Curriculum learning for data-driven modeling of dynamical systems

Michele Alessandro Bucci, Onofrio Semeraro, Alexandre Allauzen, Sergio Chibbaro, and Lionel Mathelin

1 TAU–Team, INRIA Saclay, LISN, Université Paris-Saclay, 91190 Gif-sur-Yvette, France
2 LISN-CNRS, Université Paris-Saclay, 91440 Orsay, France
3 LAMSADE, Université Paris Dauphine, Place du Maréchal de Lattre de Tassigny, 75016 Paris, France

Received 11 November 2022 / Accepted 12 February 2023 / Published online 8 March 2023
© The Author(s), under exclusive licence to EDP Sciences, SIF and Springer-Verlag GmbH Germany, part of Springer Nature 2023

Abstract The reliable prediction of the temporal behavior of complex systems is key in numerous scientific fields. This strong interest is however hindered by modeling issues: Often, the governing equations describing the physics of the system under consideration are not accessible or, when known, their solution might require a computational time incompatible with the prediction time constraints. Not surprisingly, approximating complex systems in a generic functional format and informing it ex–nihilo from available observations has become common practice in the age of machine learning, as illustrated by the numerous successful examples based on deep neural networks. However, generalizability of the models, margins of guarantee and the impact of data are often overlooked or examined mainly by relying on prior knowledge of the physics. We tackle these issues from a different viewpoint, by adopting a curriculum learning strategy. In curriculum learning, the dataset is structured such that the training process starts from simple samples toward more complex ones in order to favor convergence and generalization. The concept has been developed and successfully applied in robotics and control of systems. Here, we apply this concept for the learning of complex dynamical systems in a systematic way. First, leveraging insights from the ergodic theory, we assess the amount of data sufficient for a-priori guaranteeing a faithful model of the physical system and thoroughly investigate the impact of the training set and its structure on the quality of long-term predictions. Based on that, we consider entropy as a metric of complexity of the dataset; we show how an informed design of the training set based on the analysis of the entropy significantly improves the resulting models in terms of generalizability and provide insights on the amount and the choice of data required for an effective data-driven modeling.

1 Introduction

In the present era of mathematization of nature, the need for modeling is ubiquitous. Focusing on engineering as an example, models are invaluable for predicting quantities of interest, designing and optimizing physical systems, understanding their behavior in an environment, allowing to derive control strategies. Physical models have historically been developed from a combination of observational data and first principles, depending on the amount and type of prior knowledge. This approach is still followed today where the mathematical structure of a model is typically provided and observational data are used to inform it, i.e., learn the coefficients associated with the different terms of the retained structure. Countless variants exist but they generally follow this philosophy. Nonetheless, in many relevant cases, the physics of the phenomena is very complex and the interplay of observation with physical intuition is not sufficient to sort out a formalized model, let alone more rigorous general laws. In these cases, it is tempting to resort to a pure data-driven approach, in which some model is built up entirely from data. The data-driven approach received some attention after the work of Takens [9,35,38], that provided the theoretical backbone to predictive models based on time-series analysis of low-dimensional systems [20]. These approaches found new motivation following the significant progresses in the field of machine learning (ML), fostered by the massive augmentation of computer-based activities [15]. Even if a clear framework is still missing, the use of such algorithms to build models from data has been introduced in the last decades [see, e.g., 36,43] and, thanks to the recent blooming of data-driven techniques, has been consid-
ered in the modeling a variety of different physical phenomena [see, e.g., 5,8,31]. However, it is important to stress that the introduction of deep learning approaches has not necessarily changed the global picture of modeling. The potential versatility of highly expressive neural networks is often associated with a large amount of necessary observational data to suitably inform the associated coefficients of the model. Yet, not every situation enjoys a massive amount of observational data and can afford to inform a widely generic model. In practice, it is more likely that data are scarce due the cost and/or technical difficulty in obtaining them and complex high-dimensional systems may need a quantity of data significantly exceeding the observing capability, [3]. In that sense, the lack of data for inferring physical models might have direct consequences on the relevance of these highly expressive models. A strategy to alleviate this limitation is to simplify the postulated structure of the model so that it is less data hungry, with simpler models here understood as involving fewer parameters to be informed from the available data, hence a weaker constraint on the amount of training samples. Another strategy is to leverage prior knowledge one has on the physical system under consideration to guide the structure and the learning process, hence saving on data. For instance, known symmetries can be enforced by design in the structure or in the representation format of the input data, effectively reducing the need for data. A similar idea applies to invariances one may know the system obeys (conserved quantities such as energy, translational or rotational invariances) or properties such as stability in a given sense. This physics-aware machine learning approach has now become popular and is the motivation of numerous recent efforts [see 21,33,42,46], among many. In that sense, the modeling of chaotic dynamical systems is a natural play-field [4,28]. Remarkably, previous works seem to indicate an intriguing capability to give accurate forecasting of chaotic systems, even relatively high-dimensional [27], apparently improving over deterministic embedding techniques [20]. These results therefore deserve to be thoroughly analyzed. In the present work, we aim to shed some light on the challenges in data-driven learning dynamical systems and to explore the idea of leveraging prior information, here in the form of known organization of the dynamics, to inform the process of collecting relevant observations. Specifically, we adopt a curriculum learning strategy for dynamical systems from first principles and assess it with respect to common sampling strategies. Curriculum learning relies on the idea that machines, like humans, learn best if first exposed to “easy” situations prior to dealing with “more complex” ones when well into the learning process. A possible rationale is that curriculum learning can be interpreted as an adaptive regularization method for optimization of a non-convex objective function. By progressively introducing more complex samples, the associated cost landscape evolves from near-convex to a more complex function exhibiting many more local extrema. Sorting training examples in this way may hence favor the optimization process toward a better extremum. This concept has been mostly developed in robotics and control of systems, [34] and was formalized more recently in the context of machine learning, [2]. While it still lacks a firm theoretical ground despite some efforts [44], curriculum learning has been widely illustrated and successfully employed in a variety of applications, ranging from robotics to video games, from computer vision to natural language processing and reinforcement learning, [25,37]. What makes a situation, or training sample in the context of machine learning, “easy” or “complex” is case-specific. For instance, in classification problems, the complexity of a particular example presented to the machine may be associated with the distance to the decision surface (e.g., SVM) or with the amount of noise in the data. Here, we illustrate the impact of curriculum learning-based sampling strategies, by considering the celebrated Lorenz’63 system. While deterministic and low dimensional, the Lorenz system features many challenges encountered in more complex configurations, such as sensitivity to initial conditions and an attractor dimension $d_a$ significantly lower than the ambient dimension $d$, here $d_a \approx 2.06 < d = 3$. It is thus somehow a minimally complex unit and justifies its relevance for illustrating some challenges in learning dynamical systems. We associate the complexity of a training element with its information entropy, in the sense of Shannon, and recognize that the solution of dynamical systems in the vicinity of an unstable fixed point is well described by a linear model and exhibits a low entropy. This observation is key and allows a principled sampling strategy in collecting observations of increasing complexity from the system at hand as it evolves from an unstable fixed point. We explore and compare different strategies for sampling a dynamical system in the aim of learning its behavior and predicting its state over time. These strategies only differ by the region in the phase space where observational samples are taken from, every other aspects being identical (sampling frequency, number of samples, observed quantity, etc.). These different configurations are chosen to be typical of situations ranging from no prior information, besides an estimated upper bound of the dimension of the dynamical system up to approximate knowledge of some invariants of the system. We carry out a comprehensive statistical analysis in order to systematically test the impact that different sampling strategies have on the quality of the resulting models in terms of prediction abilities and generalizability. From an algorithmic point of view, recurrent structures such as auto-regressive models or recurrent neural networks are popular choices in such situations. In the present study, we focus on the LSTM (Long Short-Term Memory) architecture [17] for its widespread use, including applications in prediction of dynamical systems behavior [11]. Moreover, it has been recently shown that the performance given by different architectures including reservoir computing is similar [41], although for low-dimensional systems like Lorenz’63, LSTMs appear to give the best results. Given this choice, we also provide best-practice guidance concerning the possible impact
of initial conditions and the memory effects on the results of supervised learning models, besides analyzing the possibility of using less data while preserving generalizability and quality based on curriculum learning strategies. The paper is organized as follows. General principles in learning a system from data are briefly discussed in Sect. 2 to set up the stage and define notations. The general class of models we consider in this work is presented in Sect. 3. The specific configuration we retain for the present discussion is the Lorenz’63 introduced in Sect. 4, together with the different sampling strategies. The results are gathered and presented in Sect. 5 before the discussion in Sect. 6 closes the paper.

2 Theoretical background

2.1 Learning dynamical systems

The present work lies in the framework of learning a model to approximate the time evolution of a state of the system at hand from observations \( s(t) \in \mathbb{R}^{n_x} \). To do so, one needs a representation for the state and a dynamical model for describing its evolution in time. A representation of the state is given by a function \( g \) mapping observations \( s_r(t) \) over a measurement time horizon \( t \in \Gamma \), to a set of coordinates \( v \in \mathbb{R}^{d_v} \). Several mappings have been proposed in the literature, including delays by time intervals \( \Delta t \), \( v_i = s(t - i \Delta t) \) for \( i = 0, 1, \ldots \), and time derivatives of varying orders \( i \), \( v_i = s^{(i)} \), [7, 13]. To obtain the size of the coordinate space, we assume that the state of the system \( u \) lies on an attractor \( \mathbf{A} \) of dimension \( d_A < d \) that is a boundaryless bounded, smooth sub-manifold of \( \mathbb{R}^d \). In this case, the Whitney embedding theorem [45] shows that if the number of coordinates \( d_v \) is such that \( d_v > 2d_A \), then the dynamics of the state can be entirely captured by the new coordinate system. The Takens embedding theorem [35, 38] provides a way to make this result practical by asserting that one can take this number of time delays. Recent results have extended this theorem by providing guaranteed recovery properties based on the measurement operator, type of attractor, and other parameters of the problem, [10]. In this work, we assume no prior knowledge of the governing equations of the model and rely on a purely data-driven approach. The dynamical model for describing the time evolution of \( v(t) \in \mathbb{R}^{d_v} \) can be formulated in various ways, including feed-forward neural networks, owing to their high expressiveness. As an alternative, for instance in case an upper bound of the attractor dimension is unknown due to process noise or measurement noise, a recurrent model \( F_\theta \) can be employed for describing the dynamics of \( v \). The model is parameterized by \( \theta \in \mathbb{R}^{n_\theta} \) and writes

\[
\dot{v}(t) = F_\theta(v),
\]  

where a dotted quantity denotes its time-derivative. Learning a model for the system under consideration takes the form of a classical supervised learning problem based on the misfit between observations \( s_r \equiv s(u(t \in \Gamma)) \) collected through a measurement operator \( h : u \mapsto s \) over a time domain \( \Gamma = [t_0, t_f] \) and predictions, in the sense of some norm \( \| . \|_r \). It typically results in an optimization problem in terms of the model parameters, possibly including a regularization term \( r \):

\[
\theta \in \arg \inf_{\theta \in \mathbb{R}^{n_\theta}} \left\| s(t) - h \left( v(t; \tilde{\theta}) \right) \right\|_r^2 + r(\tilde{\theta}),
\]

s.t. \( \dot{v}(t; \tilde{\theta}) = F_{\tilde{\theta}}(v) \).  

2.2 Collecting data

The main issue impacting the predictions based on observations is the quantity of data actually needed to approximate the dynamics in a given sense. The issue is related to the ergodic theory of dynamical systems, whose founding idea is that the long-time statistical properties of a deterministic system can be equivalently described in terms of the invariant (time-independent) probability, \( \mu \), such that \( \mu(S) \) is the probability of finding the system in any specified region \( S \) of its phase space. If the trajectories of a \( d \)-dimensional ergodic system evolve in a bounded phase space \( \Omega \subset \mathbb{R}^d \), the Poincaré recurrence theorem [30] ensures that analogs exist as it proves that the trajectories exiting from a generic set \( S \in \Omega \) will return to such set \( S \) infinitely many times. The theorem holds for any points in \( S \), almost surely. It was originally formulated for Hamiltonian systems, but it can be straightforwardly extended to dissipative ergodic systems provided initial conditions are chosen on the attractor. In \( d \)-dimensional dissipative systems, the attractor \( \mathbf{A} \) typically has a dimension \( d_A < d \). The probability \( \mu(B^d_A(\varepsilon)) \) of finding points on the attractor which are in the \( d \)-dimensional ball of radius \( \varepsilon \) (with resolution \( \varepsilon \ll 1 \)) around \( x \in \mathbf{A} \subset \mathbb{R}^d \):

\[
\mu\left(B^d_A(\varepsilon)\right) = \int_{B^d_A(\varepsilon)} d\mu(x) \sim \varepsilon^{d_A}.
\]

When \( d_A \) is not an integer, the attractor and its probability measure are fractal. The attractor is typically multifractal [26], but we can assume without lack of generality that it is homogeneous. Given that the Poincaré theorem proves the existence of good analogs, the issue is shifted to finding out how much time is needed for a Poincaré cycle to end. This key problem was considered by Smolouchowski and resolved by Kac [19], who showed that for an ergodic system, given a set \( S \equiv B^d_A(\varepsilon) \), the mean recurrence time \( \langle \tau_S(x^*) \rangle \) relative to \( S \) is inversely proportional to the measure of the set \( S \), i.e.,

\[
\langle \tau_S(x^*) \rangle \propto \frac{1}{\mu(S)} \sim \varepsilon^{-d_A},
\]
3 Machine learning approach

3.1 Architecture

As discussed above, a set of \(d\)-dimensional observations retained to express a state of the system under consideration might not be a state vector, in the sense that it does not necessarily contain enough information to uniquely define its future evolution as an autonomous deterministic dynamical system. Even though the system were Markovian, it might not be so in the retained representation and may implicitly involve latent variables. This is typically the case in situations of partial observability or when measurements are affected by noise. As discussed in Sect. 2, under conditions rigorously studied in the embedology line of research, [9, 35], latent state variables can be substituted with observations over a recent past, that is, temporal information is traded for unavailable high-dimensional phase space information. In the present work, we consider the LSTM (Long Short-Term Memory) architecture. Compared with standard multi-layer perceptrons, LSTMs additionally feature internal variables \(\zeta\) carried-over through the recurrence and effectively acting as a “memory.” Training an LSTM then allows to derive a discrete-in-time model predicting the future “state” \(v_{k+1} \equiv v(t_k + \Delta t)\) from the current state and memory:

\[
v_{k+1} = F_\theta(v_k; \zeta_k). \tag{5}
\]

The retained architecture comprises 2 LSTM layers of 50 neurons each and one fully connected layer, with \texttt{tanh} activation functions in the LSTM cells and \texttt{sigmoids} for the state output (see the sketch in Fig. 1). The present architecture was deemed as a good compromise between accuracy and computational expenses, after an analysis with comparable architectures with one and three layers has been performed. A 1-step ahead model is learned based on a whole trajectory, or a set of trajectories, in the \(L^2\)-sense. The training is performed using the ADAM optimizer and an adaptive learning rate from 0.01 to \(10^{-5}\), with a decay factor of 0.07. The maximum number of epochs is limited to 1000.

3.2 Strategies for sampling

Learning a model here consists in estimating the best set of parameters \(\theta\) in the sense of the optimization problem defined in Eq. 2. In the context of neural networks, this acts as the loss function used for training. The regularization term may promote several properties of the optimal solution \(\theta\), such as the popular choice of low magnitude (ridge regression), \(r(\theta) \sim \|\theta\|_2\), or sparsity, \(r(\theta) \sim \|\theta\|_1\), possibly mimicked with a suitable dropout strategy. We adopt a discrete-in-time viewpoint, in a formulation consistent with the discrete model in Eq. 5. The loss function \(\mathcal{L}\) associated with the training problem then writes

\[
\mathcal{L}(\theta) = \sum_{k=1}^{K} \|s_k - h(v_k(\theta))\|_2^2 + r(\theta), \tag{6}
\]

with \(K\) such that \(T = K \Delta t\) and the 1-step ahead dynamical model \(v_{k+1} = F_\theta(v_k; \zeta_k)\). The loss is here simply defined in terms of the Euclidean distance but many alternative definitions could be employed. For instance, optimal transport-based metrics, such as Dynamic Time Warping (DTW), can prove useful when observations come at a poorly known or controlled time pace. While the way the prediction from the model is compared with observational data is pivotal, the focus in this work is instead on the relevance of the training data. The present findings are believed to hold, disregarding the definition of the loss. A standard \(L^2\) distance, implicitly assuming regularly sampled measurements, is hence retained. The measurement operator is the identity, \(h \equiv I\), so that the measurements are associated with the state of the system, \(s_k \equiv u_k, \forall k\). No noise affects the observations, so that one is in a full observability situation. In this context, the observed system is Markovian in the representation space of the observations. However, a recurrent architecture is here considered as a general structure for learning, relevant for a wide class of situations. It further has the benefit of improving upon the robustness of the learning by virtue of some degree of redundant representation. Without loss of generality, we disregard the regularization term \(r(\theta)\) in the definition of the loss function which finally writes

\[
\mathcal{L}(\theta) = \sum_{k=1}^{K} \|s_k - v_k(\theta)\|_2^2, \tag{7}
\]

s.t. \(v_{k+1} = F_\theta(v_k)\), \(v_0 = s_0\).
While this context is favorable for learning (low-dimensional deterministic system, noiseless observations, full observability, etc.), it will be seen below that the measurement strategy still significantly affects the learning performance.

3.3 Curriculum learning, metrics and complexity of the datasets

As mentioned in the Introduction, we aim at verifying the extent to which a measurement strategy inspired by the curriculum learning principle can be applied to data-driven learning of dynamical systems and how this impacts the generalizability and quality of the resulting models. The rationale is the adaptive regularization of the objective function when moving from “simpler” cost landscapes associated with simpler data toward more complex ones. From this perspective, any indicator of complexity can be used; here, we consider entropy as a measure and compute it based on singular value decomposition (SVD) \([40]\). Given an embedding for the measurements \(s\) based on time delaying

\[
s = [s(k), s(k + \tau), \ldots, s(k + (n_c - 1)\tau)], \quad (8)\]

with \(k\) an integer and \(\tau\) the time delays, the matrix \(S\) is defined as

\[
S = [s(0), s(\tau), s(2\tau), \ldots, s(N - (n_c - 1)\tau)], \quad (9)\]

with \(n_c\) indicating the number of \(\tau\). The SVD is performed on the matrix and \(M\) singular values \(\{\sigma_{i}^{\text{SVD}}\}_i\) are obtained such that the entropy \(H\) is defined as

\[
H_{\text{SVD}} = -\sum_{i=1}^{M} \sigma_{i}^{2} \log_2(\sigma_{i}^{2}), \quad (10)\]

with \(\sigma_{i}^{2} = \sigma_{i}^{\text{SVD}} / \sum_{j=1}^{M} \sigma_{j}^{\text{SVD}}\). The SVD entropy provides an estimation of the number of eigenvectors that are needed for an adequate representation of the dataset. In that sense, it measures the dimensionality of the data. With this metric, we will assess in the next section the complexity characterizing each of the training datasets that will be compared.

The second metric introduced in the article for the evaluation of the models is the correlation dimension, computed using the algorithm by Grassberger & Procaccia \([16]\). Considering a ball of radius \(\varepsilon\), the probability of finding a point scales as \(C(\varepsilon) \sim \varepsilon^{d_2}\), where \(d_2\) is an indication of the dimensionality. Using the Euclidian norm, the probability \(C(\varepsilon)\) can be efficiently computed as the correlation sum

\[
C(\varepsilon) = \lim_{N \to \infty} \frac{1}{N(N-1)} \sum_{i,j=1, i \neq j}^{N} \Phi(\varepsilon - ||s_i - s_j||), \quad (11)\]

with \(\Phi\) the Heaviside step function, \(s\) the components of the embedded variable ranging \(i, j = 1, 2 \ldots N\), with a fixed time increment; \(N\) is the number of pairs. Due to the divergence of trajectories, most pairs \((s_i, s_j)\) with \(i \neq j\) are usually uncorrelated. Here, the correlation dimension \(d_2\) is considered for the information provided on the fractal dimension of the attractor as well as the connections with the intrinsic dimension, a metric often found in machine learning \([6]\). Alternatively, the Lyapunov exponents can also be adopted as criteria.

4 Learning the Lorenz’63 system

4.1 Lorenz’63 system

To illustrate the different learning strategies, we consider the celebrated Lorenz’63 system \([24]\), originally introduced to mimic the thermal convection in the atmosphere. It consists of three coupled nonlinear ordinary differential equations (ODEs):

\[
\begin{align*}
\dot{x} &= \sigma(y - x), \\
\dot{y} &= \rho x - y - xz, \\
\dot{z} &= xy - \beta z,
\end{align*}
\]

with \(\{\sigma, \rho, \beta\} \in \mathbb{R}^3\). The system of ODEs is integrated in time with a 4th order Runge–Kutta scheme and a time step \(\Delta t = 0.01\). Constants \(\sigma, \rho, \beta\) are positive parameters, related to dimensionless scales such as the Prandtl and the Rayleigh numbers. For \(\rho > \sigma(\sigma + \beta + 3)/(\sigma - \beta - 1)\) and \(\sigma > (\beta + 1)\), the solution \(u := (x, y, z)\) exhibits a chaotic behavior. It is important for the following to note that the system features three unstable fixed points \(u_{\text{FP}}\), associated with a vanishing time-derivative:

\[
\begin{align*}
u_{\text{FP}+} &= (\sqrt{\beta(\rho - 1)}, \sqrt{\beta(\rho - 1)}, \rho - 1), \\
u_{\text{FP}0} &= (\rho, 0, 0), \\
u_{\text{FP}0} &= (\rho, 0, 0).
\end{align*}
\]

In the following, we shall consider the parameters fixed to \(\beta = 8/3, \sigma = 10, \rho = 28\) as in the original work, such that the solution is in the chaotic regime. In this regime, the Kaplan-Yorke dimension of the attractor is \(d_A \approx 2.06\).

4.2 Setup of the learning method

We turn our attention to the training set \(T = \{s_k\}_{k=1}^K\) for learning the Lorenz’63 system. It is a crucial ingredient of the learning process and largely conditions the quality of the resulting model. We consider four different training sets \((T_{\text{full}}, T_{\text{split}}, T_{\text{ens}}\text{ and } T_{\text{FP}}\)) of identical settings (sampling frequency, number of samples, measurement operator, etc.) but following different sampling strategies.
4.2.1 Ergodicity-compliant training sets, $\mathcal{I}_{\text{ergo}}$ and $\mathcal{I}_{\text{split}}$

The $\mathcal{I}_{\text{ergo}}$ dataset is a long trajectory on the attractor. We consider it the reference case as it provides sufficient statistics for learning a reliable and accurate model. With $\varepsilon = 10^{-2}$ and the Kaplan-Yorke dimension $d_A \simeq 2.06$ of the Lorenz’63 for the retained settings \{\sigma, \rho, \beta\}, [22], the required number of samples resulting from a constant sampling period of $\Delta t = 0.01$ is about $27,000$ strobes in time. A second dataset, $\mathcal{I}_{\text{split}}$, is derived from $\mathcal{I}_{\text{ergo}}$. It consists of one long trajectory on the attractor, yet split in nine chunks of equal size and reshuffled, such that each chunk is thus composed by $3000$ samples. As it will be clear when discussing the results (Sect. 5), this dataset will allow us to investigate some effects of memory.

4.2.2 Structured datasets: leveraging the fixed points, $\mathcal{I}_{\text{FP}}$

The training set $\mathcal{I}_{\text{FP}}$ exploits prior knowledge on the structure of the system at hand, namely the fixed points. As mentioned in the Introduction, it is well known that dynamical systems are organized around critical features, such as fixed points, and that invariant sets provide information on the behavior of the solution, [14]. Besides fixed points, these features include higher-dimensional invariant sets such as slow invariant manifolds. The sampling strategy here consists in acquiring observations from the system originating from its (unstable) fixed points, in contrast with other training sets obtained from the attractor. Specifically, three $3000$-sample trajectories are acquired from each of the three fixed points of the Lorenz’63 system. These trajectories originate from a small deviation from each fixed points along each of the $d = 3$ directions given by the eigenvectors of the local Jacobian matrix. Note that, if the Jacobian is not available, a set of orthonormalized deviations can also be used. The resulting training set then consists of $3 \times 3 \times 3000 = 27,000$ samples, consistent in size with the other training sets illustrated in Fig. 2.

4.2.3 Random sampling, $\mathcal{I}_{\text{ens}}$

Finally, we consider the sampling of short trajectories on the attractor. Nine trajectories are considered, each originating from a state randomly chosen on the attractor. Each trajectory is $3000$ sample long as for the other strategies introduced above so that the total size of the training set is again $9 \times 3000 = 27,000$ samples. This training set is representative of the widespread situations where information about a system come in several independent glimpses, as resulting from several measurement campaigns. It is illustrated in Fig. 2.

4.2.4 Entropy characterization

Using the definition of SVD entropy provided Eq. (10) with order $n_e = 3$ and time-delay $\tau = 1$, we charac-

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig2.png}
\caption{The figure qualitatively illustrates the different training sets used in this investigation. Each ergodic dataset ($\mathcal{I}_{\text{ergo}}$) is composed by one long trajectory (a). The datasets with trajectories emanating from the fixed point, exploiting the structure of the attractor ($\mathcal{I}_{\text{FP}}$), and the datasets randomly sampled ($\mathcal{I}_{\text{ens}}$) are composed by an ensemble of $9$ trajectories, of which one example for each is shown in (b) and (c), respectively (see also Sect. 4.2).}
\end{figure}
Fig. 3 Distribution of the SVD entropy (Eq. 10) associated with the four different datasets described in Sect. 4.2: ergodic ($\mathcal{I}_{\text{ergo}}$), ergodic-split ($\mathcal{I}_{\text{split}}$), random sampled ($\mathcal{I}_{\text{rand}}$) and structured ($\mathcal{I}_{\text{FP}}$). Only the structured dataset shows substantially lower entropy, while the other datasets share the same values and distribution.

From the physical viewpoint, the entropy for a time series is defined on the probability transition between the embedded states (unfolded through SVD). In Fig. 3, the entropy distributions are compared for each of the training sets: Quite neatly, the dataset $\mathcal{I}_{\text{FP}}$ is associated with the lowest entropy, while the remaining datasets are all characterized by the same levels. From the physical viewpoint, this behavior is explainable by considering the higher predictability of the dynamics in the vicinity of the fixed points, where the behavior of the states close to them is well approximated by the linear limit. High predictability corresponds to less required information. As we will show in the following, each of the training sets. The SVD entropy algorithm has been chosen after a comparison among different strategies for the entropy estimation. First, we considered convergence of the estimates over long trajectories emanating from different initial conditions, in order to evaluate the consistency of the results in ergodic datasets; second, a validation based on entropy estimators requiring a reduced number of data, namely the Approximate Entropy algorithm by [29], has been performed. Good agreement was found between the different entropy estimators. From the physical viewpoint, the entropy for a time series is defined on the probability transition between the embedded states (unfolded through SVD). In Fig. 3, the entropy distributions are compared for each of the training sets: Quite neatly, the dataset $\mathcal{I}_{\text{FP}}$ is associated with the lowest entropy, while the remaining datasets are all characterized by the same levels. From the physical viewpoint, this behavior is explainable by considering the higher predictability of the dynamics in the vicinity of the fixed points, where the behavior of the states close to them is well approximated by the linear limit. High predictability corresponds to less required information. As we will show in the following,

Fig. 4 Left column: Time evolution of the $y$-component of the Lorenz’63 system, both from the true (true, solid red) and the learned model (prediction, dashed blue) from different training sets and initial conditions. Right column: zoom on the [0, 2] time units interval, with the mean prediction (dashed blue line) from an ensemble of 100 trained models, and the associated 1 standard deviation. Top row: The prediction of the best (out of the ensemble set) model trained from $\mathcal{I}_{\text{short}}$ is shown. The initial condition is drawn from the training set. The prediction is seen to remain reliable up to about 4.3 Lyapunov times (5 time units) before drifting significantly from the truth. Middle row: when the same model is evaluated with an unknown initial condition (test set), the performance deteriorates and the accuracy is lost after less than four time units. When a unique ergodic-compliant trajectory $\mathcal{I}_{\text{ergo}}$ is used to train a model (bottom row), the prediction from an unknown initial condition improves significantly.
this leads to an apparent contradiction: Despite less information is contained in the structured datasets—in a Shannon sense—the possibility of training LSTM with well structured datasets characterized by increasing entropy in time allows to deploy a curriculum learning strategy.

5 Results

We now assess the quality of the models learned from various observation datasets of the Lorenz’63 system.

5.1 How much training information?

The influence of the amount of information to train a model is first considered. Two models with the same architecture (cf. Section 3) are trained from two different datasets. These datasets consist of observations of the solution of the Lorenz system along the same trajectory on the attractor, but of different length. One dataset, $T_{\text{ergo}}$, comprises 27,000 samples, while the second one $T_{\text{short}}$ is its restriction to the first 3000 samples, so that $T_{\text{short}} \subset T_{\text{ergo}}$. Once trained, the resulting models, hereafter, respectively, termed $F_{\text{ergo}}$ and $F_{\text{short}}$, are evaluated in terms of their long-term prediction capability. Results are gathered on Fig. 4. The performance of the models is here evaluated when predicting from a known (i.e., contained in their training set) and unknown (not contained) initial condition, to assess the generalization capability of the models. For all cases, memory is initialized to zero.

A known initial condition is first considered. As can be seen from the top row (left column) where it is plotted against the truth (red), the $F_{\text{short}}$ model allows a reasonable quality of the prediction (plotted in blue) over a significant time horizon of about 6 Lyapunov times (7 time units). Since training the models is achieved via a stochastic procedure, e.g., stochastic gradient descent and random initialization of the parameters, and the landscape of the loss function is not globally convex, the underlying optimization problem ends up in a local optimum and bears an aleatoric contribution. To strengthen the above conclusions to account for the statistical uncertainty of neural network modeling, we hence consider an ensemble of 100 identical independent models and evaluate their performance [1]; the final evaluation is thus based on an aggregate metric that allows to provide a robust assessment of the relative performance across different trainings based on the different datasets. The right column of Fig. 4 shows the mean and standard deviation of the prediction of the ensemble of models for the same conditions as in the left column. From the top row, one may deduce that the prediction of the ensemble of models is very consistent and that the learning is robust. With the same short training set $T_{\text{short}}$, the trained model is now evaluated on an initial condition not contained in the training data, see Fig. 4 middle row. Compared with the previous situation (top row), the performance is seen to deteriorate, both for a single model (left column) and in average (right). In particular, the time horizon before the prediction significantly differs from the truth is much shorter, indicating a poor generalizability properties. When more training data are used, $T_{\text{ergo}}$, the resulting model remains accurate over a long time horizon even when predicting from an unknown initial condition, as seen from Fig. 4 (bottom row). Again, this is a reliable finding as evidenced from the ensemble performance seen on the right column.

In Fig. 5, a complementary analysis is shown, considering the average cumulative 2-norm error over a short time period of 1 time unit. The averaging is performed considering 100 models for each of the four sampling strategies. The models are trained using random initialization of the memory. In the inset (a), the prediction emanating from trajectories included in the training set are shown. In (b), the error is shown when analyzing the test sets: larger deviations are observed for the random sampled datasets.

![Fig. 5](image-url)
and constant, at least within the short time window we consider here; the ergodic case is characterized by higher error due to a larger number of strobos back-propagated during the training in each batch. In (b), the error is shown for the testing sets. First—coherently to Fig. 4—the error increases, while time increases; secondly, larger deviations are observed for the random dataset, while the cumulative error of the other cases is more comparable.

These results illustrate the fact that the training data, relied on to learn a model, have to be informative enough of the system under consideration to allow for generalizable models to be inferred. In particular, the superior performances shown in the top row of Fig. 4 and the low error shown in Fig. 5a, can be attributed to overfitting, as the initial conditions are contained in the learning dataset.

### 5.2 Relevance of the data

Besides the mere size of the training set, a critical aspect is its relevance for informing a model in view of a given objective function. To shed light on this pivotal aspect, the four different training sets introduced in Sect. 3.2, namely the ergodic ($T_{\text{ergo}}$), the ergodic-split ($T_{\text{split}}$), the random sampled ($T_{\text{ens}}$), and the structured ($T_{\text{FP}}$) are now evaluated in terms of the quality of the associated resulting model. To avoid any bias due to the initialization of the parameters, an ensemble of 100 models is trained for each situation and the comparison is made in terms of the ensemble performance [1]. A first metric to quantify the quality of the resulting model is the $d_2$-dimension of the predicted model, computed by estimating the correlation sum in Eq. 11. This criterion is not associated with a given initial condition and is affected by the way the trained model consistently reproduces the fractal nature of the Lorenz’63 system. Results are gathered in Fig. 6 in terms of a histogram of the estimated $d_2$ dimension for the ensemble of 100 models for each of the four training sets. For a more quantitative appreciation, a threshold is defined when the dimension resulting from the trained model differs by more than 25% from the truth $d_A = 2.06$. As expected, the ergodicity-compliant training set $T_{\text{ergo}}$ leads to a good performance, with most models associated with a $d_2$ dimension close to the true value and a few degenerate models such that $d_2 \approx 0$. Only 14% of the models are beyond the threshold. Models trained from $T_{\text{FP}}$ are also seen to achieve a similar level of performance. Again, almost all these models are associated with a rather good $d_2$ dimension, with very few degenerate models. Only 8% of the models are beyond the threshold, an even better performance than models trained from $T_{\text{ergo}}$. In contrast, the $T_{\text{ens}}$ (random) and $T_{\text{split}}$ (split) dataset leads to a rather poor performance model, with a significant part of the models associated with $d_2 \approx 0$ and, respectively, 34 and 26% of the models beyond the threshold. An observation is in order in this regard. The models with $d_2 \approx 0$ are all characterized by being stationary: The dynamics of the system stabilizes toward the fixed points after a short transient. On the other hand, only a limited number of models is characterized by $d_2 \approx 1$; in this case, the temporal dynamics settles on periodic orbits. Interestingly, in most of the stable models, the fixed points are correctly predicted, also for the random datasets where they are not included explicitly; however, their stability features are not correctly predicted, as already stressed.

It may be not quite surprising that models trained with $T_{\text{ens}}$ and $T_{\text{split}}$ are associated with a similar performance. Since the only difference between the two datasets is that the samples in $T_{\text{ens}}$ originate from a collection of 9 independent segments of trajectory, randomly located on the attractor, while $T_{\text{split}}$ is made of 9 contiguous segments. Yet, such behavior for the $T_{\text{split}}$ dataset appears to be not trivial, because it contains exactly the same information of the $T_{\text{ergo}}$ one about the phase space. Further assessment of these observations can be made using a different technical tool, namely t-distributed stochastic neighbor embed-

![Fig. 6 The $d_2$ dimension is here adopted as a metric for assessing the prediction quality of the 100 models trained with each of the 4 training strategies: ergodic ($T_{\text{ergo}}$), random sampled ($T_{\text{ens}}$), structured ($T_{\text{FP}}$), and ergodic-split ($T_{\text{split}}$). The resulting $d_2$ dimension is estimated by analyzing time predictions from trajectories not belonging to the training set. For each training set, the proportion of models associated with an estimated dimension differing by more than 25% from the true dimension $d_A = 2.06$ is indicated in percent.](image-url)
Fig. 7 Illustration of the model parameters via a t-SNE representation. The LSTM memory is initialized to zero. In (a), t-distributed stochastic neighbor embedding (t-SNE) of the parameters of the trained models analyzed in Fig. 6 is shown. Each dot corresponds to a different model and its size visually indicates the accuracy: the larger the dot, the larger the error with respect to $d_A$. In (b), the radial distribution of the t-SNE projected model parameters is depicted. $r$ indicates the distance from the barycenter of the cluster in (a), while the distribution of the models is reported on the vertical axis using the same number of bins. We order from light cyan to darker blue models performing progressively worse in terms of resulting $d_2$ accuracy; note that the level set of the blue gradient is normalized for each subplot, such that only a qualitative information on the distribution along $r$ is included in the graphs.

5.3 Handling the memory

Memory is considered in this section. Instead of the null memory (initialized to zero) considered so far and classically found in many applications, an initial memory chosen at random, drawn from a multivariate Gaussian distribution $\mathcal{N}(0, I)$, is used. Compared to the null initial memory, the performances are seen to significantly increase for all models, as evidenced in Fig. 8. This is consistent with the fact that a nonzero hidden state improves the numerical conditioning of the train-
The contrast in performance between models learned from $\mathcal{I}_{\text{ergo}}$ and $\mathcal{I}_{\text{split}}$, which rely on the same information, leads to identify a key point. The LSTM model trained with $\mathcal{I}_{\text{ergo}}$ learns to best fit one long trajectory. In the case of $\mathcal{I}_{\text{split}}$, it learns to best fit nine short trajectories, which however, when stitched together, match the $\mathcal{I}_{\text{ergo}}$ long trajectory. Yet, their performances are consistently very different in the null initialization situation, emphasizing that the initial memory of the retained recurrent network (LSTM) strongly affects the resulting performance. In the case of $\mathcal{I}_{\text{ergo}}$, the impact of an incorrect initial memory is diluted within a long trajectory and weakly affects the learning quality. When short trajectories are considered instead, the relative impact of an incorrect initialization is stronger and can introduce a significant bias, hence resulting in a poor model. This observation is further supported by the model learned from $\mathcal{I}_{\text{FP}}$. While it also relies on samples originating from nine distinct, non sequential, segments, its performance is good, as evidenced by Fig. 6. What is unique about this particular training set is that the samples are from segments of trajectories originating from (unstable) fixed points, thus associated with a locally linear dynamics. This feature allows the incorrect initial memory to be progressively wiped out in time without significantly affecting the predicted dynamics.

5.4 Can we use less data?

The results shown so far lead to slightly counterintuitive conclusions: On one hand, low entropy indicates that the dynamics described by the datasets $\mathcal{I}_{\text{FP}}$ lead to a simpler training; on the other hand, simpler dynamics means high predictability and thus less information. In that sense, one might wonder whether it is possible to leverage these considerations for reducing the amount of data used during the training: In fact, the linear regions are less informative as they are redundant, in particular, when we consider “slow” trajectories emanating from the fixed points located in the two lobes of the Lorenz’63 attractor. In order to provide more effective datasets, while preserving the curriculum learning idea, we consider here shortened trajectories where the initial samples are removed. The resulting characterization of the shortened datasets, based on the SVD entropy, is contained in Fig. 9: The entropy progressively increases as the redundancy of the dataset reduces. The highest levels of entropy found for the latter case are comparable to the estimate computed for the reference dataset $\mathcal{I}_{\text{ergo}}$, as well as the cases $\mathcal{I}_{\text{ens}}$ and $\mathcal{I}_{\text{split}}$.
Fig. 10 Using the same postprocessing adopted in Fig. 7, the radial distribution of the t-SNE projected model parameters is shown for the model based on the datasets obtained starting from $T_{FP}$ and shortened of the initial states. We order from light cyan to darker blue models performing progressively worse in terms of resulting $d_2$ accuracy; note that, the level set of the blue gradient is normalized for each subplot. As a reference, we consider the case with 3000 (e). A rather compact clustering of the models, i.e., $r < 0.5$, is found up to $t = 2000$ (insets c and d), suggesting good behavior of the trained models. Shorter trajectories lead to less robustness in terms of model weights, as well as larger errors when considering $d_2$

overfitting and with low probability to produce biased results. When further reducing the data, results start to worsen until they definitively deteriorate for the last case $9 \times 1000 = 9000$. Nonetheless, we can conclude that a reduction of 33% of the samples does not lead to loss of performance for the analyzed case. While the focus of the analysis is on data selection and their impact on the quality of performance, it is worth noting that any constraint applied to the cost function can further reduce the amount of data while preserving the performance of the system, for instance by explicitly introducing information about the stability of the system (Lyapunov exponents or the Jacobian matrix) or introducing multi-objective cost functions [23,32]. In that sense, a combination of physics constraints, data selection/distillation and architecture optimization can deeply improve the prediction abilities of data-driven models with a significant reduction in the data necessary for training.

6 Concluding discussion

In this work, we apply the concept of curriculum learning for data-driven modeling of a complex dynamical system: This concept is inspired from human behavior and is based on the idea that better learning is achieved when machines are trained using data of increasing complexity. Among the possible explanations for motivating the effectiveness of this learning strategy, a possible one is related to the shape of the cost landscape, that evolves from being near-convex to more complex as the complexity of the data increases. By sorting the data in this way, the optimization process is favored toward better extrema. Motivated by this methodology, we have performed a thorough analysis of some aspects inherent to the process of learning a complex physical system from observational data using the classical Lorenz’63, owing to its simplicity and hence the possibility to carry out an exhaustive campaign of simulations. As a learning method, we have considered LSTM neural networks, an established recurrent structure often used in such a context. We addressed the following issues:

1. Is there a minimum amount of data needed to obtain a robust model able to generalize?
2. Is it possible to go beyond this limitation thanks to some “data-based” knowledge of the system using a metric?
3. What is the impact of the initial state of the retained model on its resulting performance?

The first two questions are rather fundamental. In particular, the first question concerns the amount of data required for training and it is related to what is known for classical embedding approaches, which is bound to fail when time-series data are not enough to reconstruct the entire phase space. Generally speaking, in absence of prior information on the system at hand besides an estimated upper bound of its dimension, the ergodic theory, and notably the Kac lemma, provides a reliable criterion on the minimal amount of data necessary to train a model. This amount of data grows exponentially with the dimension of the attractor. Models trained on a smaller amount of training data have been shown to consistently lead to poor performance, as illustrated both via the time-series prediction and the resulting model estimated dimension assessment criteria. Our analysis hence suggests that recent results
on dynamical systems showing a larger predictability horizon appear more as a result of overfitting than an intrinsic achievement of neural networks. The second question is addressed by comparing different training sets. In particular, a training set acquired from a system initially close to one of its unstable fixed points exhibits a low entropy, low complexity, behavior owing to the essentially linear local regime, before reaching a higher complexity, fully nonlinear, regime when getting closer to the attractor. This essentially allows to disentangle the contribution of the different regimes of the underlying system and to sequentially focus the learning effort on these different regimes from specific parts of the training set, effectively resulting in a curriculum. Interestingly, shortening the trajectories emanating from the fixed points and following insights based on the entropy contents lead to results consistent with the ones obtained with a larger amount of data. The last issue is related to the objective of introducing best practice guidelines for practitioners. The numerical test provides evidences on the effectiveness of curriculum learning, as this choice leads to consistent and accurate models even with a limited amount of data with respect to what is prescribed by ergodic theory. Moreover, the analyzed experiments have neatly shown that starting from the fixed points of the dynamical system constitutes a workaround in learning the dynamics of the system from situations where the memory is essentially harmless to the learning process. Indeed, in these memory-based models, flushing an initially incorrect memory is crucial for obtaining a relevant and faithful model able to generalize its prediction beyond the sole situations encountered during the training step. Our numerical experiments clearly show that there is an impact of the initialization of the memory on the quality of the predictions provided by the LSTM models and it is particularly severe if one learns from a set of short-termed observations. Superior performance was consistently achieved with a memory initialized at random to avoid bias. While we here focused on the Lorenz’63 system and LSTM models, we believe these findings to be widely applicable and to provide some evidence-based good practice for data-driven modeling of physical systems. Future work will focus on a finer analysis of the interplay between the structure of the dynamical system and the learning process, as well as on the choice of appropriate metrics complementing entropy levels, toward a principled learning strategy. In that sense, curriculum learning strategies can be envisioned as a valuable tool for the analysis of complex dynamical systems and open up avenues for research within the context of active learning.

Acknowledgements This work was funded by the French Agence Nationale de la Recherche via the Flowcon project (ANR-17-ASTR-0022) and the Speed project (ANR-20-CE23-0025-01). L.M. gratefully acknowledges stimulating discussions with Alex Goro-detsky (University of Michigan, US). O.S. thanks Luca de Cicco (Politecnico di Bari, Italy) for exchanges on the role of entropy metrics in curriculum learning. S.C. gratefully acknowledge fruitful discussions with Angelo Vulpiani (University La Sapienza, Italy).

Author contribution statement

MAB contributed to conceptualization, data curation, investigation, methodology, writing. OS contributed to conceptualization, investigation, methodology, writing. AA contributed to methodology, funding acquisition, writing (review and editing). SC contributed to methodology, funding acquisition, writing. LM contributed to investigation, methodology, funding acquisition, writing.

Data availability statement Data and training models used for this article will be made available on reasonable request.

References

1. R. Agarwal, M. Schwarzer, P.S. Castro, A.C. Courville, M. Bellemare, Deep reinforcement learning at the edge of the statistical precipice. Adv. Neural. Inf. Process. Syst. 34, 29304–29320 (2021)
2. Y. Bengio, J. Louradour, R. Collobert, J. Weston, Curriculum learning, in Proc. International Conference on Machine Learning, Montreal, Quebec, June 14–18 (2009)
3. G. Boffetta, M. Cencini, M. Fulcioni, A. Vulpiani, Predictability: a way to characterize complexity. Phys. Rep. 356(6), 367–474 (2002)
4. F. Borra, A. Vulpiani, M. Cencini, Effective models and predictability of chaotic multiscale systems via machine learning. Phys. Rev. E 102(5), 052203 (2020)
5. S.L. Brunton, B.R. Noack, P. Koumoutsakos, Machine learning for fluid mechanics. Annu. Rev. Fluid Mech. 52, 477–508 (2020)
6. F. Camastra, A. Staiano, Intrinsic dimension estimation: advances and open problems. Inf. Sci. 328, 26–41 (2016)
7. J.P. Crutchfield, B.S. McNamara, Equation of motion from a data series. Complex Syst. 1(417–452), 121 (1987)
8. M. de Hoop, R. Baraniuk, J. Bruna, M. Campillo, H. Jasperson, S. Mallat, T. Nguyen, L. Seydoux, Unsupervised learning for identification of seismic signals, in Geophysical Research Abstracts, vol. 21 (2019)
9. J.-P. Eckmann, D. Ruelle, Ergodic theory of chaos and strange attractors. Rev. Mod. Phys. 57, 617–656 (1985)
10. A. Eftekhar, H.L. Yap, M.B. Wakin, C.J. Rozell, Stabilizing embedding: geometry-preserving delay-coordinate maps. Phys. Rev. E 97(2), 022222 (2018)
11. F. A. Gers, D. Eck, J. Schmidhuber, Applying lstm to time series predictable through time-window approaches, in Neural Nets WIRN Vietri-01, pp. 193–200 (2002)
12. M.M. Ghazi, M. Nielsen, A. Pai, M. Modat, M.J. Cardei, S. Ourselin, L. Sorensen, On the initialization of
long short-term memory networks. ArXiv 10454, 2019 (1912).

13. J.F. Gibson, J.D. Farmer, M. Casdagli, S. Eubank, An analytic approach to practical state space reconstruction. Physica D 57(1), 1–30 (1992).

14. R. Gilmore, J.-M. Ginoux, T. Jones, C. Letellier, U.S. Freitas, Connecting curves for dynamical systems. J. Phys. A: Math. Theor. 45(25), 255101 (2010).

15. I. Goodfellow, Y. Bengio, A. Courville, Deep Learning (MIT Press, Cambridge, 2016).

16. P. Grassberger, I. Procaccia, Characterization of strange attractors. Phys. Rev. Lett. 50(5), 346 (1983).

17. S. Hochreiter, J. Schmidhuber, Long short-term memory. Neural Comput. 9(8), 1735–1780 (1997).

18. A. Jacot, F. Gabriel, C. Hongler, C. Neural tangent kernel: convergence and generalization in neural networks, in Advances in Neural Information Processing Systems, vol. 31, ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, R. Garnett (Curran Associates Inc, Red Hook, 2018).

19. M. Kac, Probability and Related Topics in Physical Sciences, vol. 1 (American Mathematical Soc, Providence, 1959).

20. H. Kantz, T. Schreiber, Nonlinear Time Series Analysis, vol. 7 (Cambridge University Press, Cambridge, 2004).

21. K. Kashinath, M. Mustafa, A. Albert, J.-L. Wu, C. Jiang, S. Esmaeilzadeh, K. Azizzadenesheli, R. Wang, A. Chattopadhyay, A. Singh, A. Maneppalli, D. Chirila, R. Yu, R. Walters, B. White, H. Xiao, H.A. Tchelepi, P. Marcus, A. Anandkumar, P. Hassanazadeh, Prabhat, Physics-informed machine learning: case studies for weather and climate modelling. Phil. Trans. Roy. Soc. A 379, 20200093 (2021).

22. N. Kuznetsov, T. Mokaev, O. Kuznetsova, E. Kudryashova, The Lorenz system: hidden boundary of practical stability and the Lyapunov dimension. Nonlinear Dyn. 102, 713–732 (2020).

23. W. La Cava, T. Helmuth, L. Spector, J.H. Moore, A probabilistic and multi-objective analysis of lexicase selection and ε-lexicase selection. Evol. Comput. 27(3), 377–402 (2019).

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

25. S. Narvekar, B. Peng, M. Leonetti, J. Sinapov, M. E. Taylor, P. Stone, curriculum learning for reinforcement learning domains: a framework and survey. J. Mach. Learn. Res., 21(1) (2020).

26. G. Paladin, A. Vulpiani, Anomalous scaling laws in multifractal objects. Phys. Rep. 156(4), 147–225 (1987).

27. J. Pathak, B. Hunt, M. Girvan, Z. Lu, E. Ott, Model-free prediction of large spatiotemporally chaotic systems from data: a reservoir computing approach. Phys. Rev. Lett. 120(2), 024102 (2018).

28. J. Pathak, Z. Lu, B.R. Hunt, M. Girvan, E. Ott, Using machine learning to replicate chaotic attractors and calculate lyapunov exponents from data. Chaos Interdiscip. J. Nonlinear Sci. 27(12), 121102 (2017).

29. S.M. Pincus, Approximate entropy as a measure of system complexity. PNAS 86(6), 2297–2301 (1991).

30. H. Poincaré, Les méthodes nouvelles de la mécanique céleste, volume 3. Gauthier-Villars et fils (1899).

31. M. Quade, M. Abel, J. Nathan Kutz, S.L. Brunton, Sparse identification of nonlinear dynamics for rapid model recovery. Chaos: Interdiscip. J. Nonlinear Sci. 28(6), 063116 (2018).

32. M. Quade, M. Abel, K. Shaﬁ, R.K. Niven, B.R. Noack, Prediction of dynamical systems by symbolic regression. Phys. Rev. E 94(1), 012214 (2016).

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

34. T.D. Sanger, Neural network learning control of robot manipulators using gradually increasing task difﬁculty. IEEE Trans. Robot. Autom. 10, 323–333 (1994).

35. T. Sauer, J.A. Yorke, M. Casdagli, Embedology. J. Stat. Phys. 65(3–4), 579–616 (1991).

36. M. Schmidt, H. Lipson, Distilling free-form natural laws from experimental data. Science 324(5923), 81–85 (2009).

37. P. Soviany, R.T. Ionescu, P. Rota, N. Sebe, Curriculum learning: a survey. Int. J. Comput. Vis. 130(6), 1526–1565 (2022).

38. F. Takens, Detecting strange attractors in turbulence. In D. Rand, L.-S. Young, editors, Dynamical Systems and Turbulence, Warwick 1980, pp. 366–381, Berlin, 1981. Springer, Berlin.

39. L. Van der Maaten, G. Hinton, Visualizing data using t-sne. J. Mach. Learn. Res., 9(11) (2008).

40. R. Varshavsky, A. Gottlieb, M. Linial, D. Horn, Novel unsupervised feature ﬁltering of biological data. Bioinformatiks 22(14), e507–e513 (2006).

41. P.R. Vlachas, J. Pathak, B.R. Hunt, T.P. Sapsis, M. Girvan, E. Ott, P. Koumoutsakos, Backpropagation algorithms and reservoir computing in recurrent neural networks for the forecasting of complex spatiotemporal dynamics. Neural Netw. 126, 191–217 (2020).

42. L. von Rueden, S. Mayer, K. Beckh, B. Georgiev, S. Gieselbach, R. Heese, B. Kirsch, J. Pfommer, A. Pick, R. Ramamurthy, M. Walczak, J. Garcke, C. Bauchhage, J. Schuecker, Informed machine learning—a taxonomy and survey of integrating knowledge into learning systems. IEEE Trans. Knowl. Data Eng. (2021). Accepted.

43. H. Voss, M. Bünner, M. Abel, Identiﬁcation of continuous, spatiotemporal systems. Phys. Rev. E 57(3), 2820 (1998).

44. D. Weinshall, G. Cohen, D. Amir, Curriculum learning by transfer learning: theory and experiments with deep networks, in Proceedings of the 35th International Conference on Machine Learning, pp. 5235–5243. PMLR (2018).

45. H. Whitney, Differentiable manifolds. Ann. Math. 37(3), 645–680 (1936).

46. Y. Zhu, N. Zabaras, P.-S. Koutsourelakis, P. Perdikaris, Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantiﬁcation without labeled data. J. Comput. Phys. 394, 56–81 (2019).

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.