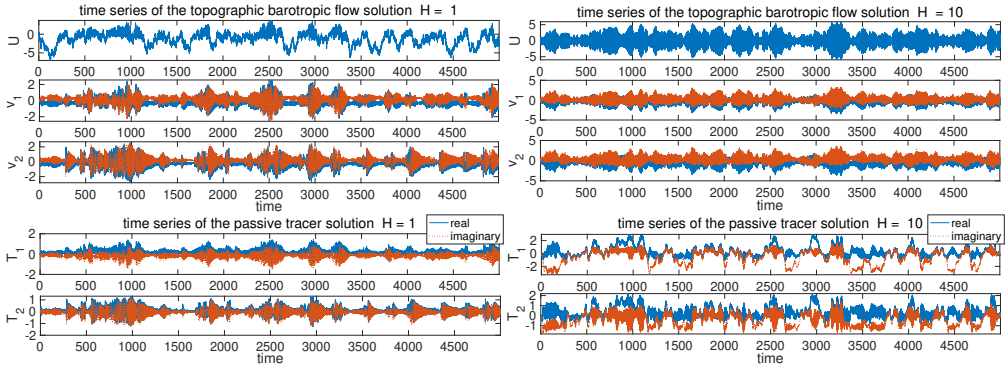
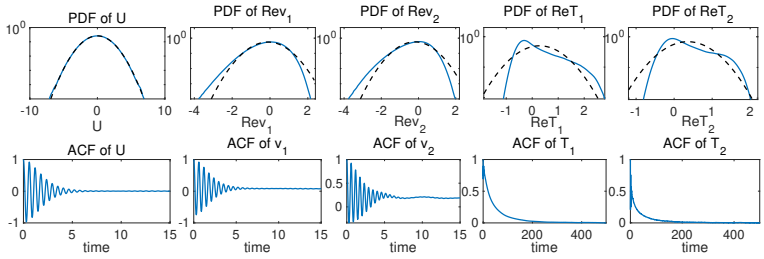


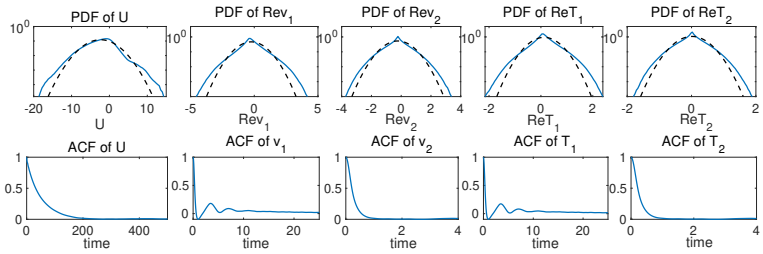
Figure 1. Time series of the flow and tracer trajectory solutions in the two parameter regimes with weak topographic stress $H = 1$ (**left**) and strong topographic stress $H = 10$ (**right**).

For a detailed comparison of the solution statistics, the probability density functions (PDFs) and autocorrelation functions (ACFs) are shown in Figure 2 for the two topographical regimes. In the strongly forced regime $H = 10$, the mean flow U displays near-Gaussian statistics, while both the flow v and tracer T modes generate highly skewed PDFs. The fat-tailed or skewed PDFs are generated due to the uncertainty in the mean flow field U interacting with the small-scale conditional Gaussian processes $\hat{v}_k$ or $\hat{T}_k$. In the autocorrelation functions, strong scale separation is also found with a fast mixing oscillating process in the flow states for both U and v, while the tracer modes for T have a much more slowly decaying mixing process. This can be also observed in the time series in Figure 1. In contrast in the weak topography case $H = 1$, the mean flow U develops fat-tailed non-Gaussian statistics in the PDF. The small-scale flow and tracer modes also display fat tails consistent with the time series. In the autocorrelation functions, the mean flow U has

a much slower mixing rate in comparison with the fast-mixing modes in both the flow and tracer. The strong scale separation makes it very difficult to learn the complete model dynamics purely from data. It requires the accurate modeling of all scales at the same time during the training of the data-driven models to correctly represent the dynamics and maintain a stable scheme.



(**a**) strong topographic stress $H = 10$



(**b**) weak topographic stress $H = 1$

Figure 2. PDFs and ACFs for flow and tracer states with strong (**upper**) and weak (**lower**) topographic stress H. The results for large-scale state U and the leading small-scale modes $\hat{v}_k$, $\hat{T}_k$ are compared. The Gaussian fits of the PDFs with the same mean and variance are plotted in dashed lines.

3. Neural Network Architecture for Correlated Dynamical Processes

In this section, we describe the general neural network architecture to learn the correlated dynamical processes in turbulent systems. Based on the model framework in Section 2, various statistical regimes can be found from the same dynamical model depending on different sets of data in the large-scale state. The main goal here is to construct an effective machine learning model to capture the complex dynamics in a uniform fashion among different statistical regimes. In the following, we propose several new structures to the basic long short-term memory (LSTM) network [43] for modeling dynamical updates with long measurement time interval Δt. A new set of loss functions based on relative entropy is also proposed to respect the turbulent nature of the extreme solution trajectories.

3.1. Architecture of the Neural Network Model

First, we provide a brief description of the main components in the neural network model designed to approximate a continuous dynamical system in the general form

$$\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}t} = f(\mathbf{x}, t), \tag{12}$$

where $\mathbf{y} \in \mathbb{R}^n$ is the target state to be predicted and $\mathbf{x} \in \mathbb{R}^d$ represents all the related input variables (explicit examples for the input and output data are shown next in Section 4 using the topographic model). f is the unknown dynamical map to be learned by the machine learning scheme from data. Adopting the general dynamical structures of the form (12), the neural network architecture is designed by (i) a residual network to approximate the increment at each time step; (ii) an LSTM chain to incorporate the correlated time sequence; and (iii) a multistage link inside each LSTM cell to capture the coupled dynamics in the model.

3.1.1. Residual Network to Capture the Dynamical Model Update

In the first place, for the machine learning approximation of the dynamical increment in (12), we adopt a residual network structure for the time increment of the unresolved dynamical functional

$$\mathbf{y}_t^M = \mathbf{y}_{t-1}^M + \Delta t f_m^M(\{\mathbf{x}_{t-m}, \cdots, \mathbf{x}_{t-1}\}). \tag{13}$$

Above, $f_m^M : \mathbb{R}^{d\times m} \to \mathbb{R}^n$ is the neural network approximation for the unknown dynamical model f. The input data consist of a correlated sequence of measurements $\{\mathbf{x}_i\}_{i=t-m}^{t-1}$ with $\mathbf{x}_i \in \mathbb{R}^d$ evaluated at m time instants ahead of the prediction time t, and the new target state $\mathbf{y}_t \in \mathbb{R}^n$ is predicted for the next measurement time t from its adjacent state $\mathbf{y}_{t-1}$. In neural network predictions, multiple previous time steps $\{\mathbf{x}_{t-m}, \cdots, \mathbf{x}_{t-1}\}$ are taken together as the input to include long-time correlations from the data. The updating time step $\Delta t = t_{n+1} - t_n$ is determined by the two adjacent measurements $\mathbf{x}_n$ and $\mathbf{x}_{n+1}$, and is usually longer than the integration step dt required for numerical stability in the direct numerical scheme of the original system (12). In addition, the input data sequence $\{\mathbf{x}_i\}$ may also include measurement error and the integrated model error from the output in the previous prediction steps.

3.1.2. LSTM Network to Approximate Time-Correlated Unresolved Processes

To accurately present the time series with a long memory and correlation time, we start with the basic structure of the LSTM network [43,47] for the realization of the increment updating functional f_m^M. The LSTM network consists of m LSTM cells $\mathbf{h}_{i+1} = \mathrm{Lc}(\mathbf{x}_i, \mathbf{h}_i; \mathbf{W})$ with the same structure and parameters $\mathbf{W}$. The cells are connected by the intermediate hidden state $\mathbf{h}_i \in \mathbb{R}^h$. The long-time correlation is represented by feeding the sequential data into each cell element accordingly. The full LSTM chain is linked by the m sequential cell structures (the connection is illustrated in Figure 3).

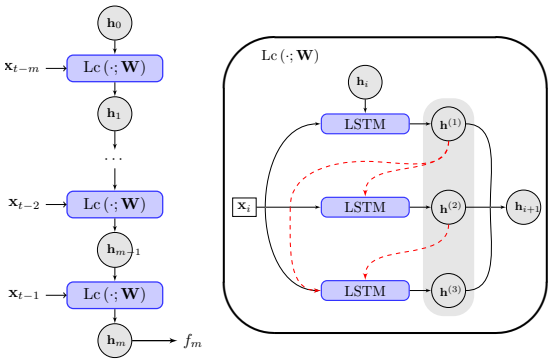


Figure 3. Illustration of the connection for the LSTM chain and the inner structure in each LSTM cell with three inner stages of the same structure.

$$\mathbf{h}_m = \mathrm{Lc}^{(m)}\{\mathbf{h}_0; \mathbf{x}_{t-m}, \cdots, \mathbf{x}_{t-i}, \cdots, \mathbf{x}_{t-1}\} \equiv \mathrm{Lc}(\mathbf{x}_{t-1}) \circ \cdots \mathrm{Lc}(\mathbf{x}_{t-i}) \cdots \circ \mathrm{Lc}(\mathbf{x}_{t-m})(\mathbf{h}_0). \tag{14}$$

In the above LSTM chain $i = 0, \cdots, m-1$, the input data $\mathbf{x}_{t-m+i}$ are fed into the corresponding i-th LSTM cell, and $\mathbf{h}_i$ is the hidden state output from the previous cell $i-1$ and input for the next adjacent cell i. For simplicity, the initial value of the hidden state is often set as zero, $\mathbf{h}_0 = 0$ (the model dependence on the initial value appears weak with a LSTM chain of moderate length m from the numerical tests). The final hidden state output $\mathbf{h}_m$ from the last step of the LSTM chain goes through another single layer of the fully connected network to give the model approximation of the dynamical increment for f

$$f_m^M = \sigma\Big(\mathbf{W}^f \mathbf{h}_m + \mathbf{b}^f\Big), \tag{15}$$

where $\mathbf{W}^f, \mathbf{b}^f$ are the linear map and bias coefficients of the final layer, and σ is a nonlinear activation function, such as the rectified linear unit (ReLU). The detailed structure for the LSTM cell with multiple gates is listed in Appendix B.1.

3.1.3. Modified Connections in the LSTM Cell Admitting Dynamical Structures

To adapt to the dynamical features using the LSTM net (14), we introduce additional modifications to the standard architecture. Considering that we usually have measurements at a larger time interval Δt, it is useful to introduce multiple inner update stages as the unresolved intermediate time steps to achieve higher-order accuracy in the final neural network output. In addition, using a single-stage update with the original LSTM will often lead to quick divergence in time iterations due to the inherent internal instability in the turbulent systems (such as the direct numerical tests in [27]).

We introduce a multistage structure in the one-step time update in each LSTM cell $\mathrm{Lc}(\mathbf{x}, \mathbf{h})$. The idea is to fill in multiple unresolved finer time stages inside the large measurement time step Δt of the available data. It is comparable to introduce a higher-order integration scheme for the discretized dynamical Equation (12)

$$\mathbf{y}_{t+\Delta t} - \mathbf{y}_t = \int_t^{t+\Delta t} f(\mathbf{x}(\tau))d\tau \approx \sum_{l=1}^{s} b_{t,l}\mathbf{f}_t^l.$$

Inside each LSTM cell for a one-step update of size Δt, we can generalize the cell structure to link concatenated basic layers of the inner connected LSTM units. It corresponds to building a multistage scheme in updating the present state to the next time step with a higher-order accuracy. Therefore, inside each LSTM cell i with input data $\mathbf{x}_i$ and the input hidden state $\mathbf{h}_i$, we can compute multiple output layers for each layer output $j = 1, \cdots, s$

$$\mathbf{h}_i^{(j)} = \mathrm{LSTM}\left(\mathbf{x}_i, \sum_{l=0}^{j-1} a_{jl}\mathbf{h}_i^{(l)}\right), \ \mathbf{h}_i^{(0)} = \mathbf{h}_i. \tag{16}$$

Above, a_{jl} are learnable model parameters for the i-th layer output. The intermediate stage outputs $\left\{\mathbf{h}^{(0)}, \cdots, \mathbf{h}^{(j-1)}\right\}$ are stacked together with the coefficients a_{jl} as the input for the next stage j. The model parameters in the LSTM units are kept the same for different stages $j = 1, \cdots, s$ since it is aimed to approximate the same dynamical functional f ultimately. Thus, the total size of the model parameters is not increased. The final hidden state output of this cell is computed with the combination of all the hidden layer outputs along the series of LSTM predictions:

$$\mathbf{h}_{i+1} = \sum_{j=1}^{s} b_j \mathbf{h}_i^{(j)}, \tag{17}$$

whereas the additional coefficients a_{jl}, b_j are added to the training parameters altogether, learned directly from the data in the optimization process.

In summary, the neural network model consists of the LSTM chain (14) with m inputs from the previous measured states $\mathbf{x}_{t-i}, i = 1, \cdots, m$ along the time series to approximate the dynamical increment $f^M \sim \mathbf{y}_t - \mathbf{y}_{t-1}$ at the next prediction time instant t. Importantly, the inner cells in the LSTM chain adopt the additional multistage scheme (17). The additional structures introduced in the model are shown to effectively improve the accuracy and robustness in both the training and prediction stages. The neural network connections are illustrated in Figure 3 for the entire LSTM chain and the inner structure of each cell.

3.2. Different Metrics for Calibrating the Loss Error

The last issue for the construction of the neural network is to define a proper loss function L measuring the error in the model prediction $\mathbf{y}^M$ compared with the target $\mathbf{y}^t$. In training the LSTM network (14), though only the last output is used for the prediction

of the next state, the intermediate cell outputs also produce meaningful predictions for the earlier states (that is, the cell i with input $\mathbf{x}_i$ can give a prediction for the state $\mathbf{y}_{i+1}$ for $i < m$). Therefore, we measure the output sequence of the last l outputs (for example, the second half $l = m/2$) in the error metric.

The proper choice of a feasible loss function L also plays a crucial role to guide the optimization procedure to an efficient convergence with emphasize on both the multiscale temporal structures and the occurrence of extreme events along the time series. We compare three different choices for the cost $L(\mathbf{x}, \mathbf{y})$ to calibrate the difference between the model output $\mathbf{x} \doteq \mathbf{y}^M$ and the truth target $\mathbf{y} \doteq \mathbf{y}^t$:

- The L_2 distance:

$$L_2(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2^2 = \frac{1}{M} \sum_{j=1}^{M} |\mathbf{x}_j - \mathbf{y}_j|^2; \tag{18a}$$

- The relative entropy loss:

$$L_{\mathrm{KL}}(\mathbf{x}, \mathbf{y}) = \frac{1}{M} \sum_{j=1}^{M} \sum_i \tilde{y}_i^{(j)} \log \frac{\tilde{y}_i^{(j)}}{\tilde{x}_i^{(j)}}; \tag{18b}$$

- Mixed loss by an L_2-relaxation with the relative entropy loss:

$$L_{\mathrm{mix}}(\mathbf{x}, \mathbf{y}) = L_{\mathrm{KL}}(\mathbf{x}, \mathbf{y}) + \alpha L_2(\mathbf{x}, \mathbf{y}). \tag{18c}$$

The L_2 distance (18a) measuring the mean square error is the most common choice of loss function, comparing the pointwise measurements of errors. In the case with data from turbulent models, small fluctuations in the solutions may end up with an unnecessarily big contribution in the L_2 loss. The pointwise measurement thus may add too much emphasis on the accumulated errors from small turbulent fluctuations. On the other hand, we aim to capture the dominant emerging features such as the extreme events. Thus, the relative entropy loss (18b) enjoys the advantage of focusing on the main coherent features of the solution invariant under small shifts in the extreme value locations. The input data in (18b) are also rescaled and measured separately from the partition functions $\tilde{x}_i^{\pm} = \frac{\exp(x_i/T_{\pm})}{\sum_i \exp(x_i/T_{\pm})}$, with $T_+ > 0, T_- < 0$ weighting the importance of the positive and negative extreme values as suggested in [31].

The mixed cost function in (18c) is shown to enjoy the advantages of the two forms of cost functions (18a) and (18b). The combination accounts for both the small-scale fluctuation and the dominant extreme events in the turbulent solution. The L_2-relaxed form is useful to fit the various small-scale structures in the solution, while the relative entropy emphasizes the extreme values in the solutions. In practice, the parameter α can take a relatively small value just as a penalty term (we pick $\alpha = 0.1$ in all the following numerical experiments as an empirical choice from different tests; it is worthwhile to carry out a systematic study for the choice of this parameter). The modified model with the above multistage inner connection in the LSTM cell is show to allow a larger learning rate in the stochastic gradient descent process, so it enjoys faster convergence and stability compared with the original models. The method is shown to be fairly robust to the choice of hyperparameters.

4. Learning Multiscale Dynamics Informed of the Physical Model

Now, we apply the general machine learning model discussed in Section 3 for the prediction of multiscale dynamical systems with extreme events. The clean structures in the unambiguous model (6) provide a desirable testbed with rich statistical features. According to the conditional Gaussian properties shown in Section 2, the neural network is designed for multiscale processes with unstable interactions. The optimized network is shown to have robust performance among different statistical regimes (explicit examples will be shown next in Section 5). The strategy is also generalizable to a wider class of complex systems with interacting scales.

4.1. General Model Setup and the Neural Network Model

For the topographic barotropic model (6), depending on the realization of the large-scale mean flow solution U, the trajectory solutions for the flow and tracer modes can be computed by integrating the exact Equations (6a) and (6c)

$$\begin{aligned}\hat{v}_k(t+\Delta t) &= \hat{v}_k(t) + \hat{F}_k^v(U(\cdot),\hat{v}_k) = \hat{v}_k(t) + \int_t^{t+\Delta t} \Big\{\Big[i\Big(k^{-1}\beta - kU\Big) - d_k\Big]\hat{v}_k - \hat{h}_k U\Big\} ds,\\ \hat{T}_k(t+\Delta t) &= \hat{T}_k(t) + \hat{F}_k^T\big(U(\cdot),\hat{v}_k,\hat{T}_k\big) = \hat{T}_k(t) + \int_t^{t+\Delta t} \big[(-\gamma_{T,k} - ikU)\hat{T}_k - \alpha\hat{v}_k\big] ds.\end{aligned} \tag{19}$$

Above, we use $\hat{F}_k^v$ and $\hat{F}_k^T$ to represent the unresolved dynamics in the fluctuation modes in different scales. The tracer modes $\hat{T}_k$ are passively advected by the advection flow field (U, v), while the flow modes $\hat{v}_k$ also give a combined contribution to the evolution of the large-scale mean flow state U. With solutions for the unresolved states, we can then compute the zonal mean flow solution by directly integrating Equation (6b) forward in time

$$U(t+\Delta t) = U(t) + \int_t^{t+\Delta t} f_U(v)d\tau + \sigma_U \sum_{j=1}^{s} \Delta W_{U,j}. \tag{20}$$

Above, the deterministic component on the right-hand side of the equation is summarized in the operator $f_U(v) = \sum_k h_k^* \hat{v}_k - d_U U$. The first term represents the topographic stress from the combined feedback from all the small-scale modes, and the second term serves as a friction from the boundary. The last white noise forcing introduces additional uncertainty to the system with different noise amplitudes σ_U. In addition, the time integration about $f_U(v)$ also includes model approximation error since we only have the model output from the neural network at two adjacent time measurements at t and $t + \Delta t$, and the data observation time Δt is often larger than the desired integration time step dt.

We then apply the general neural network framework (13) to the specific set of Equation (19) for both the flow and tracer modes. In practice, the input data can be chosen to include all state variables, including both large- and small-scale flow and tracer modes $\mathbf{x} = \{U, \hat{v}_k, \hat{T}_k\}$, and the prediction targets are set as the unresolved small-scale flow and tracer modes $\mathbf{y} = \{\hat{v}_k, \hat{T}_k\}$ at the next prediction time among all the wavenumbers k. The length of the LSTM net m can be determined according to the decorrelation time of the state so that $m\Delta t \sim T_{\text{decor}}$, where $T_{\text{decor}} = \int \mathcal{R}(s)\mathrm{d}s$ is the integrated autocorrelation function characterizing the mixing time scale.

4.1.1. Decoupled Neural Networks for Multiscale Dynamics

As we have shown in the model analysis in Section 2, one of the main challenges in learning the turbulent dynamics is to resolve the strong scale separation between the coupled processes. The smaller scale mode with a larger wavenumber usually exhibits faster mixing time and smaller variance (see from the explicit formulas and numerical examples in Section 2.3). For efficient modeling, we can decompose the high-dimensional systems according to the different scales, and propose a neural network model focusing on one specific scale structure so that the multiscale structure is better represented. The resulting neural networks also become easier to train since we decompose the large system into several smaller subsystems, requiring fewer model parameters.

Exploiting the conditional Gaussian structure, small-scale modes $\hat{v}_k, \hat{T}_k$ in (19) become decoupled conditional on the large-scale mean state U, while the mean state U is updated by the physical model (20) from the combined feedback from different scale modes. In particular, for the two-mode model, it is natural to propose two separate neural networks for the wavenumbers for capturing different scale dynamics which are coupled finally through the large-scale dynamics of U. The general framework for learning the dynamical structure of the two-mode topographic barotropic system (6) can be modeled together

using the multiscale neural network models $\{f_k^M\}$ and the explicit physical dynamics for the observed large-scale state U

$$\mathbf{y}_{n+1}(k) = \mathbf{y}_n(k) + f_k^M(\{\mathbf{x}_{n-m:n}(k)\}; \mathbf{W}); \tag{21}$$

$$U_{n+1} = U_n + \Delta t F_U(\{\mathbf{y}_{n,n+1}\}) + \sigma_U \sqrt{\Delta t}\xi_n. \tag{22}$$

Above, the neural network model f_k^M in (21) is used to learn the detailed single-scale dynamical updates in each small-scale mode. The large-scale dynamics is integrated explicitly using the physical dynamics from the output data of the neural network. Thus, the true large-scale physical dynamics can be introduced directly to the training model. $F_U(\mathbf{y})$ gives the approximation for the deterministic integration, while additional uncertainty is introduced from the white noise forcing ξ. For computing the explicit integration, we use the mid-point rule combining the input and output data

$$\Delta t F_U(\mathbf{y}_{n,n+1}) = \int_t^{t+\Delta t} f_U(v)d\tau = \frac{\Delta t}{2}[f_U(v_n) + f_U(v_{n+1})], \tag{23}$$

where v_n is the model input and v_{n+1} is the model prediction at the next time step for the numerical integration of the term $f_U(v)$ in (20). In this way, the predicted state U will incorporate information from the neural networks and the large-scale physical process.

As a remark, above, we apply the neural network to the example of the simplest two-mode model so that we are able to carry out a detailed investigation for the various features in the neural network model using the explicit model solutions. However, the idea for modeling multiscale structures can be easily generalized to higher-dimensional systems with multiple interacting scales. For example, a high-dimensional system can be decomposed into block-diagonal subsystems using the conditional Gaussian framework [24,42]. Then, the neural network model can be applied to each block approximating a series of subsystems focusing on different scales.

4.1.2. Multistep Training Loss Including Time-Dependent Data

Above, the model (21) and (22) gives a one-step prediction with a time step size Δt for the long time series. In practice, we would like to use the trained network model for long-time prediction even beyond the decorrelation time of the system. Then, there comes the problem of numerical stability and robustness from the accumulation of model approximation errors amplified through the repeated updating steps with strong internal model instability. To address this issue, a multistep model output is used in optimizing the loss function during the training process.

In training for the turbulent model with high degrees of instability, the loss function is expected to guide the neural network to gain the skill to detect the occurrence of bursting extreme events as well as the complex structures in the dynamical model. Therefore, instead of simply training the model from a one-step output, we iterate the system (21) and (22) forward in time up to n steps using the model output as the initial data for the next iteration. The general form of the loss function can be designed by the total loss along the N updates with the proper cost L in (18)

$$\mathcal{L}_n = \sum_{i=1}^{n} w_i L\left(F_M^{(i)}(\mathbf{x}; \mathbf{W}), \mathbf{y}^{(i)}\right). \tag{24}$$

Above, the LSTM model (21) is trained online, combining the output from the physics model (22) during the time iterations. $F_M^{(i)}$ is the push forward operator from recursively running the model up to $n\Delta t$, and $\mathbf{y}^{(i)}$ is the target state to be compared. The weights w_i offer a balanced calibration of the model output series. A convenient choice of the weights is to use the autocorrelation function $w_i = \left|\mathcal{R}_y(i\Delta t)\right|$ for the corresponding state in the measurement. This provides a balanced quantification for the prediction error, where the

prediction for longer future time is tolerated with the smaller weight w_i. Note that the 'multistep' training here involves time updating with n iterations, which is different from the 'multistage' neural network architecture in Section 3.1 inside one time update Δt.

To exploit the conditional Gaussian structure of the two-mode model with different statistics, we divide the data for training and prediction according to the large-scale mean flow U. Based on the explicit solutions, data in a moderate regime (with a relatively small white noise forcing σ_U) are used to train the model parameter under the loss function (24). The trained model is then applied to predict the solutions among different statistical regimes from near-Gaussian to highly skewed PDFs according to different noise levels of σ_U (see the results in Section 5.3 and Appendix B). The model is shown to be robust to model errors from multistep iterations and noise from the above multistep training strategy.

4.2. Metrics to Measure the Accuracy in Training and Prediction

Finally, the proper metrics should be proposed to calibrate the accuracy in the model training results and predictions. In the training stage with a batch of M samples, the neural network is iterated N steps for a sequence of model outputs at time $t_n = n\Delta t, n = 1, \cdots, N$. One direct way to compute the training error is through the L_2 distance among all the M training samples and in the N-step outputs. The *batch averaged relative mean square error* (BMSE) can be defined as

$$\mathrm{BMSE} = \frac{1}{M}\sum_{i=1}^{M}\frac{\sum_{n=1}^{N}\left\|\mathbf{y}_{\mathrm{m},i}^{(n)} - \mathbf{y}_{\mathrm{t},i}^{(n)}\right\|^2}{\sum_{n=1}^{N}\left\|\mathbf{y}_{\mathrm{t},i}^{(n)}\right\|^2}, \tag{25}$$

where $\mathbf{y}_{\mathrm{m},i}^{(n)}$ is the model output in the i-th sample and at the n-th step model output, and $\mathbf{y}_{\mathrm{t},i}^{(n)}$ is the corresponding target state at time step $t_n = n\Delta t$. The relative error is normalized by the L_2-norm for the true state $\left\|\mathbf{y}_{\mathrm{t},i}^{(n)}\right\|^2$ where all the samples are summed with index i, and thus the BMSE error measures the training accuracy according to the total variability among the batch samples.

In the prediction stage, we need to track the development of errors in time when the solution is achieved by recursively iterating the optimized model using the outputs. First, for the trajectory prediction for a long time series, we can calibrate the trajectory error at each single prediction time t by comparing the error with the equilibrium variance of the state. The *normalized mean square error* (NMSE) for prediction accuracy can be defined by the averaged error of M test samples

$$\mathrm{NMSE}(n) = \frac{1}{M}\sum_{i=1}^{M}\sigma_y^{-2}\left\|F_M^{(n)}(\mathbf{x}_{0,i}) - \mathbf{y}_{\mathrm{t},i}^{(n)}\right\|^2, \tag{26}$$

where $\sigma_y^2 = \left\langle |\mathbf{y}_\mathrm{t}|^2 \right\rangle$ is the variance of the predicted state at equilibrium. F_M is the optimized model operator, and the prediction at the time $t = n\Delta t$ from the initial state $\mathbf{x}_0$ is computed by iterating the model n times. Next, it is also useful to measure the statistical accuracy in mean and variance from the statistical model with ensemble solutions. Therefore, we can define the *statistical mean error* (SME) and *statistical variance error* (SVE) as

$$\mathrm{SME} = \frac{|\langle \mathbf{y}_\mathrm{m}\rangle - \langle \mathbf{y}_\mathrm{t}\rangle|^2}{\left\langle |\mathbf{y}_\mathrm{t}'|^2 \right\rangle}, \quad \mathrm{SVE} = \frac{\left|\left\langle |\mathbf{y}_\mathrm{m}'|^2 \right\rangle - \left\langle |\mathbf{y}_\mathrm{t}'|^2 \right\rangle\right|}{\left\langle |\mathbf{y}_\mathrm{t}'|^2 \right\rangle}. \tag{27}$$

Above, we use $\langle f\rangle = \frac{1}{M}\sum_{i=1}^{M} f_i$ to represent the statistical expectation computed by averaging among the samples. $\langle \mathbf{y}_\mathrm{m}\rangle, \langle \mathbf{y}_\mathrm{t}\rangle$ are the ensemble means for the model prediction and target data statistics; $\left\langle |\mathbf{y}_\mathrm{m}'|^2 \right\rangle, \left\langle |\mathbf{y}_\mathrm{t}'|^2 \right\rangle$ are the variances about the fluctuation states

$\mathbf{y}' = \mathbf{y} - \langle \mathbf{y} \rangle$. Instead of the pointwise measurement of errors in (26), the statistical errors in (27) calibrate the model's skill in capturing the representative statistical moments in an ensemble solution.

5. Predicting Extreme Events and Related Statistics using the Neural Network Model

In this section, we study the learning and prediction skill in the developed neural network model guided by the tractable model framework of the coupled topographic flow and tracer system. As analyzed in the explicit formulas in Section 2, the simple system is able to generate a variety of regimes with distinctive statistics and various extreme events. The new model architectures described in Section 3 are applied to capture the essential multiscale dynamics in both the turbulent flow and passive tracer fields at the same time, using a single set of data. In particular, we focus on two representative statistical regimes with strong $H = 10$ and weak $H = 1$ topography (as illustrated in Figure 1).

The data are collected through a long-time simulation of the original topographic model (6). The entire dataset is divided into two sections with short time trajectories for training and a new set of long time series to check the prediction accuracy:

- In the *training stage*, the time trajectory is segmented into batches of short sequences for training the neural network model parameters. Usually, the model is only updated a small number of steps of length $n\Delta t$ (say, $n = 10$ in the standard test case) for efficient training.
- In the *prediction stage*, the optimized model is used for prediction along a long time sequence. The prediction model is iterated recurrently using the previous outputs up to a long-time $N\Delta t$ (say, $N = 50,000$ iterations).

In the training stage, a huge computational overload will be generated if we measure the errors in the model outputs for too many iteration steps because we need to back-propagate all the way to the first step in computing the gradient of the final loss function (24). Therefore, a smaller number of model update n is preferred for saving memory and efficiency. In the prediction stage, on the other hand, we need to keep iterating the model output for the next forecast step to a long-time prediction time. Thus model approximation errors will accumulate in time, and this requires stability in the constructed model, especially for the highly turbulent regime with strong inherent internal instability.

In the benchmark model, we fix the standard model hyperparameters as the input LSTM chain length $m = 100$, the hidden state in the LSTM cell $h = 50$, measurement time step in the data $\Delta t = 0.1$ (which is 10 times the numerical integration step $dt = 0.01$ in the direct numerical scheme). In the multistep training loss in (24), the training model is iterated forward $n = 10$ times. This accounts for a time length $T = n\Delta t = 1$ which is still much shorter than the decorrelation time of the slow modes (see the autocorrelation functions in Figure 2). The detailed neural network model configuration and parameters are listed in Appendix B.2.

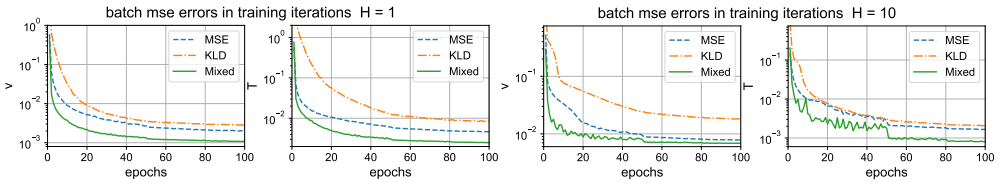
5.1. Training and Prediction with Different Loss Functions

First, we compare the training performance under different metrics of the loss functions to measure the training error. During the training procedure, we generate a long time series with 10,000 measurements in time, and divide the dataset into smaller batches with batch size 100 to train each batch with 100 iterations. Then, the model is trained repeatedly over the same data for 100 epochs. The learning rate starts with $\mathrm{lr} = 0.005$, and is decayed by 0.5 of its previous value twice, at epoch numbers 50 and 80.

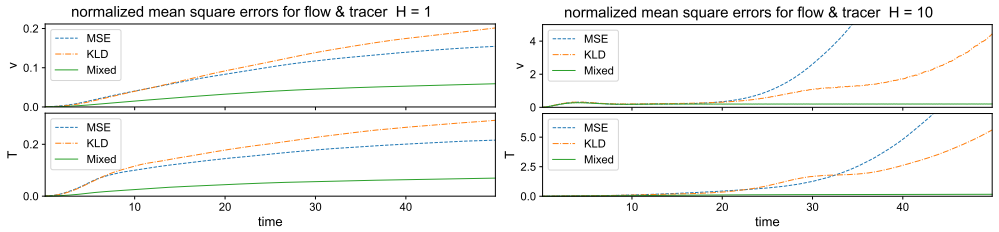
Here, the models with same architecture and hyperparameters are trained under the same set of data but optimized using the three different types of loss functions (18a)–(18c) as the optimization target. The L_2 mean square error (18a) is the most common choice for the error metric, while for training turbulent signals with intermittency and extreme events, it is found that the introduction of the KL divergence (18b) is effective for capturing the dominant extreme values [31]. In the mixed metric (18c) for loss, the KL divergence is still

taking the dominant role with the L_2 error creating an additional balancing effect with a small weight $\alpha = 0.1$.

The training errors under different loss functions are compared in the upper panel of Figure 4. We compute the batch averaged mean square errors (25) for the flow and tracer states during training iterations for the two test regimes with strong and weak topography $H = 10$ and $H = 1$. The BMSE represents the averaged relative error in the training batch, so it serves as an index for training accuracy in each epoch. In general, all the three metrics are effective to improve the model accuracy through training iterations. The mixed metric gains the highest overall training accuracy among the flow and tracer states, and decays to a smaller error value in a faster rate than that of the other two metrics. The faster convergence implies that the mixed metric loss function is easier and more efficient to train to reach a higher accuracy with much fewer required training epochs.



(**a**) training MSEs with different topography strengths H



(**b**) prediction MSEs with different topography strengths H

Figure 4. Comparison of training and prediction errors under three different loss functions L for optimization: L_2 loss (MSE), KL divergence (KLD), and the mixed loss combining these two. Training iterations of the batch averaged mean square errors are shown in the upper panel with logarithmic scale along the y coordinate. The prediction normalized mean square errors are compared in the lower panel measured at recurrent predictions of 500 time steps up to $T = 50$.

Next, to check the prediction skill and stability of the trained neural network models optimized under the three different loss functions, we compare the errors (26) under a new dataset for the prediction of a relatively long time series of $L = 500$ steps (that is, up to time $T = L\Delta t = 50$ far beyond the model decorrelation time, see the autocorrelation functions in Figure 2). The model prediction in the previous time step is used recurrently for the prediction in the next time instant; thus, prediction errors will accumulate in time. One important issue is to check whether the trained neural network model can stay robust to the increasing level of errors in the input data. The lower panel of Figure 4 compares the evolution of prediction errors during the time updates. In the weak topography case $H = 1$, the three optimized models all stay stable within finite errors during the iterative updates. Still, the optimal model trained with the mixed loss function gives the most accurate long-time accuracy with the smallest error in both flow and tracer states during the entire time. The models with the other two loss metrics generate much larger prediction errors for the longer prediction range. In the strong topography case $H = 10$, the stronger coupling between the large and small scales increases the instability in the system. The models under the MSE and KLD loss perform well only in the starting time but eventually become unstable as the errors diverge to infinity. In contrast, the model under the mixed loss function stays stable with high accuracy during the entire prediction process. This further confirms the crucial role in selecting the proper loss function for the robust performance of turbulent systems with instability.

5.2. Maintaining Model Stability by Measuring Multiple Forward Steps

Another important issue we would like to check is the number of forward steps n used in training the loss function (24)

$$\mathcal{L}_n = \sum_{i=1}^{n} w_i L(F_{i\Delta t}(\mathbf{x}), \mathbf{y}_{i\Delta t}),$$

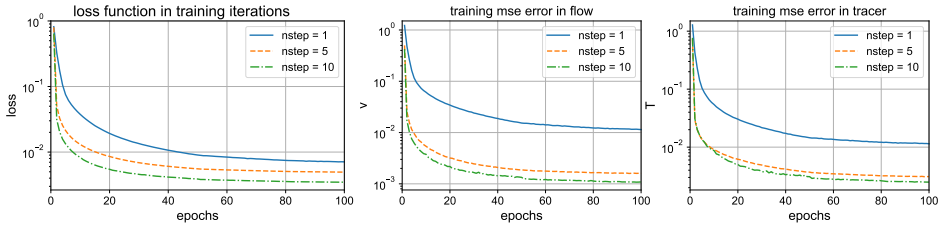
where the model i-th step output $F_{i\Delta t}(\mathbf{x})$ is compared with the target data $\mathbf{y}_{i\Delta t}$ under the most effective mixed loss metric L, and $w_i = \mathcal{R}(i\Delta t)$ is the weighting factor from the autocorrelation function. The most straightforward way is to measure the error in a single-step LSTM net output $n = 1$. However, as we have discussed in the previous section, this may lead to severe instability with exploding errors for long trajectory iterations of the model.

To address this problem, we adopt the multistep training strategy using multiple model outputs to be measured in the loss function $\mathcal{L}(\mathbf{x}, \mathbf{y})$ with $n > 1$. In order to capture the dynamical variability in a longer time range, we iterate the training model forward in time with moderate steps $n = 10$. The errors will accumulate in time, and the loss function at different forward times is compensated for by the decaying autocorrelation function $\mathcal{R}(t)$. As a cost, more memory will be consumed during the back-propagating of the entire network. However, as we will show next, the cost can be controlled by using an even smaller number of steps (such as $n = 5$) to reach the desirable training performance.

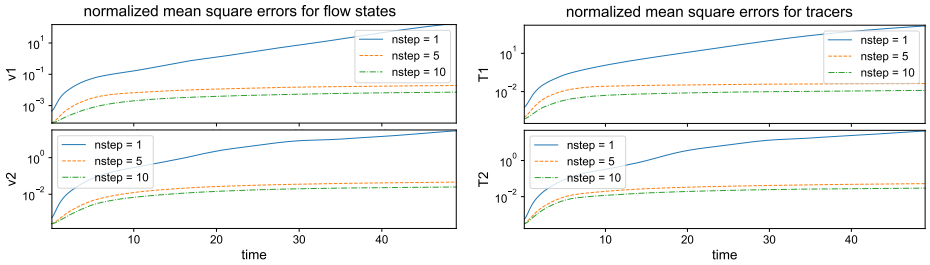
5.2.1. Training and Prediction Errors with Different Forward Time Steps

Again, we first compare the training performance in the models measuring different forward steps in the training metric. We focus on the improvement by using a multistep forward model $n = 5, 10$ in contrast with a single-step training $n = 1$. Figure 4 displays the training loss and accuracy during the training iterations. The value of the loss function decays faster to the final optimized level in the multistep training cases. And a larger forward step $n = 10$ improves the overall accuracy compared with a smaller step $n = 5$. For the training accuracy, the single-step model $n = 1$ suffers a barrier to reach high accuracy compared with the other two models with multiple steps $n = 5$ and $n = 10$. The two models with different forward steps $n = 5, 10$ can reach comparable amplitudes of final training error, while the multistep model $n = 10$ gains a higher accuracy and stability for long trajectory prediction.

Then, we compare the prediction skill in the three trained models on a new set of data for model evaluation. As previously, we run an ensemble prediction and recursively iterate the model outputs for the next step prediction up to the final prediction time $T = 50$ (with 500 iterations). The lower panel of Figure 5 compares the normalized MSEs using trained models with one-step $n = 1$ and multistep $n = 5, 10$ training. The multistep trained models gain very high accuracy in both the flow and tracer modes. The errors stay small during the entire prediction process; thus, the model approximation errors from the previous steps will not damage the accuracy in the future forecasts, implying the robustness of the trained model. On the other hand, the one-step trained model becomes insufficient to maintain the accuracy as the forecast step increases. The predicted solution stays accurate for the starting period of time, then diverges as the errors accumulate in time. Notice the logarithmic scale in the y coordinate for a much larger error in the $n = 1$ case.



(**a**) training MSEs with iterations



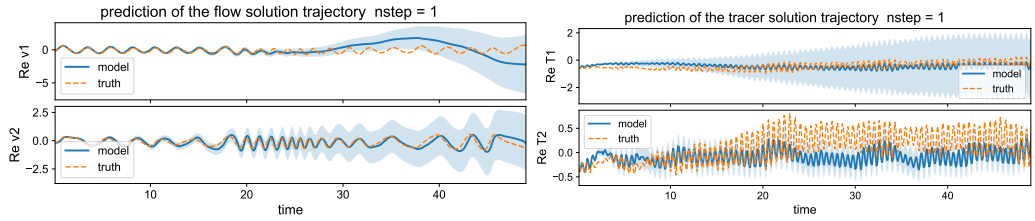
(**b**) normalized MSEs in prediction steps

Figure 5. Upper: the evolutions of loss function values and the BMSEs during training iterations for models with different training forward steps $n = 1, 5, 10$. Lower: normalized MSEs in model prediction with an ensemble prediction of $M = 500$ samples up to time $T = 50$. Performance of the trained models with different forward time steps $n = 1, 5, 10$ in optimization are compared. Logarithmic scale is applied along the y coordinate.

5.2.2. Trajectory Prediction Including Multiple Time Scales

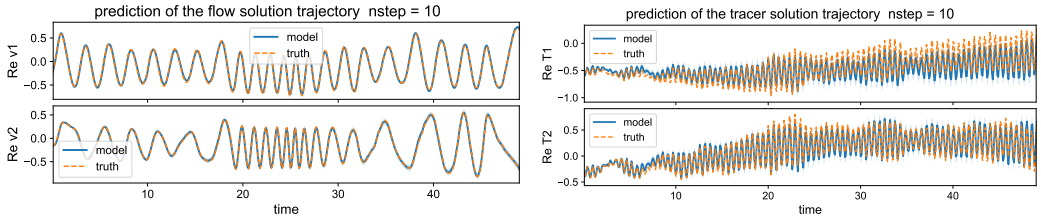
In addition, to provide a better illustration for the predicted solutions from models with different training steps, we plot one realization of the trajectories using the trained models with single and multistep training. Figure 6 shows one typical solution trajectory compared with the truth from the direct model simulation. The variance of the sample errors is plotted by the shaded area around the solution line to characterize the uncertainty in the ensemble prediction. Consistent with the previous observation from the errors, the multistep training model has good agreement with the truth in the prediction through the large number of model iterations. The multistep trained model also maintains the overall accuracy with very small variance in errors.

In contrast, the single-step trained model can only stay accurate for a short time from the initial state, while the model approximation errors quickly drive the model prediction away from the target trajectory as errors grow in time during the iterations. Large deviation from the truth is developed in time gradually, and the samples give increasingly large variance among the errors from different trajectory predictions. In particular, in the tracer time series, the solutions display two contrasting time scales with a fast oscillating small scale on top of the long-time slowly varying profile. The one-step trained model fails to track the rapid variability in the solution, and strong instability in the system leads to large error variances. The multistep training model, on the other hand, captures both time scales with very small error variance for uniformly high accuracy and stability among all the samples during the entire simulation time.



(**a**) model trained with one forward step $n = 1$

Figure 6. *Cont.*



(**b**) model trained with ten forward steps $n = 10$

Figure 6. Trajectory prediction of flow and tracer solutions compared with the truth from direction model simulations. Performance of the trained models with different forward time steps $n = 1, 10$ are compared. The sample error variance is marked by the shared area around the prediction. The multistep training case $n = 10$ has stable performance with tiny error variance.

5.3. Long-Time Prediction in Different Dynamical Regimes

In the final test, we check the model prediction for the long-time trajectory and statistical solutions of the advection flow states and the passive tracers. This is first used to confirm the long-time stability of the neural network scheme with errors accumulated in time through the time iterations; next, the same trained model is used to predict the key statistical features with different noise-forcing levels to show the scope of skills in the model. To evaluate the prediction accuracy in the neural network models, we adopt two approaches by examining both the deterministic trajectory solution and the statistical solution:

- *Trajectory prediction*: the neural network is used for the pathwise prediction for one trajectory solution from a particular initial state with uncertainties from input value and white noise forcing. Solution trajectories are solved very efficiently with $N = 50{,}000$ iterations (with time step size $\Delta t = 0.1$ to the final simulation time $T = 5000$).
- *Statistical prediction*: the neural network is used to recover from data key statistical features in leading-order statistics generated by different white noise forcing amplitudes. Instead of focusing on the pathwise solutions, it is often more useful with practical importance to learn the representative statistical structures directly.

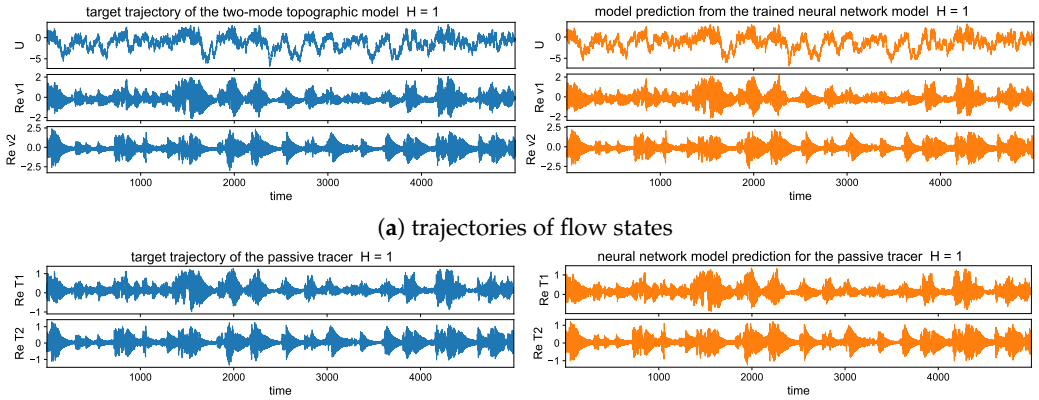
In confirming the universality of the trained model, the neural network model is trained using a limited dataset from a single white noise-forcing regime $\sigma_U \dot{W}_0$ of moderate amplitude $\sigma_U = 10\sigma_0 = \frac{1}{2\sqrt{2}}$ in the large-scale flow equation. The small-scale flow and tracer dynamics are learned based on this dataset, then the trained model is applied for the various dynamical regimes by changing the noise forcing amplitude σ_U. This can also serve as a way to confirm that the true dynamical structure is indeed learned from the model rather than being due to purely the overfitting of the data.

5.3.1. Trajectory Prediction with the Trained Model in Different Dynamical Regimes

In the trajectory prediction through the neural network model, we first check the detailed prediction skill for the multscale flow and tracer structures and the occurrence of extreme events. Even though high instability exists, preventing the long-time predictability, we can exploit the conditional Gaussian structure of the model and achieve accurate pathwise prediction. The conditional solution of the small-scale solutions $\hat{v}_k(U(\cdot))$ together with the passive tracer modes $\hat{T}_k(U(\cdot))$ can be predicted from the trained neural network efficiently given the observed solution of the large-scale process U.

The long-time trajectory predictions in the weak and strong topographic stress regimes and the noise forcing $\sigma_U = 10\sigma_0$ are shown in Figures 7 and 8. The true model solution and statistics can be found in Figures 1 and 2 in Section 2. The solutions display very strong intermittent bursts of extreme events in both small-scale flow and tracer models companied by a very fast oscillating scale. Strong time-scale separation also exists between the advection flow states (U, v) and the passive tracer T. The neural network model is shown to accurately track such features in time and keep the high accuracy during the

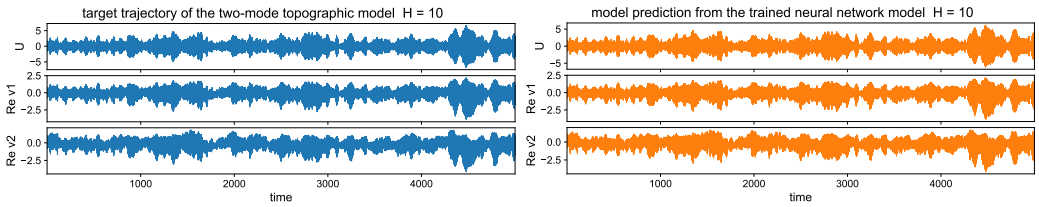
model iterations for the entire duration. This trajectory prediction shows explicitly the skill of the proposed neural network model to learn and recover the true model dynamics.



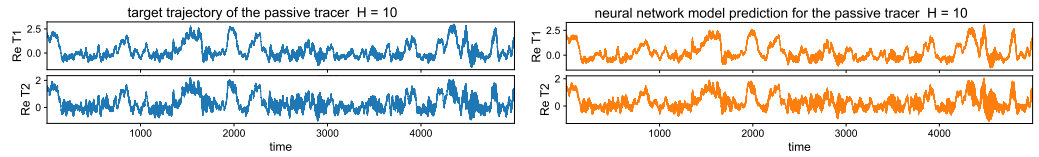
(**a**) trajectories of flow states

(**b**) trajectories of passive tracers

Figure 7. Long-time trajectory prediction of the flow and tracer solutions with weak topographic stress $H = 1$ up to $T = 5000$. The reference states from the direct model simulation (**left** panel) are compared with the model prediction (**right**) using the trained neural network model.



(**a**) trajectories of flow states



(**b**) trajectories of passive tracers

Figure 8. Long-time trajectory prediction of the flow and tracer solutions with strong topographic regime $H = 10$ up to $T = 5000$. The reference states from the direct model simulation (**left** panel) are compared with the model prediction (**right**) using the trained neural network model.

In addition, the same trained neural network model can be used for the prediction with different white noise-forcing levels σ_U in the large-scale mean flow Equation (6b). Notice that the small-scale Equation (6a) to be learned from the neural network has the same fixed dynamics, while different noise levels σ_U can induce distinct statistical features in both small- and large-scale states. This guarantees the validity of the trained model among different statistical regimes once the essential dynamics is learned from data. In Appendix B.4, we show the prediction results with a smaller or larger effect of white noise forcing $\sigma_U = \sigma_0$ and $\sigma_U = 20\sigma_0$ using the same trained neural network model to recover the unresolved small-scale solutions. It is further confirmed the universal prediction skill in the trained model among different statistical regimes, and the optimal performance is not purely through the overfitting of the training data.

5.3.2. Statistical Prediction in Leading Statistics for Different Noise Forcing

At last, we show the skill in the neural network model to recover the model statistics among various statistical regimes. By inspecting the analytical analysis results from the original model and the direct simulations of the original model, a wide variety of distinct statistics are generated under the same model framework. The same trained neural network model is then

applied to predict the statistical features under these different forcing scenarios by varying the white noise-forcing amplitude σ_U. The conditional dynamics for the small-scale flow and tracer states stay the same for different noise levels. The question is whether the trained neural network model is capable of recovering the different statistics in a uniform way from changing the noise-forcing strength σ_U for the large-scale mean state.

The model is trained based on a single set of data, being especially unaware of the strong extreme event regimes with a large amplitude from a large forcing. To measure the accuracy in the prediction of the leading statistics, we use the relative statistical error metrics (27) for measuring the accuracy in the ensemble mean and variance. In the tests, we run an ensemble prediction of $M = 5000$ trajectories and iterate the model for $N = 500$ steps. The statistics are computed from the model output in the last 200 steps when the steady state is reached. The neural network model enjoys the advantage to run a large ensemble very efficiently compared with the direct simulation.

Table 1 lists the statistical prediction for systems with different white noise-forcing strengths $\sigma_U = 10\sigma_0, 20\sigma_0$. In general, the neural network model shows uniformly high skill in recovering the leading statistics in mean and variance among the different statistical regimes. The statistical mean error (SME) calibrates the deviation in the ensemble mean from the ensemble prediction in each of the flow and tracer modes. The statistical mean error stays in small values and keeps very high accuracy. The statistical variance (SVE) calibrates the deviation in the ensemble variance compared with the truth. This characterizes the model uncertainty in each mode, and thus is a more interesting quantity to measure. The modes become more energetic with higher uncertainty, as the white noise forcing σ_U increases. The statistical errors grow with the larger value of σ_U, while they all stay as small values for an overall accurate statistical prediction.

Table 1. Statistical error in mean (SME) and variance (SVE) from the ensemble prediction of the trained neural network model. The two parameter regimes with topographic stress $H = 1$ and $H = 10$ are compared. The same trained model is used for the statistical prediction with different white noise amplitudes $\sigma_U = 10\sigma_0, 20\sigma_0$.

		$\sigma_U = 10\sigma_0$				
		U	$\hat{v}_1$	$\hat{v}_2$	$\hat{T}_1$	$\hat{T}_2$
$H = 1$	SME	3.03×10^{-4}	2.49×10^{-5}	5.53×10^{-4}	1.96×10^{-5}	7.60×10^{-4}
	SVE	7.26×10^{-3}	6.57×10^{-2}	3.12×10^{-2}	7.50×10^{-2}	3.57×10^{-2}
$H = 10$	SME	5.34×10^{-4}	2.87×10^{-4}	9.16×10^{-4}	4.29×10^{-2}	8.05×10^{-2}
	SVE	9.25×10^{-1}	8.82×10^{-2}	8.51×10^{-2}	5.31×10^{-2}	6.75×10^{-2}
		$\sigma_U = 20\sigma_0$				
		U	$\hat{v}_1$	$\hat{v}_2$	$\hat{T}_1$	$\hat{T}_2$
$H = 1$	SME	7.98×10^{-2}	8.50×10^{-3}	2.03×10^{-2}	6.18×10^{-2}	1.92×10^{-2}
	SVE	1.92×10^{-1}	2.40×10^{-1}	2.57×10^{-1}	2.51×10^{-1}	2.49×10^{-1}
$H = 10$	SME	2.84×10^{-5}	6.77×10^{-2}	2.53×10^{-1}	4.60×10^{-2}	1.44×10^{-2}
	SVE	9.00×10^{-1}	4.68×10^{-1}	4.03×10^{-1}	4.41×10^{-1}	2.03×10^{-1}

6. Summary and Discussion

We study effective machine learning strategies to predict the various anomalous statistics and the occurrence of extreme events in complex turbulent systems using an unambiguous model framework. The model is derived from the geostrophic barotropic flow and turbulent passive tracer transport [29,34] that share many similarities with natural and laboratory observations [8,54]. The coupled system is characterized by interacting multiscale processes in time and space, leading to very complicated dynamical structures. The attractive statistical features include exact formulas for flow and tracer solutions, explicit

nonlocal structures in flow and tracer modes, and the intermittent probability distributions with fat-tails and skewed PDFs. The tractable solutions of the model framework facilitate a systematic analysis of crucial multiscale properties with both mathematical theories and the development of novel numerical strategies. Detailed data-driven models are constructed to recover these statistical features from limited observed data combined with model noises.

We consider two approaches of using machine learning techniques to predict representative model solutions and statistics based on the conditional Gaussian properties: (i) a neural network to learn the unresolved small-scale dynamics and directly predict the trajectory solutions guided by the conditional Gaussian framework, and (ii) using the neural network to recover the crucial statistical moments among different regimes. New architectures are designed on the LSTM network with a residual network structure to predict the dynamical update of the unresolved small-scale states. The individual model outputs in different scales are combined in the explicit large-scale mean flow equation to inform the model with physical dynamics. Multiscale effects in large and small scale flow states as well as in the tracers are considered in the model construction and training procedures. We find the major observations in model performance using the simple two-mode topographic model:

- The trained neural network model shows uniformly high skill in learning the true dynamics. The improved model architecture enables a faster convergence rate in the training stage, and more accurate and robust predictions under different forcing scenarios. A longer time updating step is permitted allowing data measured at sparse time intervals.
- The choice of a proper loss function for the optimization of model parameters is shown to have a crucial role to improve the accuracy and stability in the final trained neural network. A mixed loss function using the relative entropy loss together with a small L_2 loss correction is shown to effectively improve the accuracy in training for complex systems with extreme events.
- A multistep training process, that is, using multiple iterative model outputs in training the loss function, is useful to improve model stability against the accumulated model errors during long-time iterations. The prediction skill of the model can be improved, and training efficiency is maintained by measuring only small update steps during the training procedure.
- The solution trajectory can be tracked by the neural network model with high accuracy and stability in a long time series prediction for the key multiscale structures with extreme events in flow and tracer states.
- Different model statistics in ensemble mean and variance can be predicted with uniform accuracy among different dynamical regimes using the same neural network model trained from a single set of data.

The promising results just set the starting point for a series of interesting future research directions for the next stage. The neural network model provides the exact structure to incorporate the conditional Gaussian framework and multiscale nonlinear dynamics. A direct generalization is to use the neural network model to prediction explicit higher-order statistics as well as the non-Gaussian PDFs in the flow and tracer fields. This framework is also ready to be generalized to a wider group of complex models with a large number of interacting modes and contributions from different scales by assigning the neural networks to capture processes with different scales. This conditional independent construction of models is easy to parallelize and thus becomes especially convenient for the implementation on GPUs. The neural network framework also shows potential to be combined with the linear and nonlinear response theories for the development of statistical data assimilation and control of high-dimensional systems [55–57]. Thus, efficient statistical model reduction strategies [26,27] can be directly applied to learn the dynamical structure from the nonlinear interactions directly from data.

Funding: This research was funded by ONR grant number N00014-24-1-2192.

Data Availability Statement: The data presented in this study are openly available in GitHub at https://github.com/qidigit/ (accessed on 30 April 2024).

Acknowledgments: The author is grateful to Andrew J. Majda for the many invaluable discussions and insightful suggestions that inspired this research. The author also would like to express thanks for the suggestions given by the anonymous reviewers.

Conflicts of Interest: The author declares no conflicts of interest.

Appendix A. Details about the Derivations for the Flow and Tracer Model Solutions

In this appendix, we show the detailed derivation of the representative statistical solutions in the topographic flow and passive tracer fields. We can rewrite the original dynamical flow and tracer Equation (6) for each spectral mode as

$$\begin{aligned} \frac{dU}{dt} &= \sum_k h_k^* \hat{v}_k \ - d_U U + \sigma_U \dot{W}_0. \\ \frac{d\hat{v}_k}{dt} &= i\omega_{v,k}(U)\hat{v}_k - l_x^2 \hat{h}_k U \ - d_{v,k}\hat{\omega}_k + \sigma_{v,k}\dot{W}_k, \\ \frac{d\hat{T}_k}{dt} &= i\omega_{T,k}(U)\hat{T}_k \ - d_{T,k}\hat{T}_k - \alpha \hat{v}_k, \end{aligned} \tag{A1}$$

where the dispersion relations for the shear flow and tracer are defined by the mean flow state

$$\omega_{v,k} = l_x\Big(k^{-1}\beta - kU\Big), \ \omega_{T,k} = -k(U+u),$$

and the small-scale velocity field is directly related with the vortical mode $\hat{\omega}_k$ through the Fourier expansion

$$u = il_y \sum_k k^{-1}\hat{\omega}_k e^{ik\mathbf{l}\cdot\mathbf{x}}, \ v = -il_x \sum_k k^{-1}\hat{\omega}_k e^{ik\mathbf{l}\cdot\mathbf{x}}.$$

Usually, for simplicity, we just set $l_x = 1$ and $l_y = 0$; thus, we can focus on the cleaner case, where the zonal flow cross-sweep is purely contributed from a large-scale solution of U, while the small-scale modes give all the shear flow fluctuations $\hat{v}_k$.

Appendix A.1. Explicit Statistical Solutions for the Flow Field

First, we can compute the trajectory solution for the shear flow v directly by integrating the second equation of (A1)

$$\hat{v}_k = -\int_0^t e^{\left(-d_{v,k}+ik^{-1}\beta\right)(t-s)} e^{-ikU[s,t]}\Big[\hat{h}_k U(s)ds + \sigma_{v,k}dW_k(s)\Big]. \tag{A2}$$

Above, we define $U[s,t] \equiv \int_s^t U(\tau)d\tau$, including the history of the large-scale solution $U\,|_{s\le t}$, and assume the initial state is unrelated for a long-time statistical solution. Furthermore, by substituting the solution (A2) back into the large-scale mean flow Equation (6b), the trajectory solution of the large-scale mean flow U can be computed as a closed system from the integro-differential equation

$$\frac{dU}{dt} + \int_0^t \sum_k e^{\left(d_{v,k}-ik^{-1}\beta\right)(t-s)} e^{-ikU[s,t]}\left[\left|\hat{h}_k\right|^2 U(s)ds + \hat{h}_k^*\sigma_{v,k}dW_k(s)\right] = -d_U U + \sigma_U \dot{W}_0. \tag{A3}$$

Thus, the solution of the mean flow U can be determined by the coupling effect from topographic stress h and the white noise forcing in both small and large scales.

Thus, the mean state for the shear flow modes $\bar{v}_k \equiv \langle \hat{v}_k \rangle$ can be computed by taking the statistical average on both sides of the above Equation (A2)

$$\langle \hat{v}_k \rangle(t) = -\hat{h}_k \int_0^t e^{\left(-d_k+ik^{-1}\beta\right)(t-s)} \Big\langle e^{-ikU[s,t]} U(s) \Big\rangle ds. \tag{A4}$$

Note that the process U is independent of the white noise for the small-scale modes $dW_k(t)$. If we take the Gaussian component (that is, the first two leading moments) of the mean flow process U, the equality using the identity from the characteristic function of a multidimensional Gaussian field expresses the coupled expectation as

$$\left\langle xe^{iy} \right\rangle = (\langle x \rangle + i\mathrm{cov}(x,y))e^{i\langle y \rangle - \frac{1}{2}\mathrm{var}(y)},$$

applied for correlated Gaussian random variables (x, y). Assuming the equilibrium steady state as $\bar{U} = \langle U \rangle$ and $r_U = \left\langle (U - \bar{U})^2 \right\rangle$, we can compute the steady-state correlation as

$$\left\langle e^{-ikU[s,t]} U(s) \right\rangle = \left(\bar{U} - ikr_U \int_0^{t-s} \mathcal{R}_U(\tau) d\tau \right) e^{-ik\bar{U}(t-s) - k^2 r_U J_U(t-s)}, \tag{A5}$$

with $J_U(t) = t * R_U = \int_0^t (t-\tau)\mathcal{R}_U(\tau)d\tau$. The autocorrelation function $\mathcal{R}_U(\tau) = \langle U(\tau)U(0) \rangle / r_U$ appears from the cross-correlations between $U(s)$ and $U[s,t]$. In the statistical steady state, the mean and variance of the time-dependent process $U[s,t]$ can be found directly as

$$\begin{aligned}
\langle U[s,t] \rangle &= \int_s^t \langle U(\tau) \rangle d\tau = \bar{U}(t-s), \\
\mathrm{var}(U[s,t]) &= \int_s^t \int_s^t \langle U'(\tau) U'(\tau') \rangle d\tau d\tau' \\
&= 2\int_0^{t-s} \int_s^{t-\tau} \langle U'(\tau) U'(0) \rangle d\tau d\tau' \\
&= 2r_U \int_0^{t-s} (t-s-\tau)\mathcal{R}_U(\tau) d\tau, \\
\mathrm{cov}(U(s), U[s,t]) &= \int_s^t \langle U'(s) U'(\tau) \rangle d\tau \\
&= \int_s^t \langle U'(0) U'(\tau - s) \rangle d\tau \\
&= r_U \int_0^{t-s} \mathcal{R}_U(\tau) d\tau.
\end{aligned}$$

Next, we compute the covariances between the flow mode $\hat{v}_k$ at different time instants. The second-order moment can be found in a similar fashion depending on the statistics and time correlations of the zonal mean flow statistics of U

$$\begin{aligned}
\langle \hat{v}_k(s) \hat{v}_k^*(t) \rangle =& \left\langle \int_0^s e^{\left(-d_k + ik^{-1}\beta\right)(s-s')} e^{-ikU[s',s]} \left[\hat{h}_k U(s') ds' + \sigma_{v,k} dW_k(s') \right] \right. \\
& \left. \cdot \int_0^t e^{\left(-d_k - ik^{-1}\beta\right)(t-s'')} e^{ikU[s'',t]} \left[\hat{h}_k^* U(s'') ds'' + \sigma_{v,k}^* dW_k^*(s'') \right] \right\rangle \\
=& e^{-d_k(t+s) - ik^{-1}\beta(t-s)} \left[\frac{|\sigma_{v,k}|^2}{2d_k} \left(e^{2d_k s} - 1 \right) \left\langle e^{ikU[s,t]} \right\rangle + \right. \\
& \left. |\hat{h}_k|^2 \int_0^t \int_0^s e^{d_k(s'+s'') - ik^{-1}\beta(s'-s'')} \left\langle e^{ik(U[s'',t] - U[s',t])} U(s') U(s'') \right\rangle ds' ds'' \right].
\end{aligned}$$

The last equality above uses the independence between the white noises and the mean flow process among non-intersected intervals. For further simplification, we can compute the equilibrium statistics as $s \to \infty, t \to \infty$ while keep the time lag $\tau = t - s$ finite. In addition, we assume stationarity in time for the zonal flow U, that is, all orders of moments are invariant in a shift in time. Then, the above correlation can be rearranged into a cleaner form as

$$\langle \hat{v}_k(0)\hat{v}_k^*(\tau)\rangle_\infty = \frac{e^{-d_k\tau - ik^{-1}\beta\tau}}{2d_k}\left[|\sigma_{v,k}|^2\left\langle e^{ikU[0,\tau]}\right\rangle + |\hat{h}_k|^2\int_{-\infty}^{\infty} e^{-(d_k+ik^{-1}\beta)r}\left\langle U(0)U(r)e^{-ikU[0,r]}\right\rangle dr\right]. \quad \text{(A6)}$$

From the last term above, we see that the third-order time correlation is required to compute the covariance between the shear flow modes. This formula is useful to compute the correlation in the shear flow modes directly from the statistics in the zonal flow state U. In particular, the equilibrium variance can be found as

$$r_{v,\infty} \equiv \left\langle |\hat{v}_k|^2\right\rangle_\infty = \frac{1}{2d_k}\left[|\sigma_{v,k}|^2 + 2|\hat{h}_k|^2\,\mathfrak{Re}\int_0^{\infty} e^{-(d_k+ik^{-1}\beta)r}\left\langle U(0)U(r)e^{-ikU[0,r]}\right\rangle dr\right]. \quad \text{(A7)}$$

Again, by using the characteristic function for coupled Gaussian fields

$$\left\langle xye^{iz}\right\rangle = [(\langle x\rangle + i\mathrm{cov}(x,z))(\langle y\rangle + i\mathrm{cov}(y,z)) + \mathrm{cov}(x,y)]e^{i\langle z\rangle - \frac{1}{2}\mathrm{var}(z)},$$

we can approximate the triad correlation as

$$\left\langle U(0)U(\tau)e^{ikU[0,\tau]}\right\rangle = \left[\mathcal{R}_U(\tau) - i\left(\bar{U} + ikr_U\int_0^\tau \mathcal{R}_U(\tau')d\tau'\right)^2\right]e^{ik\bar{U}\tau - k^2 r_U J_U(\tau)}. \quad \text{(A8)}$$

Substituting the above explicit expansion back into the variance Formula (A7), the equilibrium variance for the fluctuation modes can be rewritten as the statistics of the mean flow U

$$r_{v,\infty} = \frac{1}{2d_k}\left[|\sigma_{v,k}|^2 + 2|\hat{h}_k|^2\int_0^{\infty} e^{-d_k\tau - k^2 J_U(\tau)}(A_k(\tau)\cos\Theta_k\tau + B_k(\tau)\sin\Theta_k\tau)d\tau\right], \quad \text{(A9)}$$

with the coefficients

$$A_k = \bar{U}^2 + \mathcal{R}_U(\tau) - k^2 r_U^2\int_0^\tau \mathcal{R}_U(\tau')d\tau',\ B_k = 2k\bar{U}r_U\int_0^\tau \mathcal{R}_U(\tau')d\tau',\ \Theta_k = k\bar{U} + k^{-1}\beta.$$

Still notice that the expressions for the mean (A4) and variance (A7) only uses the Gaussian component of the random process U (that is, up to the second moments in computing the correlations); thus, errors could be introduced for extreme non-Gaussian situations.

Appendix A.2. Explicit Statistical Solutions for the Passive Tracer

Similarly to the above derivations for the advection flow solutions, we can also follow the previous steps to compute the corresponding solution for the passive tracer model (4) based on the topographic flows. Given one realization of the zonal mean flow solution U, we can first solve the tracer trajectory directly as

$$\hat{T}_k(t) = -\alpha\int_0^t e^{-\gamma_{T,k}(t-s) - ikU[s,t]}\hat{v}_k(s)ds, \quad U[s,t] = \int_s^t U(s')ds'$$

The shear flow field is generated from the small-scale state of the topographic model (A2). Combining the two formulas and rearranging the order in the integration, we obtain the trajectory solution for the tracer mode

$$\begin{aligned}\hat{T}_k &= \alpha\int_0^t e^{-\gamma_{T,k}(t-s) - ikU[s,t]}\int_0^s e^{(-d_{v,k}+ik^{-1}\beta)(s-s')}e^{-ikU[s',s]}\left[\hat{h}_k U(s')ds' + \sigma_{v,k}dW_k(s')\right]ds \\ &= \alpha e^{-\gamma_{T,k}t}\int_0^t dr\left[\hat{h}_k U(r)dr + \sigma_{v,k}dW_k(r)\right]e^{(d_k - ik^{-1}\beta)r}e^{-ikU[r,t]}\int_r^t ds e^{\gamma_{R,k}s} \quad \text{(A10)} \\ &= \frac{\alpha}{\gamma_{R,k}}\int_0^t\left[\hat{h}_k U(r)dr + \sigma_{v,k}dW_k(r)\right]\left[e^{\gamma_{R,k}(t-r)} - 1\right]e^{-\gamma_{T,k}(t-r) - ikU[r,t]}dr, \quad \text{(A11)}\end{aligned}$$

with $\gamma_{R,k} = \gamma_{T,k} - d_k + ik^{-1}\beta$. In the above formula, we make use of the property that the same wave speeds $kU[s,t]$ in the tracer and flow equations cancel each other out by exchanging the integration order. For the mean state of the tracer, we can also compute the simplified expression for the mean tracer state

$$\begin{aligned}\langle \hat{T}_k \rangle(t) &= \frac{\alpha \hat{h}_k}{\gamma_{R,k}} \int_0^t \Big[e^{\gamma_{R,k}(t-r)} - 1\Big] e^{-\gamma_{T,k}(t-r)} \Big\langle U(r) e^{-ikU[r,t]} \Big\rangle dr \\ &= -\frac{\alpha \hat{h}_k}{\gamma_{R,k}} \int_0^t \Big(e^{\left(-d_k + ik^{-1}\beta\right)s} - e^{-\gamma_{T,k}s}\Big) e^{-ik\bar{U}s - k^2 r_U J_U(s)} \left(\bar{U} - ikr_U \int_0^s \mathcal{R}_U(\tau) d\tau\right) ds.\end{aligned} \tag{A12}$$

Above, again, the cross-correlation can be computed based on the first two-order moments in (A5). The process $U(0\colon r)$ is independent of the white noise $W_k(r)$. Then, we can compute the second-order moment of the tracer

$$\Big\langle \big|\hat{T}_k\big|^2 \Big\rangle(t) = \frac{\alpha^2}{\big|\gamma_{R,k}\big|^2} e^{-2d_k t} \left[\frac{\sigma_{v,k}^2}{2\gamma_{T,k}} \Big(e^{2\gamma_{T,k}t} - 1\Big) + 2\big|\hat{h}_k\big|^2 \int_0^t dr \int_0^{t-r} d\tau \Big\langle U(r)U(r+\tau) e^{-ikU[r,r+\tau]} \Big\rangle\right]. \tag{A13}$$

Above, the first component is from the white noise forcing, and the second component is due to the contribution from the topographic effect. As a result, it will include the time correlation in the zonal mean process U. If we further assume that the process has reached the stationary state, the last time-lag correlation can be simplified using the expanded formula in (A8)

$$\Big\langle U(r)U(r+\tau) e^{-ikU[r,r+\tau]} \Big\rangle = \Big\langle U(0)U(\tau) e^{-ikU[0,\tau]} \Big\rangle.$$

Especially when there is no clear scale separation between the flow and tracer dynamics, the above formula could become very complicated. This also leads to the various very complicated statistical solutions shown in the flow and tracer fields.

Appendix B. More Details about the Neural Network Model Results

Here, we provide more details about the neural network configuration used for the tests in the main text as well as more discussions on the numerical performance.

Appendix B.1. Details about the Inner Connections in the LSTM Cell

The LSTM is designed to learn the multiscale temporal structures along time series. In the computational cell of the LSTM network, it consists of the basic building block as

$$\begin{aligned}\mathrm{LSTM}_t \coloneqq\ & f_t = \sigma_g\Big(W_f x_t + U_f h_{t-1} + V_f c_{t-1} + b_f\Big), \\ & i_t = \sigma_g(W_i x_t + U_i h_{t-1} + V_i c_{t-1} + b_i), \\ & c_t = f_t \otimes c_{t-1} + i_t \otimes \tanh(W_c x_t + U_c h_{t-1} + b_c), \\ & o_t = \sigma_g(W_o x_t + U_o h_{t-1} + V_o c_t + b_o), \\ & h_t = o_t \otimes \tanh(c_t).\end{aligned} \tag{A14}$$

Above, the $\sigma_g(x) = \frac{1}{1+e^{-x}}$ is the sigmoid activation function, and $\otimes$ represents the elementwise product. The model cell includes forget, input, and output gates f_t, i_t, o_t, and the cell state c_t. The gates are updated through a simple fully connected linear map with coefficients W, U, V, b. The hidden process $\{h_{t-m}, \cdots, h_{t-1}, h_t\}$ represents the time series of the unresolved processes after each single time step in the next immediate time instant. In the above structure (A14), we also add the previous cell state c_{t-1} to inform the gates information in f, i, o in their updates. It is shown this can also improve the model performance. The final output data are given by a final linear layer $y_t = W_y h_t$ mapping the hidden states h_t (usually with a larger dimension) back to the output state y_t.

Appendix B.2. Neural Network Algorithm for Training and Prediction in the Two-Mode Model

We summarize the neural network algorithm developed in the main text. In the numerical tests, we follow the benchmark model setup described in the following Algorithm A1.

Algorithm A1: The improved LSTM network with the fully connected inner structure (16) and (17) is used to learn the small-scale dynamics for $\hat{v}_k, \hat{T}_k$:

- Input data: $\{U, \Re\mathfrak{e}\hat{v}_k, \Im\mathfrak{m}\hat{v}_k, \Re\mathfrak{e}\hat{T}_k, \Im\mathfrak{m}\hat{T}_k\}$ for modeling dynamical updates of $\hat{v}_k$ and $\hat{T}_k$, evaluated at m previous time steps $t_{n-m+1}, t_{n-m+2}, \cdots, t_n$, with the time lag $t_n - t_{n-m+1} = m\Delta t \sim T_{\text{decorr}}$ determined by the decorrelation time;
- Output data: the unresolved modes $\{\Re\mathfrak{e}\hat{v}_k, \Im\mathfrak{m}\hat{v}_k, \Re\mathfrak{e}\hat{T}_k, \Im\mathfrak{m}\hat{T}_k\}$ at the next prediction time $t_{n+1} = t_n + \Delta t$.
  - The length of the LSTM nets are set as $m = 100$ steps for a time internal $T = 10$, that is, a sampling rate every $\Delta t = 0.1$;
  - The second half of the LSTM chain outputs, that is, $s = 50, \cdots, 100$, are measured in the loss function (24) with multistep model outputs;
  - Each LSTM cell has $s = 4$ connected inner stages (17), with the hidden state dimension $h = 50$ in the tests and a final linear map (15) to the output states.
- Solution for large-scale mean flow U is computed by combining the neural network model outputs $\{\hat{v}_k\}$ and integrating the explicit Formula (23) with the mid-point rule.

In the training stage, the parameters in the LSTM nets are optimized using the combined cost function (24) weighted from the autocorrelation function. In the prediction stage, the output states $\{U, \hat{v}_k, \hat{T}_k\}$ are computed for future time steps using the trained neural network by recursively feeding in the previous output data. The important hyperparameters to determine in the neural network include (i) the dimension of the hidden state h; (ii) the length of the input training sequence m; (iii) the inner stages of each LSTM cell s; and (iv) the push-forward time step Δt for the neural network prediction time. The standard model setup for training is listed in Table A1.

Table A1. Standard model hyperparameters for training the neural network model.

total training epochs	100
training batch size	100
starting learning rate	0.005
learning rate reduction rate	0.5
learning rate reduction at iteration step	50, 80
time step size between two measurements Δt	0.1
LSTM sequence length m	100
forward prediction steps in training n	10
hidden state size h	50
number of stages in LSTM cell s	4

Appendix B.3. Detailed Training and Prediction Results for Different Model Regimes

Here, we provide a more detailed characterization for the training and prediction accuracy in the flow and tracer states under different loss functions. In Section 5.1 of the main text, we have already shown the superiority of the mixed loss to have high skill in capturing the extreme events. Figure A1 gives a more detailed illustration for the development of errors separately in the flow and tracer states and in each wavenumber separately. The upper row shows the training iterations with the loss function value for the strong and weak topography cases $H = 10$ and $H = 1$. In both cases, the combined loss

(18c) gives the fastest convergence rate to the final saturated state in the loss error. Notice that the loss function takes different forms; thus, the absolute values are not comparable for model performance. As a further comment, the mixed metric allows an even larger training learning rate (such as lr $= 0.01$) to gain an even faster convergence rate while the other two metrics will diverge by starting with the larger learning rate. The training accuracy in the flow and tracer modes is compared in the middle and lower rows for the batch MSEs. The mixed loss function as a combined metric from the L_2 and relative entropy error provides the best overall prediction accuracy for both the flow and tracer modes among all the scales.

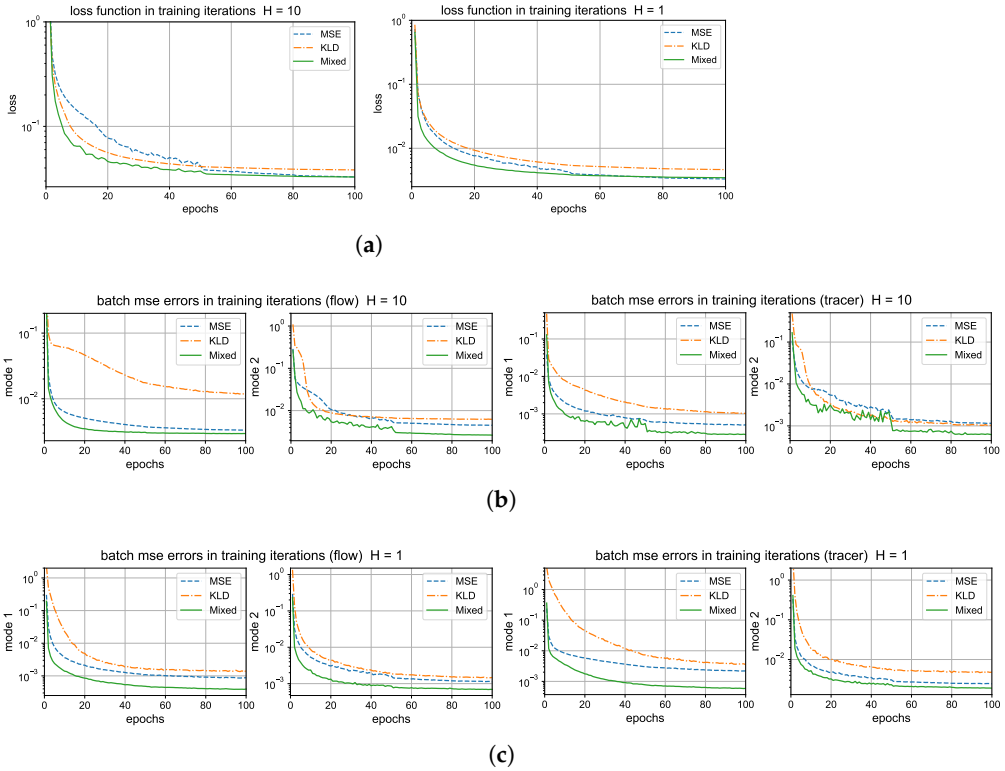
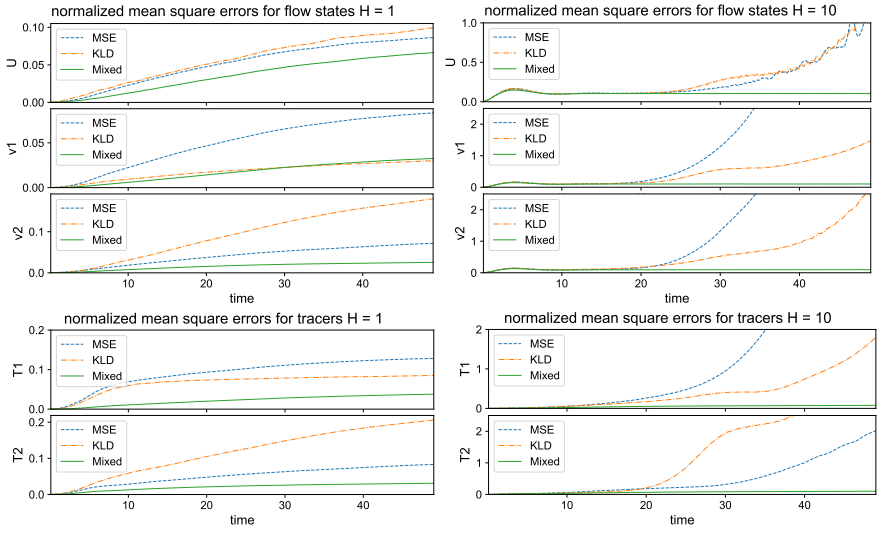


Figure A1. Training iterations of the full loss function value (**upper**) and training batch averaged mean square errors (**middle** and **lower**). Errors in each of the flow and tracer modes are compared under three different loss functions L for optimization: L_2 loss (MSE), KL divergence (KLD), and a mixed loss combining these two. Logarithmic scale is applied along the y coordinate. (**a**) Loss function with strong ($H = 10$, **left**) and weak ($H = 1$, **right**) topography. (**b**) Batch MSE for flow (**left**) and tracer (**right**) with strong topography $H = 10$. (**c**) Batch MSE for flow (**left**) and tracer (**right**) with weak topography $H = 1$.

A detailed comparison of the prediction errors in the flow and tracer modes in the prediction phase is shown in Figure A2. From the autocorrelation functions in Figure 2, the weak topography case $H = 1$ has a faster mixing rate. This implies that it is easier for the scheme to stay stable since the prediction is less dependent on the previous state and thus less prone to the input errors from the previous steps. On the other hand, in the strong topography case $H = 10$, the tracer modes have a much slower decay rates in the ACFs, inferring a more challenging case for model instability from the errors. Again, overall stability is gained from training by the mixed loss function. The other two MSE and KLD loss functions can produce reasonable predictions for the starting period of time, while the errors grow at a much faster rate without saturation, as we run the model for a longer time prediction.



(**a**) weak topography $H = 1$ (**b**) strong topography $H = 10$

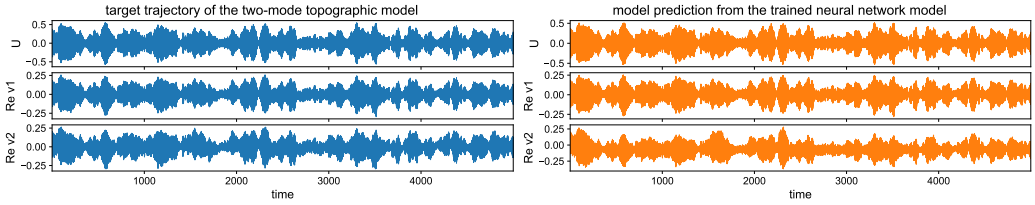
Figure A2. Comparison of prediction errors for models optimized with different loss functions in the two topography regimes $H = 1$ and $H = 10$. The normalized mean square errors are compared for both the flow states and tracer modes measured at recurrent predictions of 500 time steps up to $T = 50$.

Appendix B.4. Trajectory Prediction with Different Levels of Noises

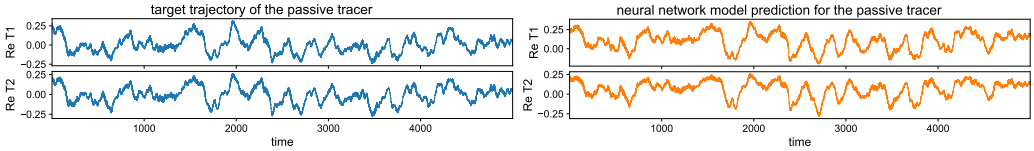
In comparison with the long-time predictions in Figures 7 and 8 for the topographic stresses $H = 1$ and $H = 10$ and with the white noise-forcing amplitude $\sigma_U = 10\sigma_0$ in the main text, here, we show the trajectory predictions in wider scenarios using the same trained neural network model while under various different prediction regimes. Two other different smaller or larger white noise-forcing amplitudes $\sigma_U = \sigma_0$ and $\sigma_U = 20\sigma_0$ are considered for the level of noises in the large-scale mean flow for U. In learning the unresolved small-scale dynamics from the data, the neural network is trained to learn the true dynamical structure. Thus, it requires that the trained model maintains the skill to recover the distinct solution structures among different statistical regimes.

Figures A3 and A4 show the prediction results with a much smaller white noise $\sigma_U = \sigma_0$ for the mean flow equation. In this case, the mean flow U has a much smaller amplitude, and the solutions in the small-scale velocity and tracer states become closer to Gaussian statistics with less frequent extreme events. From the comparison with the truth trajectories, we observe less non-Gaussian statistics far away from the highly non-Gaussian statistics shown in the main text. The neural network model maintains the high skill to capture the different representative solution structures among these distinctive statistical regimes. With both strong topography $H = 10$ and weak topography $H = 1$, the multiscale time and spatial structures as well as the change in solutions in the flow states and tracer modes are captured accurately with uniform performance among all the tested cases.

Figures A5 and A6 give the corresponding predictions with a much stronger white noise forcing $\sigma_U = 20\sigma_0$. In this case, instead, very strong non-Gaussian features and large values of extreme events appear in both the flow and tracer solutions. We observe much larger amplitudes and variance in the solutions. On the other hand, the training data do not contain such extreme cases for training the neural network model. This makes it a very challenging test case for the skill of the trained model to still capture the frequent extreme events in large amplitudes. Still, the representative structures and the locations of the extreme events in the solutions are recovered from the neural network model. Together with the uniform performance among various statistical regimes, it confirms that the neural network indeed learns the true model dynamical structures from the limited training dataset and is able to provide robust forecast against the high model instability and noise.

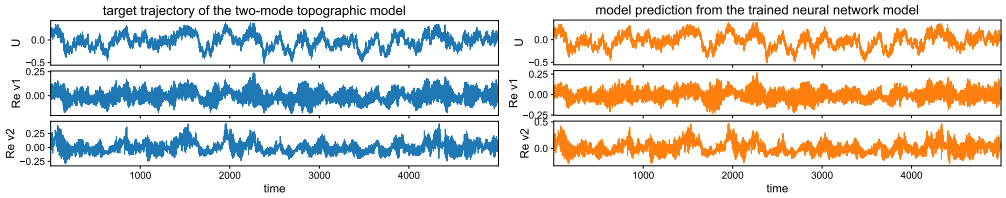


(**a**) trajectories of flow states

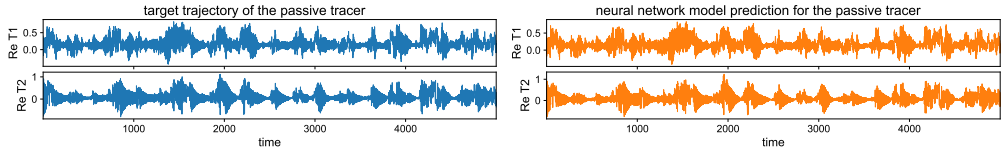


(**b**) trajectories of passive tracers

Figure A3. Long-time trajectory prediction of the flow and tracer solutions with weak white noise forcing $\sigma_U = \sigma_0$ in strong topographic regime $H = 10$. The same trained model in the main text is used for this different forcing regime with distinct statistics.

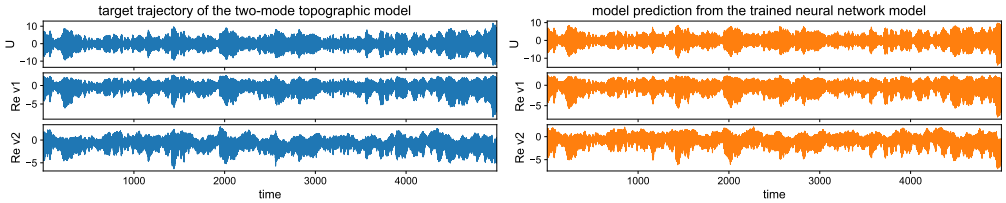


(**a**) trajectories of flow states

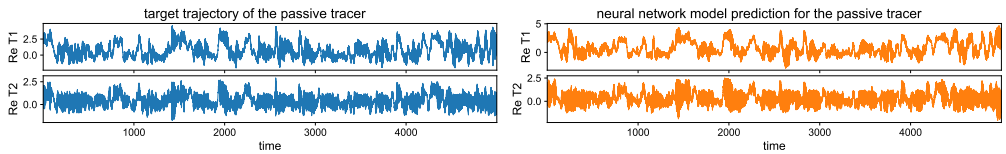


(**b**) trajectories of passive tracers

Figure A4. Long-time trajectory prediction of the flow and tracer solutions with weak white noise forcing $\sigma_U = \sigma_0$ in weak topographic regime $H = 1$. The same trained model in the main text is used for this different forcing regime with distinct statistics.

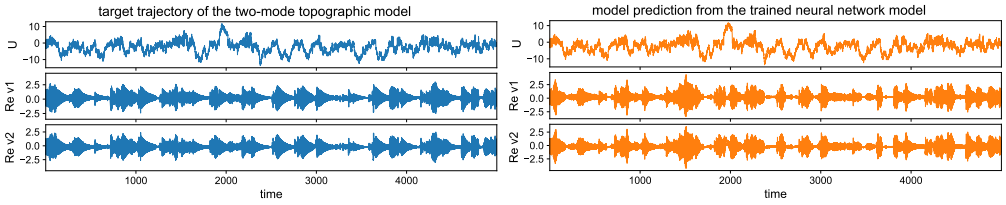


(**a**) trajectories of flow states

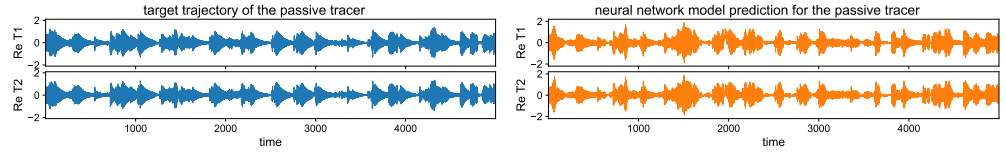


(**b**) trajectories of passive tracers

Figure A5. Long-time trajectory prediction of the flow and tracer solutions with strong white noise forcing $\sigma_U = 20\sigma_0$ in strong topographic regime $H = 10$. The same trained model in the main text is used for this different forcing regime with distinct statistics.



(**a**) trajectories of flow states



(**b**) trajectories of passive tracers

Figure A6. Long-time trajectory prediction of the flow and tracer solutions with strong white noise forcing $\sigma_U = 20\sigma_0$ in weak topographic regime $H = 1$. The same trained model in the main text is used for this different forcing regime with distinct statistics.

References

1. Majda, A.J. *Introduction to Turbulent Dynamical Systems in Complex Systems*; Springer: Berlin/Heidelberg, Germany, 2016.
2. Mohamad, M.A.; Sapsis, T.P. Sequential sampling strategy for extreme event statistics in nonlinear dynamical systems. *Proc. Natl. Acad. Sci. USA* **2018**, *115*, 11138–11143. [CrossRef] [PubMed]
3. Dematteis, G.; Grafke, T.; Vanden-Eijnden, E. Rogue waves and large deviations in deep sea. *Proc. Natl. Acad. Sci. USA* **2018**, *115*, 855–860. [CrossRef] [PubMed]
4. Bolles, C.T.; Speer, K.; Moore, M. Anomalous wave statistics induced by abrupt depth change. *Phys. Rev. Fluids* **2019**, *4*, 011801. [CrossRef]
5. Sapsis, T.P. Statistics of Extreme Events in Fluid Flows and Waves. *Annu. Rev. Fluid Mech.* **2020**, *53*, 85–111. [CrossRef]
6. Majda, A.J.; Kramer, P.R. Simplified models for turbulent diffusion: Theory, numerical modelling, and physical phenomena. *Phys. Rep.* **1999**, *314*, 237–574. [CrossRef]
7. Reich, S.; Cotter, C. *Probabilistic Forecasting and Bayesian Data Assimilation*; Cambridge University Press: Cambridge, UK, 2015.
8. Nazarenko, S.; Lukaschuk, S. Wave turbulence on water surface. *Annu. Rev. Condens. Matter Phys.* **2016**, *7*, 61–88. [CrossRef]
9. Farazmand, M.; Sapsis, T.P. A variational approach to probing extreme events in turbulent dynamical systems. *Sci. Adv.* **2017**, *3*, e1701533. [CrossRef]
10. Tong, S.; Vanden-Eijnden, E.; Stadler, G. Extreme event probability estimation using PDE-constrained optimization and large deviation theory, with application to tsunamis. *Commun. Appl. Math. Comput. Sci.* **2021**, *16*, 181–225. [CrossRef]
11. Frisch, U. *Turbulence: The Legacy of an Kolmogorov*; Cambridge University Press: Cambridge, UK, 1995.
12. Tao, W.K.; Chern, J.D.; Atlas, R.; Randall, D.; Khairoutdinov, M.; Li, J.L.; Waliser, D.E.; Hou, A.; Lin, X.; Peters-Lidard, C. A multiscale modeling system: Developments, applications, and critical issues. *Bull. Am. Meteorol. Soc.* **2009**, *90*, 515–534. [CrossRef]
13. Lucarini, V.; Faranda, D.; de Freitas, J.M.M.; Holland, M.; Kuna, T.; Nicol, M.; Todd, M.; Vaienti, S. *Extremes and Recurrence in Dynamical Systems*; John Wiley & Sons: Hoboken, NJ, USA, 2016.
14. Köppen, M. The curse of dimensionality. In Proceedings of the 5th Online World Conference on Soft Computing in Industrial Applications (WSC5), Online, 4–18 September 2000; Volume 1, pp. 4–8.
15. Daum, F.; Huang, J. Curse of dimensionality and particle filters. In Proceedings of the 2003 IEEE Aerospace Conference Proceedings (Cat. No. 03TH8652), Big Sky, MT, USA, 8–15 March 2003; Volume 4, pp. 4_1979–4_1993.
16. Rudy, S.H.; Kutz, J.N.; Brunton, S.L. Deep learning of dynamics and signal-noise decomposition with time-stepping constraints. *J. Comput. Phys.* **2019**, *396*, 483–506. [CrossRef]
17. Vlachas, P.R.; Pathak, J.; Hunt, B.R.; Sapsis, T.P.; Girvan, M.; Ott, E.; Koumoutsakos, P. Backpropagation algorithms and reservoir computing in recurrent neural networks for the forecasting of complex spatiotemporal dynamics. *Neural Netw.* **2020**, *126*, 191–217. [CrossRef] [PubMed]
18. Chattopadhyay, A.; Subel, A.; Hassanzadeh, P. Data-driven super-parameterization using deep learning: Experimentation with multiscale Lorenz 96 systems and transfer learning. *J. Adv. Model. Earth Syst.* **2020**, *12*, e2020MS002084. [CrossRef]
19. Harlim, J.; Jiang, S.W.; Liang, S.; Yang, H. Machine learning for prediction with missing dynamics. *J. Comput. Phys.* **2020**, 109922. [CrossRef]
20. Gamahara, M.; Hattori, Y. Searching for turbulence models by artificial neural network. *Phys. Rev. Fluids* **2017**, *2*, 054604. [CrossRef]
21. Maulik, R.; San, O.; Rasheed, A.; Vedula, P. Subgrid modelling for two-dimensional turbulence using neural networks. *J. Fluid Mech.* **2019**, *858*, 122–144. [CrossRef]
22. Singh, A.P.; Medida, S.; Duraisamy, K. Machine-learning-augmented predictive modeling of turbulent separated flows over airfoils. *AIAA J.* **2017**, *55*, 2215–2227. [CrossRef]

23. Harlim, J.; Mahdi, A.; Majda, A.J. An ensemble Kalman filter for statistical estimation of physics constrained nonlinear regression models. *J. Comput. Phys.* **2014**, *257*, 782–812. [CrossRef]
24. Majda, A.J.; Qi, D. Strategies for reduced-order models for predicting the statistical responses and uncertainty quantification in complex turbulent dynamical systems. *SIAM Rev.* **2018**, *60*, 491–549. [CrossRef]
25. Bolton, T.; Zanna, L. Applications of deep learning to ocean data inference and subgrid parameterization. *J. Adv. Model. Earth Syst.* **2019**, *11*, 376–399. [CrossRef]
26. Qi, D.; Harlim, J. Machine learning-based statistical closure models for turbulent dynamical systems. *Philos. Trans. R. Soc. A* **2022**, *380*, 20210205. [CrossRef]
27. Qi, D.; Harlim, J. A data-driven statistical-stochastic surrogate modeling strategy for complex nonlinear non-stationary dynamics. *J. Comput. Phys.* **2023**, *485*, 112085. [CrossRef]
28. McDermott, P.L.; Wikle, C.K. An ensemble quadratic echo state network for non-linear spatio-temporal forecasting. *Stat* **2017**, *6*, 315–330. [CrossRef]
29. Qi, D.; Majda, A.J. Predicting extreme events for passive scalar turbulence in two-layer baroclinic flows through reduced-order stochastic models. *Commun. Math. Sci.* **2018**, *16*, 17–51. [CrossRef]
30. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **2019**, *378*, 686–707. [CrossRef]
31. Qi, D.; Majda, A.J. Using machine learning to predict extreme events in complex systems. *Proc. Natl. Acad. Sci. USA* **2020**, *117*, 52–59. [CrossRef]
32. Chen, N.; Qi, D. A physics-informed data-driven algorithm for ensemble forecast of complex turbulent systems. *Appl. Math. Comput.* **2024**, *466*, 128480. [CrossRef]
33. Pedlosky, J. *Geophysical Fluid Dynamics*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2013.
34. Qi, D.; Majda, A.J. Low-dimensional reduced-order models for statistical response and uncertainty quantification: Barotropic turbulence with topography. *Phys. D Nonlinear Phenom.* **2017**, *343*, 7–27. [CrossRef]
35. Qi, D.; Majda, A.J. Rigorous statistical bounds in uncertainty quantification for one-layer turbulent geophysical flows. *J. Nonlinear Sci.* **2018**, *28*, 1709–1761. [CrossRef]
36. Weeks, E.R.; Tian, Y.; Urbach, J.; Ide, K.; Swinney, H.L.; Ghil, M. Transitions between blocked and zonal flows in a rotating annulus with topography. *Science* **1997**, *278*, 1598–1601. [CrossRef]
37. Majda, A.J.; Moore, M.; Qi, D. Statistical dynamical model to predict extreme events and anomalous features in shallow water waves with abrupt depth change. *Proc. Natl. Acad. Sci. USA* **2019**, *116*, 3982–3987. [CrossRef]
38. Hu, R.; Edwards, T.K.; Smith, L.M.; Stechmann, S.N. Initial investigations of precipitating quasi-geostrophic turbulence with phase changes. *Res. Math. Sci.* **2021**, *8*, 6. [CrossRef]
39. Moore, N.J.; Bolles, C.T.; Majda, A.J.; Qi, D. Anomalous waves triggered by abrupt depth changes: Laboratory experiments and truncated KdV statistical mechanics. *J. Nonlinear Sci.* **2020**, *30*, 3235–3263. [CrossRef]
40. Liptser, R.S.; Shiryaev, A.N. *Statistics of Random Processes II: Applications*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2013; Volume 6.
41. Chen, N.; Majda, A.J. Beating the curse of dimension with accurate statistics for the Fokker–Planck equation in complex turbulent systems. *Proc. Natl. Acad. Sci. USA* **2017**, *114*, 12864–12869. [CrossRef]
42. Chen, N.; Majda, A.J. Conditional Gaussian systems for multiscale nonlinear stochastic systems: Prediction, state estimation and uncertainty quantification. *Entropy* **2018**, *20*, 509. [CrossRef]
43. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
44. Jozefowicz, R.; Zaremba, W.; Sutskever, I. An empirical exploration of recurrent network architectures. In Proceedings of the International Conference on Machine Learning, Lille, France, 7–9 July 2015; pp. 2342–2350.
45. Kullback, S. *Information Theory and Statistics*; Courier Corporation: Chelmsford, MA, 1997.
46. Majda, A.; Abramov, R.V.; Grote, M.J. *Information Theory and Stochastics for Multiscale Nonlinear Systems*; American Mathematical Society: Providence, RI, USA, 2005; Volume 25.
47. Bianchi, F.M.; Maiorino, E.; Kampffmeyer, M.C.; Rizzi, A.; Jenssen, R. An overview and comparative analysis of recurrent neural networks for short term load forecasting. *arXiv* **2017**, arXiv:1705.04378.
48. Lesieur, M. *Turbulence in Fluids: Stochastic and Numerical Modelling*; Nijhoff: Boston, MA, USA, 1987; Volume 488.
49. Ahmed, F.; Neelin, J.D. Explaining scales and statistics of tropical precipitation clusters with a stochastic model. *J. Atmos. Sci.* **2019**, *76*, 3063–3087. [CrossRef]
50. Majda, A.J.; Chen, N. Model error, information barriers, state estimation and prediction in complex multiscale systems. *Entropy* **2018**, *20*, 644. [CrossRef]
51. Majda, A.J.; Gershgorin, B. Elementary models for turbulent diffusion with complex physical features: Eddy diffusivity, spectrum and intermittency. *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.* **2013**, *371*, 20120184. [CrossRef]
52. Majda, A.J.; Qi, D. Linear and nonlinear statistical response theories with prototype applications to sensitivity analysis and statistical control of complex turbulent dynamical systems. *Chaos Interdiscip. J. Nonlinear Sci.* **2019**, *29*, 103131. [CrossRef]
53. Majda, A.J.; Qi, D. Effective control of complex turbulent dynamical systems through statistical functionals. *Proc. Natl. Acad. Sci. USA* **2017**, *114*, 5571–5576. [CrossRef] [PubMed]
54. Müller, P.; Garrett, C.; Osborne, A. Rogue waves. *Oceanography* **2005**, *18*, 66. [CrossRef]

55. Majda, A.J.; Harlim, J. *Filtering Complex Turbulent Systems*; Cambridge University Press: Cambridge, UK, 2012.
56. Covington, J.; Qi, D.; Chen, N. Effective Statistical Control Strategies for Complex Turbulent Dynamical Systems. *Proc. R. Soc. A* **2023**, *479*, 20230546. [CrossRef]
57. Bach, E.; Colonius, T.; Scherl, I.; Stuart, A. Filtering dynamical systems using observations of statistics. *Chaos Interdiscip. J. Nonlinear Sci.* **2024**, *34*, 033119. [CrossRef] [PubMed]