Chaotic prediction using weakly-synchronised trajectories and machine learning

Accepted for publication in Communications in Nonlinear Science and Numerical Simulation

Chaotic prediction using weakly-synchronised trajectories and machine learning

Anthony S. Miller[a], Zoe Bartlett[b], Chris G. Antonopoulos[a]

[a] *University of Essex, School of Mathematics, Statistics and Actuarial Science (SMSAS), Wivenhoe Park, Colchester, CO4 3SQ, United Kingdom*
[b] *Kings College London, Department of Informatics, London, WC2B 4BG, United Kingdom*

Abstract

Machine learning approaches for forecasting chaotic dynamical systems typically rely on neural networks trained on arbitrarily selected trajectory data from a target system. However, this conventional strategy often limits predictive performance to short horizons, due to the inherent sensitivity and complexity of chaotic dynamics. Here, we show that carefully curated ensembles of initial conditions spanning a large extent of an attractor, such as the Lorenz attractor, enable longer prediction horizons. This demonstrates that prediction quality depends not only on sampling more trajectories per se, but also on capturing the right range of dynamical behaviours, driven by shadowing and structural stability. As is well known, the Lorenz attractor is structurally stable, meaning small perturbations in initial conditions lead to trajectories that "shadow" each other, leading to higher synchronisation levels among them. Motivated by this insight, we introduce a novel data filtering method based on phase synchrony analysis, using the Kuramoto Order Parameter (KOP) to select training trajectories with low synchronisation levels, still evolving chaotically on the attractor. We demonstrate the approach on feedforward neural networks applied to the Lorenz system, comparing models trained on KOP-filtered datasets with those trained on unfiltered datasets. Evaluation across varying perturbation scales shows that KOP-based filtering substantially improves predictive performance. Models trained on low-KOP data achieve longer prediction horizons, exhibit superior agreement with ground truth dynamics, and produce output probability distributions closely approximating Gaussian behaviour, significantly outperforming traditional strategies that do not provide insight into the dynamical properties of the datasets. Our results highlight the importance of considering the properties of the dataset to increase dynamical diversity in the training datasets and suggest a principled pathway for enhancing machine learning-based forecasts of chaotic systems.

Keywords: Lorenz system, Lorenz attractor, chaos, Lyapunov exponents, Kuramoto order parameter, neural networks, machine learning, prediction horizon, chaotic prediction

1. Introduction

Predicting the future state of a chaotic dynamical system from time series observations is a long-standing scientific endeavour [1–3]. Although these systems are governed by deterministic equations, they exhibit extreme sensitivity to initial conditions and thus, nearby trajectories diverge exponentially. Hence, an infinitesimal error in the initial state exponentially amplifies, making long-term predictions impossible. Early work established the foundations of attractor reconstruction and nonlinear forecasting, while more recent studies have demonstrated that data-driven and machine learning methods can substantially improve predictions in chaotic regimes [4–6]. An example of such a system is the Lorenz model [7], which has become a standard test bed for assessing techniques that attempt to forecast chaotic dynamics [8].

Recent progress in machine learning, especially in the use of neural networks, has revived interest in data-driven methods for modelling and forecasting chaotic systems [9, 10]. These approaches do not need explicit information about the underlying dynamical equations and can discover patterns directly from observational data. Yet, neural networks trained on chaotic time series data frequently struggle with long-term prediction. Trained models would drift away from the actual dynamics after a brief prediction horizon (PH), typically due to compounded errors and overfitting to specific characteristics in the training data [11]. Recent advances have also demonstrated successful prediction of extreme events in chaotic systems using neural networks. For instance, Coulibaly et al. [12] showed that neural networks can forecast extreme events following bifurcation to spatiotemporal chaos in microcavity lasers, while Pammi et al. [13] demonstrated prediction of spatiotemporal extreme events in laser systems with saturable absorbers. These studies highlight the broader applicability of machine learning approaches to capturing rare and extreme dynamical behaviours in chaotic systems. Work by Jiang et al. [14] has demonstrated that chaotic systems can be leveraged for machine learning predictions by embedding data into networks of chaotic oscillators coupled through topological structures, achieving accurate predictions across diverse applications including molecular-scale, cellular-scale and organ-scale problems. While their approach uses chaotic dynamics as a computational framework for predicting real-world data, the complementary challenge of directly

Email address: am22990@essex.ac.uk (Anthony S. Miller)

predicting the evolution of chaotic systems themselves remains fundamentally dependent on the quality and diversity of training data used to learn the underlying dynamics.

An important but overlooked determinant of prediction performance is the selection and composition of training data. In much research, training trajectories are sampled randomly or according to uniform criteria such as closeness to the attractor or long integration durations, without considering the dynamical properties of the datasets [15, 16]. While pragmatic, these methods can result in training datasets that may be synchronised or correlated to such a degree that they might undermine learning and generalisation. For example, this might be the case for coupled dynamical systems exhibiting chaotic synchronisation, in which an emphasis on diversity among training data is critical for capturing the system's complex dynamics. By diversity here we mean that the datasets should be composed of trajectories with the same dynamical, correlation and phase-synchronisation levels, so avoiding datasets where some of the trajectories are more synchronised than others.

To address this, we propose using the Kuramoto Order Parameter (KOP), defined in Eq. (2), as a quantitative selection criterion for training trajectories with low synchronisation levels. KOP is a popular measure of global synchrony, originally introduced in research on coupled oscillator systems [17]. It has since had its theoretical underpinnings and applications substantially extended, most prominently by Strogatz, who established its significance in the emergence of collective dynamics in a wide range of nonlinear systems, from neural, to chemical, to ecological networks [18] to name a few. KOP computes the collective phase coherence of a system of oscillators, quantifying how synchronised the system is at any given time. Values close to 1 reflect strong synchrony (almost full synchronisation where the oscillators are nearly phase-synchronised) and values close to 0 reflect phase dispersion (oscillators are uniformly spread on the unit circle, meaning there is no global synchronisation). Applied to chaotic systems, KOP allows for a measure of collective phase alignment and thus, provides a practical metric for filtering data according to dynamical diversity.

Our analysis concentrates on the Lorenz system [7] as a case study. By removing highly synchronised trajectories and retaining only those with low-KOP values, we create training datasets that are more geometrically and statistically diverse, and consequently, produce more generalisable and temporally stable model predictions. Most importantly, the selection criterion is explicitly based on the Kuramoto order parameter: trajectories are included only if they fall below a chosen low-KOP threshold. We apply this KOP-based selection method to both training and test datasets to guarantee consistency in dynamical structure throughout the evaluation pipeline.

To assess our method, we carry out hyperparameter tuning over a set of model architectures on typical regression metrics, such as the mean squared error. We take the top ten models derived from the hyperparameter tuning performed on 100 trajectories and train the models on 100 trajectories. We then derive another set of top 10 models from hyperparameter tuning on 1,000 trajectories and train these 10 models on data containing 1,000 trajectories. In both cases no KOP screening has taken place. Subsequently, we train each set of top ten performing models on data from 100 low-KOP trajectories and then 1,000 low-KOP trajectories. Performance is measured on perturbed initial conditions with predictions compared to ground-truth solutions found using the RK45 numerical integrator [19, 20]. We examine how the prediction error changes with time and perturbation size and investigate the statistical structure of the resulting distributions of the prediction horizons.

Our findings suggest that the inclusion of a dynamical synchrony filter, using KOP, in the data selection procedure can substantially enhance model robustness and long-term predictive accuracy. In addition, we note that low-KOP training data suppresses the characteristic rapid divergence seen in neural models trained on more synchronised or unfiltered data. Our results here demonstrate that using phase-dynamical measures, such as KOP, is a well-founded approach for training data selection in machine learning models used for the prediction of chaotic, nonlinear dynamical systems.

2. Theoretical framework and background

2.1. The Lorenz system and its chaotic attractor

The Lorenz system, originally derived from simplified atmospheric convection equations, represents a paradigmatic example of a system that exhibits deterministic chaos and serves as a cornerstone for understanding non-linear dynamical behaviour [7]. Beyond its role as a paradigmatic chaotic system, the Lorenz equations also serve as a universal model describing bifurcations in nearly conservative and quasi-reversible systems [21]. This universality underscores the broader theoretical significance of our results, as improvements in predicting Lorenz dynamics may generalize to other systems undergoing similar bifurcation structures. The Lorenz system is governed by three coupled, non-linear ordinary differential equations:

$$\frac{dx}{dt} = \sigma(y - x), \quad \frac{dy}{dt} = x(\rho - z) - y, \quad \frac{dz}{dt} = xy - \beta z \tag{1}$$

where $\sigma = 10$, $\rho = 28$, and $\beta = 8/3$ represent the classical parameter values that give rise to the so-called Lorenz attractor [22]. The system exhibits sensitive dependence on initial conditions, with nearby trajectories diverging exponentially at a rate characterised by the positive Lyapunov exponent $\lambda_1 \approx 0.9036$ [23] for the above set of parameters. This fundamental property limits predictability, as small uncertainties in initial conditions grow exponentially in time, resulting in

a finite PH that scales logarithmically with initial uncertainty [24]. Hence the Lorenz attractor is a strange attractor with a fractal dimension approximately equal to 2.06 [25] indicating that although the system evolves in a three-dimensional phase space, the long-term dynamics are confined to a lower-dimensional manifold with complex and fractal geometric structure. The Lorenz attractor is structurally stable, meaning small perturbations in initial conditions resulting in trajectories that "shadow" each other, leading to higher synchronisation levels among them. Understanding this geometric complexity and phase-synchronisation exhibited by the system, is crucial for developing effective prediction strategies, as successful models must capture both the local dynamical rules and the global topological constraints imposed by the strange attractor.

2.2. Phase synchronisation and the Kuramoto order parameter

Phase synchronisation in chaotic systems manifests as coordinated patterns of activity that differ markedly from the dynamics observed in non-synchronised trajectories [26]. In particular, chaotic synchronisation is when chaotic oscillators lock their phases (maintaining a bounded phase difference) while their amplitudes remain chaotic. This is a weaker but more realistic form of synchronisation than complete synchronisation in chaotic dynamics. In such cases, while the amplitude and full trajectories of individual components may remain chaotic, their phases can become temporally aligned, revealing a subtle but significant form of synchrony. In this study, we are not analysing a network of coupled oscillators, but rather ensembles of trajectories originating from a single chaotic system, the Lorenz model in Eq. (1). We aim to ensure sufficient variability in the ensemble used for machine learning, as excessive synchronisation among trajectories would diminish the diversity of input data and potentially bias the learning process. To detect and quantify the degree of phase alignment across these trajectories, we used the Kuramoto Order Parameter (KOP) [17, 27], a well-established measure of phase coherence. This allows us to track synchrony over time by computing the collective phase alignment among multiple trajectories, providing a continuous and interpretable metric that is sensitive to the emergence of phase synchronisation.

Particularly, following [17, 27], the instantaneous KOP, $R(t)$, is defined by:

$$R(t) = \left| \frac{1}{N} \sum_{n=1}^{N} e^{i\theta_n(t)} \right| \tag{2}$$

where $\theta_n(t)$ is the instantaneous phase of the nth oscillator at time t, and N is the number of trajectories or signals considered. Here, the concept of a population of oscillators is replaced by a set of trajectories of the Lorenz system (1) and $|\cdot|$ is the modulus of the complex number given by the sum in Eq. (2). $R(t)$ quantifies the degree of phase synchronisation among a population of oscillators (dataset of trajectories in our case) and takes values in the range $0 \leq R(t) \leq 1$. When $R(t) = 1$, all trajectories are perfectly phase-locked, meaning they share the same phase and the system is fully synchronised. At the other extreme, $R(t) = 0$ indicates complete desynchronisation, where the phases of the trajectories are uniformly distributed around the unit circle and no coherent pattern emerges. Intermediate values of $R(t)$, i.e., $0 < R(t) < 1$, reflect partial synchrony, where some level of phase coherence exists despite individual differences in trajectory dynamics. Thus, the value of $R(t)$ provides a measure of the system's collective synchronisation state in time.

To investigate the phase synchronisation of the Lorenz trajectories within the datasets using Eq. (2), it is necessary to compute the *instantaneous phase*, θ, of each component of the solutions. We discuss θ given by Eq. (4) in more detail in Sec. 2.2.1. A widely used technique utilised to compute θ is the *Hilbert transform*, which provides a simple means of constructing an *analytic signal* from a real-valued time series, which in our case, is one of the components of the solutions in the dataset. This allows for the extraction of both the instantaneous amplitude and phase, θ, of all components of the trajectories in the dataset [28, 29].

2.2.1. The Hilbert transform

The Hilbert transform $\mathcal{H}[x](t)$ of a real signal $x(t)$ is defined by:

$$\mathcal{H}[x](t) = \frac{1}{\pi} \text{P.V.} \int_{-\infty}^{\infty} \frac{x(\tau)}{t - \tau} \, d\tau, \tag{3}$$

where P.V. denotes the *Cauchy principal value* of the integral. This transformation can be understood as the convolution of the signal with the impulse response function $h(t) = \frac{1}{\pi t}$:

$$\mathcal{H}[x](t) = x(t)h(t) = \frac{1}{\pi t} x(t).$$

In the frequency domain, the Hilbert transform applies a phase shift of $-\frac{\pi}{2}$ to positive frequencies and $+\frac{\pi}{2}$ to negative frequencies, effectively multiplying the Fourier transform by $-i\text{sgn}(\omega)$, where $\text{sgn}(\omega)$ is the signum function [29].

Upon computation of the Hilbert transform $\mathcal{H}[x](t)$, we construct the *analytic signal*, $z(t)$, corresponding to $x(t)$ as:

$$z(t) = x(t) + i\mathcal{H}[x](t).$$

This complex-valued signal enables us to extract the *instantaneous amplitude*, $A(t)$, and *instantaneous phase*, $\theta(t)$, as:

$$\begin{aligned} A(t) &= |z(t)| = \sqrt{x(t)^2 + \mathcal{H}[x](t)^2} \\ \theta(t) &= \arg(z(t)) = \arctan 2(\mathcal{H}[x](t), x(t)). \end{aligned} \tag{4}$$

The arctan 2 function is essential as it provides the phase angle in the full range $[-\pi, \pi]$ by considering the signs of both the real and imaginary components, ensuring correct quadrant determination. Unlike the standard arctan function, which is limited to $[-\pi/2, \pi/2]$, the arctan 2 function avoids phase ambiguities and handles cases where the real part $x(t)$ approaches zero, making it more robust for analysing chaotic signals where the trajectory can traverse all quadrants of the complex plane.

The Hilbert transform is a powerful tool for extracting instantaneous phase and amplitude via the analytic signal representation [28–30]. Its correct interpretation relies on several structural assumptions. First, it is mathematically well-defined for narrowband or monocomponent signals; when applied to broadband, multicomponent, or strongly nonstationary signals, the resulting instantaneous phase and frequency may become ambiguous or physically misleading [26, 29, 31].

Second, the method assumes sufficient smoothness and integrability of the signal. In practical implementations, finite-length data require windowing, padding, or filtering, which introduce boundary artefacts and edge effects [30, 32].

Third, the Hilbert transform assumes adequate signal-to-noise ratio and absence of strong residual trends; without appropriate preprocessing (e.g. detrending or band-pass filtering), spurious phase dynamics may arise [26, 29].

Fourth, interpretation of instantaneous frequency derived via the analytic signal implicitly assumes approximate validity of Bedrosian-type conditions [33], which are often only approximately satisfied in empirical data.

Finally, as a linear transformation, it assumes that interacting components are sufficiently separated in frequency; in the presence of mode mixing or overlapping spectral content it cannot uniquely attribute phase dynamics to distinct underlying processes [26, 34]. For these reasons, the Hilbert-based quantities we compute should be interpreted within the assumptions of approximate narrowband structure and adequate signal-to-noise ratio.

By approximate narrowband structure, we refer to signals whose spectral energy is concentrated within a relatively small frequency interval around a dominant carrier frequency [29]. More precisely, a signal $x(t)$ is approximately narrowband if it can be represented in the form

$$x(t) = A(t)\cos\big(\phi(t)\big),$$

where the amplitude $A(t)$ varies slowly compared to the oscillatory timescale, and the instantaneous frequency $\omega(t) = \phi'(t)$ also varies slowly in time. Equivalently, the Fourier spectrum of $x(t)$ is concentrated around a central frequency ω_0, with bandwidth $\Delta\omega$ satisfying

$$\Delta\omega \ll \omega_0.$$

Under these assumptions, the analytic signal constructed via the Hilbert transform yields a well-defined and physically interpretable instantaneous phase and frequency [26, 29].

By Bedrosian-type conditions, we refer to the assumption that the amplitude envelope and the carrier oscillation of a signal occupy well-separated frequency bands [33]. For a signal written as

$$x(t) = A(t)\cos(\phi(t)),$$

these assumptions are approximately satisfied when the envelope $A(t)$ varies slowly (i.e. is concentrated at low frequencies), while the carrier $\cos(\phi(t))$ is narrowband around a dominant non-zero frequency ω_0, with minimal spectral overlap between the two. When such approximate spectral separation holds, the Hilbert transform produces a meaningful analytic signal and well-defined instantaneous phase and frequency [29, 31].

We note, however, that in more general dynamical settings the Hilbert transform may still be used even when some of the above assumptions are only partially satisfied. In practice, appropriate preprocessing (e.g. band-pass filtering, detrending, or empirical mode decomposition) can improve approximate narrowband structure and reduce spectral overlap [26, 34]. Moreover, for chaotic or weakly multicomponent oscillatory systems, the Hilbert phase can serve as a consistent *protophase* that captures the dominant rotational dynamics of the attractor, even if strict Bedrosian conditions are not met [26, 31]. In such cases, the extracted phase should be interpreted geometrically rather than strictly spectrally, and its validity can be assessed by consistency checks (e.g. phase continuity, monotonicity, or comparison with alternative phase definitions). Accordingly, while the assumptions above provide the ideal theoretical framework, controlled relaxation of these assumptions does not preclude the meaningful application of the Hilbert transform in practice.

2.2.2. Application to the Lorenz system

In the case of system (1), the state variables $x(t)$, $y(t)$, and $z(t)$ can be thought of as time series data from which a phase, θ, for each would be extracted. The $x(t)$ or $y(t)$ component is typically taken, particularly if there is interest in oscillatory activity in the vicinity of attractors. Following the calculation of the instantaneous phases $\theta_j(t)$ for all oscillators or trajectories using the Hilbert transform, the phases can subsequently be used to compute the *Kuramoto Order Parameter*:

$$R(t)e^{i\psi(t)} = \frac{1}{N}\sum_{n=1}^{N} e^{i\theta_n(t)}, \tag{5}$$

where $R(t) \in [0, 1]$ measures the *level of synchrony* (0 for complete desynchronisation and 1 for complete synchronisation), $\psi(t)$ is the *average phase* of all trajectories, and N is the number of oscillators (or trajectories, if using machine learning ensembles as in our case). This metric enables the quantification of phase coherence among the trajectories of the machine learning model, enabling a more dynamical-system-compliant evaluation of model performance [26, 35].

While the Lorenz attractor does not strictly satisfy narrowband or Bedrosian-type conditions required for ideal Hilbert phase extraction, due to its broadband chaotic spectrum and amplitude–phase coupling, after appropriate preprocessing and in regimes of coherent rotational dynamics, the analytic signal yields a consistent protophase that captures the system's effective oscillatory structure.

With respect to the Lorenz system (1), KOP enables the detection of synchronisation among various realisations of the x-component of the system, which is often employed as the variable in prediction experiments. Values of R close to 1 are characteristic of globally synchronised phase dynamics in the trajectory ensemble, and small values are representative of desynchronised dynamics. This has serious implications for machine learning algorithms whose goal is to predict the x, y or z components. Datasets with R values close to 1 may very well have inherent structural constraints on phase-space exploration and hence manifest apparently high predictability during both training and test stages as sensitivity to initial conditions is reduced due to the "shadowing" effect [36]. The filtering of trajectory ensembles using R thus, constitutes a critical first step in ascertaining that training and test datasets accurately reflect the variability of the chaotic domain of the system.

The Kuramoto order parameter does not require oscillators to be harmonic; it requires only that a meaningful phase variable can be defined for each unit. For nonlinear or non-sinusoidal oscillators, the phase can be constructed through standard phase-reduction techniques. In particular, if the system possesses a stable limit cycle, one may define a geometric (isochronal) phase via isochrons in the basin of attraction [37, 38]. This construction yields a well-defined phase variable even when the waveform is strongly non-sinusoidal. In experimental or time-series settings, the phase may be extracted from the analytic signal using the Hilbert transform under approximate narrowband conditions [26], or via alternative constructions such as Poincaré sections or protophase-to-phase transformations [39]. Once a consistent phase variable $\theta_i(t)$ is defined for each oscillator, KOP remains well-defined, independently of waveform shape or nonlinear amplitude dynamics [27]. It is worth noting that the meaningful application of KOP works best when: (i) coherent oscillatory behaviour is present (e.g. stable limit-cycle dynamics), and (ii) a consistent and comparable phase definition can be established across oscillators. In strongly nonstationary, broadband, or non-oscillatory regimes, phase extraction warrants additional care and the interpretation of $R(t)$ should be approached accordingly [26, 40].

2.3. Limitations of high-KOP training data for chaotic system modelling

The inability of high-KOP training sets to yield neural models that generalise well for chaotic systems is because neural networks make use of the artificial coherence and low variability that are characteristic of globally synchronised states, rather than learning the basic non-linear dynamical rules that govern the system's evolution. This behaviour creates a biased numerical probability density function, in which the model's learned representations tend to over-represent regular dynamical regimes that only occupy a tiny fraction of the overall dynamical space of the system [27, 41].

Geometrically, these high-KOP states are equivalent to orbits probing the attractor in a biased way, spending prolonged periods in phase space regions where there is long-term synchrony due to structural stability and "shadowing" [36], and without sufficiently exploring areas where rapid phase desynchronisation and more complicated transient behaviour dominate. Moreover, models trained on high-KOP data may fail to capture rare dynamical events or extreme behaviours [12, 13], as these phenomena typically occur in desynchronised, high-diversity regions of the attractor that are underrepresented in synchronised training sets. This form of bias for selective sampling can be seen within the framework of *ergodicity breaking* theory [42, 43].

In fully chaotic systems, ergodicity ensures that time averages along individual trajectories are equivalent to ensemble averages over the entire attractor. However, persistently high-KOP trajectories violate this principle by excessively dwelling in low-dimensional manifolds within the attractor, thus hampering the learning algorithm's ability to explore more chaotic regions. As a result, models trained on high-KOP data learn conditional distributions $P(x_{t+1}|x_t)$ that are valid only within the constrained dynamical regime of global synchrony. Consequently, the models generalise poorly to more heterogeneous or weakly synchronised regimes [44].

The temporal correlation structure in high-KOP data exacerbates this issue by introducing spurious long-range predictability. The artificial correlations discovered are beyond the inherent predictability limits of chaotic systems, and thus neural networks learn non-robust patterns that extend far beyond the synchronised regime. Such models are prone to producing overconfident forecasts and biased, non-Gaussian error distributions when inferring. Their performance is highly contingent on how extensively test trajectories are extended into equally high-KOP regimes, thus heavily restricting their applicability to the larger class of chaotic dynamical behaviours represented by the overall system [4, 45].

Incorporating a dynamic synchronisation filter using KOP in the data selection process significantly enhances the model's robustness and long-term predictive accuracy for several reasons. First, synchronisation-based filtering improves attractor coverage by retaining segments that traverse broader regions of state space while preserving coherent phase organisation. This reduces extrapolation during forecasting and improves generalisation across dynamical regimes. Second, it mitigates the influence of short-lived transients and incoherent fluctuations, thereby increasing the effective signal-to-noise ratio of the training data. Third, synchronised regimes exhibit stronger collective structure and effective dimensional reduction, which stabilises the mapping learned by the neural network and reduces overfitting. Finally, long-term forecasting in chaotic systems is highly sensitive to phase drift; training on phase-coherent intervals improves phase continuity and reduces error amplification under iterative prediction. Importantly, the filter does not constrain the dynamics artificially; rather, it selects segments that are statistically and geometrically more representative of the attractor's invariant structure.

3. Methodology

3.1. Data description: Trajectory generation and data preprocessing

To ensure a comprehensive representation of the Lorenz attractor, we generated training and prediction datasets through a two-stage process:

Training Data Generation

Starting with random initial conditions drawn from the uniform distribution over $S = (x_0, y_0, z_0) \in [-0.5, 29.5) \times [-0.5, 29.5) \times [-0.5, 29.5)$, we integrated the Lorenz system for $T = 50$ (5,000 time steps with $\Delta t = 0.01$) to allow trajectories to evolve onto the attractor. We then perturbed only the x-component of each endpoint by 10^{-2} and integrated forward for an additional $T = 20$ (2,000 time steps). This procedure yielded two training datasets: S_1 containing 100 trajectories and S_2 containing 1,000 trajectories. The ten-fold increase in dataset size was chosen deliberately, in order to assess whether an N-fold increase in the number of training trajectories results in a commensurate improvement in predictive performance, and to investigate the extent to which dataset size interacts with the KOP-filtering procedure. Hyperparameter tuning then produced 1,200 models for both S_1 and a further 1,200 models for S_2, with the top 10 models selected in each case based on the smallest Mean Squared Error (MSE) evaluated on the respective S_1 and S_2 training datasets. We then trained these top 10 neural network models for S_1 and S_2 on their respective datasets. The top 10 model architectures selected are detailed in Tables 1 and 2.

Prediction Data Generation

To evaluate model performance, we generated two groupings of initial conditions

(a) Grouping 1 (measuring the sensitivity to perturbations): Using the same initial conditions from the training datasets S_1 and S_2, we created eight prediction datasets by perturbing each component of every initial condition by 10^{-15}, 10^{-13}, 10^{-11}, 10^{-9}, 10^{-7}, 10^{-5}, 10^{-3}, and 10^{-1}. These perturbation magnitudes were applied sequentially to generate sets $P_1(S_1), P_2(S_1), \ldots, P_8(S_1)$ and $P_1(S_2), P_2(S_2), \ldots, P_8(S_2)$, allowing us to assess models sensitivity to initial condition variations.

(b) Grouping 2 (measuring the generalisation to unseen data): To test generalisation beyond the training set distribution, we generated five independent sets of 1,000 initial conditions each using different random seeds drawn from a uniform distribution over $S = (x_0, y_0, z_0) \in [-0.5, 29.5) \times [-0.5, 29.5) \times [-0.5, 29.5)$. Each set underwent the same integration procedure as the training data (integrate for $T = 50$, perturb x-component by 10^{-2}, integrate for $T = 20$) to place initial conditions on the attractor while ensuring they were distinct from any training examples.

3.2. Neural network architecture

We implemented a fully connected feedforward artificial neural network, also referred to as a multi-layer perceptron (MLP), constructed using the Keras `Sequential` API in Python. The network is designed to accept as input a three-dimensional vector representing the state variables (x, y, z) of the Lorenz system (1) at a given time step.

The architecture consists of two hidden layers and an output layer, with the number of neurons and activation functions in each layer treated as tunable hyperparameters. Specifically: (a) The first hidden layer receives the three-dimensional input and contains `_layer1` neurons, with an activation function specified by `_activation_fn_1`. (b) The second hidden layer contains `_layer2` neurons, using `_activation_fn_2` as its activation function. (c) The output layer consists of three neurons with a linear activation function, corresponding to the predicted x, y, and z coordinates at the next time step. The

specific values used for `_layer1`, `_layer2`, `_activation_fn_1` and `_activation_fn_2` are shown in Tables 1 and 2. This architecture offers flexibility through parametrisation, allowing systematic tuning of the model's complexity and non-linearity. The use of a linear activation function in the output layer is appropriate for the regression task we are employing, where the goal is to predict continuous-valued outputs from a chaotic dynamical system.

Mathematically, this can be represented as follows: Let the input vector be denoted as $\mathbf{x}_t = [x, y, z]^T \in \mathbb{R}^3$, representing the state of the Lorenz system (1) at time step t. The neural network maps this input through two hidden layers and an output layer to produce a prediction $\hat{\mathbf{x}}_{t+1} = [\hat{x}, \hat{y}, \hat{z}]^T \in \mathbb{R}^3$, corresponding to the estimated state at time step $(t+1)$:

$$\begin{aligned}
\mathbf{h}_1 &= \sigma_1\left(\mathbf{W}_1\mathbf{x}_t + \mathbf{b}_1\right), \quad \mathbf{W}_1 \in \mathbb{R}^{n_1\times 3},\ \mathbf{b}_1 \in \mathbb{R}^{n_1},\\
\mathbf{h}_2 &= \sigma_2\left(\mathbf{W}_2\mathbf{h}_1 + \mathbf{b}_2\right), \quad \mathbf{W}_2 \in \mathbb{R}^{n_2\times n_1},\ \mathbf{b}_2 \in \mathbb{R}^{n_2},\\
\hat{\mathbf{x}}_{t+1} &= \mathbf{W}_3\mathbf{h}_2 + \mathbf{b}_3, \quad \mathbf{W}_3 \in \mathbb{R}^{3\times n_2},\ \mathbf{b}_3 \in \mathbb{R}^3,
\end{aligned}$$

where $\mathbf{h}_1 \in \mathbb{R}^{n_1}$ and $\mathbf{h}_2 \in \mathbb{R}^{n_2}$ denote the outputs of the first and second hidden layers, respectively and where all vectors are column vectors (e.g., $\mathbf{x}_t \in \mathbb{R}^{3\times 1}$). Here, n_1 and n_2 represent the number of neurons in these layers, corresponding to `_layer1` and `_layer2`, $\sigma_1(\cdot)$ and $\sigma_2(\cdot)$ are the activation functions for these layers, corresponding to `_activation_fn_1` and `_activation_fn_2`, and the output layer uses a linear activation function.

This structure defines a parametric function $f_\vartheta : \mathbb{R}^3 \to \mathbb{R}^3$, where ϑ represents the set of all learnable parameters $\{\mathbf{W}_1, \mathbf{b}_1, \mathbf{W}_2, \mathbf{b}_2, \mathbf{W}_3, \mathbf{b}_3\}$. The model is trained to minimise a loss function that measures the discrepancy between the predicted, $\hat{\mathbf{x}}_{t+1}$, and true, $\mathbf{x}_{t+1}$, future states of the system. In our case, we have used the mean squared error (MSE) as a loss function and discuss the details in the following section.

Our design enables systematic comparison of model performance across different training dataset sizes (100 vs. 1,000 trajectories) and hyperparameter configurations, as shown in Table 3, providing a comprehensive evaluation of neural network architectures for capturing the complex non-linear dynamics of the Lorenz chaotic attractor.

Model	**Layer1 Neurons**	**Layer2 Neurons**	**Activation1**	**Activation2**	**Learning Rate**	**Optimiser**
Model A	30	25	Sigmoid	Sigmoid	0.001	Adamax
Model B	30	30	Sigmoid	Sigmoid	0.001	Adamax
Model C	30	30	Tanh	Sigmoid	0.0001	NAdam
Model D	30	30	Tanh	Sigmoid	0.001	Adamax
Model E	25	30	Tanh	Sigmoid	0.0001	Adam
Model F	25	30	Tanh	Tanh	0.0001	NAdam
Model G	20	30	Sigmoid	Sigmoid	0.001	Adam
Model H	30	30	Sigmoid	Tanh	0.001	NAdam
Model I	15	30	Tanh	Sigmoid	0.001	NAdam
Model J	30	30	Sigmoid	Tanh	0.001	Adamax

Table 1: The configuration parameters after hyperparameter tuning for Models A through to J trained on 100 trajectories (grouping S_1) with learning rates of 0.001 (Models A, B, D, G, H, I, J) and 0.0001 (Models C, E, F), employing Adamax [46], NAdam [47], and Adam [46] optimisers.

3.3. Model Training

In order to train the models, we firstly perform hyperparameter tuning (described in Subsec 3.3.1) to determine the top 10 architectural structures for training sets S_1 and S_2 (as described in 3.1). Once these architectural structures were defined, we then trained the models using a custom metric, detailed in 3.3.2, to determine the optimum number of training epochs to avoid overtraining and underfitting. Once we have defined the optimal structures and epochs we train the models derived and shown in Tables 1 and 2, in preparation for the prediction phase.

3.3.1. Hyperparameter tuning

To optimise the model architecture and training configuration, a structured hyperparameter search, using a grid search strategy, was conducted training sets S_1 and S_2 . Table 1 shows the configuration parameters for ten MLP models trained on the 100-trajectory training set (S_1). Similarly, Table 2 shows the configuration parameters for an additional ten MLP

Model	Layer1 Neurons	Layer2 Neurons	Activation1	Activation2	Learning Rate	Optimiser
Model K	15	30	Sigmoid	Tanh	0.0001	NAdam
Model L	25	25	Softmax	ReLU	0.0001	NAdam
Model M	25	25	Sigmoid	Tanh	0.0001	NAdam
Model N	30	20	Softmax	Tanh	0.0001	NAdam
Model O	25	30	Sigmoid	Sigmoid	0.0001	Adam
Model P	25	20	Sigmoid	Sigmoid	0.0001	NAdam
Model Q	30	30	Softmax	ReLU	0.0001	NAdam
Model R	15	30	Sigmoid	Tanh	0.0001	NAdam
Model S	30	30	Sigmoid	Tanh	0.0001	Adam
Model T	25	30	Tanh	Tanh	0.0001	NAdam

Table 2: The configuration parameters after hyperparameter tuning for Models K through to T trained on 1,000 trajectories (grouping S_2) with a consistent learning rate of 0.0001, predominantly using NAdam optimisation (8 models) with Adam employed for Models O and S.

models trained on the 1,000-trajectory training set (S_2). 1200 models were generated in the hyperparameter tuning stage for each training set S_1 and S_2 and the model performance assessed on MSE to determine in the top 10 in each case. It is worth noting that the variation in learning rates between the two tables reflects the optimisation requirements at different training dataset scales: larger datasets, and by extension larger batch sizes, produce more accurate gradient estimates, necessitating lower learning rates to prevent overfitting and ensure stable convergence [48, 49].

Hyperparameter	Values Tested
Neurons per hidden layer	5, 10, 15, 20, 25, 30
Optimisers	SGD, RMSprop, Adam, AdamW, NAdam, Adamax, Adadelta
Learning rates	10^{-2}, 10^{-3}, 10^{-4}
Activation functions	Sigmoid, ReLU, Tanh, Softmax

Table 3: The hyperparameter search space used in model tuning.

3.3.2. Custom epoch measurement metric

A common challenge in training neural networks is the selection of an appropriate number of training epochs. Often, this decision is based on heuristics or prior experience, leading to suboptimal outcomes [50, 51]. Excessive training can result in overfitting, where the model learns spurious patterns in the training data that do not generalise to unseen inputs. Conversely, insufficient training (underfitting) fails to capture the underlying dynamics of the system being modelled.

Before fully training the neural network models defined in Tables 1 and 2, we implemented a more formalised approach to finding an optimum number of training epochs to predict the x, y and z components of the Lorenz system (1). Instead of depending on early stopping procedures or ad hoc limits on epoch numbers, our novel method fuses two distinct measures that simultaneously encompass model performance and dynamical consistency within a single framework for epoch evaluation.

Our custom epoch measure metric combines the MSE and PH to capture complementary aspects of a models performance. The MSE is defined by:

$$\text{MSE} = \frac{1}{T_p} \sum_{t=1}^{T_p} \|\hat{\mathbf{x}}(t) - \mathbf{x}(t)\|^2 , \tag{6}$$

where T_p is a prediction horizon of fixed length. MSE is a standard pointwise measure that evaluates the average deviation of predicted, $\hat{\mathbf{x}}(t)$, from observed, $\mathbf{x}(t)$, values, thus measuring the accuracy of predictions. At the same time, the PH

measures how long the predicted trajectory remains within a pre-specified error tolerance ($f = 0.4$ following the approach in [11]), of the ground truth values (as provided by RK45). Thus, measuring the temporal reliability of the model.

To examine predictability, we generated predictions over a horizon of fixed length $T_p = 20$ steps and evaluated performance according to our custom MSE-PH measure.

The custom metric finds the intersection between the peak prediction horizon and minimum of the MSE. This can be seen in Fig. 1. During training we observed that PH typically increases as the model begins to learn, peaks, and then declines as overfitting sets in, while MSE continues to decrease or plateau. The intersection point where PH reaches its maximum and MSE approaches its minimum represents a trade-off between overfitting and underfitting. This point was taken as the optimal epoch count for the model.

Most importantly, during this process, we set the batch size to be the length of the number of time points we are investigating. Here we set $T_p = 20$, with $\Delta t = 0.01$, giving us a batch size of 2001. This approach removed the stochastic variation introduced by mini-batching, resulting in smoother convergence behaviour and more consistent metric tracking across epochs.

This two-metric approach guarantees that the chosen number of epochs maximises both short-term predictive accuracy and long-term dynamical coherence, resulting in a more comprehensive model selection criterion.

To ensure robustness, this procedure was repeated across the two sets of 10 MLP models for S_1 and S_2 (see Tables 1 and 2). The resulting intersection points were averaged, yielding an empirically derived optimal epoch count of 365 for the training dataset. This strategy provides a dynamically grounded, data-driven alternative to conventional heuristics, especially relevant for chaotic systems where traditional validation loss alone may not fully capture the stability of temporal predictions.

3.3.3. Final model training in preparation for prediction

After we have derived the optimal number of epochs and optimal architectural structures as described in Tables 1 and 2, we train the models, reverting to a batch size of 32, removing the custom metric and measure the performance using the MSE alone. All neural network models were implemented using the `Keras` API with a `TensorFlow` backend in Python.

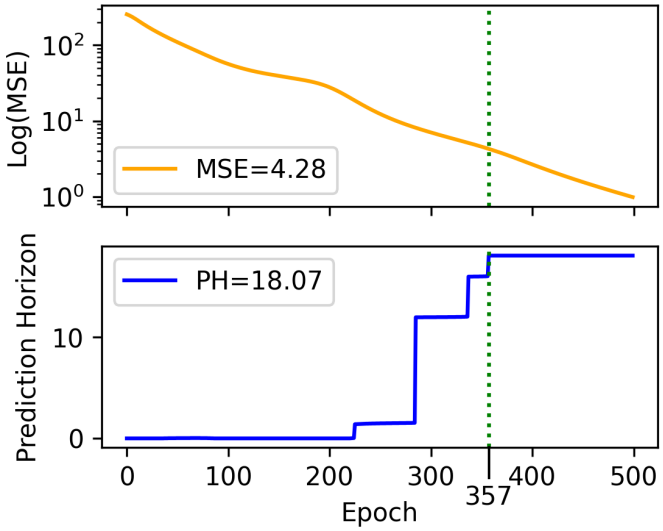
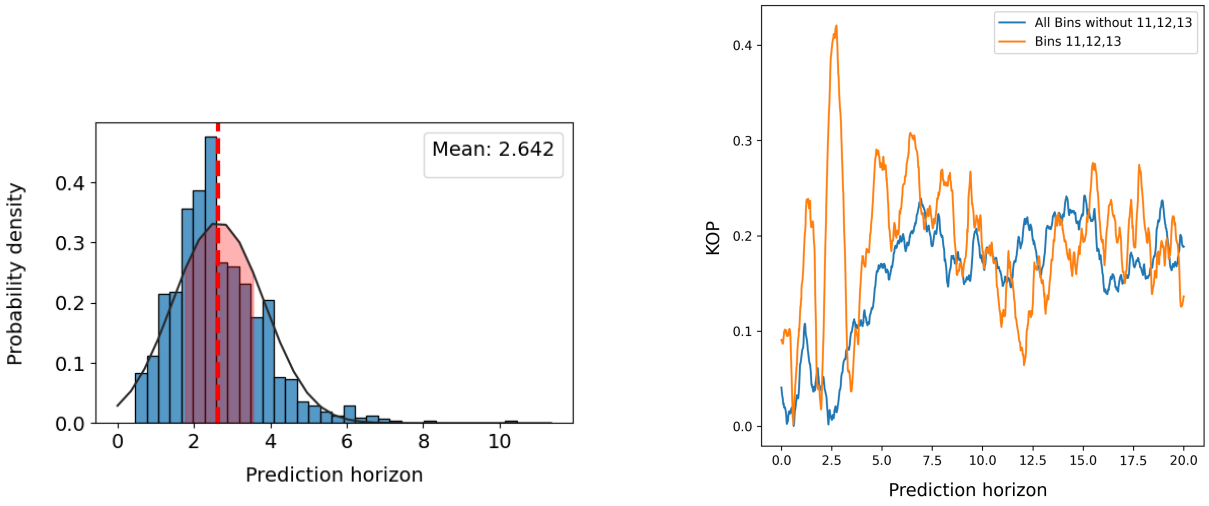


Figure 1: Evolution of log-scaled MSE and PH during training of Model A on 2000 trajectories over 500 epochs. While MSE decreases steadily, PH increases and then saturates, indicating a trade-off between learning and overfitting. The vertical green line marks the empirically selected optimal epoch (epoch 357 in this instance), where PH is near its peak (18.07 Lyapunov Times) and MSE is low (4.28). This criterion, averaged over 10 independently initialised MLPs, yielded a robust stopping point at 365 epochs.

3.4. Kuramoto order parameter filtered datasets

Initial training and model development were carried out using datasets without any filtering or evaluation based on their KOP values. Although these baseline models yielded different prediction horizons, a closer inspection of their probability density functions (PDFs) revealed substantial skewing to the left of the mean prediction horizon, particularly within specific histogram bins as can be seen in Fig. 2(a). Gaussian fitting of the output PDFs demonstrated this skew clearly, with three bins to the left of the mean exhibiting heightened asymmetry and departure from the Gaussian PDF. Trajectory analysis further showed that these bins corresponded to trajectories with elevated KOP values as opposed to the KOP values of the remaining trajectories in the datasets, seen in Fig. 2(b), suggesting a relation between elevated KOP values and skewness to the left of the mean of the PDF of the prediction horizons, seen in Fig. 2(a).



(a) (b)

Figure 2: PDF and KOP results from Model P where the initial conditions were perturbed by 10^{-5}. Panel (a): An example of a PDF plot of prediction horizons. Here we use 33 bins, which highlight the three bins to the left of the mean (red dashed line), leading to a skewing of the distribution of the prediction horizons, non-conforming to the Gaussian distribution. Panel (b): KOP values of the predicted trajectories from Model P within bins 11, 12 & 13 compared to all other bins from the histogram in panel (a).

This discovery, an example of which is shown in Fig. 2(b), highlighted the necessity to look into the synchronisation patterns of the trajectories in the dataset and motivated the construction of a revised dataset, explicitly curated to contain only low-KOP dynamics as computed by using Eq. (2). By low-KOP dynamics, we mean datasets of trajectories with a very low Kuramoto order parameter R. In this work, we have opted for $R < 0.053$ to ensure there is almost no synchronisation in the trajectory datasets.

Models A-J (Table 1) and K-T (Table 2) were retrained using the low KOP-adapted dataset, while preserving their original architectural hyperparameters, such as neuron counts, activation functions, and structural layers. This retraining phase aimed to isolate and assess the effect of KOP suppression on output prediction accuracy.

To test the generalisability of these low-KOP-trained models, we used the same approach as before, and created datasets by perturbing the x, y and z components of the training trajectories by varying degrees of external perturbation (10^{-15}, 10^{-13}, 10^{-11}, …, 10^{-1}). Preliminary testing demonstrated notable improvements in the Gaussian fitting of the output PDFs, meaning that the numerical PDF conformed better to the Gaussian PDF. These preliminary results indicated enhanced distributional consistency when models are trained on low-KOP data, that is data with $R < 0.053$.

To further substantiate these findings, five additional datasets were generated using different random seeds, all verified to maintain minimal KOP ($R < 0.053$). The combination of these differing datasets gave rise to three distinct cases: (1) models trained and tested without any explicit control over KOP, (2) models trained on low-KOP datasets and tested on datasets with externally perturbed trajectories, and (3) models trained and evaluated exclusively on independently generated low-KOP datasets.

The improved distributional alignment observed in these experiments (for example see the PDF in Fig. 3) supports the conclusion that datasets with trajectories with high-KOP values contribute to skewed PDFs. Hence, we will continue with refined data curation strategies that lead to low-KOP dynamics. This approach is expected to generate datasets with numerical PDFs conforming better to the Gaussian PDF.

3.5. Computation of the probability density functions

To evaluate how effectively our machine learning models capture the statistical behaviour of the Lorenz system (1) over finite-time forecasts, we analysed the PDFs of the prediction horizons. Instead of examining the PDFs of the raw state variables $x(t)$, $y(t)$, or $z(t)$ over long time series, we focused on the distribution of values specifically within the fixed-length forecast windows produced by the models. This approach captures how uncertainty and spread evolve within prediction intervals, providing insight into the model's ability to emulate the system's inherent variability over short- to medium-term horizons.

In particular, we employed a histogram-based method to estimate these numerical PDFs. For a given prediction horizon, we aggregated the predicted values across an ensemble of realisations or initial conditions and constructed numerical PDFs for each time step in the horizon. At a particular forecast step, given a dataset $\{x_i\}_{i=1}^{T}$, the probability of having a

data point in bin k is given by:

$$\hat{p}(x_k) = \frac{n_k}{Th}, \tag{7}$$

where n_k is the number of values in bin k, h is the bin width, and T is the total number of predicted samples at that time step. The set of all $\hat{p}(x_k)$ probabilities is the numerical PDF of the predicted horizons. The normalisation in Eq. (7) ensures that the resulting histogram integrates to one, conforming to the properties of a PDF.

3.5.1. Selection of the optimal number of bins: The Freedman–Diaconis rule

An essential step in histogram-based PDF estimation is the choice of bin width or number of bins. An inappropriate bin width can either obscure important features or introduce noise. We used the *Freedman–Diaconis rule* [52] to determine an optimal bin width and consequently, the number of bins, adaptively based on the statistical dispersion of the forecasted values. The Freedman–Diaconis rule defines the bin width h as:

$$h = 2\frac{\text{IQR}(x)}{T^{1/3}}, \tag{8}$$

where $\text{IQR}(x)$ is the interquartile range of the data. This choice is robust against outliers and well suited to the often skewed and multi-modal distributions encountered in chaotic systems. The corresponding number of bins is then calculated as:

$$\text{Number of bins} = \left\lceil \frac{\max(x) - \min(x)}{h} \right\rceil, \tag{9}$$

where $\lceil\ \rceil$ denotes the ceiling function, which rounds up to the nearest integer to ensure a whole number of bins. The number of bins in Eq. (9) is the maximum number of bins discussed in Subsec. 3.5 and used throughout the paper.

This data-driven approach ensures that the histogram reflects meaningful features in the predictive distribution, rather than artefacts of sampling or resolution.

3.5.2. Implementation in Python

In implementation, we used Python's `numpy.histogram` function to compute numerically the PDFs of the resulting prediction horizons, specifying the number of bins according to the Freedman-Diaconis rule (9). In particular, we collected the prediction horizons resulting from the use of the different models in Tables 1 and 2 and computed 10 PDFs for the models in Table 1 and another 10 PDFs for the models in Table 2, following the idea in Subsec: 3.5. In the paper, we present those cases of PDFs that conformed well with the Gaussian PDF, denoted by solid black curves in the figures, superimposed on the numerical PDFs represented by histograms (for example, see the PDF in Fig. 2(a)).

3.6. Prediction framework and evaluation

To evaluate the predictive accuracy of the MLP neural network, we calculate the relative error between the ground truth trajectory (computed using RK45) and the predicted trajectory over time, for multiples of the Lyapunov time of the system. The Lyapunov time is a characteristic timescale that quantifies how quickly nearby trajectories in a dynamical system diverge due to sensitivity to initial conditions. It is defined in terms of the largest Lyapunov exponent [53, 54], which is a measure of the average exponential rate of divergence of infinitesimally close trajectories in phase space [53, 54]. The Lyapunov time is defined by:

$$LT = \frac{1}{\lambda_{\max}}, \tag{10}$$

where LT is the Lyapunov time and $\lambda_{\max}$ is the largest Lyapunov exponent of the system. It represents the time it takes for a small perturbation in the initial conditions to grow by a factor of e, due to exponential divergence:

$$\delta(t) \approx \delta_0 e^{\lambda_{\max} t}, \tag{11}$$

where δ_0, $\delta(t)$ are the initial separation and separation at time t of two trajectories. After a time LT, the deviation becomes approximately:

$$\delta(LT) \approx \delta_0 e. \tag{12}$$

This exponential growth implies that the system becomes unpredictable beyond a few Lyapunov times.

The relative error at time t is defined as:

$$E(t) = \frac{\|\mathbf{x}_{\text{true}}(t) - \mathbf{x}_{\text{pred}}(t)\|_2}{\|\mathbf{x}_{\text{true}}(t)\|_2}, \tag{13}$$

where $\mathbf{x}_{\text{true}}(t) = [x_{\text{true}}(t), y_{\text{true}}(t), z_{\text{true}}(t)]^T$ represents the ground truth state vector obtained from the RK45 integration, $\mathbf{x}_{\text{pred}}(t) = [x_{\text{pred}}(t), y_{\text{pred}}(t), z_{\text{pred}}(t)]^T$ represents the predicted state vector from the neural network, and $\|\cdot\|_2$ denotes the Euclidean norm.

This relative error metric provides a normalised measure of the prediction accuracy that accounts for the magnitude of the true state vector. The error is computed at each time step of the trajectory, allowing for analysis of how the neural network's predictive capability degrades over the Lyapunov time scale. For the Lorenz system (1) with standard parameters $\sigma = 10$, $\rho = 28$, $\beta = 8/3$, the maximum Lyapunov exponent $\lambda_{\max} \approx 0.9036$ that corresponds to a Lyapunov time (10) of approximately 1.1 time units, beyond which small errors in initial conditions lead to exponential divergence in chaotic trajectories, and thus to unpredictability.

Following the approach in Pathak et al. [11], we define a prediction as tolerable if the relative error remains below a fixed threshold. Specifically, a threshold value of $f = 0.4$ is adopted, beyond which the predicted trajectory is considered to have diverged significantly from the ground truth trajectory and is no longer deemed reliable. This threshold serves as an operational definition for the temporal limit of predictive validity in chaotic systems.

4. Results

We evaluate the performance of our models in Tables 1 and 2, using a combination of prediction horizon statistics, direct trajectory comparisons against the RK45 ground truth, and distributional metrics that account for the sensitivity of the initial conditions. All models were trained on the Lorenz system (1) using unfiltered and KOP-filtered trajectory data (on both S_1 and S_2), where filtering was conducted using $R < 0.053$ as a KOP threshold.

4.1. Prediction horizon comparison

Following [11], PH was defined as the first time point where the L_2 error between the predicted and reference trajectories exceeded the fixed threshold $f = 0.4$. In our simulations, we have seen that across multiple perturbation scales, models trained on low-KOP trajectories consistently outperformed those trained on unfiltered or high-KOP datasets, as demonstrated by comparing the probability density functions in Fig. 6 (unfiltered data) with those in Figs. 7 and 8 (KOP-filtered data). For instance, Model Q trained on unfiltered data at perturbation level 10^{-5} achieved a mean prediction horizon of 2.270 Lyapunov times (Fig. 6(a)), whereas the same model trained on low-KOP filtered data achieved a mean prediction horizon of 2.89 Lyapunov times (Fig. 7(a)), representing a 27% improvement. Results for all models at perturbation level 10^{-5} when trained on 1,000 low-KOP trajectories, are summarised in Table 5.

This result was consistent across ten independent training runs, underscoring the robustness of the screening approach. These findings suggest that using only weakly synchronised trajectories, i.e. filtered ones with $R < 0.053$, the numerical PDFs better conform to the Gaussian PDF and that the mean prediction horizons increase.

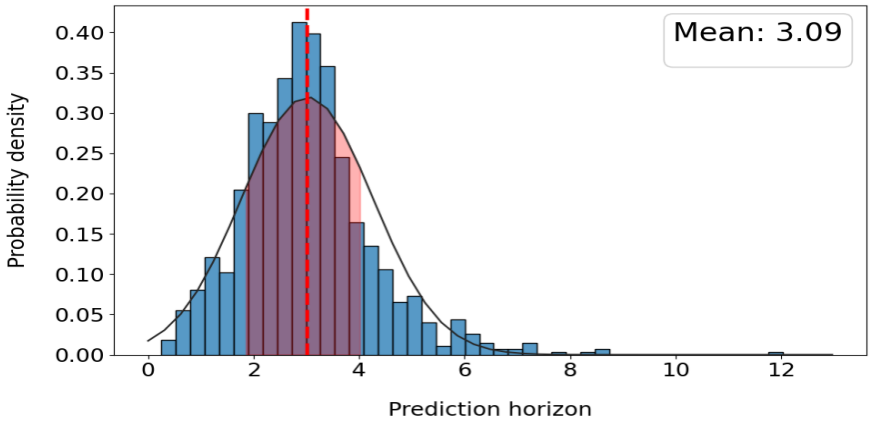


Figure 3: PDF plot of PH from Model R: A low-KOP value set at 0.053 was used for testing, yielding a near-Gaussian fit with a mean of 3.09 (denoted by the red dashed line) and standard deviation of 1.29. The pink bars denote prediction horizons within one standard deviation around the mean.

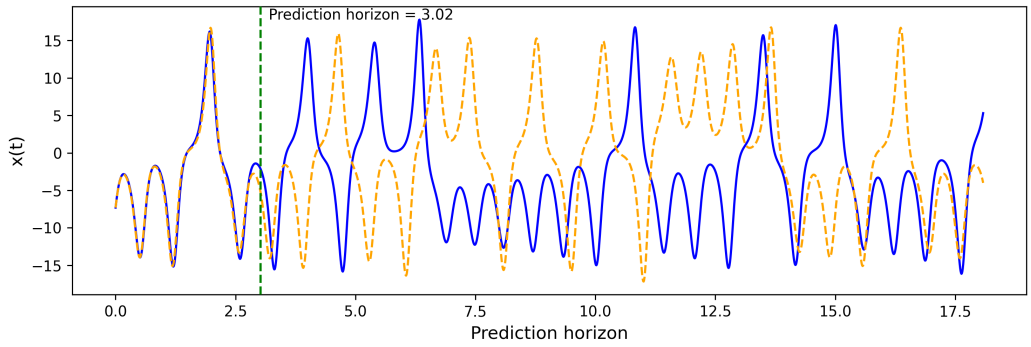
We note that Table 4 does not include a KOP-filtered comparison column equivalent to that presented in Table 5 for S_2. This asymmetry reflects a practical constraint of the S_1 dataset: with only 100 trajectories available, applying the low-KOP selection criterion ($R < 0.053$) did not yield a sufficiently distinct filtered subset. The limited pool size meant that the synchronisation properties of the retained trajectories were not statistically separable from those of the full unfiltered set, producing negligible differences in prediction performance. KOP-based filtering therefore requires a minimum trajectory pool of sufficient size to ensure that a genuinely low-synchrony subset can be isolated, a condition met by S_2 (1,000 trajectories) but not by S_1. This practical threshold represents an important consideration for the application of KOP filtering in future work.

Model	Mean PH	Std Dev
Model A	1.63	1.00
Model B	1.11	0.97
Model C	2.56	1.05
Model D	0.79	0.43
Model E	1.35	0.85
Model F	0.92	0.52
Model G	0.84	0.59
Model H	1.30	0.62
Model I	1.63	0.93
Model J	1.13	0.69

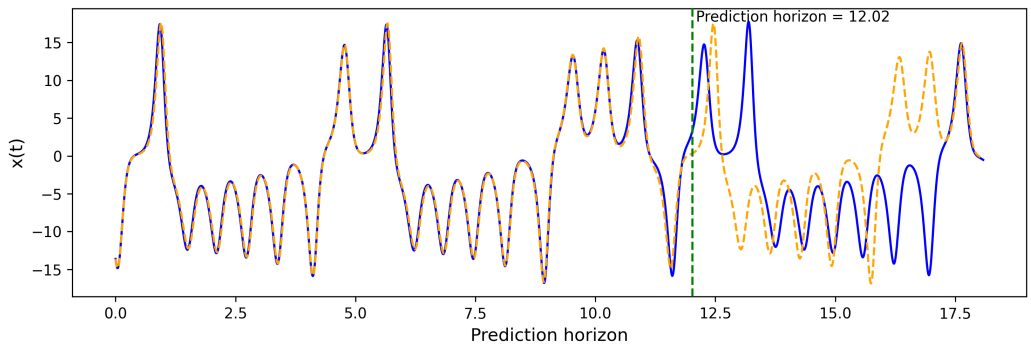
Table 4: Prediction horizon statistics for Models A–J trained on unfiltered data (100 trajectories - S_1) at perturbation level 10^{-5}. Mean and standard deviation are reported in Lyapunov times.

4.2. Trajectory agreement and robustness

Comparative plots of model predictions against the RK45 ground truth reveal that low-KOP-trained models track the true trajectory with greater accuracy (i.e., for longer), particularly over short to medium timescales (i.e. before exponential divergence dominates). Figure 5 demonstrates this behaviour for Model R, showing the relative error $E(t)$ for both the mean prediction horizon (Fig. 5a and an extended prediction horizon of 12.02 Lyapunov times (Fig. 5b. To further illustrate the improvement in trajectory tracking, we present direct comparisons between ground truth and predicted trajectories in Fig. 4, which shows the x-component evolution for both unfiltered and KOP-filtered models.



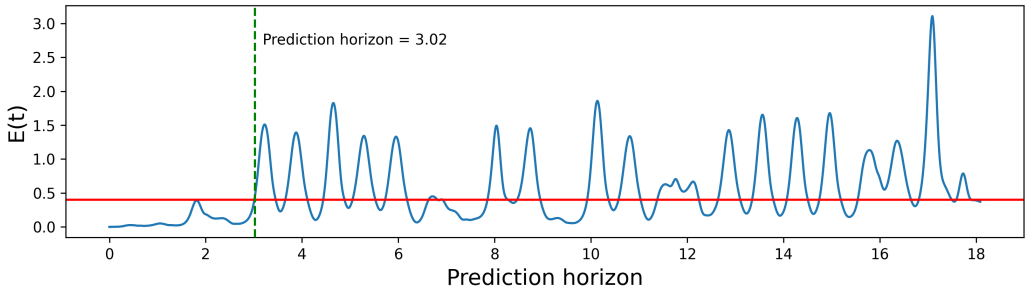
(a)



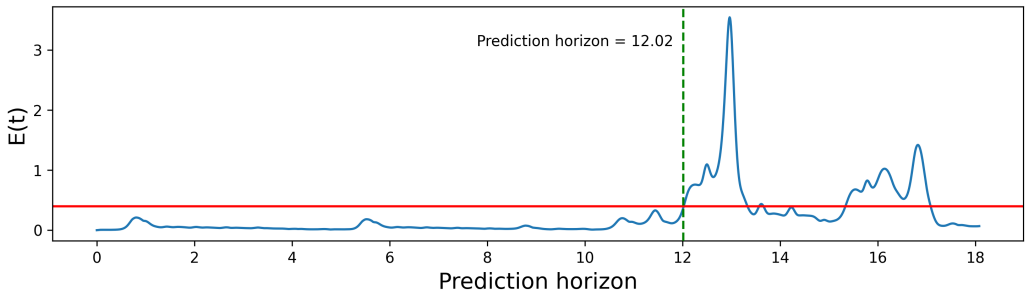
(b)

Figure 4: Panel (a) shows the ground truth (blue) and predicted trajectories (orange) for the x component evolution of a selected initial condition from the **unfiltered** set of predicted trajectories (grouping 1). Here the predicted trajectory breaks down after only 3.02 Lyapunov times. Whilst initially staying close to the ground truth, the predicted trajectory fails to follow the swapping from the positive 'wing' to the negative 'wing'. Panel (b) similarly shows the ground truth and predicted trajectories for a selected initial condition from the **low-KOP filtered** set of trajectories (grouping 1). In this case, the predicted trajectory follows the ground truth closely for 12.02 Lyapunov times but thereafter fails to follow the ground truth as before. Note that here we have used Model R in Table 2 in both cases.

Moreover, the standard deviation of PH across multiple test conditions was significantly reduced in these models, as shown in Tables 4 and 5. This indicates a higher degree of robustness to small variations in initial conditions, an



(a)



(b)

Figure 5: Relative error $E(t)$ from Model R: Panel (a) shows the relative error for the prediction horizon of 3.02 Lyapunov times in Fig. 4a and panel (b) shows the relative error for the prediction horizon of 12.02 Lyapunov times. Note that here we have used Model R in Table 2.

essential quality when forecasting chaotic systems. In contrast, models trained on unfiltered data often exhibited erratic generalisation behaviour, suggestive of implicit bias toward over-synchronised attractor substructures.

The effect of KOP filtering on prediction performance is not uniform across all models, and in some cases reveals an interesting tension between mean prediction horizon improvement and variability reduction. Model R provides a particularly instructive example: KOP filtering yields a substantial improvement in mean PH of 33.2%, increasing from 2.32 to 3.09 Lyapunov times, yet simultaneously produces an increase in standard deviation of 15.2%, rising from 1.12 to 1.29 Lyapunov times. This apparent contradiction can be understood by considering that low-KOP trajectories do not simply select more uniform or easier-to-predict regions of the attractor, but rather trajectories that are more dynamically informative. Training on a filtered ensemble that better captures the underlying attractor geometry exposes the model to a more diverse set of phase space trajectories, elevating mean prediction quality whilst broadening the distribution of outcomes. This suggests that the relationship between KOP filtering, trajectory diversity, and prediction performance is more nuanced than a simple improvement across all metrics, and warrants further investigation into the interplay between ensemble composition and model generalisation.

4.3. Probability density distribution analysis of prediction horizons and generation of initial conditions for prediction

Histograms of PH across varying perturbation amplitudes offer further insight into the model's predictive behaviour. Models trained on unfiltered data exhibited highly skewed distributions, often marked by premature divergence and pronounced kurtosis. Such characteristics are indicative of overfitting to specific trajectories or to dynamically coherent regions within the attractor, thereby limiting the model's ability to generalise across the full phase space.

For instance, Fig. 6(a) and (c) show the PDFs for Model Q at perturbation levels 10^{-5} and 10^{-15} respectively, where clear left-skewing is evident, with several bins to the left of the mean contributing to the asymmetry.

By contrast, models trained on low-KOP filtered data yielded substantially smoother and more symmetric distributions (Figs. 7 and 8), thus conforming better to the Gaussian PDF. Comparing Fig. 6(a) with Fig. 7(a) directly illustrates this improvement: the same Model Q at the same perturbation level (10^{-5}) transitions from a skewed distribution with mean 2.270 to a near-Gaussian distribution with mean 2.886 when trained on low-KOP data. This effect was particularly pronounced at perturbations of 10^{-5} and below, where the filtered models demonstrated significantly greater predictive consistency relative to the unfiltered baseline. These results suggest that training on low-KOP data facilitates a more faithful encoding of the attractor's geometric structure, thereby enhancing the model's generalisation capacity and robustness to small perturbations.

Figure 8 further demonstrates the robustness of this approach across independently generated datasets. All five sets of low-KOP filtered data (panels (a) through (j)) show consistent near-Gaussian PDFs with comparable mean prediction horizons, indicating that the improvement is not an artefact of a particular random seed but rather a fundamental consequence of the KOP-based filtering approach.

Model	Unfiltered Data		KOP-Filtered Data ($R < 0.053$)		Mean PH Improvement (%) / Degradation (%)	Std Dev Reduction (%)
	Mean PH	Std Dev	Mean PH	Std Dev		
Model K	3.34	1.32	2.19	1.04	-34.4	+21.2
Model L	2.36	0.98	2.83	1.25	+19.9	-27.6
Model M	3.10	1.43	2.03	0.90	-34.5	+37.1
Model N	2.40	0.91	2.63	1.30	+9.6	-42.9
Model O	2.47	1.19	2.14	1.02	-13.4	+14.3
Model P	2.64	1.19	1.86	0.88	-29.6	+26.1
Model Q	2.27	1.26	2.89	1.20	+27.3	+4.8
Model R	2.32	1.12	3.09	1.29	+33.2	-15.2
Model S	1.97	0.92	2.91	1.19	+47.7	-29.3
Model T	3.91	1.44	1.97	1.17	-49.6	+18.8

Table 5: Comparison of prediction horizon statistics for Models K–T trained on unfiltered vs. KOP-filtered datasets at perturbation level 10^{-5} with 1,000 trajectories (S_2). Mean and standard deviation are reported in Lyapunov times. For mean PH, positive values indicate improvement and negative values indicate degradation. For std dev, positive values indicate a reduction in variability and negative values indicate an increase in variability when using KOP-filtered data. Note that improvements in mean PH and std dev reduction are not always concurrent, reflecting the nuanced effect of KOP filtering on prediction quality and trajectory diversity.

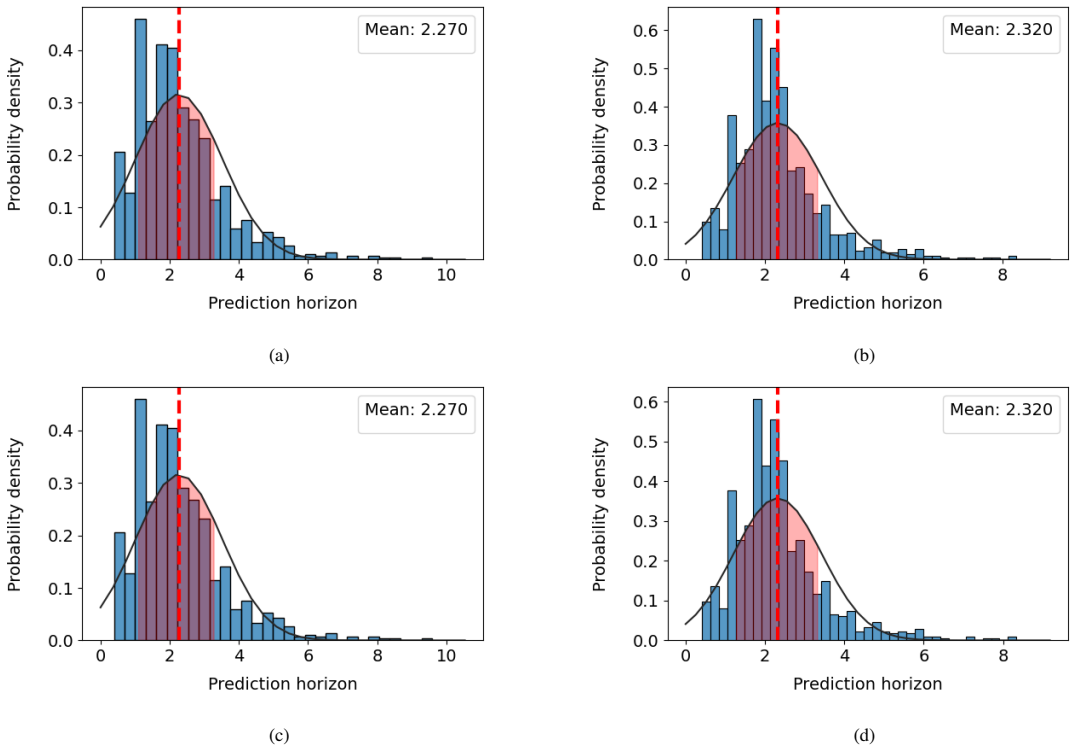


Figure 6: Probability density functions of prediction horizons for neural network models Q and R in Table 2 evaluated under differently perturbed initial conditions. The left column (panels (a) and (c)) presents PDFs computed from predictions on synchronised (high-KOP or unfiltered) data for Model Q at 10^{-5} and 10^{-15} perturbations, while the right column (panels (b) and (d)) shows predictions on synchronised (high-KOP or unfiltered data) from Model R again at 10^{-5} and 10^{-15} perturbations. In the cases with higher synchronisation there is evidence of greater skewness.

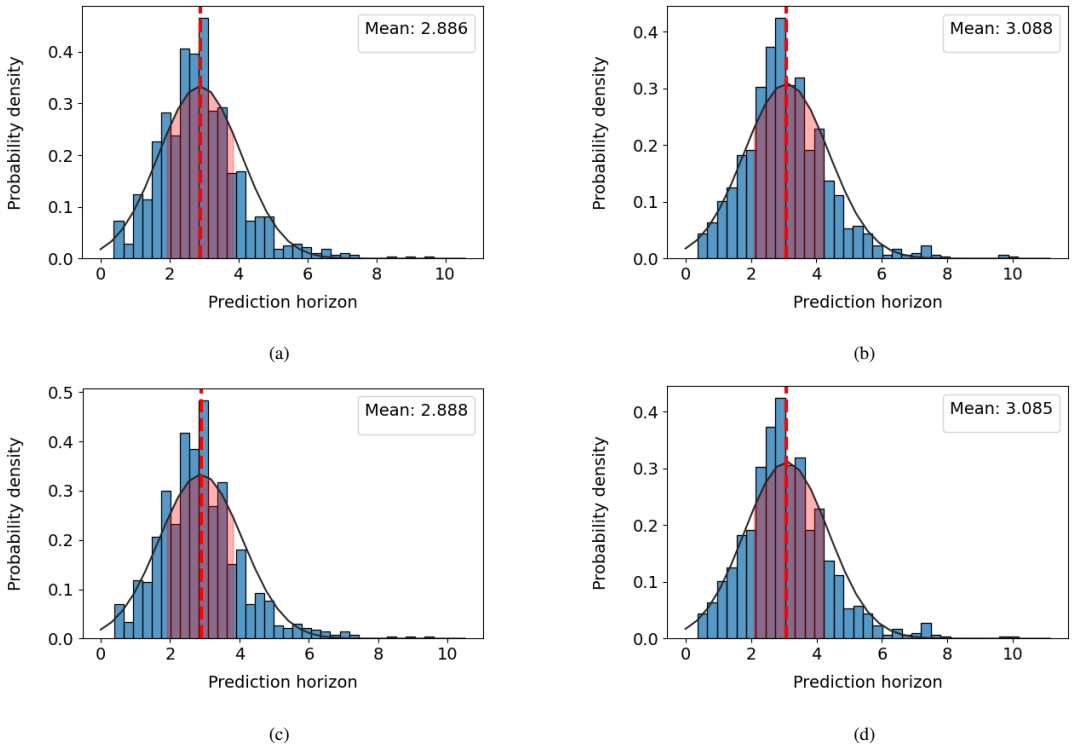


Figure 7: Probability density functions of prediction horizons for neural network models Q and R in Table 2 evaluated under differently perturbed initial conditions where the data have been subject to low-KOP filtering. The left column (panels (a) and (c)) presents PDFs computed from predictions on synchronised (low-KOP or filtered) data for Model Q at 10^{-5} and 10^{-15} perturbations, while the right column (panels (b) and (d)) shows predictions on synchronised (low-KOP or filtered) from Model R again at 10^{-5} and 10^{-15} perturbations. In contrast to 6, which shows evidence of skewness, the numerical PDFs here align much more closely with a Gaussian distribution denoted by black solid curves. The low-KOP filtered data, removes skewness, leading to a higher mean prediction horizon for the same models.

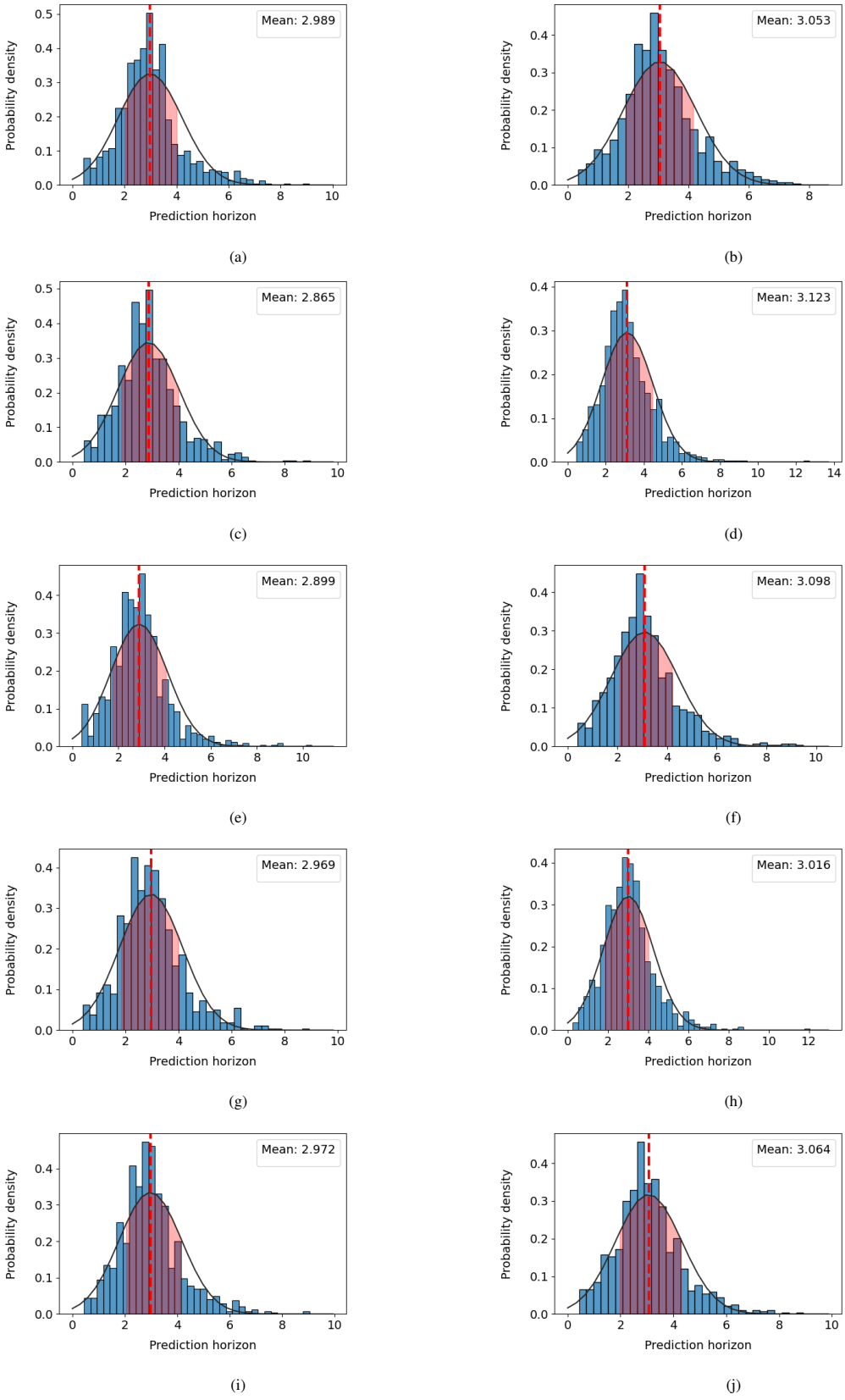


Figure 8: Probability density functions of prediction horizons for neural network models Q and R evaluated under low-KOP (low synchrony) conditions. The left column (panels (a), (c), (e), (g), (i)) presents PDFs computed for Model Q low-KOP data and the right column (panels (b), (d), (f), (h), (j)) presents PDFs computed for Model R low-KOP data. Each row corresponds to a different "set" of data filtered for low-KOP, where each "set" of data is generated using a different random seed before being screened for low synchrony.

Next, to evaluate the generalisation and robustness of the KOP-filtering approach across different initial condition distributions, we applied our trained models to the prediction datasets from both groupings described in Subsec. 3.1. Grouping 1 consists of initial conditions derived by perturbing the training trajectories across a range of perturbation magnitudes (10^{-15} to 10^{-1}), enabling assessment of model sensitivity to initial condition variations. Grouping 2 consists of independently generated initial conditions using different random seeds, providing a test of generalisation to truly unseen data. The results of these evaluations are presented in the following section.

4.4. Influence of initial conditions on prediction horizon probability density functions

Here, we investigate the effect of different sets of initial conditions on the conformance of the PDFs to the Gaussian PDF and the conclusions that can be drawn. As we have seen in Sec. 3.4, using unfiltered datasets, results in skewed PDFs, as can be seen in Fig. 2. This is because there is a considerable amount of trajectories in the unfiltered datasets with different phase-synchronisation properties that give rise to smaller prediction horizons and thus, to skewness. As the Lorenz system (1) is well-known for exhibiting chaotic synchronisation due to "shadowing" and structural stability, we checked the datasets of trajectories for phase-synchronisation using the well-known Kuramoto order parameter in Eq. (5). We wanted to understand whether there is a group of trajectories that synchronises for considerably long integration times compared with others in the same dataset that stay almost desynchronised throughout integration.

We present the results for the computations of the extents of the x, y and z components of grouping 1 in Table 6 (see Subsec. 3.1 for grouping definitions). To compute the extents, we computed the global minima and maxima of the three components separately, and then subtracted the global minimum from the global maximum for each component. These results for the different perturbation levels (in column one) are shown in column two in Table 6, where each entry in the triplet is the extent in the x, y and z directions. The triplets in column three in the Table show the extents of the attractor in the three directions computed by calculating the global minima and global maxima of the ground-truth trajectories integrated numerically using RK45. Analogous results for grouping 2 are reported in Table 7, where column one is the index of each set, column two is the extents of the initial conditions for the set in column one, and column three is the extents of the Lorenz attractor for each set. We observe that in both groupings, the sizes of the cubes that contain the initial conditions for training and prediction are comparable to the size of the cube that contains the attractor.

We then used the sets of initial conditions in the two groupings and based on the evidence in Sec. 3.4, we applied the low-KOP filtering discussed in 2.2, ending up with a subset of $1,000$ trajectories for both groupings, with very low-KOP synchronisation levels (i.e, with $R < 0.053$). Next, we retrained the models using these low-KOP datasets and ran the PH calculations to compute the new PDFs, presented in Figs. 7 and 8. By comparing Fig. 6 for unfiltered datasets with Fig. 7 for filtered datasets, we observe that (a) there is no longer as much skewness in the PDFs of the low-KOP datasets and (b), the PDFs conform better to the Gaussian PDF.

4.4.1. The role of synchronisation in predictability

While the geometric extent of trajectories is predetermined by the initial ensemble, not all trajectories within these bounds are equally valuable for training predictive models. Our analysis reveals that synchronised trajectories due to shadowing and structural stability of the Lorenz system, where phase relationships become temporarily coordinated, significantly degrade prediction accuracy. Trajectories exhibiting high synchronisation (high-KOP values) represent dynamical behaviour that is not as diverse as that of low-KOP trajectories. This explains why random sampling, which may inadvertently select synchronised trajectory segments, fails to capture the full dynamical diversity necessary for robust prediction. Conversely, trajectories with low synchronisation (low-KOP values) provide maximally independent samples of the attractor's dynamics, enabling neural networks to learn the complete repertoire of dynamical behaviours within the geometrically invariant bounds of the attractor.

4.4.2. Implications for chaotic dynamics

This geometric invariance coupled with the effects of synchronisation in the Lorenz system, has profound implications for our understanding of chaotic dynamics. Despite exponential divergence of individual trajectories, the collective spatial bounds are essentially predetermined by the initial ensemble spread, with the chaotic dynamics redistributing trajectories within these bounds.

The geometric invariance and synchronisation fundamentally challenge conventional approaches to sampling chaotic systems. Since the geometric extent is predetermined by the initial ensemble distribution, and synchronisation reduces the effective dynamical diversity of the training data, improving prediction models requires strategic filtering to select desynchronised trajectories that capture independent dynamical behaviours within these fixed bounds, precisely what KOP filtering achieves. The invariance between initial and final extents represents a fundamental constraint on chaotic dynamics, revealing that while chaos ensures sensitive dependence at the trajectory level, it operates within geometrically invariant bounds at the ensemble level. This distinction between local unpredictability and global geometric constraint, combined with the effect of synchronisation diminishing predictive model performance, has important implications for both theoretical understanding and practical applications in neural network training for chaotic system prediction.

Perturbation Levels	Extents of Perturbed ICs	Extents of the Attractor
10^{-15}	35.11504786896894 47.40301630245102 40.828271056049026	34.706582678284185 45.957618428155065 38.0962690353708
10^{-13}	35.11504786896894 47.40301630245102 40.82827105604903	34.70657620572245 45.957665671895974 38.09621670874611
10^{-11}	35.11504786896894 47.40301630245102 40.82827105604903	34.70547164307637 45.95740295279348 38.09224745240886
10^{-9}	35.11504786896894 47.40301630245102 40.828271056049026	34.70691816658985 45.954832024216145 38.10253189243813
10^{-7}	35.11504786896894 47.40301630245102 40.828271056049026	34.684124375626 45.91967432161095 38.08102177091558
10^{-5}	35.11504786896894 47.40301630245102 40.82827105604903	34.716189493588175 45.9759830445415 38.197805660485116
10^{-3}	35.11504786896894 47.40301630245102 40.828271056049026	34.68993064554021 45.929148570908524 38.190395499986174
10^{-1}	35.11504786896894 47.40301630245102 40.828271056049026	34.73082473525957 45.99767224899058 38.13025584849451

Table 6: Perturbation levels (first column), where "Extents of Perturbed ICs" (second column) and "Extents of the Attractor" (third column) are the extents of the attractor computed numerically by using the perturbed and ground-truth trajectories, respectively. Here we have used the trajectories in grouping 1.

Extended integration times or additional trajectories produce negligible changes to these global extents ($< 3\%$ variation), confirming the robustness of this geometric invariance property. This invariance between initial and final extents, together with our findings on synchronisation effects, represents fundamental constraints on chaotic dynamics that directly inform optimal strategies for data selection in machine learning applications.

4.5. Discussion on Gaussian probability density function fitting

The use of KOP as a screening metric provides a systematic method for filtering data by considering the underlying phase synchrony of the trajectories in the dataset. From a statistical mechanics perspective, low-KOP values correspond to a regime in which constituent oscillators exhibit little phase coherence, such that dominant collective modes are non-existent or weak [27, 41]. In such states, the collective behaviour is dominated by stochastic or uncorrelated fluctuations, which, through the central limit theorem, naturally give rise to unimodal, symmetric distributions [55]. This theoretical relation explains why data filtered for low-KOP more consistently satisfy Gaussian assumptions and contain fewer higher-order statistical artefacts under downstream analyses.

In contrast, data drawn from higher-KOP regimes exhibit stronger synchronisation due to shadowing and structural stability or emergent collective organisation across the system [56]. Such regimes impose long-range dependencies and dynamic coupling that can impose multimodal or asymmetric characteristics on the underlying distribution. Importantly, strong synchronisation may not be visually apparent in raw time series, but is expressed in the statistical profile of phase-based observables. The imposition of these non-Gaussian characteristics frustrates inference procedures that assume distributional symmetry, leading to distortions in predictive modelling and statistical testing.

Sets	Extents of Set ICs	Extents of the Attractor
	35.22487805225061	34.66489871984085
1	50.78933389792624	45.8812774423619
	42.174927473992675	38.1680942209734
	33.94038522465193	34.68839554830721
2	44.23195444253331	45.92812552458531
	40.48183287773829	38.05162073928093
	35.919366596930914	34.66995045600162
3	45.375317908342865	45.88417560223456
	41.06281358057437	38.030072511754405
	35.262351788717204	34.7243650451591
4	48.11354553123794	45.991412921908136
	38.94726060167497	38.0847171379738
	36.62992503763881	34.646957138988746
5	44.195327290969715	45.83561906673275
	40.2293728590208	37.91065988750479

Table 7: Different "Sets" of initial conditions (first column), where "Extents of Perturbed ICs" (second column) and "Extents of the Attractor" (third column) are the extents of the attractor computed numerically by using the initial conditions in the sets (first column) and ground-truth trajectories, respectively. Here we have used the trajectories in grouping 2.

Thus, KOP filtering is not merely a preprocessing step, but directly shapes the statistical geometry of the data. As shown in Fig. 8, low-KOP screening suppresses latent synchronisation effects, yielding smoother, unimodal, and near-Gaussian distributions compared to the more irregular profiles in Fig. 6. Across datasets, this screening isolates a more statistically homogeneous regime, in which unimodality and isotropy assumptions are more likely to hold. These properties provide a robust justification for KOP-based selection and strengthen the interpretability and generalisability of downstream models across different perturbation scales.

5. Conclusions

In this paper, we have examined the role of phase synchrony filtering in enhancing the predictive capabilities of neural networks applied to chaotic dynamical systems. We have demonstrated that the conventional approach of training machine learning models on arbitrarily selected trajectory data fundamentally limits the model's ability to capture the full dynamical diversity of chaotic attractors. The Kuramoto Order Parameter (KOP) emerges as a powerful metric for identifying trajectories that exhibit the dynamical diversity essential for robust model training. Our systematic comparison between filtered and unfiltered datasets reveals that low-KOP trajectories, characterised by weak phase synchrony, provide neural networks with training examples that are more representative of the underlying chaotic dynamics.

We have shown how KOP-based filtering directly impacts model performance across multiple evaluation criteria and perturbation levels. Networks trained on low-KOP data consistently achieved extended prediction horizons compared to those trained on conventional datasets. Moreover, these models demonstrate superior conformity to ground truth dynamics and produce output distributions that conform more closely to the Gaussian distribution. The robustness of these improvements across varying perturbation scales demonstrates that the benefits of KOP filtering are not simply a result of favourable initial conditions but reflect fundamental improvements in how neural networks learn to represent chaotic dynamics. This enhanced performance arises from the reduced structural bias inherent in low-synchrony training data, which forces neural networks to develop a more generalised internal representation of the system's behaviour.

The implications of our findings extend beyond the specific case of the Lorenz system we studied here. The success of phase synchrony analysis in improving neural network performance suggests that dynamical screening represents a fundamental principle for data-driven modelling of nonlinear systems. Traditional approaches assume all training data are equally valuable, but our method shows that careful selection based on physical principles can significantly improve performance. By choosing trajectories with specific dynamical properties, e.g., low-KOP, we can build models that maintain accuracy over longer time periods. This improvement could benefit many fields requiring long-term predictions, from weather forecasting to ecosystem modelling.

It is important to acknowledge, however, that the results presented here rest on several working assumptions. The phase extraction underpinning our KOP computation relies on the Hilbert transform, which performs optimally under approximate narrowband and Bedrosian-type conditions, as discussed in 2.2.1. While the Lorenz system exhibits sufficiently coherent oscillatory structure for these assumptions to hold in practice, their validity should be carefully assessed before applying this framework to other dynamical systems. Furthermore, the applicability of KOP as a filtering mechanism is not guaranteed to generalise to all chaotic systems. Systems with more complex attractor geometries, higher-dimensional phase spaces, or fundamentally different synchronisation properties may require alternative or complementary phase-based metrics. We also note that KOP-based filtering requires a sufficiently large trajectory pool to be effective; our experiments suggest that datasets of the order of 100 trajectories may be too small to yield a meaningfully distinct low-synchrony subset under the criterion $R < 0.053$, and that a minimum pool size of the order of 1,000 trajectories is needed for the filtering to produce a statistically separable low-synchrony subset. Investigating the extent to which KOP-based filtering transfers to such systems represents an important direction for future work.

In summary, our results show that strategic training data curation based on phase synchrony analysis represents a significant advancement in machine learning approaches to chaotic systems prediction. By using the Kuramoto Order Parameter to identify trajectories with high dynamical diversity, we achieve substantial improvements in prediction accuracy and duration. While our work focuses on improving mean prediction horizons through dynamical diversity, the KOP-filtering approach may prove particularly valuable for predicting extreme events and rare dynamical behaviours in chaotic systems [12, 13]. Future work could investigate whether low-KOP training data enhances the detection and forecasting of extreme events, which typically emerge from desynchronised, high-diversity attractor regions. These findings open new possibilities for applying similar filtering techniques to other chaotic systems where long-term forecasting remains challenging.

Data availability statement

The source code to reproduce the results in this paper is freely available in the GitHub repository [57].

Acknowledgments

References

[1] N. H. Packard, J. P. Crutchfield, J. D. Farmer, R. S. Shaw, Geometry from a time series, Phys Rev Lett 45 (9) (1980) 712–716. doi:10.1103/PhysRevLett.45.712.

[2] F. Takens, Detecting strange attractors in turbulence, in: D. A. Rand, L. S. Young (Eds.), Dynamical Systems and Turbulence, Warwick 1980, Vol. 898 of Lecture Notes in Mathematics, Springer, 1981, pp. 366–381. doi:10.1007/BFb0091924.

[3] J. D. Farmer, J. J. Sidorowich, Predicting chaotic time series, Phys Rev Lett 59 (8) (1987) 845–848. doi:10.1103/PhysRevLett.59.845.

[4] J. Pathak, Z. Lu, B. R. Hunt, M. Girvan, E. Ott, Model-free prediction of large spatiotemporally chaotic systems from data: a reservoir computing approach, Phys Rev Lett 120 (2) (2018).

[5] Y. Fan, J. Jiang, K. Lin, Long-term prediction of chaotic systems with machine learning, Phys Rev Res 2 (1) (2020).

[6] M. Khodkar, P. Hassanzadeh, A data-driven, physics-informed framework for forecasting the spatiotemporal evolution of chaotic dynamics with nonlinearities modeled as exogenous forcings, J Comput Phys 440 (2021). doi:10.1016/j.jcp.2021.110412.

[7] E. N. Lorenz, Deterministic nonperiodic flow, J Atmos Sci 20 (2) (1963) 130–141.

[8] T. Sauer, J. A. Yorke, M. Casdagli, Embedology, J Stat Phys 65 (1991) 579–616. doi:10.1007/BF01053745.

[9] P. R. Vlachas, W. Byeon, Z. Y. Wan, T. P. Sapsis, P. Koumoutsakos, Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks, Proc R Soc A 474 (2213) (2018).

[10] S. L. Brunton, J. N. Kutz, Data-driven science and engineering: machine learning, dynamical systems, and control, Cambridge University Press, 2019.

[11] J. Pathak, B. Hunt, M. Girvan, Z. Lu, E. Ott, Hybrid forecasting of chaotic processes: using machine learning in conjunction with a knowledge-based model, Chaos Interdiscip J Nonlinear Sci 28 (4) (2018).

[12] S. Coulibaly, F. Bessin, M. G. Clerc, A. Mussot, Extreme events following bifurcation to spatiotemporal chaos in a spatially extended microcavity laser, Chaos Solit Fractals 160 (2022). doi:10.1016/j.chaos.2022.112199.

[13] V. P. Pammi, M. G. Clerc, S. Coulibaly, S. Barbay, Spatiotemporal extreme events in a laser with a saturable absorber, Phys Rev Lett 130 (22) (2023). doi:10.1103/PhysRevLett.130.223801.

[14] J. Jiang, L. Chen, L. Ke, B. Dou, Y. Zhu, Y. Shi, et al., Machine learning predictions from unpredictable chaos, J R Soc Interface 22 (2025). doi:10.1098/rsif.2025.0441.

[15] B. Lusch, J. N. Kutz, S. L. Brunton, Deep learning for universal linear embeddings of nonlinear dynamics, Nat Commun 9 (1) (2018) 4950.

[16] S. Pannasch, J. Gross, B. Lindner, E. Schöll, Training recurrent neural networks to predict chaotic systems in the presence of noise, Front Appl Math Stat 7 (2021) 65.

[17] Y. Kuramoto, Self-entrainment of a population of coupled nonlinear oscillators, in: H. Araki (Ed.), International Symposium on Mathematical Problems in Theoretical Physics, Vol. 39 of Lecture Notes in Physics, Springer, 1975, pp. 420–422.

[18] S. H. Strogatz, From Kuramoto to Crawford: exploring the onset of synchronization in populations of coupled oscillators, Physica D: Nonlinear Phenomena 143 (1–4) (2000) 1–20. doi:10.1016/S0167-2789(00)00094-4.

[19] J. R. Dormand, P. J. Prince, A family of embedded Runge-Kutta formulae, J Comput Appl Math 6 (1) (1980) 19–26. doi:10.1016/0771-050X(80)90013-3.

[20] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, et al., SciPy 1.0: fundamental algorithms for scientific computing in Python, Nat Methods 17 (2020) 261–272. doi:10.1038/s41592-019-0686-2.

[21] M. Clerc, P. Coullet, E. Tirapegui, Lorenz bifurcation: instabilities in quasi-reversible systems, Phys Rev Lett 83 (19) (1999) 3820–3823. doi:10.1103/PhysRevLett.83.3820.

[22] C. Sparrow, The Lorenz equations: bifurcations, chaos, and strange attractors, Springer-Verlag, 1982.

[23] A. Wolf, J. B. Swift, H. L. Swinney, J. A. Vastano, Determining Lyapunov exponents from a time series, Phys D Nonlinear Phenom 16 (3) (1985) 285–317.

[24] E. N. Lorenz, The predictability of a flow which possesses many scales of motion, Tellus 21 (3) (1969) 289–307. doi:10.3402/tellusa.v21i3.10086.

[25] P. Grassberger, I. Procaccia, Measuring the strangeness of strange attractors, Phys D Nonlinear Phenom 9 (1–2) (1983) 189–208.

[26] A. Pikovsky, M. Rosenblum, J. Kurths, Synchronization: a universal concept in nonlinear sciences, Cambridge University Press, 2001.

[27] Y. Kuramoto, Chemical oscillations, waves, and turbulence, Springer, 1984.

[28] D. Gabor, Theory of communication, J Inst Electr Eng – Part III Radio Commun Eng 93 (26) (1946) 429–457.

[29] B. Boashash, Estimating and interpreting the instantaneous frequency of a signal. I. Fundamentals, Proc IEEE 80 (4) (1992) 520–538.

[30] L. Cohen, Time-frequency analysis, Prentice Hall, 1995.

[31] D. Vakman, On the analytic signal, the Teager–Kaiser energy algorithm, and other methods for defining amplitude and frequency, IEEE Trans Signal Process 44 (4) (1996) 791–797.

[32] S. L. Hahn, Hilbert transforms in signal processing, Artech House, 1996.

[33] E. Bedrosian, A product theorem for Hilbert transforms, Proc IEEE 51 (1963) 868–869.

[34] N. E. Huang, et al., The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis, Proc R Soc A 454 (1998) 903–995.

[35] M. G. Rosenblum, A. S. Pikovsky, J. Kurths, Phase synchronization of chaotic oscillators, Phys Rev Lett 76 (11) (1996) 1804–1807.

[36] K. Palmer, Shadowing in dynamical systems: theory and applications, Vol. 501 of Mathematics and Its Applications, Springer, 2000. doi:10.1007/978-1-4757-3210-8.

[37] J. Guckenheimer, Isochrons and phase sensitivity analysis, J Math Biol 1 (1975) 259–273.

[38] A. T. Winfree, The geometry of biological time, Springer, 2001.

[39] B. Kralemann, L. Cimponeriu, M. Rosenblum, A. Pikovsky, R. Mrowka, Phase dynamics of coupled oscillators reconstructed from data, Phys Rev E 77 (2008).

[40] T. Schreiber, Measuring information transfer, Phys Rev Lett 85 (2000) 461–464.

[41] J. A. Acebrón, L. L. Bonilla, C. J. P. Vicente, F. Ritort, R. Spigler, The Kuramoto model: a simple paradigm for synchronization phenomena, Rev Mod Phys 77 (1) (2005) 137–185. doi:10.1103/RevModPhys.77.137.

[42] R. G. Palmer, Broken ergodicity, Adv Phys 31 (6) (1982) 669–735. doi:10.1080/00018738200101438.

[43] C. Stahl, O. Hart, R. Nandkishore, Towards absolutely stable ergodicity breaking in two and three dimensions, Phys Rev B 111 (2) (2025) L020302. doi:10.1103/PhysRevB.111.L020302.

[44] E. Ott, Chaos in dynamical systems, Cambridge University Press, 2002.

[45] H. Jaeger, H. Haas, Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication, Science 304 (5667) (2004) 78–80.

[46] D. P. Kingma, J. Ba, Adam: a method for stochastic optimization, in: International Conference on Learning Representations (ICLR), 2015.
URL https://arxiv.org/abs/1412.6980

[47] T. Dozat, Incorporating Nesterov momentum into Adam, in: Proceedings of the 4th International Conference on Learning Representations (ICLR) Workshop, 2016, pp. 1–4.

[48] S. Jastrzebski, Z. Kenton, D. Arpit, N. Ballas, A. Fischer, Y. Bengio, et al., Three factors influencing minima in SGD, arXiv:1711.04623 (2017).

[49] S. Li, Y. He, Q. Ho, Y. Wang, Surge phenomenon in optimal learning rate and batch size scaling, arXiv preprint arXiv:2404.07167 (2024).

[50] S. Walczak, N. Cerpa, Heuristic principles for the design of artificial neural networks, Inf Softw Technol 41 (2) (1999) 107–117. doi:10.1016/S0950-5849(98)00116-5.

[51] I. Goodfellow, Y. Bengio, A. Courville, Deep learning, MIT Press, 2016.
URL http://www.deeplearningbook.org

[52] D. Freedman, P. Diaconis, On the histogram as a density estimator: L^2 theory, Z Wahrscheinlichkeitstheorie Verw Geb 57 (4) (1981) 453–476. doi:10.1007/BF01025868.

[53] G. Benettin, L. Galgani, A. Giorgilli, J.-M. Strelcyn, Lyapunov characteristic exponents for smooth dynamical systems and for Hamiltonian systems; a method for computing all of them. Part 1: theory, Meccanica 15 (1) (1980) 9–20. doi:10.1007/BF02128236.

[54] G. Benettin, L. Galgani, A. Giorgilli, J.-M. Strelcyn, Lyapunov characteristic exponents for smooth dynamical systems and for Hamiltonian systems; a method for computing all of them. Part 2: numerical application, Meccanica 15 (1) (1980) 21–30. doi:10.1007/BF02128237.

[55] M. Breakspear, S. Heitmann, A. Daffertshofer, Generative models of cortical oscillations: neurobiological implications of the Kuramoto model, Front Hum Neurosci 4 (2010) 190.

[56] J. Cabral, E. Hugues, O. Sporns, G. Deco, Role of local network oscillations in resting-state functional connectivity, Neuroimage 57 (1) (2011) 130–139.

[57] A. Miller, https://github.com/antrallas/weakly-synchronised-trajectories (2025).