Deep KKL: Data-driven Output Prediction for Non-Linear Systems

Steeven Janny, Vincent Andrieu, Madiha Nadri, Christian Wolf

Deep KKL: Data-driven Output Prediction for Non-Linear Systems

Steeven Janny[1], Vincent Andrieu[2], Madiha Nadri[2], Christian Wolf[3]

***Abstract*— We address the problem of output prediction, ie. designing a model for autonomous nonlinear systems capable of forecasting their future observations. We first define a general framework bringing together the necessary properties for the development of such an output predictor. In particular, we look at this problem from two different viewpoints, control theory and data-driven techniques (machine learning), and try to formulate it in a consistent way, reducing the gap between the two fields. Building on this formulation and problem definition, we propose a predictor structure based on the Kazantzis-Kravaris/Luenberger (KKL) observer and we show that KKL fits well into our general framework. Finally, we propose a constructive solution for this predictor that solely relies on a small set of trajectories measured from the system. Our simulations show that our solution allows to obtain an efficient predictor over a subset of the observation space.**

I. Introduction

A. Context

We investigate the prediction (forecasting) of future observed outputs of a non-linear dynamical system, which is not necessarily observable, and for which we have access to an initial part of the trajectory, as well as to a training set of additional representative trajectories sampled with different initial conditions. This task shares many similarities with system identification, as both problems require to design a model for a specific plant in order to represent its dynamics. Yet, in output prediction, we solely consider the system output rather than the full state representation of the system. For a long time in the literature, common solutions for this class of problem relied on explicitly modeling the physical phenomena exhibited by the dynamical system. The resulting models are then required to be as exhaustive as possible to minimize the prediction error by taking into account every part of the dynamics coming into play. Output predictors are central in diverse applications, like observability (e.g Kalman filtering [1], [2], [3], Luenberger observer [4]) or model predictive control [5].

Recently, data-driven approaches based on machine learning emerged as a valuable alternative to methods based on handcrafted models for a large range of applications, where modeling is difficult, laborious or impossible. These procedures learn the dynamics directly from a set of observations of the system. In its most modern form, Deep Learning, high-capacity deep neural networks are trained from massive amounts of data, with impact on many applications in control theory by complementing classical methods [6], [7] or even replacing them, for instance through Deep Reinforcement Learning [8]. Depending on the concrete application and the amount of available data, recent work tends to demonstrate that neural networks may benefit from hybridization with more classical modeling techniques. Examples are combinations with physical models [9], [10], classical control techniques [11], [12], or adding inductive biases to neural networks encoding domain knowledge such as projective geometry [13], path planning in graphs [14], or even objectives inspired from animal development [15].

In this paper, we develop a framework for designing an output predictor for forecasting the observed output of an unknown dynamical system. While designed from a control theoretic point of view, it is easily transferable to methods based on Deep Learning. Moreover, under some assumptions, we show that an upper bound of the prediction error can be computed for predictors complying with our definition.

As a use case of this general approach, we develop an output predictor based on the Kazantzis-Kravaris/Luenberger observer (KKL) [16] for non-linear systems. Building on theoretical work [17], [18] and [19], we develop a data-driven approach to compute a KKL output predictor without any knowledge of the dynamics which generated the observations. Our method mainly relies on Deep Learning to identify relevant regularities in the training data and extracts a predictor from them. We illustrate some of the capabilities of the model across a variety of simulations. We also highlight the limitations, which are due to this constructive solution for KKL. We compare the proposition with two types of deep networks classically used in the field of machine learning for time series forecasting: *Recurrent Neural Networks* (RNNs) [20] and a more modern variant called *Gated Recurrent Units* (GRUs) [21].

In the same spirit, recent development around the Koopman operator [22], [23] proposes to identify a transformation that projects the state of a system into an infinite dimensional latent space, in which the dynamics is fully linear, and then exploits this representation to explain the output. The Koopman operator shares with our work the idea of using Deep Learning to find a latent representation of a non-linear system from a set of observed data. Nonetheless, there are few keys differences with our contribution:

- Koopman theory gives an infinite-dimensional transformation into a fully linear system. So any finite-dimensional transformation results as an approximation. By relaxing the constraint on output linearity, KKL guarantee the existence of a finite-dimensional transformation under very weak assumptions.

[1]Steeven Janny is with the Université Lyon, INSA Lyon, CNRS, LIRIS, Villeurbanne, France. E-mail: steeven.janny@insa-lyon.fr.

[2]Madiha Nadri and Vincent Andrieu are with Université Lyon, CNRS, LAGEPP, Villeurbanne, France.

[3]Christian Wolf is with Université Lyon, INSA Lyon, CNRS, LIRIS, Villeurbanne, France

- The latent space created by the Koopman operator contains information about the full state, while our proposition requires only the *observable* part of the state to be embedded. Thus, our contribution does not require neither a measurement of the complete state, nor the observability of the system.
- Koopman requires the mapping from the state to the latent representation and its inverse, whereas KKL only requires the identification of the inverse mapping.
- In contrast to our contribution, methods based on the Koopman operator do not take benefit from access to the first steps of the observed trajectory. Their predictions are solely based on the initial state of the system.

B. The output prediction problem

Consider an unknown dynamical system of dimension $n \in \mathbb{N}$ with measured output:

$$\dot{x} = f(x) \ , \ y = h(x) \ , \tag{1}$$

with $f : \mathbb{R}^n \mapsto \mathbb{R}^n$ a smooth vector field and $h : \mathbb{R}^n \mapsto \mathbb{R}$ a smooth observation function. For each $x \in \mathbb{R}^n$, we assume that there exists a unique solution to (1), denoted at time t by $X(x_0, t)$, with x_0 as initial condition. This solution is defined for all time (i.e. we assume forward and backward completeness). We introduce $\mathbb{Y}$, the set of all possible output functions that can be generated by this dynamical system from the set of initial conditions. Formally,

$$\mathbb{Y} = \{y : \mathbb{R}^+ \mapsto \mathbb{R}, \exists x_0 \ , \ y(t) = h(X(x_0, t))\}. \tag{2}$$

The problem we want to solve is the following: *Given a current time h can we infer the future value of an experiment y in $\mathbb{Y}$ given that we know $y(t)$, for t in $[0, h]$?* Note that we may not solve this problem for all y in $\mathbb{Y}$ but at least for those in a particular subset Y of $\mathbb{Y}$.

We address this problem by first defining a framework encapsulating the observation dynamics into a larger dynamical model, said **generative model** with a **contraction** property. This is similar to the idea of an *internal model* [24], as a generative model is a process simulating the system response, with the exception that our definition is not necessarily motivated from control purposes. Under some assumptions, we propose an upper bound of the prediction error over time for such a model.

In a second step, we suggest a possible solution via the Kazantzis-Kravaris/Luenberger (KKL) observer formalism. After proving the existence of a generative model under this particular form, we verify that it also respects the hypothesis required for our upper bound. To demonstrate the feasibility of this solution, and inspired by [25], we design a learning algorithm to discover such KKL models. In our simulations, the KKL-based predictor exhibits remarkable forecasting capabilities, excellent generalization and robustness to noise.

II. Prediction via embedding into an output dependent uniform contraction

A. Uniform contraction and generating model

Consider now a dynamical system in the form:

$$\dot{z} = G(z, y), \tag{3}$$

where z in $\mathbb{R}^m$ and y in $\mathbb{R}$. We denote by $Z(z_0, t, y)$ the solution of (3) at time t initiated from an initial condition z_0. This solution depends only on the values of y for t in $[0, h]$, i.e. it is causal.

Definition 1: [26] System (3) is said to define a **uniform exponential contraction** if there exist two positive constants k and λ s.t. for all locally integrable functions $y : \mathbb{R}_+ \mapsto \mathbb{R}$ and all (z_a, z_b) the two solutions $Z(z_a, t, y)$ and $Z(z_b, t, y)$ initiated respectively from z_a and z_b at t=0 satisfy:

$$|Z(z_a, t, y) - Z(z_b, t, y)| \leq k e^{-\lambda t} |z_a - z_b| . \tag{4}$$

Remark 1: We are interested in this type of dynamical systems because they *forget* their initial conditions. This will be made precise in Proposition 1.

Consider an autonomous system with measured output:

$$\dot{z} = g(z) \ , \ y = \psi(z) \ , \tag{5}$$

where $g : \mathbb{R}^m \to \mathbb{R}^m$ and $\psi : \mathbb{R}^m \to \mathbb{R}$ and where the solution initiated from z in $\mathbb{R}^m$ and evaluated at time t is denoted by $\mathcal{Z}(z, t)$. Let Y be a subset of $\mathbb{Y}$.

Definition 2: A **Generating Model** (GM) for Y is defined as a couple (g, ψ) such that for all y in Y there exists z_0^y in $\mathbb{R}^m$ such that $y(t) = \psi(\mathcal{Z}(z_0^y, t))$.

For instance, (f, h) is a generating model for the entire set $\mathbb{Y}$. A generating model allows to explain an output y in Y via a dynamical system. If we know the initial condition z_0^y associated to y, future values can be predicted by integration of the GM starting from z_0^y.

B. Prediction based on contraction and generating model

We wish to predict the future of any experiments in $Y \subset \mathbb{Y}$. To this end, the following definition provides two necessary conditions.

Definition 3: An **Output Predictor** for $Y \subset \mathbb{Y}$ is defined as a couple (G, ψ) such as

- $\dot{z} = G(z, y)$ is a uniform exponential contraction with parameter (k, λ) as in Definition 1;
- the couple (g, ψ) with $g(z) = G(z, \psi(z))$ is a generating model for Y.

The behavior of an output predictor is outlined in Figure 1. Let h be the number of known timesteps of y and p the number of predicted timesteps. For an output $y \in Y$, we note z_0^y the exact initial condition such that $\psi(Z(z_0^y, t, y)) = y(t)$ and z_0 the (random) initial condition used in the predictor. The prediction is decomposed into three steps:

1) First, the known part of the observation $y(t), t \in [0, h]$ is combined with the contraction property so that $Z(z_0, t, y)$ gets close to $Z(z_0^y, t, y)$. This is the **closed-loop** behavior of the predictor.
2) Then, the autonomous dynamical model $\dot{z} = g(z)$ produces predictions in the latent space $z(t)$, $t \in [h, h + p]$. We refer to this behavior as **open-loop**, since the real observation y is not used as a feedback.
3) Finally, the predicted latent state variables $z(t)$ are input to ψ to compute the output $\hat{y}(t) = \psi(z(t))$.

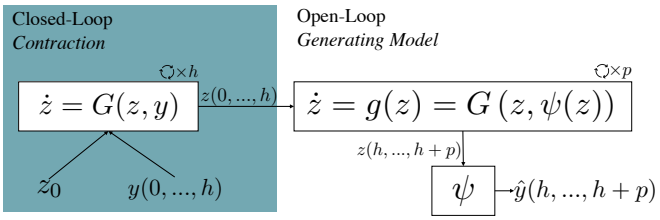

Fig. 1. Computation graph for output predictors. The known part of the observation $y(t)$ for $t < h$ is used to make the latent state $Z(z_0, t, y)$ converge to $Z(z_0^y, t, y)$. During the prediction step, we open the loop and let the autonomous system $\dot{z} = g(z)$ perform the prediction.

Furthermore, if the dynamics of the latent representation g, and the map ψ are Lipschitz, one can compute an upper bound of the prediction error due to an error on the initial condition z_0.

Proposition 1: Assume there exist $G : \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}^m$ and $\psi : \mathbb{R}^m \to \mathbb{R}$, both C^1, such that (G, ψ) defines an output predictor for Y with:

$$\left|\frac{\partial g}{\partial z}(z)\right| \leq L_1 \ , \ \left|\frac{\partial \psi}{\partial z}(z)\right| \leq L_2 \ , \tag{6}$$

with $g(z) = G(z, \psi(z))$, L_1 and L_2 in $\mathbb{R}^+$, then for all experiments $y \in Y$, known in the time interval $[0, h]$, the prediction $\hat{y}$ at the prediction horizon $p > 0$ is given as:

$$\hat{y}(h+p) = \psi(\mathcal{Z}(Z(0, h, y), p)) \ , \tag{7}$$

and satisfies

$$|\hat{y}(h+p) - y(h+p)| \leq kL_2 e^{-\lambda h + L_1 p} |z_0^y| \ . \tag{8}$$

The proof of proposition 1 is detailed in Appendix VII-A.

Remark 2: The prediction mismatch is upper-bounded by a term, which has the following properties:

- As the prediction horizon p increases, the prediction error grows as well. This growth is exponential and depends mainly on the Lipschitz constant of g denoted L_1.
- As h increases, we obtain more information on the output before predicting. For each fixed prediction horizon, the upper-bound exponentially goes to zero for increasing p.

III. A POSSIBLE SOLUTION VIA KKL

A. KKL as an output predictor

In what follows, we derive the KKL observer structure to build an output predictor in the sense of Definition 3. For the sake of following mathematical consideration, the state space is reduced to a compact subset $\mathcal{O} \subset \mathbb{R}^n$, and we assume that it is invariant along the dynamics, ie. for all x_0 in $\mathcal{O}$:

$$X(x_0, t) \in \mathcal{O} \ , \forall t \ \in \mathbb{R}.$$

We introduce $Y_{\mathcal{O}} \subset \mathbb{Y}$, the set of output functions that can be generated by this dynamical system when restricting x_0 to be in $\mathcal{O}$:

$$Y_{\mathcal{O}} = \left\{ y : \mathbb{R}^+ \mapsto \mathbb{R}, \exists x_0 \in \mathcal{O} \ , \ y(t) = h\left(X(x_0, t)\right) \right\} \ . \tag{9}$$

Inspired by the KKL observers, see [16] or [17], we consider the particular case in which the contracting model given in (3) is defined on $\mathbb{R}^m$ for some $m \in \mathbb{N}$ and is in the form:

$$G(z, y) = Az + by \ , \tag{10}$$

with $A \in \mathbb{R}^{m \times m}$ a Hurwitz matrix and $b \in \mathbb{R}^m$ such that (A, b) is a controllable pair. The dynamical model (3) with G defined in (10), trivially defines a uniform contraction since for all $(z_a, z_b) \in \mathbb{R}^m \times \mathbb{R}^m$ and a given $y \in \mathbb{Y}$:

$$|Z(z_a, t, y) - Z(z_b, t, y)| = e^{At} |z_a - z_b| \ . \tag{11}$$

Since A is Hurwitz, it yields the existence of k and λ such that (4) holds. To show that this formalism also defines a GM, we need to find A, b and a function ψ such that $\dot{z} = Az + b\psi(z)$ generates the output. With the use of Proposition 1, 2 and 3 from [18], we have the following statement:

Theorem 1: With $m = 2n + 2$, there exist a Hurwitz matrix A and a vector b with (A, b) controllable and a continuous mapping $\psi : \mathbb{R}^m \mapsto \mathbb{R}$ such that with G defined in (10), (G, ψ) defines an output predictor for $Y_{\mathcal{O}}$.

Thus, this result confirms that a linear contraction in the form (10) may define an output predictor. The proof of Theorem 1 is given in the Appendix.

Remark 3: Going through the proof, it turns out that almost any complex couple (A, b) of dimension $m' = n + 1$ can be chosen to prove the existence of ψ, as long as A is Hurwitz and (A, b) controllable. One can readily extend the m'-dimensional complex case to our m-dimensional real equation by choosing $m = 2m'$.

B. Lipschitz KKL predictor

The bounds on the prediction error obtained in Proposition 1 depend on the Lipschitz constants of ψ and g where $g(z) = Az + b\psi(z)$. However, the mapping ψ obtained from Theorem 1 may not be globally Lipschitz. In [19] some sufficient conditions have been obtained to construct a globally Lipschitz mapping ψ based on some geometric observability assumptions. Inspired by the result obtained in [27] it can be shown that when the dynamical system to predict is observable, a global Lipschitz mapping ψ may be obtained. Consequently, Proposition 1 may be employed.

Proposition 2: Assume that h is a globally Lipschitz mapping. Assume moreover that the following two observability conditions are satisfied.

- *Backward Distinguishability:* for all (x_1, x_2) in $\mathcal{O}^2$ such that $x_1 \neq x_2$, there exists $t \leq 0$ such that $h(X(x_1, t)) \neq h(X(x_2, t))$.
- *Backward Infinitesimal Distinguishability:* for all (x, v) in $\mathcal{O} \times \mathbb{R}^n$ such that $v \neq 0$, there exists $t \leq 0$ such that

$$\frac{\partial h(X(x, t))}{\partial x} v \neq 0$$

then there exist a Hurwitz matrix A, a vector b with (A, b) controllable, a mapping $\psi : \mathbb{R}^m \mapsto \mathbb{R}$ and a positive real number L_2 such that

1) with G defined in (10) (G, ψ) defines an output predictor for $Y_{\mathcal{O}}$;
2) the function ψ has bounded derivative. i.e.

$$\left|\frac{\partial \psi}{\partial z}(z)\right| \leq L_2 \ , \ \forall z \in \mathbb{R}^m;$$

3) the conclusion of Proposition 1 holds with $L_1 = \|A\| + \|b\| L_2$.

Actually, the assumptions of the former proposition can be weakened by assuming that there exists an (unknown) change of coordinates, such that in such a coordinate system (1) takes a triangular form

$$\left\{ \begin{array}{l} \dot{x}_1 = f_1(x_1) \\ \dot{x}_2 = f_2(x_1, x_2) \end{array} \right. , \quad y = h(x_1) \ , \tag{12}$$

for which the couple (f_1, h) satisfies the observability assumptions of the proposition. In that case, the former proposition may be applied. Assuming the existence of this change of coordinates is very similar to the assumptions made in [19] to obtain that this mapping is globally Lipschitz.

IV. Learning ψ with deep networks

In what follows, we propose a constructive method to find ψ based on Deep Learning. We suppose to have access to two different types of data: (i) during a training phase, we have access to a representative training set of sample trajectories $Y_D \subset \mathbb{Y}$ to learn $\psi_\theta(z)$, where we now have made explicit in the notation the dependency of ψ on learned parameters θ; (ii) for each experiment, as described in the previous sections, we have access to the initial output trajectory $y(t) \in Y$ for $t < h$, and are required to forecast the future output up to time $h + p$ (where p is the prediction horizon).

A. Modeling ψ_θ

We model function ψ_θ as a *Multilayer Perceptron* (MLP) where θ in $\Theta \subset \mathbb{R}^q$ is the set of parameters to be learned. This class of models is known to have universal approximation power under mild conditions either for infinitely wide [28] or infinitely deep (ie. layered) [29] model architectures, and they also have the advantage that methods exist to limit the Lipschitz constants of the class of learned functions (see [30], [31] for example).

Since the previous section proves the existence of $\psi\theta$ regardless of the choice of (A, b) (as long as A is Hurwitz), we decided to learn A freely and fix $b = (1 \, ... 1)$, which reduces the number of degrees of freedom of the model. All parameters are trained by gradient descent to minimize:

$$\begin{aligned} (\theta, A) = \arg\min_{\theta, A} & \sum_{y \in Y_D} \sum_{t=0}^{h+p} \|y(t) - \psi_\theta(z(t))\|^2 \\ \text{subject to } \dot{z}(t) = & \left\{ \begin{array}{l} Az(t) + by(t) \text{ if } t < h \\ Az(t) + b\psi_\theta(z(t)) \text{ else} \end{array} \right. \end{aligned} \tag{13}$$

For the sake of implementation simplicity, we used a discrete formulation of the dynamics for our simulations.

	RNN	GRU	KKL
Van Der Pol	0.0057	0.0343	0.**0013**
Lotka-Volterra	**0.0885**	0.1780	0.1064
Lorenz	0.0441	0.0480	**0.0262**
Mean-Field	0.2254	0.2044	**0.0012**

TABLE I : MSE on testing set with $h = 5$ and $p = 95$. The accuracy of Deep KKL is at least equal to those of the classic GRU and RNN.

B. Data-sets and baselines

We compare our proposition to two classical types of deep neural networks designed for time series, namely *Recurrent Neural Networks* (RNNs) [20]

$$z_{t+1} = \tanh(W_1 z_t + W_2 y_t + b) \tag{14}$$

and *Gated Recurrent Units* (GRUs) [21]

$$\begin{array}{l} r_{t+1} = \sigma(W_{r1} y_t + W_{r2} z_t + b_r) \\ x_{t+1} = \sigma(W_{x1} y_t + W_{x2} z_t + b_x) \\ n_{t+1} = \tanh(W_{n1} y_t + r_{t+1} * (W_{n2} z_t + b_{n2}) + b_{n1}) \\ z_{t+1} = (1 - x_{t+1}) * n_{t+1} + x_{t+1} * z_t. \end{array} \tag{15}$$

where $\sigma(x) = (1 + e^{-x})^{-1}$ is the sigmoid function and $*$ is the Hadamard product. These models contain inductive biases in the form of a recurrent memory vector z_t, which allows to propagate hidden state over time t. In other words, they define latent dynamical systems $z_{t+1} = G(z_t, y_t)$. The function ψ_θ has the same structure for each of the two variants.

To our knowledge, no proof exists that RNNs and GRUs define proper output predictors in the sense of Definition 3. Depending on the learned matrix W_1, the function learned by the RNN may define a contraction (since $\tanh$ is monotonic), but there is no rigorous proof that ψ exists for this formalism.

We evaluate our proposition on four different problems that exhibits chaotic behavior: *Van Der Pol* oscillator [32], *Lorenz* attractor [33], *Lotka-Volterra* predation equations [34] and a *Mean Field* model [35] for a fluid flow past a cylinder. Appendix VII-D provides details about these systems.

V. Numerical Simulations

A. Global performances

Table I reports the Mean Squared Error (MSE) on prediction for each model on all four datasets, namely:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{Np} \sum_{y \in Y_T} \sum_{t=h}^{h+p} (y(t) - \hat{y}(t))^2 \tag{16}$$

where Y_T is the test set of trajectories, of cardinality N. To evaluate the temporal generalization capacities of all models, they were evaluated on a more difficult task than the one they were trained on. They were trained on predicting p=25 future measurements by exploiting h=25 previous measurements. However, during testing, the MSE of Table I was calculated over p=95 predictions after having seen only h=5 initial time steps. The results show that KKL generalizes efficiently over this broader horizon, despite the drastic decrease in the amount of data supplied as input (see Figure 2).

On our test systems, the accuracy of Deep KKL is at least equal to those of the classic GRU and RNN, in spite of its

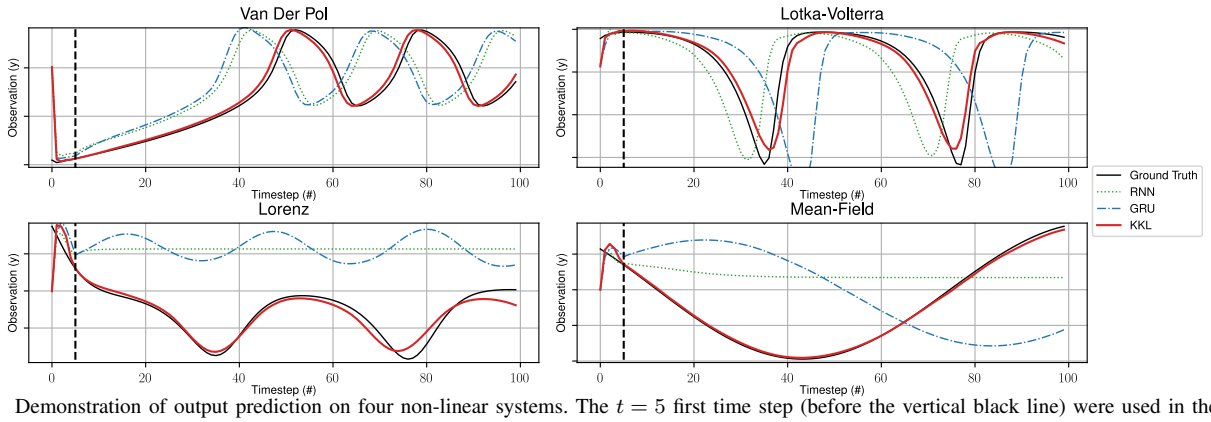

Fig. 2. Demonstration of output prediction on four non-linear systems. The $t = 5$ first time step (before the vertical black line) were used in the closed loop behavior of each models, then the open-loop predicts the $p = 95$ following measurements

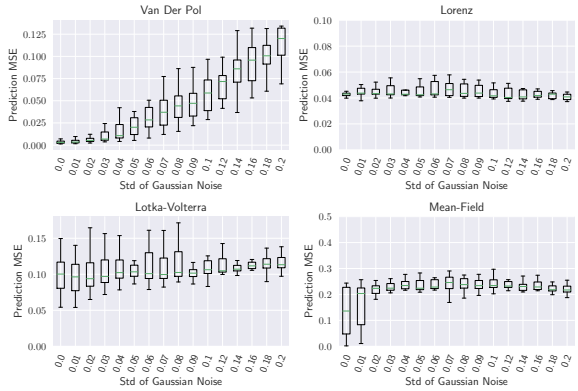

Fig. 3. Boxplot of MSE on the test set Y_T according to the amount of noise added during training. Observation measurements lie in $[-1, 1]$. Deep KKL is capable to deal with a reasonable amount of noise in the training data.

inherent simplicity. By our simulations, we show that Deep KKL is efficient for output prediction on systems of small dimension, while offering a structure more suitable for the elaboration of guarantees. Nevertheless, in practice, the RNN and GRU deep models are rarely used in their simple form, and are generally stacked, i.e. multi-layered, where one layer takes as input the state of the previous layer. We do not claim, that on systems with very complex dynamics (stochasticity / uncertainty, large dimensions, strong non-linearity, etc.) Deep KKL will be competitive with more complex and expressive models (eg. [9], [36]). However, in our examples, Deep KKL takes advantage of its simpler structure and manages to perform better. This seems to indicate that for systems of moderate complexity, the use of high-capacity deep models does not seem to be a guarantee of better results.

B. Noise Robustness

In an experimental setup, measurements are inevitably disturbed by noise and errors, either due to mechanical disturbances on the systems or electronic noise associated to the measurement, etc. We decided to evaluate these settings by training our model on noisy observations. In practice, we altered the measured output $y \in Y_D$ with Gaussian noise of zero mean and varying standard deviation.

Figure 3 shows the evolution of prediction error made by

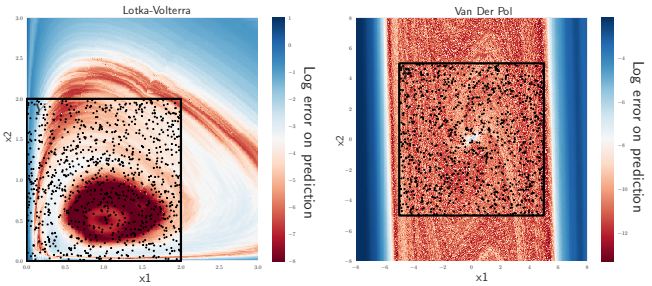

Fig. 4. Generalization on unseen domain $\mathbb{Y}$ of Deep KKL for *Van Der Pol* and *Lotka-Volterra* equations. Each dot represents log-MSE on a trajectory starting from the corresponding initial condition $(x_1\ x_2)$. The black square represents the domain of the training set, training trajectories are black dots.

Deep KKL as a function of the amount of noise added to the training set. Our proposed method is still able to learn with a reasonable amount of noise on the training data.

C. Limitations due to Learning

On top of the initialization error detailed in Proposition 1, using Deep Learning implies another source of error due to the fact that for a given θ in Θ the estimation ψ_θ is merely an approximation of the true ψ on Y, which leads to errors in the open-loop phase of the prediction process. The universal approximation theorem of neural networks [37] guarantees that if we allow the set of necessary parameters to be arbitrary large, then for an arbitrary choice of a constant $\delta > 0$, there exists a set of parameters θ in Θ such that

$$|\psi(z) - \psi_\theta(z)| \leq \delta \ , \quad z \in \mathbb{R}^m. \tag{17}$$

The evaluation of the constant bound $\delta > 0$ is difficult, since we do not have access to the ground truth ψ. The errors $|\psi(z) - \psi_\theta(z)|$ can have multiple reasons, and we will here ignore aspects of learnability [38], and concentrate on how a given error obtained by ψ_θ impacts the prediction error over time. We formalize this as the following proposition.

Proposition 3: Consider $Y \subset \mathbb{Y}$. Assume that (A, b, ψ) exists such that (G, ψ) with G defines in (10) is a KKL output predictor for Y. Assume moreover that:

$$\left| \frac{\partial \psi}{\partial z}(z) \right| \leq L_2 \ . \tag{18}$$

and that θ in Θ and $\delta > 0$ satisfy (17). Then for all experiments $y \in Y$, known in the time interval $[0, h]$, a prediction $\hat{y}_\theta$ at the prediction horizon $p > 0$ given as:

$$\hat{y}_\theta(h+p) = \psi_\theta(\mathcal{Z}_\theta(Z(0,h,y),p)) \ , \tag{19}$$

where $\mathcal{Z}_\theta(z_0, p)$ is the solution initiated from z_0 at time p of

$$\dot{z}_\theta = Az_\theta + b\psi_\theta(z_\theta) \ , \tag{20}$$

satisfies

$$\begin{aligned} |y(h+p) - \hat{y}_\theta(h+p)| \leq kL_2 e^{-\lambda h + L_1 p}|z_0^y| \\ + \delta\left(\sqrt{e^{L_3 p} - 1} + 1\right) \ , \end{aligned} \tag{21}$$

for some positive numbers k, λ, L_1, L_3 depending on L_2, A and b.

The proof for this proposition is given in appendix VII-E.

We complete this theoretical analysis by an experimental evaluation, in particular visualization of the generalization capabilities of our model. A central question in machine learning is how a model can generalize from the data it has seen during training, and thus how it performs on unseen data. Of particular interest is the distinction between ID (in-distribution) and OOD (out-of-distribution) cases, the latter describing the performance of the model on samples taken from large parameter spaces unseen during training. We explore this question and visualize the behavior of Deep KKL on a larger domain than the set from which the training trajectories have been sampled.

In Figure 4, we compute the Log-MSE of Deep KKL on a grid of trajectories from the *Van der Pol* oscillator and *Lotka-Volterra* equations. For each point on the heat-map, we generate the true trajectories from the corresponding initial condition $x(t=0) = (x_1 \quad x_2)^T$ by integrating the corresponding ODE. Then, we use Deep KKL to predict the output of this system and compare the trajectories. The black square represents the set from which the trajectories in Y_D were sampled.

There is evidence for excellent in-distribution generalization, as Deep KKL generalizes well inside the set parameter space covered by Y_D, of course beyond the samples of Y_D themselves. However, we observe limited, but not full OOD generalization, with failure cases when certain parameters are extended beyond the range seen during training.

VI. Conclusion

We have proposed a theoretical framework for predicting the output of dynamical systems, making it possible to easily define a device capable of representing the dynamics of the observations, and resting solely on two properties. Our proposal is illustrated in a KKL observer combined with learning a solution on a subspace of the observation space with deep neural networks. Our simulations validate our theoretical results, and demonstrate that Deep KKL is capable of representing the dynamics of chaotic systems of low dimension. However, the use of a learning methods inevitably generates a certain error in the estimates of ψ. Therefore, we proposed a quantification of the effect of this error on the predictions over time.

Future work will address learnability and sample complexity and explore the derivation of sufficient conditions on the training set Y_D and on the working set Y required for low estimation error δ.

VII. Appendix

A. Proof of Proposition 1

Note that

$$Z(z_0^y, h, y) = \mathcal{Z}(z_0^y, h) \ . \tag{22}$$

Hence, with the contraction property (1), it gives :

$$|Z(0,h,y) - \mathcal{Z}(z_0^y, h)| \leq ke^{-\lambda h}|z_0^y| \ . \tag{23}$$

Due to the Lipschitz property, it yields for all (z_a, z_b) and all $p \geq 0$

$$|\mathcal{Z}(z_a, p) - \mathcal{Z}(z_b, p)| \leq e^{L_1 p}|z_a - z_b| \ . \tag{24}$$

Setting $z_a = Z(0,h,y)$ and $z_b = \mathcal{Z}(z_0^y, h)$, the former inequality becomes

$$\begin{aligned} |\mathcal{Z}(Z(0,h,y),p) - \mathcal{Z}(z_0^y, h+p)| \\ \leq e^{L_1 p}|Z(0,h,y) - \mathcal{Z}(z_0^y, h)| \ , \\ \leq ke^{-\lambda h + L_1 p}|z_0^y| \ . \end{aligned}$$

Since (g, ψ) is a generating model, and since (18) holds, it yields

$$\begin{aligned} |\hat{y}(t+p) - y(t+p)| \\ = |\psi(\mathcal{Z}(Z(0,h,y),p)) - \psi(\mathcal{Z}(z_0^y, h+p))|, \\ \leq L_2 ke^{-\lambda h + L_1 p}|z_0^y| \ . \end{aligned} \tag{25}$$

■

B. Proof of Theorem 1

Theorem 1 mostly relies on the results obtained in [17] in the context of observer designs and [18] in the context of output regulation. The proof of this statement relies on the existence of a C^1 function $T : \mathcal{O} \mapsto \mathbb{R}^m$ mapping x to z which satisfies the differential equation :

$$L_f T(x) = AT(x) + bh(x) \quad \forall x \in \mathcal{O} \ , \tag{26}$$

where $L_f T$ is the Lie derivative of T along f. The functions ψ and T need to satisfy the equality

$$\psi(T(x)) = h(x) \quad \forall x \in \mathcal{O} \ . \tag{27}$$

Given a Hurwitz matrix A, as shown in [17], the following function T

$$T(x) = \int_{-\infty}^{0} e^{-At} bh(X(x,t))dt \ , \tag{28}$$

is well defined for x in $\mathcal{O}$ and satisfies (26). It can be shown that T is C^1 if the eigenvalues of A are smaller than a specific value depending on the Lipschitz constant of f. The proof of this results is detailed in [17] (see Theorem 2.4). To find a function ψ such that (27) is satisfied, we need to ensure that T contains enough information to represents

the observation y. This requirement can be expressed as a pseudo-injectivity with regards to h :

$$\forall (x_1, x_2) \in \mathcal{O} \quad T(x_1) = T(x_2) \Rightarrow h(x_1) = h(x_2). \quad (29)$$

It is shown in [18, Proposition 2] that this condition is satisfied provided $m = 2(n+1)$ and A is the real representation of a Hurwitz diagonal matrix. Finally, [18, Proposition 3] states the existence of ψ.

In conclusion, if the dimension of $z \in \mathbb{R}^m$ is greater or equal to $m = 2n+2$, then there exists a continuous function $\psi : \mathbb{R}^m \mapsto \mathbb{R}$ such that for any experiments y in $Y_{\mathcal{O}}$, there exists z_0^y such that:

$$\begin{array}{l} \dot{z} = Az + by \quad z(0) = z_0^y \\ \psi\big(Z(z_0^y, t, y)\big) = y(t) \quad \forall t \end{array} \quad (30)$$

C. Proof of Proposition 2

The proof of Proposition 2 relies mostly on the results presented in [27]. We follow the steps of the proof of Theorem 1. However, it is shown in [27, Proposition 3.5] and [27, Proposition 3.6] that if $m = 2n + 2$, there exist (A, b) such that the function T given in (28) is injective and full rank in $\mathcal{O}$. Employing [27, Lemma 3.2], we obtain the existence of a positive real number L_T such that

$$L_T |T(x_1) - T(x_2)| \geq |x_1 - x_2|, \quad \forall (x_1, x_2) \in \mathcal{O} \ . \quad (31)$$

Hence, denoting L_h the Lipschitz constant of h, for all (z_1, z_2) in $T(\mathcal{O})^2$, it yields

$$\begin{aligned} |h(T^{-1}(z_1)) - h(T^{-1}(z_2))| &\leq L_h |T^{-1}(z_1) - T^{-1}(z_2)| \ , \\ &\leq L_h L_T |z_1 - z_2| \ . \end{aligned}$$

Defining ψ as a global Lipschitz extension of $h \circ T^{-1}$ to $\mathbb{R}^m$ yields the first and second part with $L_2 = L_h L_T$ of the proposition. The third part of the Proposition is simply obtained by noticing that with $g(z) = Az + b\psi(z)$,

$$\left| \frac{\partial g}{\partial z}(z) \right| = \left| A + b \frac{\partial \psi}{\partial z}(z) \right| \leq |A| + |b| L_2.$$

D. Training and Architecture details

1) Creating the dataset: We used the following systems to evaluate our proposition. δt is the final sampling time, and $\mathcal{D}$ the set from where the initial conditions were sampled.

- **Van der Pol Oscillator** [32] : $\delta t = 0.25$ and $x_0 \in \mathcal{D} = [-5, 5]^2$

$$\left\{ \begin{array}{ll} \dot{x}_1 & = x_2 \\ \dot{x}_2 & = (1 - x_1^2) x_2 - x_1 \end{array} \right. \quad (32)$$

- **Lorenz Attractor** [33] $\delta t = 0.02$ and $\mathcal{D} = [-20, 20] \times [-1, 1]^2$

$$\left\{ \begin{array}{ll} \dot{x}_1 & = 10(x_2 - x_1) \\ \dot{x}_2 & = 24 x_1 - x_2 - x_1 x_3 \\ \dot{x}_3 & = x_1 x_2 - \frac{8}{3} x_3 \end{array} \right. \quad (33)$$

- **Lotka-Volterra Equations** [34] $\delta t = 0.25$ and $\mathcal{D} = [0, 2]^2$

$$\left\{ \begin{array}{ll} \dot{x}_1 & = x_1(\frac{2}{3} - \frac{3}{4} x_2) \\ \dot{x}_2 & = x_2(x_1 - 1) \end{array} \right. \quad (34)$$

- **Mean-Field** [35] We set $\delta t = 0.05$ and sample the initial conditions such that $x_1 = r \cos\theta$, $x_2 = r \sin\theta$ and $x_3 = x_1^2 + x_2^2$ with $r \in [0, 1.1]$ and $\theta \in [0, 2\pi]$, as suggested by [22].

$$\left\{ \begin{array}{ll} \dot{x}_1 & = 0.1 x_1 - x_2 - 0.1 x_1 x_3 \\ \dot{x}_2 & = x_1 + 0.1 x_2 - 0.1 x_2 x_3 \\ \dot{x}_3 & = -10(x_3 - x_1^2 - x_2^2) \end{array} \right. \quad (35)$$

For each model, we tried to predict the observation $y = h(x) = x_1$. We used 1000 trajectories for the training set and 200 for the validation and testing set respectively. These trajectories are generated by solving the differential equation numerically using RK4 solver with a resolution $10\times$ superior than the final sampling. Finally, the observations have been re-scaled so that the training set lies between -1 and 1.

2) Training details: ψ_θ is an MLP with 3 hidden layers of 128 neurons each. We used ReLU activation functions. Canonically, the dimension of the latent space is equal to $2n+1$ where n is the dimension of the system. Each model is trained with Adam optimizer for 800 epochs, with 64 trajectories per batches.

The learning rate is set to 10^{-4}. During training, the model takes as input the $h = 25$ first time steps of the output and outputs the $p = 25$ following time step. Hyper-parameters were optimized over the validation set. For testing, we reduced h to 5 time steps, and increased p to 95.

E. Proof of Proposition 3

The idea of the proof is to compare $\hat{y}_\theta$ obtained from ψ_θ with the prediction $\hat{y}$ defined in (7) obtained employing the nominal mapping ψ. Note that

$$|\psi(z) - \psi_\theta(z_\theta)| \leq |\psi(z) - \psi(z_\theta)| + |\psi(z_\theta) - \psi_\theta(z_\theta)| \ . \quad (36)$$

With (17) and knowing that ψ is L_2-Lipschitz

$$|\psi(z) - \psi_\theta(z_\theta)| \leq L_2 |z - z_\theta| + \delta \ . \quad (37)$$

On the other hand, A being Hurwitz, there exist P a positive definite matrix and $\lambda > 0$ such that

$$AP + A^T P \leq -2\lambda P \ .$$

For two vectors (u, v) in $\mathbb{R}^m$, let us denote $\langle u, v \rangle_P = u^\top P v$ and $\|u\|_P = u^T P u$. Along the solutions of the system (20) and (5) with $g(z) = Az + b\psi(z)$ it yields

$$\begin{aligned} \frac{\partial}{\partial t} \|z - z_\theta\|_P^2 &= (z - z_\theta)^T (AP + A^T P)(z - z_\theta) \\ &\quad + 2 \langle z - z_\theta, b(\psi(z) - \psi_\theta(z_\theta)) \rangle_P \ . \end{aligned} \quad (38)$$

Since $\langle u, v \rangle_P \leq \frac{2\lambda \|u\|_P}{2} + \frac{\|v\|_P}{4\lambda}$, it gives

$$\begin{aligned} \frac{\partial}{\partial t} \|z - z_\theta\|_P^2 &\leq -2\lambda \|z - z_\theta\|_P^2 + \Big(\lambda \|z - z_\theta\|_P^2 \\ &\quad + \frac{\|b\|_P^2}{\lambda} |\psi(z) - \psi_\theta(z_\theta)|^2 \Big) \ . \end{aligned} \quad (39)$$

Again with (17) and ψ Lipschitz, it yields

$$\frac{\partial}{\partial t} \|z - z_\theta\|_P^2 \leq \frac{\|b\|_P^2}{2\lambda} \Big(L_2 \|z - z_\theta\|_P^2 + \delta \Big) \ . \quad (40)$$

With Grönwall inequality, it yields,

$$\|\mathcal{Z}(z,p) - \mathcal{Z}_\theta(z,p)\|_P^2 \leq \frac{\delta^2}{L_2^2}\left(e^{\frac{L_2^2\|b\|_P^2}{2\lambda}p} - 1\right) \ , \forall (z,p). \tag{41}$$

This implies with $\hat{y}$ defined in (7) :

$$|\hat{y}(h+p) - \hat{y}_\theta(t+p)| \leq \delta\left(\sqrt{e^{\frac{L_2^2\|b\|_P^2}{2\lambda}h} - 1} + 1\right).$$

However,

$$\begin{aligned}|y(h+p) - \hat{y}_\theta(h+p)| \leq \\ |y(h+p) - \hat{y}(h+p)| + |\hat{y}(h+p) - \hat{y}_\theta(h+p)|,\end{aligned} \tag{42}$$

and employing Proposition 1 it finally implies

$$\begin{aligned}|y(h+p) - \hat{y}_\theta(h+p)| \leq kL_2 e^{-\lambda h + L_1 p}|z_0^y| \\ + \delta\left(\sqrt{e^{\frac{L_2^2\|b\|_P^2}{2\lambda}p} - 1} + 1\right) \ ,\end{aligned} \tag{43}$$

where k is obtained from P and $L_1 = \|A\| + L_2\|b\|$ and $L_3 = \frac{L_2^2\|b\|_P^2}{2\lambda}$. This concludes the proof.

■

REFERENCES

[1] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, 1960.

[2] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, 2004.

[3] E. Laroche, E. Sedda, and C. Durieu, "Methodological insights for online estimation of induction motor parameters," *Control Systems Technology, IEEE Transactions on*, 2008.

[4] D. G. Luenberger, "Observing the state of a linear system," *IEEE Transactions on Military Electronics*, 1964.

[5] J. Richalet, A. Rault, J. Testud, and J. Papon, "Model predictive heuristic control: Applications to industrial processes," *Automatica*, 1978.

[6] S. S. Pon Kumar, A. Tulsyan, B. Gopaluni, and P. Loewen, "A deep learning architecture for predictive control," *10th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2018*, 2018.

[7] J. Peralez, F. Galuppo, P. Dufour, C. Wolf, and M. Nadri, "Data-driven multimodel control waste for heat recovery system on a heavy duty truck engine," in *2020 59th IEEE Conference on Decision and Control (CDC)*, 2020.

[8] L. Buşoniu, T. de Bruin, D. Tolić, J. Kober, and I. Palunko, "Reinforcement learning for control: Performance, stability, and deep approximators," *Annual Reviews in Control*, 2018.

[9] E. de Bezenac, A. Pajot, and P. Gallinari, "Deep learning for physical processes: Incorporating prior scientific knowledge," in *International Conference on Learning Representations*, 2018.

[10] V. L. GUEN, Y. Yin, J. DONA, I. Ayed, E. de Bezenac, N. THOME, and patrick gallinari, "Augmenting physical models with deep networks for complex dynamics forecasting," in *International Conference on Learning Representations*, 2021.

[11] G. Shi, X. Shi, M. O'Connell, R. Yu, K. Azizzadenesheli, A. Anandkumar, Y. Yue, and S.-J. Chung, "Neural lander: Stable drone landing control using learned dynamics," in *Conference: 2019 International Conference on Robotics and Automation (ICRA)*, 2019.

[12] K. Zeng, R. Mottaghi, L. Weihs, and A. Farhadi, "Visual Reaction: Learning to Play Catch with Your Drone," in *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[13] E. Beeching, O. S. J.Dibangoye, and C. Wolf, "Egomap: Projective mapping and structured egocentric memory for deep rl," in *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD)*, 2020.

[14] E. Beeching, J. Dibangoye, O. Simonin, and C. Wolf, "Learning to reason on uncertain topological maps," in *European Conference on Computer Vision (ECCV)*, 2020.

[15] A. Aubret and L. Matignon and S. Hassas, "A survey on intrinsic motivation in reinforcement learning," in *arXiv:1908.06976*, 2019.

[16] N. Kazantzis and C. Kravaris, "Nonlinear observer design using lyapunov's auxiliary theorem," *Systems & Control Letters*, 1998.

[17] V. Andrieu and L. Praly, "On the Existence of a Kazantzis-Kravaris/Luenberger Observer," *SIAM Journal on Control and Optimization*, 2006.

[18] L. Marconi, L. Praly, and A. Isidori, "Output Stabilization via Nonlinear Luenberger Observers," *SIAM Journal on Control and Optimization*, 2007.

[19] A. Isidori, L. Praly, and L. Marconi, "About the Existence of Locally Lipschitz Output Feedback Stabilizers for Nonlinear Systems," *SIAM Journal on Control and Optimization*, 2010.

[20] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, 1986.

[21] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, B. Fethi, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder decoder for statistical machine," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014.

[22] S. L. B. Bethany Lusch, J. Nathan Kutz, "Deep learning for universal linear embeddings of nonlinear dynamics," *Nature Communications*, 2018.

[23] C. W. Rowley, I. Mezic, S. Bagheri, P. Schlatter, and D. S. Henningson, "Spectral analysis of nonlinear flows," *Journal of Fluid Mechanics*, 2009.

[24] B. Francis and W. Wonham, "The internal model principle of control theory," *Automatica*, 1976.

[25] L. d. C. Ramos, F. Di Meglio, V. Morgenthaler, L. F. F. da Silva, and P. Bernard, "Numerical design of luenberger observers for nonlinear systems," in *2020 59th IEEE Conference on Decision and Control (CDC)*, 2020.

[26] W. Lohmiller and J.-J. E. Slotine, "On contraction analysis for nonlinear systems," *Automatica*, 1998.

[27] V. Andrieu, "Convergence speed of nonlinear luenberger observers," *SIAM Journal on Control and Optimization*, 2014.

[28] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, vol. 4, pp. 251–257, 1991.

[29] Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang, "The Expressive Power of Neural Networks: A View from the Width," in *NeurIPS*, 2017.

[30] P. L. Bartlett, D. J. Foster, and M. J. Telgarsky, "Spectrally-normalized margin bounds for neural networks," in *Advances in Neural Information Processing Systems*, 2017.

[31] K. Scaman and A. Virmaux, "Lipschitz regularity of deep neural networks: Analysis and efficient estimation," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018.

[32] B. van der Pol Jun. D.Sc, "On "relaxation-oscillations"," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 1926.

[33] E. N. Lorenz, "Deterministic nonperiodic flow," *Journal of Atmospheric Sciences*, 1963.

[34] V. Volterra and M. Brelot, *Leçons sur la théorie mathématique de la lutte pour la vie*. Gauthier-Villars et cie., 1931.

[35] B. R. Noack, K. Afanisiev, M. Morzynski, G. Tadmor, and F. Thiele, "A hierarchy of low-dimensional models for the transient and post-transient cylinder wake," *Journal of Fluid Mechanics*, 2003.

[36] F. Baradel, N. Neverova, J. Mille, G. Mori, and C. Wolf, "Cophy: Counterfactual learning of physical dynamics," in *ICLR*, 2020.

[37] B. C. Csáji *et al.*, "Approximation with artificial neural networks," *Faculty of Sciences, Etvs Lornd University, Hungary*, 2001.

[38] L. Valiant, "A theory of the learnable," in *Communications of the ACM*, vol. 27(11), 1984.