Journal of Physics:
Complexity

PAPER • **OPEN ACCESS**

The high-frequency and rare events barriers to neural closures of atmospheric dynamics

To cite this article: Mickaël D Chekroun *et al* 2024 *J. Phys. Complex.* **5** 025004

IOP Publishing J. Phys. Complex. 5 (2024) 025004 https://doi.org/10.1088/2632-072X/ad3e59

Journal of Physics: Complexity

OPEN ACCESS

RECEIVED
20 September 2023

REVISED
18 March 2024

ACCEPTED FOR PUBLICATION
12 April 2024

PUBLISHED
22 April 2024

PAPER

The high-frequency and rare events barriers to neural closures of atmospheric dynamics

Mickaël D Chekroun[1,2,*], Honghu Liu[3], Kaushik Srinivasan[4] and James C McWilliams[5]

[1] Department of Atmospheric and Oceanic Sciences, University of California, Los Angeles, CA, United States of America
[2] Department of Earth and Planetary Sciences, Weizmann Institute of Science, Rehovot 76100, Israel
[3] Department of Mathematics, Virginia Tech, Blacksburg, VA 24061, United States of America
[4] Department of Atmospheric and Oceanic Sciences, University of California, Los Angeles, CA 90095-1565, United States of America
[5] Department of Atmospheric and Oceanic Sciences and Institute of Geophysics and Planetary Physics, University of California, Los Angeles, CA 90095-1565, United States of America
[*] Author to whom any correspondence should be addressed.

E-mail: mchekroun@atmos.ucla.edu

Keywords: neural networks, parameterization, slow-fast systems, atmospheric dynamics

Abstract

Recent years have seen a surge in interest for leveraging neural networks to parameterize small-scale or fast processes in climate and turbulence models. In this short paper, we point out two fundamental issues in this endeavor. The first concerns the difficulties neural networks may experience in capturing rare events due to limitations in how data is sampled. The second arises from the inherent multiscale nature of these systems. They combine high-frequency components (like inertia-gravity waves) with slower, evolving processes (geostrophic motion). This multiscale nature creates a significant hurdle for neural network closures. To illustrate these challenges, we focus on the atmospheric 1980 Lorenz model, a simplified version of the Primitive Equations that drive climate models. This model serves as a compelling example because it captures the essence of these difficulties.

1. Introduction

Atmospheric and oceanic flows constrained by Earth's rotation satisfy an approximately geostrophic momentum balance on larger scales, associated with slow evolution on time scales of days, but they also exhibit fast inertia-gravity wave oscillations. The problems of identifying the slow component (e.g. for weather forecast initialization [1–4]) and of characterizing slow-fast interactions are central to geophysical fluid dynamics, and the former was first coined as a slow manifold problem by Leith [5]. The L63 model [6] famous for its chaotic strange attractor is a paradigm for the geostrophic component, while the L80 model [7] is its paradigmatic successor both for the generalization of slow balance and for slow-fast coupling.

The explosion of machine learning (ML) methods provides an unprecedented opportunity to analyze data and accelerate scientific progress. A variety of ML methods have emerged for solving dynamical systems [8–10], predicting [11] or discovering [12] them from data. For larger scale problems, much effort has been devoted lately to the learning of neural subgrid-scale parameterizations in coarse-resolution climate models [13] but yet the lack of interpretability and reliability prevents a widespread adoption so far [14, 15].

In parallel, the learning of stable neural parameterizations of small scales or neglected variables has progressed remarkably for the closure of fluid models in turbulent regimes such as the forced Navier–Stokes equations or quasi-geostrophic flow models on a β-plane; see [16–22].

While neural networks show promise for climate modeling, the full Primitive Equations (PE) remain a challenge. This study identifies potential hurdles in achieving efficient neural closures for PE. We leverage the L80 model, a simplified version of the PE, as a illustrative example to highlight these fundamental issues.

In that respect, the L80 model exhibits a fascinating dynamical transition. For small Rossby numbers, its solutions evolve slowly over time and remain entirely slow, dominated by large-scale Rossby waves [23].

However, as the Rossby number increases, faster oscillations become superimposed on these slow background motions [24, 25]. This spontaneous emergence of high-frequency components, linked to inertia-gravity waves (IGWs) riding on the slower geostrophic flow, significantly complicates the closure problem in atmospheric models [25, 26].

Multiscale dynamics, characterized by the intricate interplay of slow and fast processes without clear separation, are not unique to the L80 model. Similar regimes have been observed in fully resolved Primitive equation (PE) models, where fronts and jets generate complex multiscale interactions [27, 28] as well as in cloud-resolving models, where large-scale convectively coupled gravity waves emerge spontaneously [29]. Tropical convection regions, where organized activity produces gravity waves with a broad spectrum, ranging from 10 km to over 1000 km wavelengths [30] provide another instance of such multiscale dynamics. Finally, inertia-gravity waves have also been observed in continental shallow convection, where they contribute to organized mesoscale patterns over vegetated areas [31].

Inertia-gravity waves can hold surprising amounts of energy even at large scales. For example, Rocha *et al* [32] found that IGWs contribute nearly half of the near-surface kinetic energy in specific ocean regions at scales ranging from 10 to 40 km. This overlap between wave and turbulence scales in geophysical kinetic energy spectra creates a challenge: perturbation methods like Wentzel-Kramers-Brillouin (WKB) [33] become inapplicable across all scales [34].

Such regimes where slow and fast dynamics overlap were shown to constitute critical challenges for closure methods in the L80 model. Solutions in these regimes blend slow background motion with sudden bursts of IGWs carrying a significant portion of the total energy. These 'high-low frequency (HLF)' solutions disrupt the expected slaving relationships satisfied at lower Rossby numbers, leading to a major breakdown in closure techniques relying on a separation between the slow and fast variables [25].

A recent study by [26] proposes a promising solution to closure problems in such HLF regimes without timescale separation and where slow Rossby variables are influenced by high-frequency waves. This approach hinges on the Balance equation (BE) [23, 35] as rooted in the works of Monin [36], Charney and Bolin [1, 37], and Lorenz [38], which allows for a nonlinear separation of variables. As demonstrated in [26], the BE isolates, for large Rossby numbers, the fast, non-geostrophic component of the flow as residual dynamics off the BE manifold. Building on the BE separation, it was shown in [26] that this fast motion can be effectively parameterized using networks of nonlinear stochastic oscillators (NSOs). These NSOs are designed to match the characteristic patterns of variability observed in the fast motion, leveraging the concept of resonances discussed in [39–41]. The resulting stochastic closure shows then high-accuracy skills in reproducing the multiscale dynamics.

This work emphasizes the limitations of (standard) neural networks (alone) for achieving such accurate closures for HLF regimes, highlighting their struggle to simultaneously capture the slow, balanced motion while restoring the high-frequency oscillations. Section 2 discusses the limitations of neural networks for parameterizing the L80 model's slow motion, emphasizing in particular their sensitivity to rare event statistics (section 3). Section 4 highlights the fundamental challenges faced by neural networks in capturing both the slow and high-frequency content of the L80 solutions, ultimately hindering accurate closure.

2. Learning slow neural closure: sensitivity

The L80 model, obtained by Lorenz in [7] as a nine-dimensional truncation of the PE onto three Fourier modes with low wavenumbers, can be written as:

$$\begin{aligned}
a_i \frac{\mathrm{d}x_i}{\mathrm{d}t} &= -\nu_0 a_i^2 x_i - c\left(a_i - a_k\right) x_j y_k + c\left(a_i - a_j\right) y_j x_k + a_i b_i x_j x_k - 2c^2 y_j y_k + a_i\left(y_i - z_i\right),\\
a_i \frac{\mathrm{d}y_i}{\mathrm{d}t} &= -a_k b_k x_j y_k - a_j b_j y_j x_k + c\left(a_k - a_j\right) y_j y_k - a_i x_i - \nu_0 a_i^2 y_i,\\
\frac{\mathrm{d}z_i}{\mathrm{d}t} &= g_0 a_i x_i - b_k x_j\left(z_k - h_k\right) - b_j\left(z_j - h_j\right) x_k + c y_j\left(z_k - h_k\right) - c\left(z_j - h_j\right) y_k - \kappa_0 a_i z_i + F_i,
\end{aligned} \tag{1}$$

whose model parameters are described in [7, 25].

The above equations are written for each cyclic permutation of the set of indices $(1,2,3)$, namely, for (i,j,k) in $\{(1,2,3),(2,3,1),(3,1,2)\}$. The model variables $(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$ are amplitudes for the divergent velocity potential, stream-function, and dynamic height, respectively.

In this model, the square root of the constant forcing F_1 can be interpreted as the Rossby number; see [23] and [25, equation (2.4)]. Transitions to chaos occur as the Rossby number Ro is increased [23, 25]. As mentioned above, at small Rossby numbers, the solutions to the L80 model are dominated by Rossby waves and thus remain entirely slow for all time. As identified in [25], when the Rossby number is further increased beyond a critical Rossby number Ro^*, fast IGW oscillations emerge spontaneously and are superimposed on

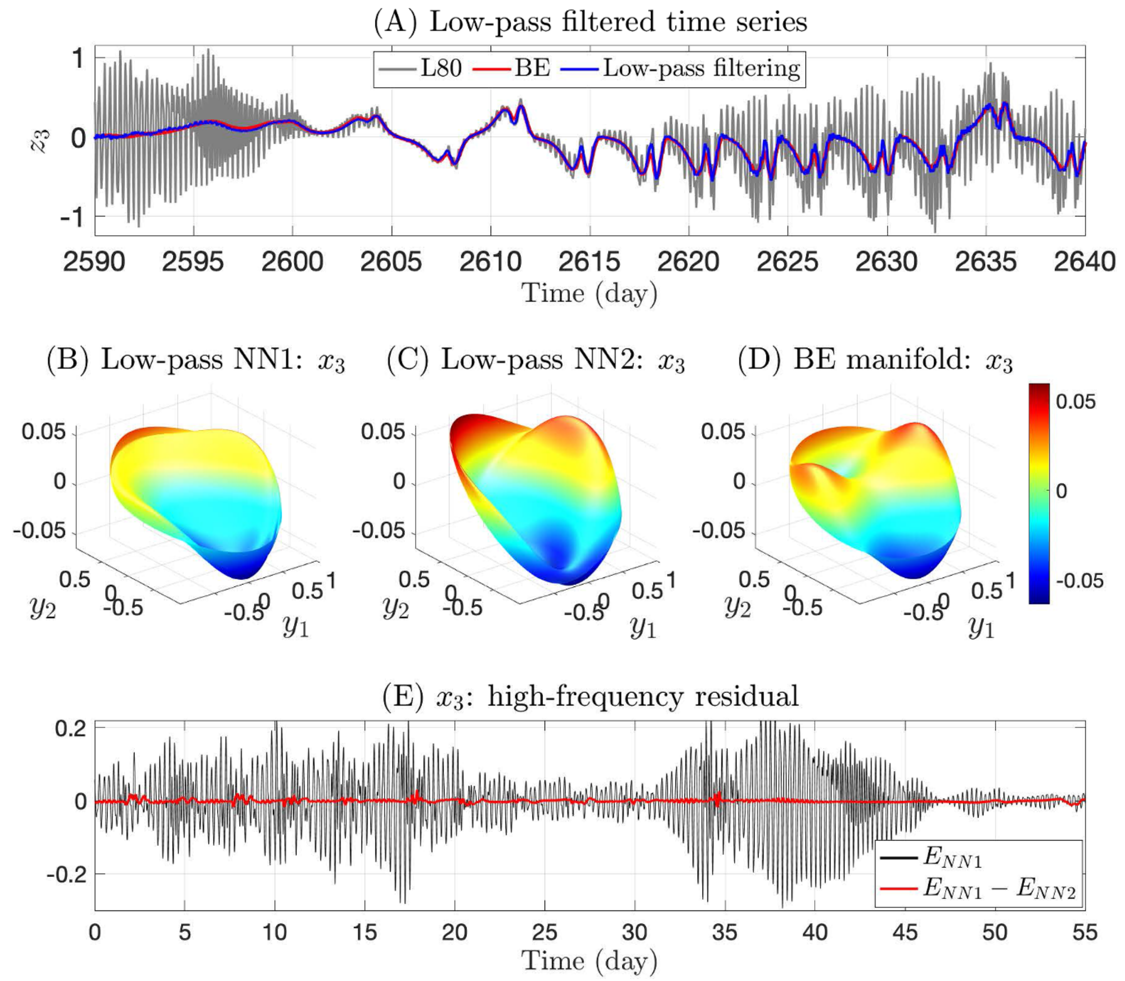

Figure 1. Panel (A): Illustration, for the z_3-variable, of the BE manifold's ability in capturing the L80 model's slow motion. See [23] and appendix B for a derivation. Panels (B) and (C): Neural parameterizations $\boldsymbol{\mathcal{X}}^3_{\boldsymbol{\theta}}$ for the x_3-variable, as learnt through random selection (NN_1)/predefined selection (NN_2). Visualized here as mappings from (y_1, y_2) onto the unit sphere in $\mathbb{R}^3$. Panels (D): Same visualization adopted for the BE manifold. Panel (E): High-frequency residual $E_{NN_1}(t)$ for x_3 (black) given by (5) and its difference with $E_{NN_2}(t)$ (red).

the slow component of the solutions. For such regimes, the aforementioned BE manifold on which balanced solutions lie [23, 25, 35] is no longer able to parameterize fully the L80 dynamics since a substantial portion of it, associated with the IGWs, evolves transversally to the BE manifold [26, figure 3]. These regimes with energetic bursts of IGWs lie beyond the parameter range explored by Lorenz in his original 1980 article [7] and beyond other regimes with exponential smallness of IGW amplitudes as studied in subsequent Lorenz 86 models [42–45] and the full primitive equations [46] at smaller Rossby numbers [47].

The HLF solutions considered in this study are obtained for such a critical parameter regime where $Ro > Ro^*$. They correspond to those of [26, figure 7]; see appendix A for details. We first analyze the ability of neural parameterizations to learn the slow motion of the L80 dynamics in the HLF regime. To do so, we preprocess the target variables $\boldsymbol{x}$ and $\boldsymbol{z}$ to be parameterized by applying a low-pass filter in order to extract the slow motion. In that respect, a simple moving average is adopted with a window size equal to T_{GW}, the dominant period of the gravity waves. The results are shown in figure 1(A) for the z_3-variable for which we observe that the low-pass filtered solution almost coincides this way with the BE parameterization $\boldsymbol{z}_{BE}(t) = G(\boldsymbol{y}(t))$ with $\boldsymbol{y}(t)$ denoting the $\boldsymbol{y}$-component of the HLF solution to the L80 model.

The L80 model has an inherent structure that can be exploited for closure. Studies have shown that the BE manifold, constructed in two steps (parameterizing $\boldsymbol{z}$ as a function of $\boldsymbol{y}$ and then $\boldsymbol{x}$ as a function of $\boldsymbol{y}$ and the parameterized $\boldsymbol{z}$), achieves excellent closure across various parameter regimes [23] (see appendix B and [25] for details). To leverage this existing knowledge and facilitate comparison with the BE manifold, we design our neural network parameterizations with a similar structure. Specifically, we first learn a feedforward neural network (multilayer perceptron, MLP) denoted as $\boldsymbol{\mathcal{Z}}_{\boldsymbol{\theta}}$, which takes the (unfiltered) variable $\boldsymbol{y}$ as input and predicts the filtered $\boldsymbol{z}$-variable (equation (2)). Then, we train a second MLP, $\boldsymbol{\mathcal{X}}_{\boldsymbol{\theta}}$, that takes both $\boldsymbol{y}$ and the output of $\boldsymbol{\mathcal{Z}}_{\boldsymbol{\theta}}$, $(\boldsymbol{y}, \boldsymbol{\mathcal{Z}}_{\boldsymbol{\theta}}(\boldsymbol{y}))$, as input in order to predict the filtered $\boldsymbol{x}$-variable.

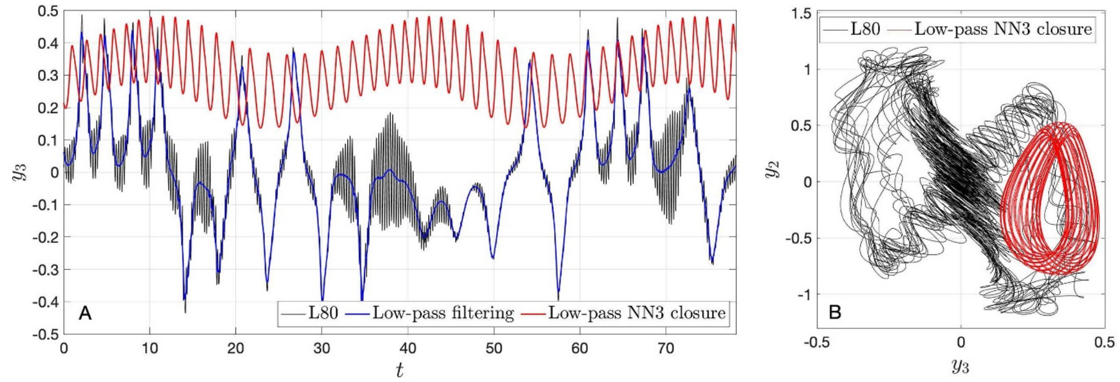

Figure 2. False quasiperiodicity produced by a slow neural closure. Here, the slow neural closure equation (6) is driven by $\mathcal{Z}_{\boldsymbol{\theta}}$ and $\mathcal{X}_{\boldsymbol{\theta}}$ that are trained using a low-pass filtered version of $\boldsymbol{y}(t)$ (blue curve in panel (A)) unlike the closure defined in equation (6) where the slow neural closure is trained using the unfiltered $\boldsymbol{y}$-variable.

The structure of our MLPs is standard. Each neural parameterization, e.g. $\boldsymbol{z}$ in terms of $\boldsymbol{y}$, is sought by means of an MLP with L hidden layers of p neurons each. It boils down to find

$$\mathcal{Z}_{\boldsymbol{\theta}}\left(\boldsymbol{y}\right) = \mathcal{N}_{\text{out}} \circ \mathcal{N}_L \circ \cdots \circ \mathcal{N}_1 \circ \mathcal{N}_{\text{in}}\left(\boldsymbol{y}\right), \tag{2}$$

in which $\mathcal{N}_{\text{in}}$ (resp. $\mathcal{N}_{\text{out}}$) constitutes the input (resp. output) layer, while $\mathcal{N}_k$ is a mapping from $\mathbb{R}^p$ (the space of neurons) onto itself, given by $\mathcal{N}_k(\xi) = \Psi_k(\mathbf{W}_k\xi + \mathbf{b}_k)$ (ξ in $\mathbb{R}^p$) where Ψ_k is a p-dimensional elementwise function, i.e. a function that applies a (scalar) activation function to each of its inputs individually, and the $\mathbf{W}_k$ and $\mathbf{b}_k$ denote respectively the weight matrices and bias vectors to be learnt. In (2), the subscript $\boldsymbol{\theta}$ denotes the collection of these parameters. In this work, the nonlinear activation function is a simple tanh function, and the input and output layers consist just of linear normalization and reversal operations. It turns out that NNs with one hidden layer and 5 neurons are sufficient to obtain loss functions with a small residual; see table 1.

Based on our approach paralleling the BE manifold construction, we learn our neural parameterizations for the L80 model, through the following consecutive minimizations. First, given a discrete set of time instants t_j, one minimizes

$$\mathcal{L}_{\boldsymbol{\theta}}\left(\boldsymbol{z};\boldsymbol{y}\right) = \sum_j \left\|\boldsymbol{z}_{t_j} - \mathcal{Z}_{\boldsymbol{\theta}}\left(\boldsymbol{y}_{t_j}\right)\right\|^2, \tag{3}$$

in which $\boldsymbol{z}$ is filtered (in time) while $\boldsymbol{y}$ is not, followed by the minimization of

$$\mathcal{L}_{\boldsymbol{\theta}}\left(\boldsymbol{x};\left(\boldsymbol{y},\mathcal{Z}_{\boldsymbol{\theta}_1^*}\left(\boldsymbol{y}\right)\right)\right) = \sum_j \left\|\boldsymbol{x}_{t_j} - \mathcal{X}_{\boldsymbol{\theta}}\left(\boldsymbol{y}_{t_j}, \mathcal{Z}_{\boldsymbol{\theta}_1^*}\left(\boldsymbol{y}_{t_j}\right)\right)\right\|^2, \tag{4}$$

with $\boldsymbol{x}$ filtered and where $\mathcal{Z}_{\boldsymbol{\theta}_1^*}$ denotes the optimal parameterization obtained after minimization of (3).

We emphasize the importance of including the unfiltered $\boldsymbol{y}$-component of the HLF solution in the training data, even though it contains rapid oscillations. This unfiltered data is indeed crucial for the network to learn a proper representation of the slow motion. If we replace the unfiltered $\boldsymbol{y}$-component with a filtered version (like the blue curve for y_3 in figure 2(A), the resulting closure fails. It produces an unrealistic quasi-periodic behavior that does not resemble even the L80 model's quasi-periodic behaviors documented in [23] for nearby parameter settings (see red curves in figure 2).

To assess whether a neural parameterization is successful in capturing the slow motion, we evaluate also the following *high-frequency (HF) residual*

$$E^j_{NN}\left(t\right) = x_j\left(t\right) - \mathcal{X}^j_{\boldsymbol{\theta}_2^*}\left(\boldsymbol{y}\left(t\right), \mathcal{Z}_{\boldsymbol{\theta}_1^*}\left(\boldsymbol{y}\left(t\right)\right)\right), \tag{5}$$

in which the $x_j(t)$ and $\boldsymbol{y}(t)$ are *both* unfiltered. For an NN with small residual, $E^j_{NN}(t)$ is typically void of slow oscillations (see figure 1(E) with mean $\langle E^j_{NN}\rangle \approx 0$ for each $1 \leqslant j \leqslant 3$.

Table 1. Loss function evaluations for two neural networks. The loss functions (3) for $\boldsymbol{z}$ and (4) for $\boldsymbol{x}$, are minimized using two neural networks, NN_1 and NN_2 providing each a parameterization $(\boldsymbol{\mathcal{Z}}_{\boldsymbol{\theta}}, \boldsymbol{\mathcal{X}}_{\boldsymbol{\theta}})$, differing only in the way the training, validation, and testing sets are selected. In each case, the aspect ratios between these sets are the same.

Epochs	10	50	100	300	500	1000
NN_1 loss for $\boldsymbol{z}$ (random) $(\times 10^{-3})$	11.17	9.26	9.26	9.26	9.26	9.26
NN_2 loss for $\boldsymbol{z}$ (predefined) $(\times 10^{-3})$	13.70	10.66	9.28	9.05	9.05	9.05
NN_1 loss for $\boldsymbol{x}$ (random) $(\times 10^{-4})$	1.76	1.38	1.35	1.33	1.32	1.32
NN_2 loss for $\boldsymbol{x}$ (predefined) $(\times 10^{-4})$	1.62	1.37	1.33	1.31	1.31	1.31

Figure 1 illustrates this feature with two neural networks, NN_1 and NN_2, trained using different strategies for selecting training, validation, and testing data. Even though both networks achieve good parameterization results offline (similar to the BE manifold), their underlying structures differ visually from the BE manifold.

To explore these differences, we focus on specific components ($\boldsymbol{\mathcal{X}}^j_{\boldsymbol{\theta}_2^*}$ for x_j and $\boldsymbol{\mathcal{Z}}^j_{\boldsymbol{\theta}_1^*}$ for z_j) of the neural parameterizations. We plot these components as level sets on a three-dimensional sphere to reveal their geometric properties. This visualization is particularly useful since $\boldsymbol{\mathcal{Z}}^j_{\boldsymbol{\theta}_1^*}$ and $\boldsymbol{\mathcal{X}}^j_{\boldsymbol{\theta}_2^*}$ are scalar fields depending on three variables. For a given radius, the level sets of $\boldsymbol{\mathcal{Z}}^j_{\boldsymbol{\theta}_1^*}$ (resp. $\boldsymbol{\mathcal{X}}^j_{\boldsymbol{\theta}_1^*}$) on the three-dimensional sphere, $y_1^2 + y_2^2 + y_3^2 = r^2$, can be visualized as a 2D surface that maps (y_1, y_2) to z_j (resp. x_j). Figures 1(B)–(D) show these level sets for radius $r = 1$.

Interestingly, these visualizations reveal significant differences in the minimizers (and consequently, the parameterization formulas) of NN_1 and NN_2, even though their loss function values differ only by 1% (table 1) and their high-frequency residuals are similar (red curve in figure 1(E).

These geometric offline differences hide more profound consequences when the neural parameterizations are used online, for closure. As explained below, the sensitivity of online predictions that are tied to sampling issues is indeed observed. In that respect, recall that a common practice to train NNs is to divide the dataset into three subsets. The first subset is the training set, which is used for computing the loss function's gradient and updating the network weights and biases.

The second subset is the validation set. It corresponds to the second dataset over which the prediction skills of the fitted model are assessed. The error on the validation set is monitored during the training process to provide an unbiased evaluation while tuning the model's hyperparameters. When the network begins to overfit the data, the error on the validation set typically begins to rise after an initial decrease. The network parameters are saved at the minimum. It gives then the 'final model' that is tested over the test set that is typically a holdout dataset not used as a validation nor a training set.

The parameterization NN_1 shown in figure 1(A) is learnt through a random selection while NN_2 is learnt through a predefined selection. In each case, ratios for training, testing, and validation are 0.7, 0.15, and 0.15, respectively. The total length of the training is 700 days. Given the same input and target data, the minimal values of the loss functions (3) and (4) for NN_1 and NN_2 are reported in table 1, across epochs. Already after 500 epochs, one observes that the loss function evaluations differ only by 1% between the random or predefined selection protocol of the training, validation, and testing sets.

We now discuss the sensitivity issue of online predictions driven by such neural parameterizations that are close in terms of their loss function scoring. This point is illustrated in figure 3. There, we show online prediction corresponding to a given slow NN-parameterization $(\boldsymbol{\mathcal{X}}_{\boldsymbol{\theta}_2^*}, \boldsymbol{\mathcal{Z}}_{\boldsymbol{\theta}_1^*})$ learnt by minimization of the loss functions (equations (3) and (4)), namely the solution to the slow neural closure

$$a_i \frac{\mathrm{d}y_i}{\mathrm{d}t} = -a_k b_k \boldsymbol{\mathcal{X}}^j_{\boldsymbol{\theta}_2^*}\left(\boldsymbol{y}, \boldsymbol{\mathcal{Z}}_{\boldsymbol{\theta}_1^*}(\boldsymbol{y})\right) y_k - a_j b_j y_j \boldsymbol{\mathcal{X}}^k_{\boldsymbol{\theta}_2^*}\left(\boldsymbol{y}, \boldsymbol{\mathcal{Z}}_{\boldsymbol{\theta}_1^*}(\boldsymbol{y})\right) + c\left(a_k - a_j\right) y_j y_k - a_i \boldsymbol{\mathcal{X}}^i_{\boldsymbol{\theta}_2^*}\left(\boldsymbol{y}, \boldsymbol{\mathcal{Z}}_{\boldsymbol{\theta}_1^*}(\boldsymbol{y})\right) - \nu_0 a_i^2 y_i. \tag{6}$$

This closed equation in the $\boldsymbol{y}$-variable is obtained by replacing the x_ℓ-variables in the $\boldsymbol{y}$-equation of the L80 model (equation (1)) by their neural parameterizations, either NN_1 or NN_2.

The attractor corresponding to the slow NN_1-closure (with random selection) differs clearly from that of slow NN_2-closure (with predefined selection) in spite of convergence and closeness of the loss functions at their respective minimal value; see figure 3(B). Both predict periodic orbits with different attributes, one self-intersecting in the (y_2, y_3)-plane (NN_1), the other without intersection point (NN_2).

A closer inspection at these topological differences reveals in the time domain that the slow NN_1-closure is able to capture more accurately the low-frequency content of certain temporal patterns exhibited by the

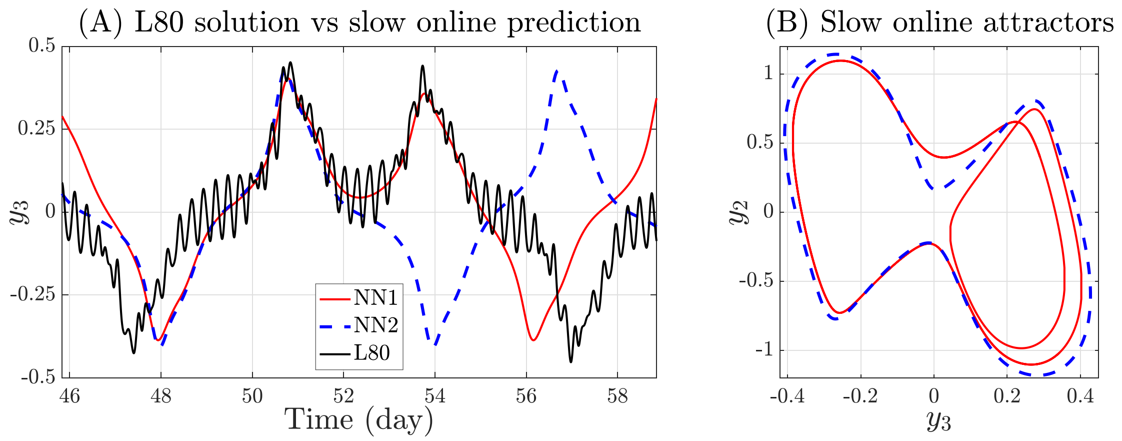

Figure 3. Sensitivity of the slow neural closures. Here, NN_1 and NN_2 differ only in their training modalities. NN_1 is learnt from random selection of the training, validation, and testing sets, and NN_2 from a predefined selection with the same aspect ratios; see Text. The corresponding loss functions differ by 1% (see table 1), while the dynamical differences of the online predictions are substantial.

HLF solutions of the L80 model compared to the slow NN_2-closure; blue vs red curves in figure 3(A). We argue below that such a sensitivity between online solutions takes its root in the rare events tied to the irregular transitions exhibited by the HLF solutions to the L80 model that spoils the offline learning.

In contrast, at lower Rossby numbers, for regimes devoid of fast oscillations such as shown in figure 4(D) below corresponding to $F_1 = 6.97 \times 10^{-2}$ in the L80 model, neural closures of high-accuracy are easily accessible with skills comparable to those obtained with the BE manifold; see figure 5. As explained below, the reasons for this success lie in the absence of high-frequencies in the solutions to parameterize and in the absence of rare events in the statistics of lobe transitions.

3. Irregular transitions, rare events and learning consequences

The significant sensitivity observed in capturing the low-frequency content with nearby neural parameterizations (as measured by their loss functions) requires further investigation. Since these variations in figure 3 solely stem from how training, validation, and test sets are chosen, we conduct in this section a statistical analysis of key features of the L80 dynamics in HLF regimes. Our focus is on the irregular lobe transitions exhibited by HLF solutions. For comparison, we also analyze lobe transitions in the slow chaotic regime of figure 4(D), where neural parameterizations perform well and learn the closure effectively. Notably, figure 5 demonstrates that for the slow chaotic regime, high-accuracy neural closures are readily achievable, with skills comparable to those obtained using the BE manifold.

To gain a deeper understanding of lobe transition statistics in the slow chaotic and HLF regimes, we performed high-resolution simulations of the L80 model for each regime. Each simulation spanned a 500 year period, integrating the L80 dynamics with a timestep of 0.75 minutes. This corresponds roughly to an interval of size $730,000 \times T_{GW}$, where T_{GW} is the dominant period of the gravity waves in the model.

In each regime, the L80 attractor exhibits two lobes. This is shown in the (y_2, y_3)-projection for the HLF regime (figure 4(A)) and in the (y_1, y_3)-projection for the slow chaos regime (figure 4(D)). The latter evokes the Lorenz 63 ‘butterfly attractor’ [6], consistent with the L80 dynamics devoid of fast motion for this Rossby number (geostrophic motion). The former attractor, more fuzzy, exemplifies the presence of fast dynamics riding the slow, geostrophic motion.

In each case, these lobes are essentially separated by the vertical line $y_3 = 0$. Numerical integration of the L80 model reveals that the visit of the right lobe comes with $y_3(t)$ getting greater than some threshold value y_b, while the visit of the left lobe comes with $y_3(t)$ getting smaller than $y_a = -y_b$. A close inspection of the solution in the HLF case reveals that the choice of $y_b = 0.2$ constitutes a good one to identify the sojourn of the dynamics within one lobe from the other. This choice leads furthermore to an interval $(-y_b, y_b)$ that

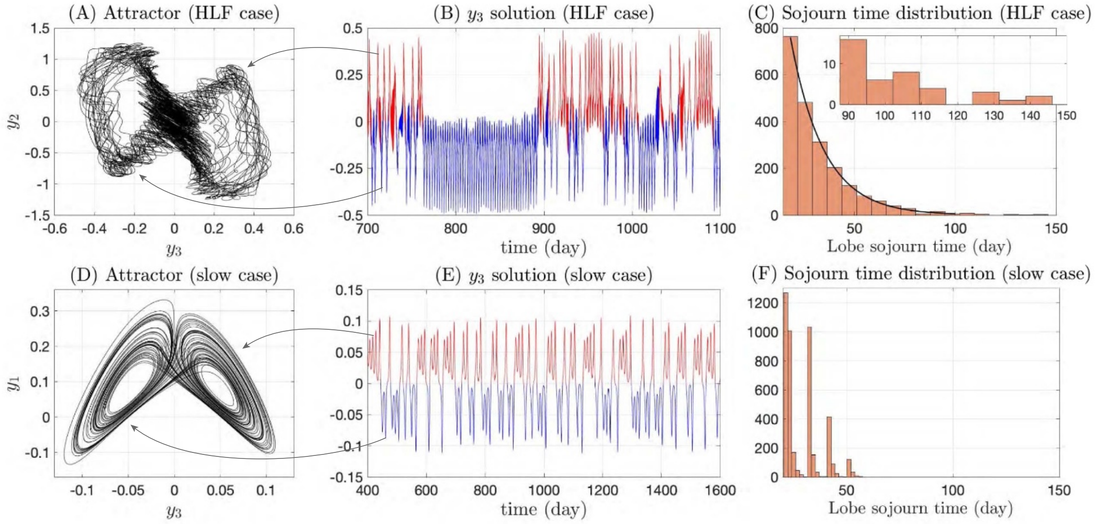

Figure 4. Panel (A): Attractor in the HLF case. Panel (B): The sojourn episodes within one particular lobe are marked by different colours. Here, the parameters are those used in Lorenz's original paper [7] except $F_1 = 0.3027$ in equation (1). Panel (C): Lobe sojourn time distributions. The exponential fit is calculated over 500 yr-long simulation of equation (1) and is shown by the black curve $f(t) = ae^{bt}$ with $a = 2292$ and $b = -6.05 \times 10^{-2}$ with t in day. The inset in panel (C) shows a magnification of the distribution for the rare and large sojourn times. Panels E and F: Same as panels B and C except that $F_1 = 6.97 \times 10^{-2}$, corresponding to the slow chaotic regime shown in panel (D) in which the solutions are void of fast oscillations. In this regime, no rare event statistics emerge.

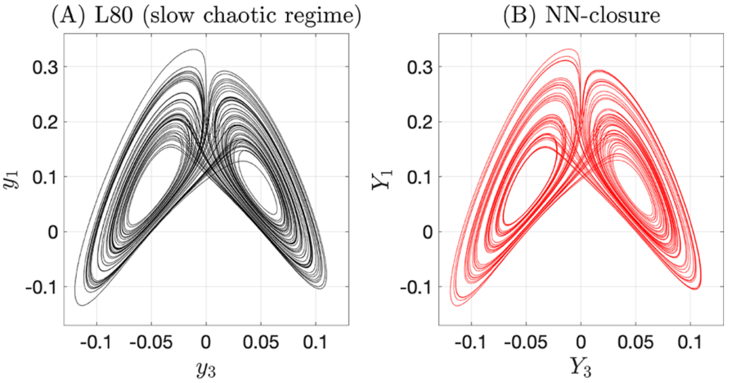

Figure 5. The L80 attractor vs. its NN-closure in the slow chaos regime. Here $F_1 = 6.97 \times 10^{-2}$ in the L80 model, which corresponds to the slow chaos case shown in figure 4(D) and in [25, figure 7].

provides a good bound of the bursts of fast oscillations crossing the vertical line $y_3 = 0$ in the (y_2, y_3)-plane ('gray' zone).

To count the transitions from one lobe to the other one thus proceeds as follows. Given our 500 yr long simulation of $y_3(t)$ we first find the local maxima and minima that are above y_b and below y_a, respectively. No transition occurs between consecutive such local maxima or minima. A transition occurs only when a local maximum above y_b is immediately followed by a local minimum below y_a or vice versa. If a local maximum is immediately followed by a local minimum, the intermediate time instant at which the trajectory goes below zero is identified as the transition instant, and the other way around if a local minimum is immediately followed by a local maximum. These transition times characterized this way allow us to count the sojourn times in a lobe and display the distribution of these sojourn times shown in figures 4(C) and (F).

These lobe sojourn time distributions reveal a striking difference between the HLF and slow chaotic regimes. In the HLF case, we observe indeed that the solution can stay in one lobe for a period of time that can be arbitrarily long (see solution's segment between $t = 763$ and $t = 893$ shown in blue in figure 4(B) albeit of probability of occurrence vanishing exponentially as shown in figure 4(C). As a comparison, the transitions between the attractor's lobes occur at a much more regular pace in the slow chaotic regime (see figure 4(E) in which the solutions to the L80 model are void of fast oscillations. In this case, the distribution of sojourn times drops quickly below a 60 day duration barrier (figure 4(F)).

These rare events, following an exponential distribution, pose a significant challenge for developing reliable slow neural closures. They introduce diversity in the temporal patterns of the time series, which contributes to the sensitivity issues observed in figure 3. A random training set might be skewed towards one lobe duration more than a predefined set, leading to confusion in the learning process for the neural network.

4. The high-frequency barrier to neural closure

Section 3 demonstrated that using neural networks to parameterize the slow dynamics of HLF solutions can lead to sensitivity issues in online prediction (figure 3). This sensitivity arises from rare events associated with irregular lobe sojourn durations, as shown in figures 4(B) and (C). In this section, we explore another challenge: the direct parameterization of the unfiltered $\boldsymbol{x}$-components of HLF solutions. These components contain a complex mixture of both slow and fast motions, posing significant difficulties for closure with neural networks.

To illustrate this point, we learn an MLP for $\boldsymbol{x}(t)$, denoted by $\boldsymbol{\mathcal{V}}_{\boldsymbol{\theta}}$, with (the unfiltered) $\boldsymbol{y}(t)$-variable of the L80 model (equation (1)), as input, and the *unfiltered* $\boldsymbol{x}$-component, $\boldsymbol{x}(t)$, as output. Note that unlike the slow NN-parameterizations above, the parameterization $\boldsymbol{\mathcal{V}}_{\boldsymbol{\theta}}$ aims at parameterizing $\boldsymbol{x}(t)$ directly as a nonlinear mapping of $\boldsymbol{y}(t)$ without conditioning on $\boldsymbol{z}(t)$ nor filtering of any sort. The corresponding closure, called a vanilla NN-closure, consists then of equation (6) in which $\boldsymbol{\mathcal{X}}_{\boldsymbol{\theta}_2^*}(\boldsymbol{y}, \boldsymbol{\mathcal{Z}}_{\boldsymbol{\theta}_1^*}(\boldsymbol{y}))$ is replaced by $\boldsymbol{\mathcal{V}}_{\boldsymbol{\theta}^*}(\boldsymbol{y})$, obtained after minimization of the following L^2-loss function

$$\mathcal{L}_{\boldsymbol{\theta}}\left(\boldsymbol{x};\boldsymbol{y}\right) = \sum_j \left\|\boldsymbol{x}_{t_j} - \boldsymbol{\mathcal{V}}_{\boldsymbol{\theta}}\left(\boldsymbol{y}_{t_j}\right)\right\|^2, \tag{7}$$

for which the target variable $\boldsymbol{x}(t)$ is *unfiltered*, i.e. containing a mixture of fast and slow oscillations. To address this more challenging problem we use MLPs with a larger capacity either with more neurons and/or layers.

Interestingly, our experiments show that a neural network with just one hidden layer and 20 neurons achieves the best closure results. Figure 6 compares simulated time series from four different vanilla NN-closure settings. The setting with one hidden layer and 20 neurons partially captures the complexity of the HLF solution's temporal patterns (figures 7(A) and (B)). However, it entirely misses the high-frequency content associated with IGWs, as evident from the power spectral density (PSD) comparison in figure 8.

While increasing the complexity of a neural network (more hidden layers or neurons) can reduce the loss function during training, it does not guarantee better performance in the actual closure. For example, a vanilla neural network ($\boldsymbol{\mathcal{V}}_{\boldsymbol{\theta}}$) with 5 hidden layers and 20 neurons per layer predicts an unrealistic, small-amplitude periodic orbit when used online in the neural closure through time-stepping (figure 9(B)). Additionally, it exaggerates high-frequency content in the solutions it generates offline (see figure 9(C) and table 2).

Our results highlight the limitations of using a vanilla neural network closure to directly capture the fast dynamics of the L80 system using the 'slow' variable $\boldsymbol{y}$. This approach relies on potentially complex, non-linear functions encoded by MLPs, but struggles to represent the system's multiscale dynamics accurately. This issue is similar to the spectral bias problem observed in standard neural networks for function fitting [48], where they prioritize capturing low-frequency features. However, the challenge here is more complex. The goal is to learn the neglected 'fast' variables and their high-frequency content offline, so the online solution through the NN-closure can reproduce both the mixture of slow and fast motions of the original system. This includes capturing global geometric features like the attractor's shape and symmetry. As shown in figure 7(C), vanilla NN-closures often distort these features compared to the true L80 attractor.

To address the limitations of feedforward neural networks (vanilla NNs) to close the L80 dynamics in HLF regimes, one route to explore would be to incorporate memory effects using architectures like Long-Short Term Memory (LSTM) networks [48]. LSTMs have demonstrably achieved model reduction in

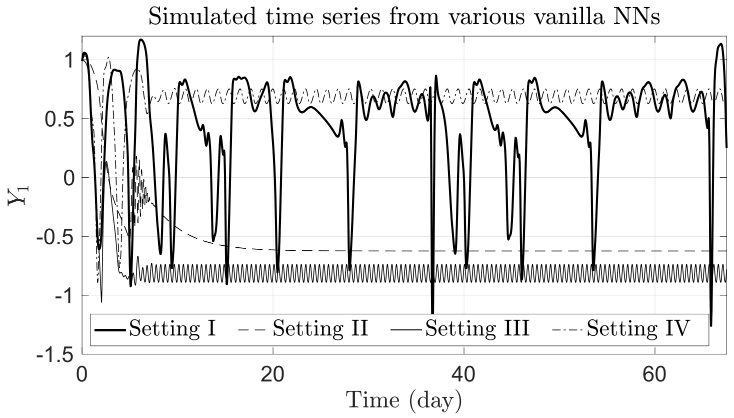

Figure 6. Simulated time series from vanilla NN-closures in four different settings. Setting I (same as used for the results shown in figure 7): one hidden layer with 20 neurons (thick solid line). Setting II: two hidden layers with 5 neurons in each layer (dashed line). Setting III: two hidden layers with 10 neurons in each layer (light solid line). Setting IV: two hidden layers with 20 neurons in each layer (dash-dotted line). The corresponding loss function values are given in table 2.

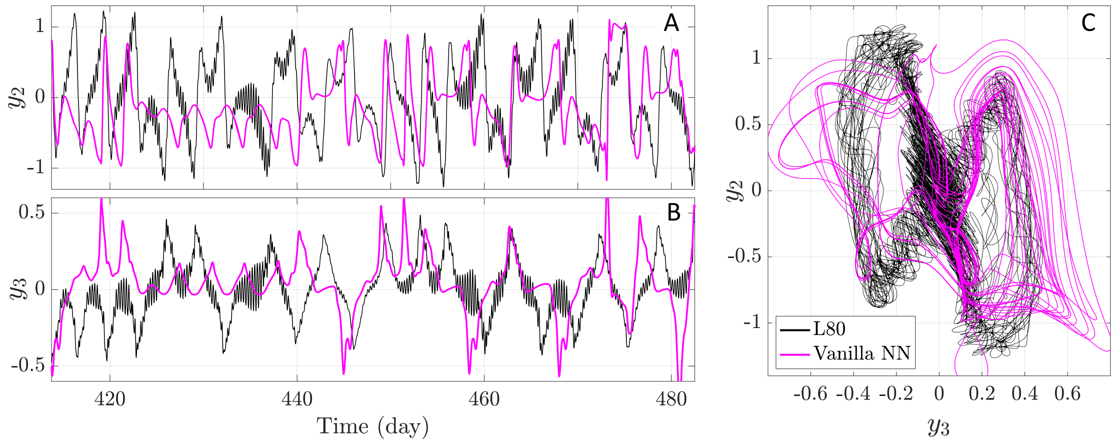

Figure 7. Vanilla NN-closure vs L80 dynamics. Failure to capture the high-frequency content and symmetry of the L80 attractor. Here, is used the best performing vanilla neural network (NN1) from Setting I in figure 6.

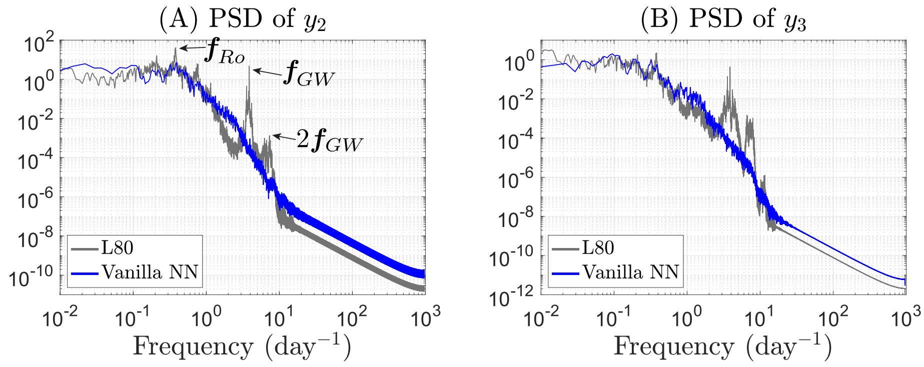

Figure 8. Power spectral density (PSD) comparison. This figure compares the PSD of variables y_2 (panel (A)) and y_3 (panel (B)) for the L80 model (gray curve) and the best performing vanilla neural network closure (blue curve) from Setting I in figure 6. While the vanilla closure captures the overall spectral background of the L80 solutions well, it misses the important peaks at frequencies f_{GW} and f_{Ro} (and their harmonics). These frequencies correspond to inertia-gravity waves and Rossby waves, respectively.

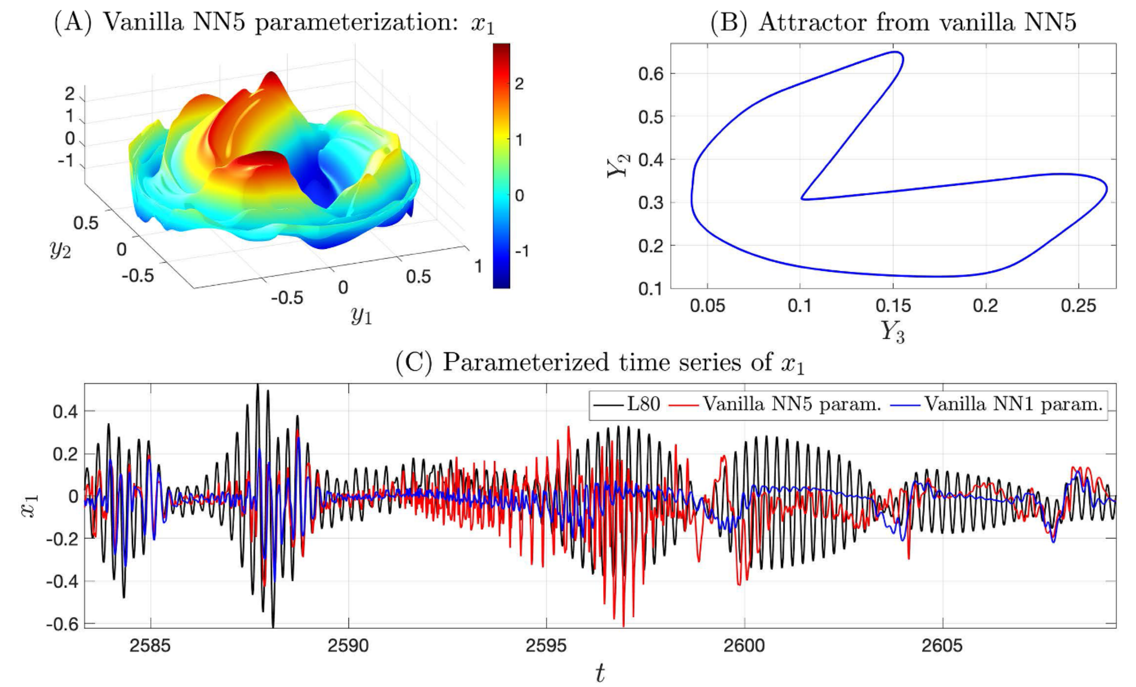

Figure 9. Panel (A): This panel shows the neural parameterization ($\boldsymbol{\mathcal{V}}_{\boldsymbol{\theta}}$) with 5 layers and 20 neurons per layer (denoted as NN5) for variable x_1. We use the same visualization style as figures 1(B)–(D). Notice the sharp gradients in the manifold, reflecting NN5's attempt to capture the high-frequency details of the HLF solutions. Panel (B): This panel displays the corresponding solution (Y_2, Y_3) obtained using the NN5 closure. Panel (C): Compared to the best performing vanilla neural network (NN1) from Setting I in figure 6, NN5 exaggerates the high-frequency content in the offline parameterization.

Table 2. Loss function evaluations. In this table are reported the loss values corresponding to the vanilla NN-closures shown in figure 6. Note that the underlying loss function is that defined in equation (7).

Epochs	10	50	100	300	500
Setting I loss ($\times 10^{-2}$)	2.62	2.54	2.52	2.49	2.49
Setting II loss ($\times 10^{-2}$)	2.74	2.67	2.66	2.64	2.64
Setting III loss ($\times 10^{-2}$)	2.72	2.45	2.44	2.43	2.43
Setting IV loss ($\times 10^{-2}$)	2.42	2.33	2.32	2.30	2.30

various contexts (e.g. [49–51]). This success can be attributed to theoretical underpinnings from dynamical systems theory (Takens' delay embedding theorem [52]) and statistical mechanics (Mori-Zwanzig formulation [22, 53–56]). Additionally, we mention recent approaches combining Takens' embedding with Koopman operator theory and sparse regression to obtain linear representations of nonlinear dynamics [57].

However, as highlighted in [22], memory effects might not be crucial for achieving effcient closure of solutions in the HLF regime. Studies have shown that using the BE manifold for capturing the geostrophic motion and a network of stochastic oscillators for IGWs can achieve high accuracy without recurrent architectures like LSTMs [26]. This, along with the challenges of rare events discussed earlier, raises questions about whether LSTMs or other recurrent networks are necessary to reproduce the intricate multiscale dynamics of $\boldsymbol{y}$ using a closed model (like in [26]) built with these components (LSTMs).

5. Discussion

Our findings, particularly the interplay between rare events and the multiscale nature in HLF regimes, highlight the challenges that machine learning can face for accurate closure of geophysical flows in which geostrophic and ageostrophic motions interact strongly. As extreme weather events and non-Gaussian statistics become more prevalent with climate change [58–62], this study underscores that significant hurdles remain despite the recent advancements in neural parameterizations. Reliable parameterizations that robustly capture rare events are crucial. In this regard, incorporating rare event algorithms [63–67] could be beneficial. By simulating rare events offline, these algorithms could improve the sampling of distribution tails, leading to better trained neural networks.

This study contributes new insights into the challenges of closing the Lorenz 80 model using data-driven methods, particularly in high Rossby number regimes ($Ro > Ro^*$). Compared to other Lorenz models, like the less challenging Lorenz 96 model [68], the L80 system has received less attention for closure tasks. However, the recent stochastic closure approach by [26] for these demanding regimes provides a valuable benchmark for future research. We hope this work encourages further exploration of the L80 model as a meaningful testbed for developing and comparing closure ideas.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: https://doi.org/10.7910/DVN/72F33A.

Acknowledgments

The authors sincerely thank the reviewers for their insightful comments and careful reading of our manuscript. Their feedback has been invaluable in helping us improve the clarity and focus of the article. This work has been partially supported by the Office of Naval Research (ONR) Multidisciplinary University Research Initiative (MURI) Grant N00014-20-1-2023, by the National Science Foundation Grant DMS-2108856, and by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (Grant Agreement No. 810370). We also acknowledge the computational resources provided by Advanced Research Computing at Virginia Tech.

Appendix A. HLF solutions and the slow motion learning

The high-low frequency (HLF) solutions used in this article are those reported in [26, figure 7]. These solutions are obtained from the parameters used in Lorenz's original paper [7] except F_1 chosen to be $F_1 = 3.027 \times 10^{-1}$ as identified in [25]; see the Materials and Methods section in [26] for details.

As shown in figure 10, for this parameter regime, the HLF solutions contain a mixture of slow and fast oscillations in each variable $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$ of the L80 model that causes serious difficulties for closure [26]. The dominant frequency of the Rossby wave content in the HLF solutions is $f_{Ro} = 0.31\,\mathrm{d}^{-1}$ ($T_{Ro} = 3.2\,\mathrm{d}$) and that of the inertia-gravity wave (IGW) content is $f_{GW} = 3.76\,\mathrm{d}^{-1}$ ($T_{GW} = 6.3\,\mathrm{h}$).

To learn a neural parameterization of the slow motion, the weights and biases of the NNs are updated according to a Levenberg-Marquardt (LM) optimization [69]. The LM algorithm is known to be efficient for small or medium-scaled problems [70, chapter 12], especially when the loss function is just a mean squared error, which is the case here. This algorithm is sufficient to obtain loss functions with small residuals; see table 1.

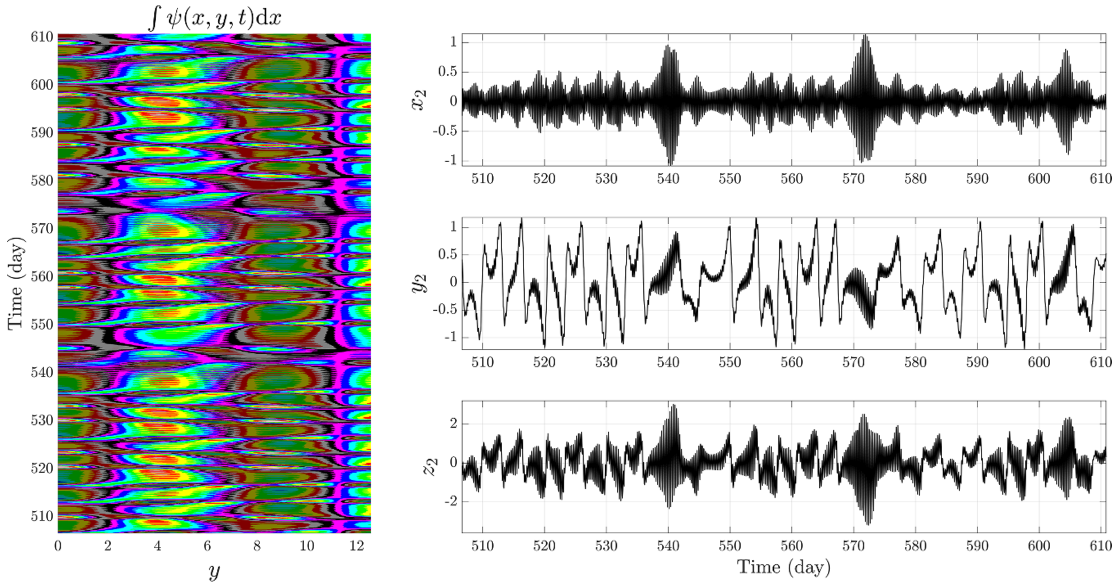

Figure 10. HLF solutions. Left panel: Hovmöller plot of the streamfunction (averaged over the x-direction). Right panels: a few corresponding time series. Note the energetic bursts of fast oscillations corresponding to spontaneous bursts of IGWs. A vanilla NN consists of seeking a feedforward neural network (FNN) mapping the $\boldsymbol{y}$-components of the L80 model (equation (1)) to the $\boldsymbol{x}$-components. These fast energetic bursts are a serious barrier to learning with FNNs. The streamfunction ψ is constructed from the $\boldsymbol{y}$-components of the L80 model solution according to $\psi(x,y,t) = \sum_{j=1}^{3} y_j(t)\cos(\alpha_j^1 x)\cos(\alpha_j^2 y)$ where the spatial variables x and y (not to be confused with $\boldsymbol{x}$ and $\boldsymbol{y}$ in the L80 model) takes value in a square domain $[0,L]\times[0,L]$ with $L = 4\pi$ and the vectors $\boldsymbol{\alpha}_j = (\alpha_j^1, \alpha_j^2)$ $(j = 1,2,3)$ are chosen to satisfy the conditions given by [7, Eqs. (16)–(17)]. Following [7], we chose $\boldsymbol{\alpha}_1 = (\sqrt{2}/2, \sqrt{2}/2)$, $\boldsymbol{\alpha}_2 = ((\sqrt{2}-\sqrt{6})/4, (\sqrt{2}+\sqrt{6})/4)$, and $\boldsymbol{\alpha}_3 = -(\boldsymbol{\alpha}_1 + \boldsymbol{\alpha}_2)$.

Appendix B. The BE manifold and BE closure

For consistency, we recall from [25] the derivation of the BE manifold that serves as our parameterization baseline. Mathematically, the BE manifold aims at reducing the L80 model to a 3D system of ODEs, by means of nonlinear parameterization of the variables $\boldsymbol{x} = (x_1, x_2, x_3)^{\mathrm{T}}$ and $\boldsymbol{z} = (z_1, z_2, z_3)^{\mathrm{T}}$, in terms of the variable $\boldsymbol{y} = (y_1, y_2, y_3)^{\mathrm{T}}$; see [23]. By analyzing the order of magnitudes of the different terms in the x_i-equations and after rescaling following [25], we arrive to the following parameterization of the $\boldsymbol{z}$-variable in terms of the rotational $\boldsymbol{y}$-variable

$$z_i = G_i(\boldsymbol{y}) = y_i - \frac{2c^2}{a_i} y_j y_k. \tag{B1}$$

Further algebraic manipulations show that under an invertibility condition of a matrix $M(\boldsymbol{y}, G(\boldsymbol{y}))$ conditioned on the $\boldsymbol{y}$-variable, one obtains (implicitly) $\boldsymbol{x}$ as a function Φ of $\boldsymbol{y}$ given by

$$\Phi(\boldsymbol{y}) = [M(\boldsymbol{y}, G(\boldsymbol{y}))]^{-1} \begin{pmatrix} d_{1,2,3}(\boldsymbol{y}, G(\boldsymbol{y})) \\ d_{2,3,1}(\boldsymbol{y}, G(\boldsymbol{y}))) \\ d_{3,1,2}(\boldsymbol{y}, G(\boldsymbol{y}))) \end{pmatrix}, \tag{B2}$$

where the $d_{i,j,k}$ are given explicitly; see [23, 25]. The function $\Phi(\boldsymbol{y}) = (\Phi_1(\boldsymbol{y}), \Phi_2(\boldsymbol{y}), \Phi_3(\boldsymbol{y}))^{\mathrm{T}}$ corresponds to the *BE manifold*, it is aimed to provide a nonlinear parameterization between $\boldsymbol{x}$ and $\boldsymbol{y}$ when the latter exists.

The BE closure is then

$$\frac{\mathrm{d}y_i}{\mathrm{d}\tau} = -a_i^{-1} a_k b_k \Phi_j(\boldsymbol{y}) y_k - a_i^{-1} a_j b_j y_j \Phi_k(\boldsymbol{y}) + c a_i^{-1}(a_k - a_j) y_j y_k - \Phi_i(\boldsymbol{y}) - \nu_0 a_i y_i, \tag{B3}$$

for which (i,j,k) lies in $\{(1,2,3), (2,3,1), (3,1,2)\}$.

ORCID iDs

Mickaël D Chekroun https://orcid.org/0000-0002-4525-5141
Honghu Liu https://orcid.org/0000-0002-9226-0744

References

[1] Bolin B 1955 Numerical forecasting with the barotropic model *Tellus* **7** 27–49

[2] Baer F and Tribbia J J 1977 On complete filtering of gravity modes through nonlinear initialization *Mon. Weather Rev.* **105** 1536–9

[3] Machenhauer B 1977 On the dynamics of gravity oscillations in a shallow water model, with applications to normal mode initialization *Beitr. Phys. Atmos* **50** 253–71

[4] Daley R 1981 Normal mode initialization *Rev. Geophys.* **19** 450–68

[5] Leith C E 1980 Nonlinear normal mode initialization and quasi-geostrophic theory *J. Atmos. Sci.* **37** 958–68

[6] Lorenz E N 1963 Deterministic nonperiodic flow *J. Atmos. Sci.* **20** 130–41

[7] Lorenz E N 1980 Attractor sets and quasi-geostrophic equilibrium *J. Atmos. Sci.* **37** 1685–99

[8] Sirignano J and Spiliopoulos K 2018 DGM: a deep learning algorithm for solving partial differential equations *J. Comput. Phys.* **375** 1339–64

[9] Raissi M, Perdikaris P and Karniadakis G 2019 Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations *J. Comput. Phys.* **378** 686–707

[10] Bar-Sinai Y, Hoyer S, Hickey J and Brenner M 2019 Learning data-driven discretizations for partial differential equations *Proc. Natl Acad. Sci.* **116** 15344–9

[11] Pathak J, Hunt B, Girvan M, Lu Z and Ott E 2018 Model-free prediction of large spatiotemporally chaotic systems from data: a reservoir computing approach *Phys. Rev. Lett.* **120** 024102

[12] Brunton S L, Proctor J L and Kutz J N 2016 Discovering governing equations from data by sparse identification of nonlinear dynamical systems *Proc. Natl Acad. Sci.* **113** 3932–7

[13] Rasp S, Pritchard M and Gentine P 2018 Deep learning to represent subgrid processes in climate models *Proc. Natl Acad. Sci.* **115** 9684–9

[14] Gentine P, Pritchard M, Rasp S, Reinaudi G and Yacalis G 2018 Could machine learning break the convection parameterization deadlock? *Geophys. Res. Lett.* **45** 5742–51

[15] Brenowitz N D, Beucler T, Pritchard M and Bretherton C 2020 Interpreting and stabilizing machine-learning parametrizations of convection *J. Atmos. Sci.* **77** 4357–75

[16] Bolton T and Zanna L 2019 Applications of deep learning to ocean data inference and subgrid parameterization *J. Adv. Model. Earth Syst.* **11** 376–99

[17] Maulik R, San O, Rasheed A and Vedula P 2019 Subgrid modelling for two-dimensional turbulence using neural networks *J. Fluid Mech.* **858** 122–44

[18] Kochkov D, Smith J, Alieva A, Wang Q, Brenner M P and Hoyer S 2021 Machine learning–accelerated computational fluid dynamics *Proc. Natl Acad. Sci.* **118** e2101784118

[19] Zanna L and Bolton T 2020 Data-driven equation discovery of ocean mesoscale closures *Geophys. Res. Lett.* **47** e2020GL088376

[20] Subel A, Guan Y, Chattopadhyay A and Hassanzadeh P 2023 Explaining the physics of transfer learning in data-driven turbulence modeling *PNAS Nexus* **2** gad015

[21] Srinivasan K, Chekroun M D and McWilliams J C 2024 Turbulence closure with small, local neural networks: forced two-dimensional and β-plane flows *J. Adv. Model. Earth Syst.* e2023MS003795

[22] Lucarini V and Chekroun M D 2023 Theoretical tools for understanding the climate crisis from Hasselmann's programme and beyond *Nat. Rev. Phys.* **5** 744–65

[23] Gent P R and McWilliams J C 1982 Intermediate model solutions to the Lorenz equations: strange attractors and other phenomena *J. Atmos. Sci.* **39** 3–13

[24] Vautard R and Legras B 1986 Invariant manifolds, quasi-geostrophy and initialization *J. Atmos. Sci.* **43** 565–84

[25] Chekroun M D, Liu H and McWilliams J C 2017 The emergence of fast oscillations in a reduced primitive equation model and its implications for closure theories *Comput. Fluids* **151** 3–22

[26] Chekroun M D, Liu H and McWilliams J C 2021 Stochastic rectification of fast oscillations on slow manifold closures *Proc. Natl Acad. Sci. USA* **118** e2113650118

[27] Plougonven R and Snyder C 2007 Inertia–gravity waves spontaneously generated by jets and fronts. Part I: different baroclinic life cycles *J. Atmos. Sci.* **64** 2502–20

[28] Polichtchouk I and Scott R 2020 Spontaneous inertia-gravity wave emission from a nonlinear critical layer in the stratosphere *Q. J. R. Meteorol. Soc.* **146** 1516–28

[29] Tulich S, Randall D and Mapes B 2007 Vertical-mode and cloud decomposition of large-scale convectively coupled gravity waves in a two-dimensional cloud-resolving model *J. Atmos. Sci.* **64** 1210–29

[30] Lane T P 2015 Convectively generated gravity waves *Encyclopedia of Atmospheric Sciences* 2nd edn (Elsevier) pp 171–9

[31] Dror T, Chekroun M D, Altaratz O and Koren I 2021 Deciphering organization of GOES–16 green cumulus, through the EOF lens *Atmos. Chem. Phys.* **21** 12261–72

[32] Rocha C B, Chereskin T K, Gille S T and Menemenlis D 2016 Mesoscale to submesoscale wavenumber spectra in drake passage *J. Phys. Oceanogr.* **46** 601–20

[33] Bender C M and Orszag S 1999 *Advanced Mathematical Methods for Scientists and Engineers: Asymptotic Methods and Perturbation Theory* vol 1 (Springer Science & Business Media)

[34] Young W R 2021 Inertia-gravity waves and geostrophic turbulence *J. Fluid Mech.* **920** F1

[35] McWilliams J and Gent P 1980 Intermediate models of planetary circulations in the atmosphere and ocean *J. Atmos. Sci.* **37** 1657–78

[36] Monin A 1952 Change of pressure in a barotropic atmosphere *Akad. Nauk. Izv. Ser. Geofiz.* **4** 76–85

[37] Charney J 1955 The use of the primitive equations of motion in numerical prediction *Tellus* **7** 22–26

[38] Lorenz E 1960 Energy and numerical weather prediction *Tellus* **12** 364–73

[39] Chekroun M D, Neelin J D, Kondrashov D, McWilliams J C and Ghil M 2014 Rough parameter dependence in climate models: the role of Ruelle-Pollicott resonances *Proc. Natl Acad. Sci. USA* **111** 1684–90

[40] Chekroun M D, Tantet A, Dijkstra H A and Neelin J D 2020 Ruelle–Pollicott resonances of stochastic systems in reduced state space. Part I: theory *J. Stat. Phys.* **179** 1366–402

[41] Tantet A, Chekroun M D, Dijkstra H A and Neelin J D 2020 Ruelle-Pollicott resonances of stochastic systems in reduced state space. Part II: stochastic hopf bifurcation *J. Stat. Phys.* **179** 1403–48

[42] Lorenz E N 1986 On the existence of a slow manifold *J. Atmos. Sci.* **43** 1547–58

[43] Lorenz E N and Krishnamurthy V 1987 On the nonexistence of a slow manifold *J. Atmos. Sci.* **44** 2940–50

[44] Camassa R 1995 On the geometry of an atmospheric slow manifold *Physica* D **84** 357–97

[45] Vanneste J 2008 Exponential smallness of inertia–gravity wave generation at small rossby number *J. Atmos. Sci.* **65** 1622–37

[46] Temam R and Wirosoetisno D 2011 Slow manifolds and invariant sets of the primitive equations *J. Atmos. Sci.* **68** 675–82

[47] Vanneste J 2013 Balance and spontaneous wave generation in geophysical flows *Annu. Rev. Fluid Mech.* **45** 147–72

[48] Hochreiter S and Schmidhuber J 1997 Long short-term memory *Neural Comput.* **9** 1735–80

[49] Gupta A and Lermusiaux P F J 2021 Neural closure models for dynamical systems *Proc. R. Soc.* A **477** 20201004

[50] Harlim J, Jiang S, Liang S and Yang H 2021 Machine learning for prediction with missing dynamics *J. Comput. Phys.* **428** 109922

[51] Lu P Y, Ariño Bernad J and Soljačić M 2022 Discovering sparse interpretable dynamics from partial observations *Commun. Phys.* **5** 206

[52] Takens F 1981 Detecting strange attractors in turbulence *Dynamical Systems and Turbulence, Warwick 1980* ed D Rand and L-S Young (Springer) pp 366–81

[53] Zwanzig R 1961 Memory effects in irreversible thermodynamics *Phys. Rev.* **124** 983–92

[54] Mori H 1965 Transport, collective motion and brownian motion *Prog. Theor. Phys.* **33** 423–55

[55] Chorin A J, Hald O H and Kupferman R 2002 Optimal prediction with memory *Physica* D **166** 239–57

[56] Santos Gutiérrez M, Lucarini V, Chekroun M D and Ghil M 2021 Reduced-order models for coupled dynamical systems: data-driven methods and the Koopman operator *Chaos* **31** 053116

[57] Brunton S L, Brunton B W, Proctor J L, Kaiser E and Kutz J N 2017 Chaos as an intermittently forced linear system *Nat. Commun.* **8** 19

[58] Raveh-Rubin S and Wernli H 2015 Large-scale wind and precipitation extremes in the Mediterranean: a climatological analysis for 1979–2012 *Q. J. R. Meteorol. Soc.* **141** 2404–17

[59] Trenberth K E, Fasullo J T and Shepherd T G 2015 Attribution of climate extreme events *Nat. Clim. Change* **5** 725–30

[60] Swain D, Langenbrunner B, Neelin J and Hall A 2018 Increasing precipitation volatility in twenty-first-century california *Nat. Clim. Change* **8** 427–33

[61] Galfi V M and Lucarini V 2021 Fingerprinting heatwaves and cold spells and assessing their response to climate change using large deviation theory *Phys. Rev. Lett.* **127** 058701

[62] Seneviratne S I *et al* 2021 Weather and climate extreme events in a changing climate *IPCC 2021: Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change* ed V Masson-Delmotte *et al* (Cambridge University Press) ch 11, pp 1513–766

[63] Cérou F and Guyader A 2007 Adaptive multilevel splitting for rare event analysis *Stoch. Anal. Appl.* **25** 417–43

[64] Dematteis G, Grafke T, Onorato M and Vanden-Eijnden E 2019 Experimental evidence of hydrodynamic instantons: the universal route to rogue waves *Phys. Rev.* X **9** 041057

[65] Ragone F, Wouters J and Bouchet F 2018 Computation of extreme heat waves in climate models using a large deviation algorithm *Proc. Natl Acad. Sci. USA* **115** 24–29

[66] Gálfi V M, Lucarini V, Ragone F and Wouters J 2021 Applications of large deviation theory in geophysical fluid dynamics and climate science *Riv. Nuovo Cimento* **44** 291–363

[67] Simonnet E, Rolland J and Bouchet F 2021 Multistability and rare spontaneous transitions in barotropic β-plane turbulence *J. Atmos. Sci.* **78** 1889–911

[68] Rasp S 2019 Lorenz '96 is too easy! Machine learning research needs a more realistic toy model (available at: https://raspstephan.github.io/blog/lorenz-96-is-too-easy/)

[69] Hagan M and Menhaj M 1994 Training feedforward networks with the Marquardt algorithm *IEEE Trans. Neural Netw.* **5** 989–93

[70] Wilamowski B and Irwin J 2018 *Intelligent Systems* (CRC Press)