Variational Deep Learning for the Identification and Reconstruction of Chaotic and Stochastic Dynamical Systems from Noisy and Partial Observations

Duong Nguyen, Said Ouala, Lucas Drumetz, Ronan Fablet

Variational Deep Learning for the Identification and Reconstruction of Chaotic and Stochastic Dynamical Systems from Noisy and Partial Observations

Duong Nguyen, *Member, IEEE,* Said Ouala, *Member, IEEE,*
Lucas Drumetz, *Member, IEEE,* and Ronan Fablet, *Member, IEEE.*

***Abstract*—The data-driven recovery of the unknown governing equations of dynamical systems has recently received an increasing interest. However, the identification of the governing equations remains challenging when dealing with noisy and partial observations. Here, we address this challenge and investigate variational deep learning schemes. Within the proposed framework, we jointly learn an inference model to reconstruct the true states of the system from series of noisy and partial data and the governing equations of these states. In doing so, this framework bridges classical data assimilation and state-of-the-art machine learning techniques and we show that it generalizes state-of-the-art methods. Importantly, both the inference model and the governing equations embed stochastic components to account for stochastic variabilities, model errors and reconstruction uncertainties. Various experiments on chaotic and stochastic dynamical systems support the relevance of our scheme w.r.t. state-of-the-art approaches.**

***Index Terms*—dynamical systems identification, variational inference, data assimilation, neural networks.**

I. Introduction

The identification of the governing equations of dynamical processes, usually stated as Ordinary Dynamical Equations (ODE) or Partial Differential Equations (PDE), is critical in many disciplines. For example, in geosciences, it provides the basis for the simulation of climate dynamics, short-term and medium-range weather forecast, short-term prediction of ocean and atmosphere dynamics, etc. In aerodynamics or in fluid dynamics, it is at the core of the design of aircrafts and control systems, of the optimization of energy consumption, etc.

Classically, the derivations of governing equations are based on some prior knowledge of the intrinsic nature of the system [1]–[4]. The derived models can then be combined with the measurements (observations) to reduce errors, both in the model and in the measurements. This approach forms the discipline of Data Assimilation (DA). However, in many cases, the underlying dynamics of the system are unknown or only partially known, while a large number of observations are available. This has motivated the development of learning-based approaches [5], where one aims at identifying the governing equations of a process from time series of measurements. Recently, the ever increasing availability of data thanks to developments in sensor technologies, together with advances in Machine Learning (ML), has made this issue a hot topic. Numerous methods have successfully captured the hidden dynamics of systems under ideal conditions, *i.e.* noiseless and high sampling frequency using a variety of data-driven schemes, including analog methods [6], sparse regression schemes [7], reservoir computing [8], [9] and neural approaches [10]–[14]. However, real life data often involve noisy and irregularly-sampled data as for instance encountered in the the monitoring of ocean and atmosphere dynamics from satellite-derived observation data [15]–[17]. In such situations, the above-mentioned approaches are most likely to fail to uncover the unknown governing equations.

To address this challenge, we need to jointly solve the reconstruction of the hidden dynamics and the identification of the governing equations. This may be stated within a data assimilation framework [18] using state-of-the-art assimilation schemes such as the Ensemble Kalman Smoother (EnKS) [19]. Deep learning approaches are also particularly appealing to benefit from their flexibility and computational efficiency. Here, we investigate a variational deep learning framework. More precisely, we state the considered issue as a variational inference problem with an unknown transition distribution associated with the underlying dynamical model. The proposed method generalizes learning-based schemes such as [10], [20]–[23] and also explicitly relates to data assimilation formulations. Importantly, it can account for errors and uncertainties both within the dynamical prior and the inference model. Overall, our key contributions are:

- a general deep learning framework for learning dynamical systems from noisy and partial observations using variational inference and random-n-step-ahead forecasting, which can be considered as two complementary regularization strategies to improve the data-driven identification of governing equations of dynamical systems;
- insights on the reason why many existing methods for learning dynamical systems do not work when the available data are not perfect, *i.e.* noisy and/or irregularly-sampled;
- numerical experiments with chaotic systems which support the relevance of the proposed framework to improve the learning of governing equations from noisy and partial observation datasets compared to state-of-the-art schemes;

Duong Nguyen, Said Ouala, Lucas Drumetz and Ronan Fablet are with IMT Atlantique, Lab-STICC, 29238 Brest, France (email: {van.nguyen1, said.ouala, lucas.drumetz, ronan.fablet}@imt-atlantique.fr)

This work was supported by Labex Cominlabs (grant SEACS), CNES (grant OSTST-MANATEE), Microsoft (AI EU Ocean awards), ANR Projects Melody and OceaniX and GENCI-IDRIS (Grant 2020-101030).

- numerical experiments which demonstrate that our method can capture the characteristics of dynamical systems where the stochastic factors are significant.

The paper is organized as follows. In Section II, we formulate the problem of learning non-linear dynamical systems. We review state-of-the-art methods and analyze their drawbacks in Section III. Section IV presents the details of the proposed framework, followed by the experiments and results in Section V. We close the paper with conclusions and perspectives for future work in Section VI.

II. Problem formulation

Consider a dynamical system, described by a Ordinary Differential Equation (ODE) as follows:

$$\frac{\mathrm{d}\mathbf{z}_t}{\mathrm{d}t} = f\big(\mathbf{z}_t\big) \tag{1}$$

where $\mathbf{z}_t \in \mathbb{R}^{d_z}$ is a geometrical point, called the *state* of the system (d_z is the dimension of $\mathbf{z}_t$), $f : \mathbb{R}^{d_z} \longrightarrow \mathbb{R}^{d_z}$ is a *deterministic* function, called the *dynamical model*.

We aim at learning the dynamics of this system from some observation dataset, that is to say identifying the governing equations f, given a series of observations $\mathbf{x}_{t_k}$:

$$\mathbf{x}_{t_k} = \Phi_{t_k}\big(\mathcal{H}\big(\mathbf{z}_{t_k}\big) + \boldsymbol{\varepsilon}_t\big) \tag{2}$$

where $\mathcal{H} : \mathbb{R}^{d_z} \longrightarrow \mathbb{R}^{d_x}$ is the observation operator, usually known (d_x is the dimension of $\mathbf{x}_{t_k}$), $\boldsymbol{\varepsilon}_t \in \mathbb{R}^{d_x}$ is a zero-mean additive noise and $\{t_k\}_k$ refers to the time sampling, typically regular such that $t_k = t_0 + k.\delta$ with respect to a fine time resolution δ and a starting point t_0. We introduce a masking operator Φ_{t_k} to account for the fact that observation $\mathbf{x}_{t_k}$ may not be available at all time steps t_k ($\Phi_{t_k,j} = 0$ if the j^{th} variable of $\mathbf{x}_{t_k}$ is missing). For the sake of simplicity, from now on in this paper, we use the notation $\mathbf{x}_k$ for $\mathbf{x}_{t_k}$ and $\mathbf{x}_{k+n}$ for $\mathbf{x}_{t_{k+n.\delta}}$.

From Eq.(1) and (2), we derive a state-space formulation:

$$\mathbf{z}_{k+n} = \mathcal{F}^n\big(\mathbf{z}_k\big) + \boldsymbol{\omega}_{k+n} \tag{3}$$

$$\mathbf{x}_k = \Phi_k\big(\mathcal{H}\big(\mathbf{z}_k\big) + \boldsymbol{\varepsilon}_k\big) \tag{4}$$

where $\mathbf{z}_{k+n}$ results from an integration of operator f from state $\mathbf{z}_t$: $\mathcal{F}^n\big(\mathbf{z}_k\big) = \mathbf{z}_k + \int_k^{k+n} f\big(\mathbf{u}_t\big)\mathrm{d}\mathbf{u}_t$. $\boldsymbol{\omega}_k \in \mathbb{R}^{d_z}$ is a zero-mean noise process, called the *model error*. $\boldsymbol{\omega}_k$ may come form the neglected physics, numerical approximations and/or errors of the modeling. $\boldsymbol{\varepsilon}_k$ is the *observation error* (or *measurement error*). Note that f is continuous, the discretisation only happens because we want to calculate the integral F over the interval $[t_k, t_{k+n}]$.

Within this formulation, maximising the log likelihood $\ln p(\mathbf{x}_{0:T})$ results in finding the governing equations f from a starting point $\mathbf{z}_k$ to a time $k+n$ where we have observation $\mathbf{x}_n$.

III. Related work

The identification of dynamical systems has attracted attention for several decades and closely relates to data assimilation (DA) for applications to geoscience. Proposed approaches typically consider some parametric model for operator $\mathcal{F}$, for example, a linear function in [24] or Radial Basis Functions (RBFs) in [25]. While data assimilation mostly focuses on the reconstruction of the hidden dynamics given some observation series, a number of studies have investigated the situation where the dynamical prior is unknown. They typically learn the unknown parameters of the model using an iterative Expectation-Maximization (EM) procedure. The E-step involves a DA scheme (*e.g.* the Kalman filter [26], the Extended Kalman filter [27], the Ensemble Kalman filter [28], the particle filter [29], etc.) to reconstruct the true states $\{\mathbf{z}_k\}$ from observations $\{\mathbf{x}_k\}$, whereas the M-step retrieves the parameters of $\mathcal{F}$ best describing the reconstructed state dynamics. Those methods address the fact that the observations are not ideal. They may also account for model uncertainties ($\boldsymbol{\omega}_k$ in Eq. (3)). However, since they rely on analytic solutions, the choices of the candidates for $\mathcal{F}$ are generally limited. For a comprehensive introduction as well as an analysis of the limitations of those methods, the reader is referred to [30].

Recently, the domain of dynamical systems identification has received a new wave of contributions. Advances in machine learning opens new means for learning the unknown dynamics of the systems. In line of work, one of the pioneering contribution is the Sparse Identification of Nonlinear Dynamics (SINDy) presented in [7]. SINDy assumes that the governing equation of a dynamical model can be decomposed into only a few basic functions such as polynomial functions, trigonometric functions, exponential functions, etc. The method creates a dictionary of such possible functions and uses sparse regression to retrieve the corresponding weighting coefficients of each basic function. Under ideal conditions, SINDy can find the exact solution of Eq. (1). The key advantage of SINDy is its solutions are interpretable, *i.e.* the parametric form of the governing equations can be recovered. Another advantage is that the solution comprises only a few terms, which improves the generalization properties of the learnt models. However, SINDy requires the time derivative $\frac{d\mathbf{x}_t}{dt}$, which might be highly corrupted by noise for noisy and partial observation datasets, to be observed. Therefore, SINDy's performance may be strongly affected by noisy data. Besides, it requires some prior knowledge of the considered system to create a suitable library of the basic functions.

Analog methods [31]–[33], including the Analog Data Assimilation (AnDA) presented in [6], propose a non-parametric approach for data assimilation. AnDA implicitly learns Eqs. (3) and (4) by remembering every seen pairs $\{state, successor\} = \{\mathbf{x}_t, \mathbf{x}_{k+1}\}$ and storing them in a catalog. To predict the evolution of a new query point $\mathbf{x}_t$, AnDA looks for k similar states in the catalog, the prediction is then a weighted combination of the corresponding successors of these states. The performance of this method heavily depends on the quality of the catalog. If the catalog contains enough data and the data are clean, AnDA provides a good and straightforward solution for data assimilation. However, since AnDA relies on a k-Nearest Neighbor (k-NN) approach, it may be strongly affected by noisy data especially when considering high-dimensional systems.

A number of neural-network-based (NN-based) methods have been introduced recently. These methods leverage deep

neural networks as universal function approximators. They vary from direct applications of standard NN architectures, such as LSTMs in [34], ResNets in [12], etc. to some more sophisticated designs, dedicated for dynamical systems and often referred to as Neural ODE schemes [10], [11], [35], [36]. The reservoir computing, whose idea is derived from Recurrent Neural Networks (RNNs), used in [8] and [9] can also be regarded as a NN-based model. As illustrated in [10], [11], [35], [36], through the combination of a parametrization for differential operator f and some predefined integration schemes (*e.g.*, explicit Runge-Kutta 4 scheme (RK4) in [10], black-box ode solvers in [35], [36]), the Neural ODE schemes provides significantly better forecasting performance, especially when dealing with chaotic dynamics. Powered by deep learning, these methods can successfully capture the dynamics of the system under ideal conditions (noise-free and regularly sampled with high frequency). However, they have the following limits: i) the network requires regularly-sampled data[1] and ii) when dealing with noisy observations, no regularization techniques have been proved effective to prevent overfitting in learning the hidden dynamics.

Overall, the above-mentioned learning-based methods do not apply or fail when the observations are corrupted by noise and irregularly-sampled. Their learning step is stated as the minimization of a short-term prediction error of the observations:

$$loss = \sum_t g(||\mathbf{x}_{k+n}^{pred} - \mathbf{x}_{k+n}||_2) \tag{5}$$

where $\mathbf{x}_{k+n}^{pred} = \mathcal{F}^n(\mathbf{x}_k)$ is the predicted observation at $k+n$ given the current observation $\mathbf{x}_k$, $||.||_2$ denotes the L2 norm, g is a function of $||\mathbf{x}_{k+n}^{pred} - \mathbf{x}_{k+n}||_2$. As shown in Fig. 1, with this family of cost functions, the model tends to overfit the observations (the blue curve or the green and yellow curves) instead of learning the true dynamics of the system (the red curve). Another reason why the these methods fail is because they violate the Markovian property of the system. Note that the process of the true states $\mathbf{z}_{0:T}$ of the system is Markovian (*i.e.*, given $\mathbf{z}_t$, $\mathbf{z}_{k+1}$ does not depend on $\mathbf{z}_{0:k-1}$). However, when the data are damaged by noise, the process of the observations $\mathbf{x}_{0:T}$ is not Markovian. Given $\mathbf{x}_t$, we still need the information contained in $\mathbf{x}_{0:k-1}$ to predict $\mathbf{x}_{k+1}$. For this reason, applying Markovian architectures like SINDy, AnDA, DenseNet, BiNN, etc. directly on the observations $\mathbf{x}_{0:T}$ would not succeed. Models with memory like LSTMs may capture the non-Markovian dynamics in the training phase, however, in the simulation phase, they still need the memory, which implies that the learnt dynamics do not have the Markovian property of the true dynamics of the system.

In this paper, we consider a variational deep framework which derives from a variational inference for state-space formulation (Eqs. 3, 4). This framework accounts for uncertainty components in the dynamical prior as well as in the observation model. Similarly to DA schemes [18], [20]–[23], it jointly solves for the reconstruction of the hidden dynamics and the identification of the governing equations. Importantly, it benefits from the computational efficiency and modeling flexibility of deep learning frameworks for the specification of the dynamical prior and the inference model as well as for the use of a stochastic regularization during the training phase through a randomized n-step-ahead prediction loss. The proposed framework generalizes our recent works presented in [20] and [21] and similar works, which have been developed concurrently in [18], [22] and [23]. As will be detailed in the next section, [18], [22] and [23] are specific instances of the proposed framework with some specific settings, such as the model error covariance matrix is constant (we relax this hypothesis), the inference scheme is the Ensemble Kalman Smoother (we exploit both strategies: Ensemble Kalman Smoother and NN-based schemes), the optimization technique is based on EM (we exploit both EM and gradient-based techniques).

[1]Latent ODE ([36] can apply for data sampled irregularly in time, however, data my be sampled irregularly in space also.

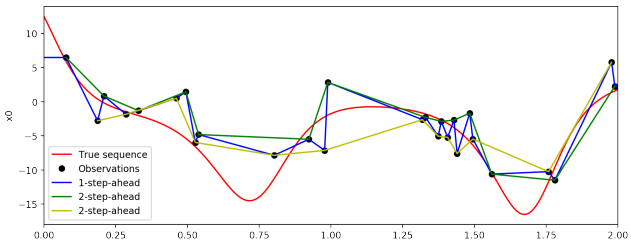

Fig. 1: Problems of learning dynamical systems from imperfect data. This figure plots the first component of the Lorenz-63 system, when the observation operator is the identity matrix. The observation is noisy, partial and irregular. If the learning algorithm is applied directly on the observations (black dots), which are noisy and irregularly sampled, and linear interpolation is used to create regularly-sampled data, the dynamics seen by the network are the blue curve (for 1-step-ahead forecasting models) or the green and the yellow curve (for 2-step-ahead forecasting models, these two curves correspond to two possible starting points of the sequence) instead of the true dynamics (the red curve).

IV. Proposed framework

In this section, we detail the proposed variational deep learning framework for the data-driven identification of the governing equations of dynamical systems from noisy and irregularly-sampled observations. We first present the general framework based on variational inference. We then introduce the considered NN-based parameterizations for the dynamical prior and the inference model along with the implemented learning scheme. We further discuss how the proposed framework relates to previous work.

A. Variational inference for learning dynamical systems

Given a series of observations $\mathbf{x}_{0:T} = \{\mathbf{x}_0, .., \mathbf{x}_k\}$, instead of looking for a model $\mathcal{F}$ that minimizes a loss function in a family of short-term prediction error functions as in Eq. (5), we aim to learn operator $\mathcal{F}$ such that it maximizes the log likelihood $\ln p(\mathbf{x}_{0:T})$ of the observed data. We assume

that $\mathbf{x}_{0:T}$ are noisy and/or irregularly-sampled observations of the true states $\mathbf{z}_{0:T}$, like in Eqs. (3) and (4). We can derive the log-likelihood $\ln p(\mathbf{x}_{0:T})$ from the marginalization of $\ln p(\mathbf{x}_{0:T}, \mathbf{z}_{0:T})$ over $\mathbf{z}_{0:T}$:

$$\ln p(\mathbf{x}_{0:T}) = \ln \int p(\mathbf{x}_{0:T}, \mathbf{z}_{0:T}) d\mathbf{z}_{0:T} \tag{6}$$

With the exception of some simple cases, the computation in Eq. (6) is intractable because the posterior distribution $p(\mathbf{z}_{0:T}|\mathbf{x}_{0:T})$ is intractable [37]. To address this issue, Variational Inference (VI) proposes approximating $p(\mathbf{z}_{0:T}|\mathbf{x}_{0:T})$ by a distribution q which maximizes the Evidence Lower BOund (ELBO)[2]:

$$\mathcal{L}(\mathbf{x}_{0:T}, p, q) = \int q(\mathbf{z}_{0:T}|\mathbf{x}_{0:T}) \ln \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}, \mathbf{z}_{0:T})}{q(\mathbf{z}_{0:T}|\mathbf{x}_{0:T})} d\mathbf{z}_{0:T} \tag{7}$$

Based on the state-space formulation in Eqs. (3) and (4), we consider the following parameterization for the joint likelihood $p(\mathbf{x}_{0:T}, \mathbf{z}_{0:T})$:

$$p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}, \mathbf{z}_{0:T}) = p_{\boldsymbol{\theta}}(\mathbf{z}_{0:T}) p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}|\mathbf{z}_{0:T}) \tag{8}$$

$$p_{\boldsymbol{\theta}}(\mathbf{z}_{0:T}) = p_{\boldsymbol{\theta}}(\mathbf{z}_0) \prod_{t=1}^{n-1} p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{z}_{k-1}) \prod_{t=n}^{T} p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{z}_{k-n}) \tag{9}$$

$$p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}|\mathbf{z}_{0:T}) = \prod_{k=0}^{T} p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{z}_k) \tag{10}$$

$$q_{\boldsymbol{\phi}}(\mathbf{z}_{0:T}|\mathbf{x}_{0:T}) = \prod_{t=0}^{T} q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{z}_k^f, \mathbf{x}_{0:T}) \tag{11}$$

with $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ are the sets of parameters of p and q, respectively; $\mathbf{z}_k^f$ is the state forecast by $\mathcal{F}$.

The distributions in Eqs. (9), (10) and (11), are respectively the classic distributions of a state-space formulation: 1) the transition (or the dynamic, or the prior) distribution one-step ahead $p_{\boldsymbol{\theta}}(\mathbf{z}_{k+1}|\mathbf{z}_k)$ or n-step ahead $p_{\boldsymbol{\theta}}(\mathbf{z}_{k+n}|\mathbf{z}_k)$; 2) the emission (or the observation) distribution $p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{z}_k)$; and 3) the inference (or the posterior) distribution $q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{z}_k^f, \mathbf{x}_{0:T})$. To better constrain the time consistency of the learnt dynamics, the considered dynamical prior embeds a n-step-ahead forecasting model. Given an initialization $\mathbf{z}_0$, it first applies a one-step-ahead prior to propagate the initial state to the first n time steps. The application of the n-step-ahead prior follows to derive the joint distribution over the entire time range $\{0, .., T\}$. This n-step-ahead prior is regarded as a mean to further regularize the time consistency of the learnt dynamical model.

By explicitly separating the transition, the inference and the generative processes, the proposed framework is fully consistent with the underlying state-space formulation and the associated Markovian properties. Especially, the prior $p_{\boldsymbol{\theta}}(\mathbf{z}_{k+n}|\mathbf{z}_k)$ will embed a Markovian architecture; by contrast, the posterior $q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{z}_k^f, \mathbf{x}_{0:T})$ shall capture the non-Markovian characteristics of the observed data. Given the learnt model, the generation of simulated dynamics only relies on the dynamical prior $p_{\boldsymbol{\theta}}(\mathbf{z}_{k+1}|\mathbf{z}_k)$ to simulate state sequences, which conform to the Markovian property. Overall, for a given observation dataset, the learning stage comes to maximize Eq. (7) w.r.t. both $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$, which comprise all the parameters of inference and the forward model, *i.e.* the parameters of $\mathcal{F}$, $\mathcal{H}$, $\boldsymbol{\omega}_{k+n}$ and $\boldsymbol{\epsilon}_k$.

So far we have presented the general form of the variational inference framework for learning dynamical systems from noisy and potentially partial observations. In the following sub-sections, we will analyse some specific instances of the proposed framework and provide insights into the implicit hypotheses behind methods in the literature.

B. Parametrisation of the forward model $p_{\boldsymbol{\theta}}$

Model $p_{\boldsymbol{\theta}}$ involves two sets of parameters: (i) $\boldsymbol{\theta}_z$—the parameters of the transition distributions $p_{\boldsymbol{\theta}}(\mathbf{z}_{k+n}|\mathbf{z}_k)$ and (ii) $\boldsymbol{\theta}_x$—the parameters of the emission distribution $p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{z}_k)$.

Regarding the later, we assume the observation noise to be a white noise process with a multivariate covariance $\mathbf{R}$ similarly to [22] and [23] such that $p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{z}_k)$ is a conditional multivariate normal distribution:

$$p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{z}_k) = \mathcal{N}(\mathcal{H}(\mathbf{z}_k), \mathbf{R}) \tag{12}$$

We may consider different experimental settings: especially, a known observation operator $\mathcal{H}$ and an unknown covariance $\mathbf{R}$ as well as unkown observation operator $\mathcal{H}$ and covariance $\mathbf{R}$.

Regarding the n-step-ahead dynamical prior $p_{\boldsymbol{\theta}}(\mathbf{z}_{k+n}|\mathbf{z}_k)$ (including $n = 1$), we consider a conditional Gaussian distribution where the mean path is driven by the the governing equation $\mathcal{F}$: $\mathbf{z}_{k+n} = \mathcal{F}^n(\mathbf{z}_k)$ and we denote by $\mathbf{Q}_{k+n}$ the covariance matrix (usually called the *model error covariance* in DA) representing the dispersion around the mean path [28]. Any state-of-the-art architecture for learning dynamical systems can be used to model $\mathcal{F}$. Here, we consider NN-based methods associated with explicit integration schemes. To account for second-order polynomial model, as proposed in [10], we consider a bilinear fully-connected architecture to model f in Eq. (1) and an NN implementation of the RK4 integration scheme to derive the flow operator Eq. (3). Regarding the covariance dynamics, the covariance matrix $\mathbf{Q}_{k+n}$ is approximated by a diagonal matrix $\mathbf{Id}_k^f$, with $\mathbf{I}$ is the identity matrix and $\mathbf{d}_k^f$ is the output of a MultiLayer Perceptron (MLP):

$$\mathbf{d}_{k+n}^f = MLP^{var\_dyn}(\mathbf{z}_k, \mathcal{F}^n(\mathbf{z}_k)) \tag{13}$$

C. Parameterization of the inference model $q_{\boldsymbol{\phi}}$

There is no restriction the parameterization of posterior $q_{\boldsymbol{\phi}}$. However, the parameterization clearly affects the performance of the overall optimisation. Here, we investigate two strategies for $q_{\boldsymbol{\phi}}$: 1) an Ensemble Kalman Smoother (EnKS) [19] and 2) an LSTM Variational Auto Encoder (LSTM-VAE). The former one is a classic DA scheme that is widely used in many domains in which dynamical systems play an important role, for example in Geosciences [40]. We use the implementation presented in [19]. The latter is a modern NN architecture, which has been proven effective for modeling stochastic sequential data [41] [42]. The backbone of LSTM-VAE is a

[2]For the sake of simplicity, here we present only ELBO, however, one can use ELBO's variants such as IWAE [38] or FIVO [39] instead.

bidirectional LSTM which captures the long-term correlations in data. Specifically, we parameterize the inference scheme as follows. The forward LSTM is given by:

$$\mathbf{h}_k^f = lstm(\mathbf{h}_{k-1}^f, MLP^{enc}(\mathbf{x}_{k-1}^f)) \tag{14}$$

and the backward LSTM by:

$$\mathbf{h}_k^b = lstm(\mathbf{h}_{k+1}^b, \mathbf{h}_k^f, MLP^{enc}(\mathbf{x}_k^f)) \tag{15}$$

where $\mathbf{h}_k^f$, $\mathbf{h}_k^b$ are the hidden states of the forward and the backward LSTM, respectively; $lstm$ is the recurrence formula of LSTM [43]; MLP^{enc} is an encoder parameterized by an MLP. We parameterize the posterior $q_{\boldsymbol{\phi}}$ by a conditional Gaussian distribution with mean $\boldsymbol{\mu}_k^q$ and a diagonal covariance matrix $\mathbf{Id}_k^q$:

$$q_{\boldsymbol{\phi}}(\mathbf{z}_k) = \mathcal{N}(\boldsymbol{\mu}_k^q, \mathbf{d}_k^q \mathbf{I}) \tag{16}$$

$$\boldsymbol{\mu}_k^q, \mathbf{d}_k^q = MLP^{dec}(\mathcal{F}^n(\mathbf{z}_{k-n}), \mathbf{h}_k^f, \mathbf{h}_k^b) \tag{17}$$

with MLP^{dec} is a decoder parameterized by an MLP. Note that in Eq. (17), $q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{x}_{0:T})$ depends on $\mathcal{F}^n(\mathbf{z}_{k-n})$. This idea is inspired by DA, where $\mathcal{F}^n(\mathbf{z}_{k-n})$ is analogous to the forecasting step and $q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{x}_{0:T})$ to the analysis step, which depends on the forecasting step. The whole model, called Data-Assimilation-based ODE Network (DAODEN) is illustrated in Fig. 2. To our knowledge, the latter is the first end-to-end RNN-based stochastic model introduced for the identification of in the dynamical systems from noisy and partial observations. In this respect, the model used in [34] is a purely deterministic RNN-based network. However, similar architectures have been used in Natural Language Processing (NLP) such as the Variational Recurrent Neural Network presented in [41], the Sequential Recurrent Neural Network presented in [42]. Fig. 2 shows how DAODEN differs from those works. The main difference is that the transition $\mathbf{z}_k \rightarrow \mathbf{z}_{k+1}$ is independent of observation $\mathbf{x}_k$ (*i.e.* the dynamic is autonomous). Besides, the emission $\mathbf{z}_k \rightarrow \mathbf{x}_k$ is also independent of the historical state $\mathbf{z}_0, .., \mathbf{z}_{k-1}$. These differences relate to domain-related priors. In dynamical systems' theory and associated application domains such as geoscience, the underlying dynamics follow physical principles. Therefore, they are autonomous and are not affected by the measurements (the observations). As a consequence, $\mathbf{z}_{k+1}$ does not depend of $\mathbf{x}_{1:k-1}$ conditionally to $\mathbf{z}_k$. At a given time t, observation $\mathbf{x}_k$ is a measurement of state $\mathbf{z}_k$ of the system, this measurement does not depend on any other state $\mathbf{z}_{t' \neq t}$, *i.e.* given $\mathbf{z}_k$, $\mathbf{x}_k$ and $\mathbf{z}_{t'}$ are independent with any $t' \neq t$. For this reason, architectures used in NLP like VRNN, SRNN can not apply for dynamical systems identification.

D. Objective function

Following a variational Bayesian setting, the learning phase comes to minimize a loss given the opposite of the ELBO:

$$loss_{ELBO} = -\mathcal{L}(\mathbf{x}_{0:T}, p_{\boldsymbol{\theta}}, q) \tag{18}$$

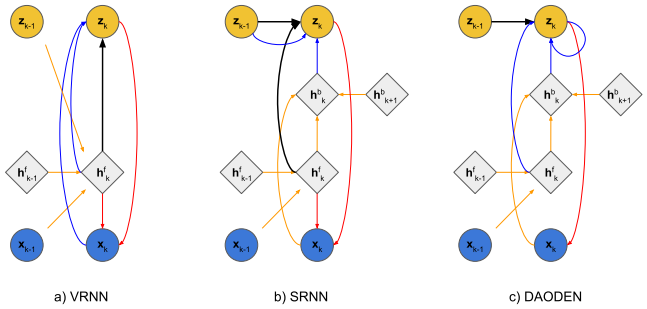

Fig. 2: Architecture of VRNN, SRNN and DAODEN when $n = 1$. We denote as $\mathbf{x}_k$ the observations, $\mathbf{z}_k$ the system's states, $\mathbf{h}_k^f$ the latent state of the forward LSTM and $\mathbf{h}_k^b$ the latent state of the backward LSTM. The black, red, blue and orange arrows denote respectively the transition of the system's states, the emission of the observations, the inference of the system's states and recurrence of the LSTMs, respectively. In VRNN (a) and SRNN (b), the dynamic $\mathbf{z}_k \rightarrow \mathbf{z}_{k+1}$ is not independent of the observation $\mathbf{x}_k$. The generation of the observation is also entangled with the recurrence of the LSTMs.

Instead of solving Eq. (7), one can solve its Maximum A Posteriori (MAP) solution by restricting $q_{\boldsymbol{\phi}}$ to Dirac distributions:

$$\mathcal{L}_{MAP} = \sum_{k=0}^{k} \ln p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{z}_k^*) + \ln p_{\boldsymbol{\theta}}(\mathbf{z}_0^*) + \sum_{t=0}^{n-2} \ln p_{\boldsymbol{\theta}}(\mathbf{z}_{k+1}^*|\mathbf{z}_k^*) + \sum_{t=0}^{k-n} \ln p_{\boldsymbol{\theta}}(\mathbf{z}_{k+n}^*|\mathbf{z}_k^*) \tag{19}$$

with $\mathbf{z}_k^* = \mathbb{E}\left[q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{x}_{0:T})\right]$ if $q_{\boldsymbol{\phi}}$ is parameterized by EnKS and $\mathbf{z}_k^* = q(\mathbf{z}_k|\mathbf{x}_{0:T}) = \delta(\mathbf{z}_k|\mathbf{x}_{0:T})$ if $q_{\boldsymbol{\phi}}$ is parameterized by a neural network. To do so, we remove the covariance part in Eq. (17) (the LSTM-VAE becomes an LSTM Auto Encoder (LSTM-AE)) :

$$\mathbf{z}_k^* = \boldsymbol{\mu}_k^q = dec(\mathcal{F}^1(\mathbf{z}_{k-1}), \mathbf{h}_k^f, \mathbf{h}_k^b) \tag{20}$$

The MAP loss function, which relates to the weak-constraint 4D-Var in DA [44], is given by:

$$loss_{MAP} = -\mathcal{L}_{MAP}(\mathbf{x}_{0:T}, p_{\boldsymbol{\theta}}, q) \tag{21}$$

This is the objective function used in [18], [22] and [23]. However, these models suppose that $\mathbf{Q}_k$ is time invariant, *i.e.* $\mathbf{Q}_k = \mathbf{Q}$.

One may further assume that the covariance matrices of the transition distribution $p_{\boldsymbol{\theta}}(\mathbf{z}_{k+n}^*|\mathbf{z}_k^*)$ and the covariance matrices of the observation distribution $p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{z}_k^*)$ are diagonal and

constant, both in time and in space, Eq. (19) then becomes[3]:

$$\mathcal{L}_{determ} = -\lambda \sum_{t=0}^{k} ||\boldsymbol{\phi}_k(\mathcal{H}(\mathbf{z}_k^*)) - \mathbf{x}_k||_2^2 - \sum_{t=0}^{n-2} ||\mathcal{F}^1(\mathbf{z}_k^*) - \mathbf{z}_{k+1}^*||_2^2 - \sum_{t=0}^{k-n} ||\mathcal{F}^n(\mathbf{z}_k^*) - \mathbf{z}_{k+1}^*||_2^2 \quad (22)$$

The associated loss function is given by:

$$loss_{determ} = -\mathcal{L}_{determ}(\mathbf{x}_{0:T}, p_{\boldsymbol{\theta}}, q) \quad (23)$$

which is the objective function used in [20] and [21]. We may note that if $\mathbf{x}_k = \mathbf{z}_k$, (23) becomes the shorkt-terms prediction error widely used in the literature [8]–[10], [12]. In other words, [8]–[10], [12] implicitly suppose that the observations are ideal.

E. Optimization strategy

There are two optimization strategies: 1) alternatively optimize $\boldsymbol{\theta}$ then $\boldsymbol{\phi}$ (Expectation-Maximization-like or EM-like) to minimize the loss function or 2) jointly optimize the loss function over $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$.

For models whose posterior $q_{\boldsymbol{\phi}}$ is implemented by an EnKS, since EnKS uses analytic formulas and the NN-based parametrization of $p_{\boldsymbol{\theta}}$ is usually optimized by Gradient Descent (GD) techniques, we consider an alternated EM procedure as optimization strategy for the whole model. In the E-step, the EnKS computes the posterior $q_{\boldsymbol{\phi}}$, represented by an ensemble of states $\mathbf{z}_k^{(i)}$. Given this ensemble of states, the M-step minimizes the loss function over $\boldsymbol{\theta}$ using a stochastic gradient descent.

For DAODENs, we can fully benefit from the resulting end-to-end architecture, as both the forward model $p_{\boldsymbol{\theta}}$ and the posterior $q_{\boldsymbol{\phi}}$ are parameterized by neural networks, to jointly optimize all model parameters using a stochastic gradient descent. This gradient descent may regarded as a particular case of EM where the M-step takes only one single gradient step. For NN-based models, gradient descent strategies usually work better than EM ones [45].

F. Random-n-step-ahead training

Within the considered framework, we noted experimentally that the model may overfit the data, when the number of the forecasting steps is fixed. For example, if the observation operator $\mathcal{H}$ is an identity matrix, a possible overfitting situation is when the inference scheme also becomes an identify operator: $\mathbb{E}\left[q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{x}_{0:T})\right] \rightarrow \mathbf{x}_k$. In such situations, the dynamics seen by the dynamical sub-modules would be the noisy dynamics.

To deal with these overfitting issues, we further exploit the flexibility of the proposed n-step-ahead dynamical prior during the training phase. At each mini-batch iteration in the training phase, we draw a random value of n between 1 and a predefined n-step-ahead_max. We then apply gradient descent with the sampled value of n. The resulting randomized training procedure is detailed in Alg. 1. This randomized procedure is regarded as a regularization procedure to fit a time-consistent dynamical operator $\mathcal{F}$. We noted in previous works that neural ODE schemes may tend not to distinguish well the dynamical operator from the integration scheme [46]. Here, through the randomization of parameter n, we constrain the end-to-end architecture to apply for different prediction horizons, which in turn constrain the identification of dynamical prior f. Asymptotically, the proposed procedure would be similar to a weighted sum of loss (18) computed for different values of n, which have been proposed for the data-driven identification of governing equations in the noise-free case [36].

[3]The derivation of (22) can be found in our previous paper [20].

Algorithm 1: Random-n-step-ahead training.

Result: The set of parameters $\{\boldsymbol{\theta}, \boldsymbol{\phi}\}$ of the learnt model.
Inputs: $\mathbf{x}_{0:T}$, $\mathbf{z}_0$, the initial values of $\{\boldsymbol{\theta}, \boldsymbol{\phi}\}$, n-step-ahead_max, n_iteration_max;
iter = 0;
while *iter* $<$ *n_iteration_max* **do**
  t = 0;
  n-step-ahead = randint(1,n-step-ahead_max);
  while $t < k - n$ **do**
    if $t <$ *n-step-ahead* -2 **then**
      $n = 1$;
    else
      $n =$ n-step-ahead;
    $\mathbf{z}_{k+n}^f = \mathcal{F}^n(\mathbf{z}_k)$;
    $\mathbf{d}_{k+n}^f = MLP^{var\_dyn}(\mathbf{z}_k, \mathcal{F}^n(\mathbf{z}_k))$;
    $p_{\boldsymbol{\theta}}(\mathbf{z}_{k+n}|\mathbf{z}_k) = \mathcal{N}(\mathbf{z}_{k+n}^f, \mathbf{d}_{k+n}^f)$;
    Calculate $q_{\boldsymbol{\phi}}(\mathbf{z}_{k+n}|\mathbf{z}_{k+n}^f, \mathbf{x}_{0:T})$;
    $\mathbf{z}_{k+n} \sim q_{\boldsymbol{\phi}}(\mathbf{z}_{k+n}|\mathbf{x}_{0:T})$;
    $p_{\boldsymbol{\theta}}(\mathbf{x}_{k+n}|\mathbf{z}_{k+n}) = \mathcal{N}(\mathcal{H}(\mathbf{z}_{k+n}), \mathbf{R})$;
  Calculate $loss$;
  Optimize $loss$ w.r.t. $\{\boldsymbol{\theta}, \boldsymbol{\phi}\}$;

G. Initialization by optimization

In this section, we present the initialization technique used for in the experiments in this paper. Although this technique is not compulsory, it helps improve the stability of the training.

To calculate the state of the system at any given time t, we need both the true dynamics and the precise initial condition $\mathbf{z}_0$. If we use DAODEN, we also have to initialize $\mathbf{h}_0^f$ and $\mathbf{h}_{k+1}^b$. The common approach is "wash out" [47], *i.e.* to initialize $\mathbf{h}_0^f$ and $\mathbf{h}_{k+1}^b$ to zeros or random values and run the LSTMs until the effect of the initial values disappears. However, these initialization techniques are not suitable for learning dynamical systems, because during the washout period, the network is not stable, especially when using an explicit integration scheme (here is the RK4). These instabilities may make the training fail. The value of the objective function also varies highly during this period, leading to an unreliable outcome of the final loss.

Sharing a similar idea with [48] and [36], we use a different initialization strategy. We add two auxiliary networks, a Forward Auxiliary Net to provide $\mathbf{h}_0$ and $\mathbf{z}_0$, and an Backward

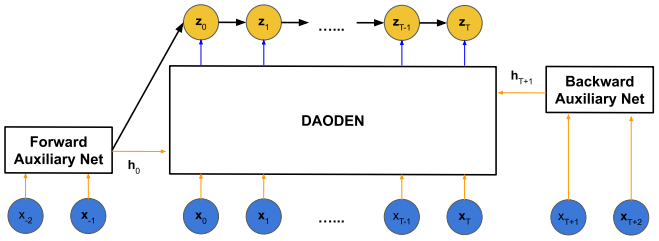

Fig. 3: Initialization by optimization. An auxiliary network is added for the initialization of $\mathbf{x}_1$ and $\mathbf{h}_1$.

Auxiliary Net to provide $\mathbf{h}_{k+1}$ for the main model. Each auxiliary network is an LSTM. We use one segment at the beginning of the sequence and one segment at the end of the sequence for the inputs of these networks.

V. Experiments and results

In this section, we report numerical experiments to evaluate the proposed framework. We include a comparison with respect to state-of-the-art methods. Beyond the application to deterministic dynamics as considered in previous works [8], [9], [12], [18], [20]–[23], we also investigate an application to stochastic dynamics and a reduced-ordre modeling, where observation the operator $\mathcal{H}$ is unknown. As case-study models, we focus on Lorenz-63 and Lorenz-96 dynamics, which provides a benchmarking basis w.r.t. previous works.

A. Benchmarked dynamical models

We report numerical experiments for three chaotic dynamical systems: the Lorenz 63 (L63) systems [1], the Lorenz 96 (L96) system [49] and a stochastic Lorenz 63 (L63s) system [50]. The details of the L63, the L96 and the L63s are presented in the Appendices. Note that these models are chaotic, *i.e.* they are are highly sensitive to initial conditions such that a small difference in a state may lead to significant changes in future. Because of this chaotic nature, applying directly standard deep neural network architectures would not be successful.

We chose the L63 as a benchmarking system because of its famous butterfly attractor. The system involves in 3-dimensional states, making it easy to visualize for a qualitative interpretation. Experiments on the L96 provides means to evaluate how the proposed schemes can scale up to higher-dimensional systems. The last system—the L63s, is to show the benefit of stochastic models over deterministic ones.

For each system, we generated 200 sequences of length of 150 using 200 different initial conditions $\mathbf{z}_0$ with time step $\delta = 0.01$, $\delta = 0.05$ and $\delta = 0.01$ for the L63, L96 and L63s, respectively[4]. In total, each training set of each system comprises 30000 points in total. Those training set are relatively small in comparison with those in [51] (512000 points), [22] (40000 points). Another setting when we generated only 1 long sequence of length of 4000 from 1 initial condition $\mathbf{z}_0$, then split it into smaller segments of length of 150 also gave similar results[5] (not reported in this paper).

For the test set, we generated 50 sequences of length of 150 using 50 different initial conditions $\mathbf{z}_0$ which are not observed in the training set. Let us recall that the true hidden states $\mathbf{z}_{0:T}$ of sequences are never used during the training phase. As in [18], [20]–[23], we first consider an experimental setting where $\mathcal{H}$ is an identity operator, and ε_t a zero-mean Gaussian white noise. We tested several signal-to-noise ratio values $r = \frac{std_\epsilon}{std_{\mathbf{z}}}$.

TABLE I: Implementations of the proposed framework.

Model name	$p_\theta(\mathbf{z}_{t+1}\|\mathbf{z}_t)$	$q_\phi(\mathbf{z}_t\|\mathbf{x}_{0:T})$	Objective function	Optimizer
BINN_EnKS	BiNN	EnKS	Eq. (23)	EM
DAODEN_determ	BiNN	LSTM-AE	Eq. (23)	GD
DAODEN_MAP	BiNN	LSTM-AE	Eq. (21)	GD
DAODEN_full	BiNN	LSTM-VAE	Eq. (18)	GD

B. Baseline schemes

In the reported experiments, we considered different state-of-the-art schemes for benchmarking purposes, namely the Analog Data Assimilation (AnDA) [6], the Sparse Identification of Nonlinear Dynamics (SINDy) [7], the Bilinear Neural Network (BiNN) [10], and the Latent ODE [36]. AnDA and SINDy are unique as their own, while BiNN and Latent ODE represent all the NN-based methods in the literature. As explained earlier in this paper, regardless of the network architecture, as long as the objective function does not take into account the fact that the observations are noisy and potentially partial, the method would not work. BiNN and Latent ODE embed the true solution of the L63 and the L96, under ideal conditions, they should work as good as other NN-based ODE model, such as those in [11], [12], [34], etc. The different between BiNN and Latent ODE is BiNN uses an explicit integration scheme (the RK4), while Latent ODE uses a black-box ODE solver. Latent ODE also uses an additional network to infer the initial condition $\mathbf{z}_0$.

Since VRNN and SRNN are not designed for dynamical systems identification (no autonomous dynamics in the hidden space), we do not consider these architectures in this paper.

C. Instances of the proposed framework

We synthesize in Table. I the different configurations of the proposed framework we implemented in our numerical experiments. We may point out that BiNN_EnKS configuration is similar to [23]. All configurations use a BiNN with a fourth-order Runge-Kutta scheme to parameterize $\mathcal{F}$. As presented above, other architectures can also be used to parameterize $\mathcal{F}$, we choose BiNN to highlight the performance of learning dynamical systems with and without inference schemes (by comparing the performance of BiNN and models following the proposed framework). The parameters of each model are

[4]This is the setting used in [51]

[5]This is the setting used in [6], [9], [22] [23]

presented in the Appendices. We provide the code that can reproduce the result in this paper: https://github.com/CIA-Oceanix/DAODEN. Interested users are highly encouraged to try those models above on different dynamical systems or to replace the dynamical sub-module by different learning methods to see the improvement of its performance on noisy and partial observations.

In this paper, unless specified otherwise the n-step-ahead_max was set to 4 for DAODEN models and 1 for baseline models (1-step-ahead is the default setting in the original papers of those methods). As in [23], for BiNN_EnKS, we suppose that we know $\mathbf{R}$. However, for DAODEN, we do not need the exact value of $\mathbf{R}$, instead we used a fixed value of $\mathbf{R}$ that was from 1 to 2 times larger than the true value of $\mathbf{R}$, the results were similar.

D. Evaluation metrics

We evaluate both the short-term and long-term performance of the learnt models using the following metrics:

- The Root Mean Square Error (RMSE) of the short-term forecast at $t_n = t_0 + n.\delta$:

$$e_n = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(\mathbf{z}_k^{pred} - \mathbf{z}_k^{true})^2} \tag{24}$$

  with $\mathbf{z}_{0+n}^{pred} \triangleq \mathcal{F}^n(\mathbf{z}_0)$ and $\mathbf{z}_0$ is the first state of each sequence in the test set.
- The reconstruction capacity given the observation, denoted as rec:

$$rec = \sqrt{\frac{1}{T}\sum_{t=T}^{n}(\mathbf{z}_t^{*} - \mathbf{z}_t^{true})^2} \tag{25}$$

  with $\mathbf{z}_t^* = \mathbb{E}\left[q_{\boldsymbol{\phi}}(\mathbf{z}_t|\mathbf{x}_{0:T})\right]$.
- The first time (in Lyapunov unit) when the RMSE reach half of the standard deviation of the true system, denoted as $\pi_{0.5}$.
- The capacity to maintain the long-term topology of the system, evaluated via the first Lyapunov exponent λ_1 calculated in a forecasting sequence of length of 20000 time steps, using the method presented in [52]. The true λ_1 of the L63 is 0.91 and the true λ_1 of the L96 is 1.67.

For each metric, we compute the average of the results on 50 sequences in the test set.

As Lorenz dynamics may interpreted in terms of geophysical dynamics, we may also give some physical interpretation to the considered metrics. For example, in geosciences, for experiments on the L96 system with δ=0.05 (correspond to 6 hours in real-world time), e_4 would relate to the precision of a weather forecast model for the next day, $\pi_{0.5}$ indicates how long the forecast is still meaningful, λ_1 indicates whether a model can be used for long-term forecast such as the simulation of climate change and rec indicates the performance of a model to reconstruct the true states of a system when the observations are noisy and partial, such as reconstructing the sea surface condition from satellite images.

E. L63 case-study

In this section we report the results on the L63. We first assess the identification performance on noisy but complete observations (*i.e.* $\boldsymbol{\phi}_t$ is an identity matrix at all time steps) of the L63 system, then extend to cases where the observations are sampled partially, both in time and in space.

Table II shows the performance of the considered model on noisy L63 data. We compare the performance of the 4 proposed models with the baselines', w.r.t the short-term prediction error and the capacity to maintain the long-term topology. All the models based on the proposed framework outperform the baselines by a large margin. This asserts the ability of the proposed framework to deal with noisy observations. In Fig. 4 we show the first component of a L63 sequence in the test set reconstructed by the inference scheme of DAODEN_determ. $q_{\boldsymbol{\phi}}$ is expected to infer a mapping that converts data from the corrupted observation space (black dots) to the true space of the dynamics (the red curve). In this space, data-driven methods can successfully learn the governing equations of the system. The reconstructed sequence is very close to the true sequence.

At first glance, we can see that no model is better than all the others in all 4 criteria. This is aligned with the finding of [53]. BiNN_EnKS and DAODEN_full have very good forecasting score, however, the performance of BiNN_EnKS in reconstructing the true states is not as good as DAODEN models. The dynamics learnt by DAODEN models are also more synchronized to the true dynamics (indicated by $\pi_{0.5}$) than those learnt by BiNN_EnKS. This might suggest that NN-based models (here are LSTM-AE and LSTM-VAE) can be an alternative for classic inference schemes like EnKS, which are among the state-of-the-art methods in data assimilation.

In Fig. 5, we show the attractors generated by the learnt models. AnDA is more suitable for data assimilation than for forecasting. When the noise level is small (r=8.5% and r=16.7%), SINDy and BiNN can still capture the dynamics of the system.When the noise level is significant (r=33.3% and r=66.7%), the attactors generated by SINDy and BiNN are distorted, which indicates that the learnt models are not valid for long-term simulations. On the other hand, all the models of the proposed framework successfully reconstructed the butterfly topology of the attractor, even when the noise level is high.

In real life, we cannot always measure a process regularly with a high sampling frequency. Hence, we address here the problem of learning dynamical systems from not only noisy but also partial observations[6]. Specifically, we consider a case study where the noisy L63 data are sampled irregularly, both in time and in space, with a missing rate of 87.5% (see Fig. 6. For this configuration, baseline schemes do not apply. We report in Table. III and Fig. 7 the performance of the different configurations of the proposed framework.

[6]The term "partial" in this context means the observations are not complete at every time step. Some components of the observations may be missing, in both spatial and temporal dimension; however, all the components of states of the system are seen at least once. For the cases where some components of the systems are never observed, please refer to [13], [54]

TABLE II: Performance of models trained on noisy L63 data. For each index, the best score is marked in **bold** and the second best score is marked in *italic*.

Model		r 8.5%	16.7%	33.3%	66.7%
AnDA	e_4	0.351±0.184	0.777±0.350	1.683±0.724	3.682±1.346
	rec	0.416±0.019	0.941±0.037	2.134±0.076	4.876±0.168
	$\pi_{0.5}$	0.820±0.480	0.380±0.172	0.249±0.174	0.104±0.116
	λ_1	26.517±7.665	27.146±42.927	76.267±28.150	127.047±0.881
SINDy	e_4	0.068±0.052	0.149±0.106	0.311±0.196	0.694±0.441
	$\pi_{0.5}$	0.490±0.261	0.165±0.085	0.077±0.049	0.034±0.034
	λ_1	0.898±0.008	0.840±0.035	0.840±0.035	nan±nan
BiNN	e_4	0.045±0.030	0.119±0.085	0.283±0.185	0.684±0.408
	$\pi_{0.5}$	3.608±1.364	2.053±0.666	0.975±0.488	0.308±0.125
	λ_1	0.900±0.011	0.868±0.010	0.122±0.208	-0.422±0.047
Latent-ODE	e_4	0.051±0.027	0.062±0.034	*0.065±0.042*	*0.213±0.084*
	$\pi_{0.5}$	2.504±1.332	2.336±1.472	2.852±1.352	2.118±1.129
	λ_1	0.892±0.018	0.877±0.018	0.885±0.015	0.675±0.027
BiNN_EnKS	e_4	**0.019±0.016**	**0.024±0.023**	**0.037±0.024**	0.276±0.160
	rec	0.323±0.024	0.431±0.042	0.598±0.093	1.531±0.332
	$\pi_{0.5}$	2.807±1.128	3.004±1.355	2.996±1.641	*2.081±1.214*
	λ_1	0.856±0.031	0.869±0.024	0.826±0.065	0.868±0.014
DAODEN_determ	e_4	0.049±0.031	0.056±0.034	0.077±0.048	0.268±0.201
	rec	0.216±0.125	0.269±0.110	**0.448±0.199**	*0.873±0.216*
	$\pi_{0.5}$	*3.519±1.282*	*3.488±1.327*	**3.470±1.562**	1.803±1.104
	λ_1	0.882±0.036	0.895±0.021	0.911±0.013	0.793±0.021
DAODEN_MAP	e_4	0.038±0.027	0.038±0.038	0.101±0.070	0.233±0.088
	rec	*0.209±0.096*	**0.234±0.065**	0.525±0.253	**0.817±0.330**
	$\pi_{0.5}$	3.271±1.270	3.219±1.260	2.993±1.413	**2.650±1.382**
	λ_1	0.860±0.047	0.876±0.029	0.916±0.012	0.920±0.008
DAODEN_full	e_4	*0.023±0.015*	*0.027±0.016*	0.072±0.045	**0.187±0.127**
	rec	**0.178±0.050**	*0.258±0.066*	*0.469±0.168*	1.003±0.380
	$\pi_{0.5}$	**3.533±1.139**	**3.496±1.215**	*3.426±1.512*	1.897±0.918
	λ_1	0.869±0.036	0.858±0.028	0.881±0.024	0.884±0.013

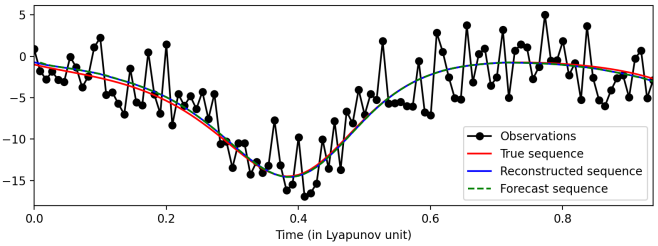

Fig. 4: An example of the the first dimension of the L63 system reconstructed by the inference module of DAODEN_determ, $r = 33\%$. Given the noisy observations (black dots), the inference module $q_{\phi}(\mathbf{z}_t|\mathbf{x}_{0:T})$ reconstructs a clean sequence of the hidden state (blue curve), which is very close to the true unknown dynamic (red curve). Given this sequence, the transition network (BiNN) can successfully learn the governing laws of the system, as it can do under ideal conditions. The green dash shows the forecast $\mathbf{z}^*_{t+1} = \mathcal{F}^1(\mathbf{z}^*_t)$ given the output $\mathbf{z}^*_t$ of q_{ϕ}.

We can see that if the noise level is not significantly high (r=33.3% or r=66.7%), all the models are able to capture the dynamical characteristics of the data. When the noise level is small, BiNN_EnKS tends to perform better than DAODEN. On the other hand, DAODEN models, especially DAODEN_full work well in these case. This may come from the capacity of LSTM to capture long-term correlations in data.

F. L96 case-study

In this section present experiments on the L96 system. The objective is to prove that the proposed framework can apply in high-dimensional spaces. We choose the deterministic and the full version of DAODEN as the candidate model. The results of models trained on noisy observations are shown in Table. IV. DAODEN models outperforms state-of-the-art methods both in terms of short-term prediction and long-term topology. In Fig. 8 we show the error between the true sequence and the sequence generated by the DAODEN_determ learnt on noisy observation with $r = 19.4\%$. Both sequences have the same starting point.

G. L63s case-study

Whereas most related works are designed for ODE only, (*i.e.* the governing equations are deterministic), the proposed framework accounts for stochastic perturbations, hence it can apply to Stochastic Differential Systems (SDEs). Using the stochastic Lorenz-63 system (L63s) presented in [50], we illustrate in this experiment the ability of DAODEN_full scheme to infer such stochastic governing equations from noisy observation data. We may recall that DAODEN_full scheme embeds

TABLE III: Performance of models trained on noisy and partial L63 data. The observations are sampled irregularly, both in time and in space, with a missing rate of 87.5%. For each index, the best score is marked in **bold** and the second best score is marked in *italic*.

Model		r 8.5%	16.7%	33.3%	66.7%
BiNN_EnKS	e_4	*0.129±0.081*	**0.143±0.065**	0.350±0.204	*0.973±0.649*
	rec	**0.721±0.204**	**1.062±0.401**	2.342±1.622	6.675±1.410
	$\pi_{0.5}$	1.873±1.034	**2.146±1.048**	*1.616±1.042*	*0.290±0.153*
	λ_1	0.801±0.016	0.782±0.012	0.304±0.147	-1.588±0.009
DAODEN_determ	e_4	0.135±0.082	0.170±0.105	*0.290±0.202*	25.034±19.821
	rec	1.300±1.525	1.448±1.332	1.985±1.474	4.222±2.191
	$\pi_{0.5}$	2.399±1.360	*2.140±1.110*	1.441±0.823	0.022±0.087
	λ_1	0.905±0.014	0.888±0.013	0.809±0.018	-0.011±0.014
DAODEN_MAP	e_4	0.175±0.119	0.325±0.235	0.459±0.343	9.105±7.136
	rec	1.352±0.997	1.705±1.434	*1.972±1.247*	*3.704±1.180*
	$\pi_{0.5}$	**2.628±1.448**	1.706±1.125	1.505±0.949	0.064±0.216
	λ_1	0.894±0.010	0.844±0.016	0.736±0.017	0.453±0.030
DAODEN_full	e_4	**0.089±0.062**	*0.158±0.104*	**0.162±0.104**	**0.254±0.142**
	rec	*1.052±0.612*	*1.268±0.718*	**1.685±0.928**	**2.725±1.356**
	$\pi_{0.5}$	*2.590±1.193*	1.943±0.904	**1.984±0.949**	**1.347±1.014**
	λ_1	0.892±0.011	0.846±0.013	0.859±0.013	0.720±0.019

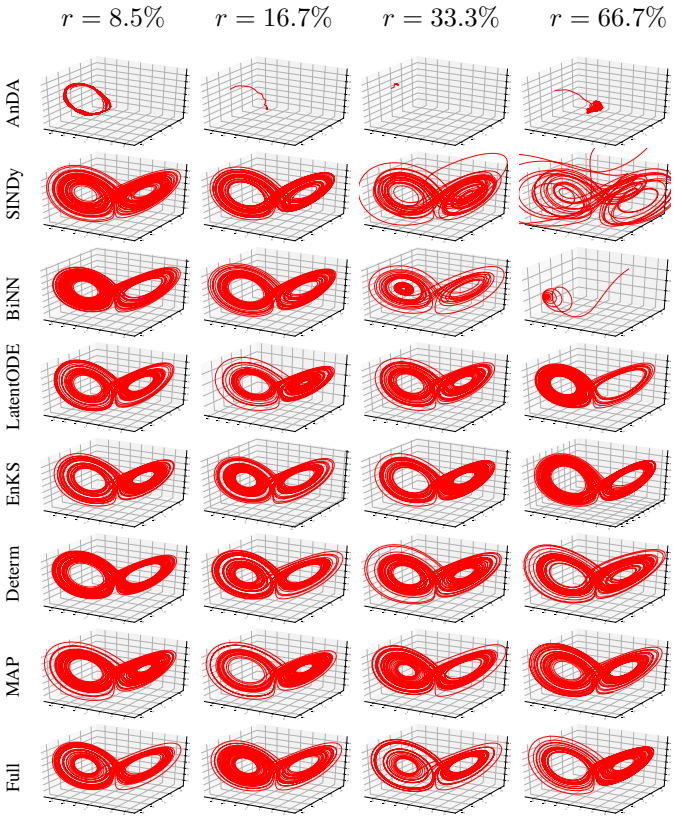

Fig. 5: Attactors generated by models trained on noisy data.

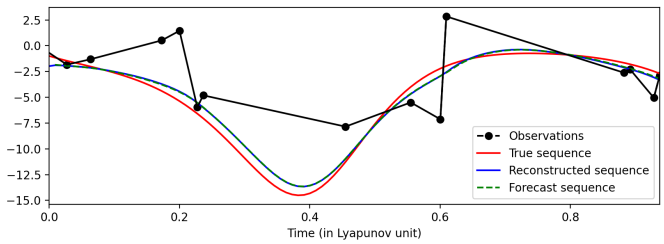

Fig. 6: An example of the the first dimension of the L63 system reconstructed by the inference module of DAODEN_determ. The observations are noisy ($r = 33\%$) and irregularly sampled with a missing rate of 87.5%.

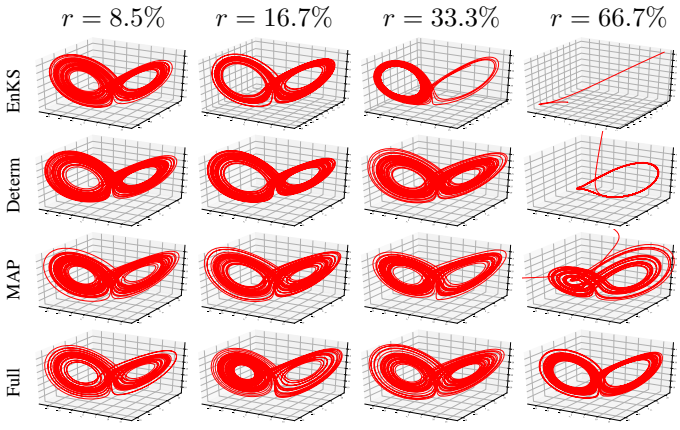

Fig. 7: Attractors generated by models trained on partially observed data scenario 1 and 2.

a parametric form of the covariance of perturbation $\boldsymbol{\omega}_t$ given by (3). Note that this parameterization is consistent with its true parametrization for L63s [50].

Here, we run experiments similar to Section V-E using L63s datasets with an additive Gaussian noise with $r = 33.3\%$. We then run the identification of the governing equations using both a deterministic parametrization (*e.g.*, BiNN_EnKS and DAODEN_determ) and the fully-stochastic scheme DAODEN_full. For weak stochastic perturbations, *i.e.* $\omega_t \neq 0$ (typically, γ larger than 8.0 in (28) in Appendix A-C). deterministic models like BiNN_EnKS or DAODEN_determ can still successfully capture the dynamics of the system (not reported in this paper). However, when $\boldsymbol{\omega}_t$ plays in important

TABLE IV: Performance of models trained on noisy L96 data. For each index, the best score is marked in **bold**.

Model		r 19.4%	r 38.8%
AnDA	e_4	0.582±0.106	1.140±0.174
	$\pi_{0.5}$	1.491±0.481	0.768±0.281
	λ_1	53.362±0.734	92.733±0.883
SINDy	$e4$	0.309±0.048	0.767±0.117
	$\pi_{0.5}$	0.628±0.166	0.150±0.047
	λ_1	1.444±0.048	1.316±0.045
BiNN	e_4	0.310±0.046	0.788±0.112
	$\pi_{0.5}$	2.503±0.565	1.111±0.274
	λ_1	1.409±0.019	1.041±0.016
DAODEN_determ	e_4	**0.048±0.006**	0.157±0.022
	$\pi_{0.5}$	**4.790±0.960**	**3.178±0.779**
	λ_1	1.624±0.022	1.601±0.023
DAODEN_full	e_4	0.067±0.014	**0.145±0.030**
	$\pi_{0.5}$	4.076±1.084	3.146±0.962
	λ_1	1.543±0.026	1.348±0.020

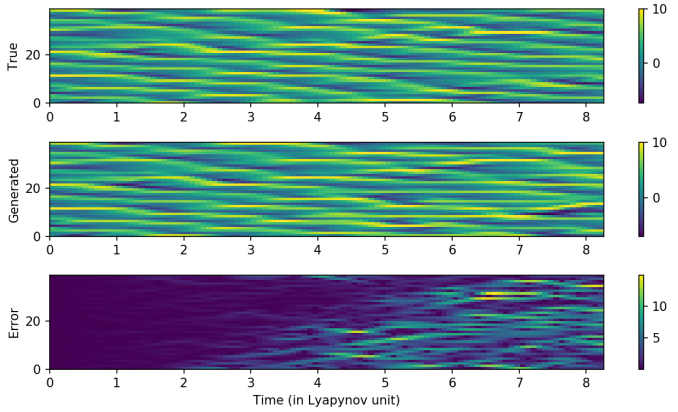

Fig. 8: The true L96 sequence (top), the sequence generated by the model trained on noisy data with $r = 19.4\%$ (middle) and the error between the true and the generated sequence (bot).

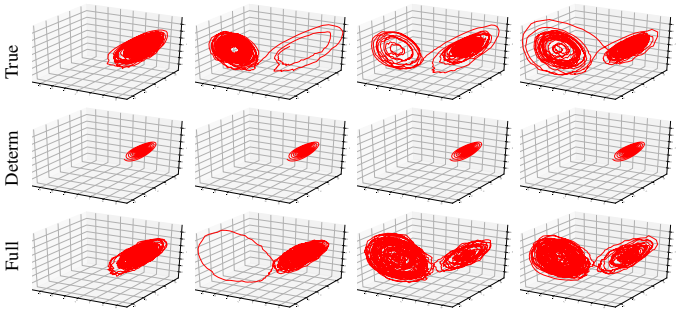

Fig. 9: Several attractors generated by the true L63s models (top), by DAODEN_determ (middle) and by DAODEN_full (bottom). The true L63s and DAODEN_full system are stochastic, hence each runtime we obtain a different sequence, even with the same initial condition. The models were trained on noisy observations with $r = 33.3\%$.

role in controlling the large-scale statistical characterization of the system, deterministic models fail as illustrated in Fig. 9 for L63s dynamics with $\gamma = 5.0$. By contrast, the fully-stochastic model successfully uncover the stochastic dynamics in both situations. In Fig. 9 top, we depict four different L63s trajectories from the same initial conditions. Due to the stochastic perturbation, the trajectories may strongly differ but all show a wide spreadout within the attractor. When considering a deterministic model (Fig. 9 middle), the four trajectories are strictly similar as there is no stochastic perturbation. Besides, the deterministic model simulates trajectories trapped on one side of the attractor, which cannot reproduce the spread of the true model. As illustrated in Fig. 9 bottom, DAODEN_full scheme succeed in capturing this stochastic patterns by embedding the stochastic factors of the system in the dispersion matrix $\mathbf{Q}_t$. Using a Monte Carlo technique, as presented in Alg. 2 in Appendix C, to forecast the state of the dynamics, we can obtain sequences with similar characteristics to the true L63s system.

H. Dealing with an unkown observation operator

In previous experiments, the observation operator $\mathcal{H}$ was known. We may also address the situation where it is unknown. It may for instance refer to the identification of some lower-dimensional governing equations of high-dimensional observations. Reduced-order modeling may also regarded as a situation where one looks for a lower-dimensional representation of some higher-dimensional dynamical system.

As case-study, we consider an experimental setting with Lorenz-63 dynamics similar to [51]. The 128-dimensional observation space derives from a 3-dimensional space, where the system is governed by L63 ODE, according to a polynomial of $\mathbf{z}_t$ and $\mathbf{z}_t^3$ with six spatial modes of Legendre coefficients (for details, see [51]). Whereas noise-free are considered in [51], we report here experiments with a Gaussian additive noise with r=19.4%. Fig. 10 shows the observations in a high-dimensional space. The inference scheme in [51] is an NN-based encoder, this architecture does not take into account the sequential correlations in the data, hence when the observations are noisy, it can not apply (because $p(\mathbf{z}_t|\mathbf{x}_t)$ is intractable). Moreover, [51] supposes that the time derivative $\frac{d\mathbf{x}_t}{dt}$ is observed. This assumption may not be true for many real-life systems. Our model, on the other hand, uses a state-space assimilation formulation. The inference scheme in our model is a sequential model, and we do not need the time derivative of the data.

The unknown observation operator $\mathcal{H}$ was parameterized by the same MLP architecture as the one used in [51]. We run this experiment with DAODEN_determ. Fig. 11 shows that the proposed framework successfully captures the low-dimensional attractor of the observed high-dimensional observation sequences. This is further supported by the first Lyapunov exponent of the learnt model $\lambda_1 = 0.92$ which is close to the true value (0.91). Because there are several possible solutions for this problem (any affine transformation of the original L63 is a solution), the coordinates of the learnt system are different, however, the topology is the same.

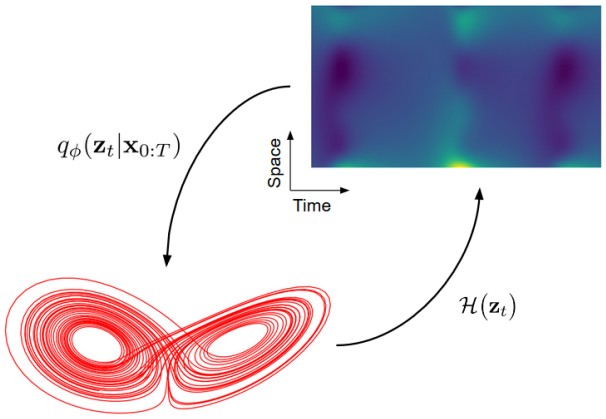

Fig. 10: Higher-dimensional Legendre observations governed by lower-dimensional L63 dynamics. Following [51], the observations (top right) are in a 128-dimensional space, while L63 dynamics (bottom left) are in a 3-dimensional space. The observation operator involves a non-linear mapping according to Legendre polynomials [51].

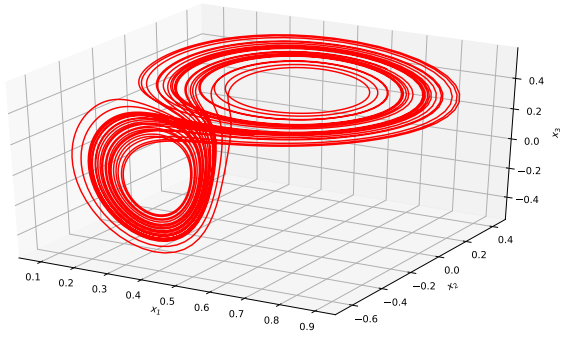

Fig. 11: Low-dimensional attractor generated by the proposed model trained from noisy higher-dimensional Legendre observations of L63 dynamics. This attractor recovers the topology of L63 dynamics. We let the reader refer to the main text for details on this experiment.

VI. Conclusions

This paper a novel deep learning scheme for the identification of governing equations of a given system from noisy and partial observation series. We combine a Bayesian formulation of the data assimilation to state-of-the-art deep learning architectures. Compared with related work [22], [23], we account for stochastic dynamics rather than only deterministic ones and derive an end-to-end architecture using a variational deep learning model, which fully conforms to the state-space formulation considered in data assimilation. Through numerical experiments for chaotic and stochastic dynamics, we have demonstrated that we can extend the observation configurations where we can recover hidden governing dynamics from noisy and partial data w.r.t. the state-of-the-art, including for high-dimensional systems governed by lower-dimensional dynamics.

Beyond the generalization of previous work through a variational Bayesian formulation, the proposed framework involves two key contributions w.r.t. state-of-the-art data assimilation schemes. We first show that neural network architectures bring new means for the parameterization of both the dynamical models and the inference scheme. Especially, our experiments support the relevance of LSTM-based architectures as alternatives to state-of-the-art data assimilation schemes such as Ensemble Kalman methods. Future work shall further explore these aspects and could benefit from the resulting end-to-end architecture to improve reconstruction performance [53]. For deep learning practitioners, our experiments point out that assimilation schemes and random n-step-ahead forecasting can be loosely considered as regularization techniques to prevent overfitting. We have also proved that the stochastic implementation of the proposed framework can capture characteristics of stochastic dynamical systems from noisy data. These results open new research avenues for dealing with real dynamical systems, for which the stochastic perturbations often play a significant role in driving long-term patterns.

References

[1] E. N. Lorenz, "Deterministic Nonperiodic Flow," *Journal of the Atmospheric Sciences*, vol. 20, no. 2, pp. 130–141, Mar. 1963.

[2] R. C. Hilborn, *Chaos and nonlinear dynamics: an introduction for scientists and engineers*. Oxford University Press on Demand, 2000.

[3] J. C. Sprott and J. C. Sprott, *Chaos and time-series analysis*. Citeseer, 2003, vol. 69.

[4] M. W. Hirsch, S. Smale, and R. L. Devaney, *Differential equations, dynamical systems, and an introduction to chaos*. Academic press, 2012.

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

[6] R. Lguensat, P. Tandeo, P. Ailliot, M. Pulido, and R. Fablet, "The Analog Data Assimilation," *Monthly Weather Review*, vol. 145, no. 10, pp. 4093 – 4107, Oct. 2017.

[7] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proceedings of the National Academy of Sciences*, vol. 113, no. 15, pp. 3932–3937, Apr. 2016.

[8] J. Pathak, Z. Lu, B. R. Hunt, M. Girvan, and E. Ott, "Using Machine Learning to Replicate Chaotic Attractors and Calculate Lyapunov Exponents from Data," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 27, no. 12, p. 121102, Dec. 2017, arXiv: 1710.07313.

[9] J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott, "Model-Free Prediction of Large Spatiotemporally Chaotic Systems from Data: A Reservoir Computing Approach," *Physical Review Letters*, vol. 120, no. 2, p. 024102, Jan. 2018.

[10] R. Fablet, S. Ouala, and C. Herzet, "Bilinear Residual Neural Network for the Identification and Forecasting of Geophysical Dynamics," in *2018 26th European Signal Processing Conference (EUSIPCO)*, Sep. 2018, pp. 1477–1481, iSSN: 2219-5491.

[11] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Multistep Neural Networks for Data-driven Discovery of Nonlinear Dynamical Systems," *arXiv:1801.01236 [nlin, physics:physics, stat]*, Jan. 2018, arXiv: 1801.01236.

[12] T. Qin, K. Wu, and D. Xiu, "Data Driven Governing Equations Approximation Using Deep Neural Networks," *arXiv:1811.05537 [cs, math, stat]*, Nov. 2018, arXiv: 1811.05537.

[13] I. Ayed, E. de Bézenac, A. Pajot, J. Brajard, and P. Gallinari, "Learning Dynamical Systems from Partial Observations," *arXiv:1902.11136 [physics]*, Feb. 2019, arXiv: 1902.11136.

[14] P. R. Vlachas, W. Byeon, Z. Y. Wan, T. P. Sapsis, and P. Koumoutsakos, "Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks," *Proceedings. Mathematical, Physical, and Engineering Sciences*, vol. 474, no. 2213, May 2018.

[15] D. W. Pierce, "Distinguishing coupled ocean–atmosphere interactions from background noise in the North Pacific," *Progress in Oceanography*, vol. 49, no. 1-4, pp. 331–352, 2001.

[16] C. Johnson, N. K. Nichols, and B. J. Hoskins, "Very large inverse problems in atmosphere and ocean modelling," *International journal for numerical methods in fluids*, vol. 47, no. 8-9, pp. 759–771, 2005.

[17] J. Isern-Fontanet and E. Hascoët, "Diagnosis of high-resolution upper ocean dynamics from noisy sea surface temperatures," *Journal of Geophysical Research: Oceans*, vol. 119, no. 1, pp. 121–132, 2014.

[18] M. Bocquet, J. Brajard, A. Carrassi, and L. Bertino, "Data assimilation as a learning tool to infer ordinary differential equation representations of dynamical models," *Nonlinear Processes in Geophysics*, vol. 26, no. 3, pp. 143–162, Jul. 2019.

[19] G. Evensen and P. J. van Leeuwen, "An Ensemble Kalman Smoother for Nonlinear Dynamics," *Monthly Weather Review*, vol. 128, no. 6, pp. 1852–1867, Jun. 2000.

[20] D. Nguyen, S. Ouala, L. Drumetz, and R. Fablet, "EM-like Learning Chaotic Dynamics from Noisy and Partial Observations," Mar. 2019.

[21] ——, "Assimilation-Based Learning of Chaotic Dynamical Systems from Noisy and Partial Data," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2020, pp. 3862–3866, iSSN: 2379-190X.

[22] J. Brajard, A. Carrassi, M. Bocquet, and L. Bertino, "Combining data assimilation and machine learning to emulate a dynamical model from sparse and noisy observations: a case study with the Lorenz 96 model," *Geoscientific Model Development Discussions*, vol. 2019, pp. 1–21, 2019.

[23] M. Bocquet, J. Brajard, A. Carrassi, and L. Bertino, "Bayesian inference of chaotic dynamics by merging data assimilation, machine learning and expectation-maximization," *Foundations of Data Science*, vol. 2, no. 1, pp. 55–80, 2020, arXiv: 2001.06270.

[24] Z. Ghahramani and G. E. Hinton, "Parameter estimation for linear dynamical systems," Technical Report CRG-TR-96-2, University of Totronto, Dept. of Computer Science, Tech. Rep., 1996.

[25] Z. Ghahramani and S. T. Roweis, "Learning Nonlinear Dynamical Systems Using an EM Algorithm," in *Advances in Neural Information Processing Systems 11*, M. J. Kearns, S. A. Solla, and D. A. Cohn, Eds. MIT Press, 1999, pp. 431–437.

[26] G. Welch and G. Bishop, "An introduction to the Kalman filter," 1995.

[27] M. Hoshiya and E. Saito, "Structural identification by extended Kalman filter," *Journal of engineering mechanics*, vol. 110, no. 12, pp. 1757–1770, 1984.

[28] G. Evensen, *Data Assimilation: The Ensemble Kalman Filter*. Springer Science & Business Media, Aug. 2009, google-Books-ID: 2_zaTb_O1AkC.

[29] A. Doucet and A. M. Johansen, "A tutorial on particle filtering and smoothing: Fifteen years later," *Handbook of nonlinear filtering*, vol. 12, no. 656-704, p. 3, 2009.

[30] H. U. Voss, J. Timmer, and J. Kurths, "Nonlinear dynamical system identification from uncertain and indirect measurements," *International Journal of Bifurcation and Chaos*, vol. 14, no. 06, pp. 1905–1933, Jun. 2004.

[31] B. Nagarajan, L. Delle Monache, J. P. Hacker, D. L. Rife, K. Searight, J. C. Knievel, and T. N. Nipen, "An Evaluation of Analog-Based Postprocessing Methods across Several Variables and Forecast Models," *Weather and Forecasting*, vol. 30, no. 6, pp. 1623–1643, Dec. 2015.

[32] P. L. McDermott and C. K. Wikle, "A model-based approach for analog spatio-temporal dynamic forecasting," *Environmetrics*, vol. 27, no. 2, pp. 70–82, 2016.

[33] Z. Zhao and D. Giannakis, "Analog forecasting with dynamics-adapted kernels," *Nonlinearity*, vol. 29, no. 9, pp. 2888–2939, Aug. 2016.

[34] K. Yeo and I. Melnyk, "Deep learning algorithm for data-driven simulation of noisy dynamical system," *Journal of Computational Physics*, vol. 376, pp. 1212–1231, Jan. 2019.

[35] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, "Neural Ordinary Differential Equations," *arXiv:1806.07366 [cs, stat]*, Jun. 2018, arXiv: 1806.07366.

[36] Y. Rubanova, R. T. Q. Chen, and D. K. Duvenaud, "Latent Ordinary Differential Equations for Irregularly-Sampled Time Series," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d. Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 5320–5330.

[37] C. Bishop, *Pattern Recognition and Machine Learning*, ser. Information Science and Statistics. New York: Springer-Verlag, 2006.

[38] Y. Burda, R. Grosse, and R. Salakhutdinov, "Importance Weighted Autoencoders," *arXiv:1509.00519 [cs, stat]*, Nov. 2016, arXiv: 1509.00519.

[39] C. J. Maddison, D. Lawson, G. Tucker, N. Heess, M. Norouzi, A. Mnih, A. Doucet, and Y. W. Teh, "Filtering Variational Objectives," in *Advances in Neural Information Processing Systems*, May 2017, pp. 6576–6586.

[40] S. P. Khare, J. L. Anderson, T. J. Hoar, and D. Nychka, "An investigation into the application of an ensemble Kalman smoother to high-dimensional geophysical systems," *Tellus A: Dynamic Meteorology and Oceanography*, vol. 60, no. 1, pp. 97–112, Jan. 2008.

[41] J. Chung, K. Kastner, L. Dinh, K. Goel, A. Courville, and Y. Bengio, "A Recurrent Latent Variable Model for Sequential Data," in *Advances in neural information processing systems*, Jun. 2015, pp. 2980–2988.

[42] M. Fraccaro, S. r. K. Sø nderby, U. Paquet, and O. Winther, "Sequential Neural Models with Stochastic Layers," in *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2016, pp. 2199–2207.

[43] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[44] P. Courtier, J.-N. Thépaut, and A. Hollingsworth, "A strategy for operational implementation of 4D-Var, using an incremental approach," *Quarterly Journal of the Royal Meteorological Society*, vol. 120, no. 519, pp. 1367–1387, 1994.

[45] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[46] S. Ouala, A. Pascual, and R. Fablet, "Residual Integration Neural Network," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 3622–3626, iSSN: 2379-190X.

[47] H. Jaeger, *Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the" echo state network" approach*. GMD-Forschungszentrum Informationstechnik Bonn, 2002, vol. 5.

[48] N. Mohajerin and S. L. Waslander, "Multistep Prediction of Dynamic Systems With Recurrent Neural Networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3370–3383, Nov. 2019.

[49] E. N. Lorenz, "Predictability: A problem partly solved," in *Seminar on predictability*, vol. 1, 1996.

[50] B. Chapron, P. Dérian, E. Mémin, and V. Resseguier, "Large-scale flows under location uncertainty: a consistent stochastic framework," *Quarterly Journal of the Royal Meteorological Society*, vol. 144, no. 710, pp. 251–260, 2018.

[51] K. Champion, B. Lusch, J. N. Kutz, and S. L. Brunton, "Data-driven discovery of coordinates and governing equations," *arXiv:1904.02107 [stat]*, Mar. 2019, arXiv: 1904.02107.

[52] A. Wolf, J. B. Swift, H. L. Swinney, and J. A. Vastano, "Determining Lyapunov exponents from a time series," *Physica D: Nonlinear Phenomena*, vol. 16, no. 3, pp. 285–317, Jul. 1985.

[53] R. Fablet, L. Drumetz, and F. Rousseau, "Joint learning of variational representations and solvers for inverse problems with partially-observed data," *arXiv:2006.03653 [cs, eess, stat]*, Jun. 2020, arXiv: 2006.03653.

[54] S. Ouala, D. Nguyen, L. Drumetz, B. Chapron, A. Pascual, F. Collard, L. Gaultier, and R. Fablet, "Learning Latent Dynamics for Partially-Observed Chaotic Systems," *arXiv:1907.02452 [cs, stat]*, Jul. 2019, arXiv: 1907.02452.

APPENDIX A
DYNAMICAL SYSTEMS

A. The Lorenz-63 system

The Lorenz-63 system (L63), named after Edward Lorenz, is a 3-dimensional dynamical system that model the atmospheric convection [1]. The L63 is governed by the following ODE:

$$\begin{aligned}\frac{\mathrm{d}\mathbf{x}_{t,1}}{\mathrm{d}t} &= \sigma\left(\mathbf{x}_{t,2}-\mathbf{x}_{t,1}\right)\\ \frac{\mathrm{d}\mathbf{x}_{t,2}}{\mathrm{d}t} &= \left(\rho-\mathbf{x}_{t,3}\right)\mathbf{x}_{t,1}-\mathbf{x}_{t,2}\\ \frac{\mathrm{d}\mathbf{x}_{t,3}}{\mathrm{d}t} &= \mathbf{x}_{t,1}\mathbf{x}_{t,2}-\beta\mathbf{x}_{t,3}\end{aligned} \tag{26}$$

When $\sigma = 11$, $\rho = 28$ and $\beta = 8/3$, this system has a chaotic behavior, with the Lorenz attractor shown in Fig. 12.

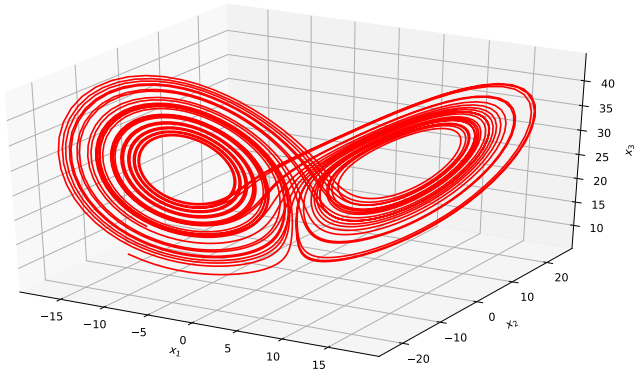

Fig. 12: The attractor of the Lorenz–63 system when $\sigma = 10$, $\rho = 28$ and $\beta = 8/3$.

Some characteristics of the L63 with the above set of parameters are as follows:

- The system is chaotic, a minor change in the initial condition will lead to a completely different trajectory in long term.
- The attractor of the L63 has a "butterfly form", the particles frequently change side of the attractor. The density of the particles in two sides of the attractor is also similar.

B. The Lorenz-96 system

The Lorenz-96 system (L96) [49] is a periodic 40-dimensional dynamical system governed by the following ODEs:

For $i = 1, ..N_x$:

$$\frac{\mathrm{d}\mathbf{x}_{t,i}}{\mathrm{d}t} = (\mathbf{x}_{t,i+1} - \mathbf{x}_{t,i-2})\mathbf{x}_{t,i-1} - \mathbf{x}_{t,i} + F \qquad (27)$$

with $N_x = 40$, $\mathbf{x}_{t,-1} = \mathbf{x}_{t,N_x-1}$, $\mathbf{x}_{t,0} = \mathbf{x}_{t,N_x}$ and $\mathbf{x}_{t,N_x+1} = \mathbf{x}_{t,1}$.

We choose $F = 8$ to have chaotic system.

C. The stochastic Lorenz-63 system

The stochastic Lorenz-63 system (L63s) is presented in [50]. It is a modified version of the L63 to model situations where the large-scale characteristics of a physical event may be changed because of accumulated perturbations in fine scales. The governing equations of the L63s are as follow:

$$\begin{aligned}
\mathrm{d}\mathbf{x}_{t,1} &= \left(\sigma\left(\mathbf{x}_{t,2} - \mathbf{x}_{t,1}\right) - \frac{4}{2\gamma}\mathbf{x}_{t,1}\right)\mathrm{d}t \\
\mathrm{d}\mathbf{x}_{t,2} &= \left(\left(\rho - \mathbf{x}_{t,3}\right)\mathbf{x}_{t,1} - \mathbf{x}_{t,2} - \frac{4}{2\gamma}\mathbf{x}_{t,2}\right)\mathrm{d}t + \frac{\rho - \mathbf{x}_{t,3}}{\gamma^{0.5}}\mathrm{d}B_t \\
\mathrm{d}\mathbf{x}_{t,3} &= \left(\mathbf{x}_{t,1}\mathbf{x}_{t,2} - \beta\mathbf{x}_{t,3} - \frac{8}{2\gamma}\mathbf{x}_{t,3}\right)\mathrm{d}t + \frac{\mathbf{x}_{t,2}}{\gamma^{0.5}}\mathrm{d}B_t
\end{aligned} \qquad (28)$$

with B_t a Brownian motion.

In the L63s, the noise level is controlled by γ. The data used in this paper were generated with $\sigma = 11$, $\rho = 28$ and $\beta = 8/3$ and $\gamma = 5$. With this set of parameters, the particles are easily trapped in one side of the attractor, as shown in Fig. 9 in the paper.

APPENDIX B
MODEL SETUP

A. Models used for the L63 and the L63s

All the four models (BiNN_EnKs, DAODEN_determ, DAODEN_MAP and DAODEN_full) use the same dynammical sub-module: a BiNN. The architecture of this network is presented in Table. V. The terms Linear and Bilinear are for the Linear and the Bilinear modules implemented in Pytorch.

TABLE V: Architecture of the BiNN used for the L63 and the L63s.

Parameter	**Value**
Number of Linear cells	1
Linear cell size	$[3, 3]$
Linear cell activation	Linear
Number of Bilinear cells	3
Bilinear cell size	$[3, 3, 3]$
Bilinear activation	Linear

For BiNN_EnKS, we used the EnKS implementation suggested in [19]. The size of the ensemble was choosen as 50.

As shown in Fig. 2 in the paper, the inference scheme of DAODEN models is an LSTM-based network. The parameters of the inference sub-module of DAODEN_full is presented in Table. VI. All the encoders and the decoders are MLPs. Similar architectures were used for DAODEN_determ and DAODEN_MAP, by removing the variance parts.

TABLE VI: Architecture of the inference scheme of DAODEN_full used for the L63 and the L63s.

Parameter	**Value**
LSTM layers	2
LSTM hidden state dimension	9
MLP^{enc} size	$[3, 7, 3]$
MLP^{enc} activation	ReLU
MLP^{dec} size	$[21, 7, 6]$
MLP^{dec} activation	ReLU

B. Models used for the L96

For the L96, we used the convolutional version of BiNN, as presented in [23].

The architecture of the inference scheme is presented in Table. VII.

TABLE VII: Architecture of the inference scheme of DAODEN_determ used for the L96.

Parameter	**Value**
LSTM layers	2
LSTM hidden state dimension	80
MLP^{enc} size	$[40, 80, 40]$
MLP^{enc} activation	ReLU
MLP^{dec} size	$[200, 80, 40]$
MLP^{dec} activation	ReLU

C. Models used for the L63 with Legendre observations

The dynamical sub-module of the DAODEN_determ model used in Section V-H is the same as the one presented in Section B-B. The architecture of the inference scheme used in Section V-H is presented in Table. VIII.

TABLE VIII: Architecture of the inference scheme of DAODEN_determ used for the L63 with Legendre observations

Parameter	**Value**
LSTM layers	2
LSTM hidden state dimension	9
MLP^{enc} size	$[128, 64, 32, 3]$
MLP^{enc} activation	Sigmoid
MLP^{dec} size	$[21, 32, 64, 128]$
MLP^{dec} activation	Sigmoid

APPENDIX C
SIMULATION OF STOCHASTIC DYNAMICS

To simulate a stochastic sequence given the learnt stochastic model ($\mathcal{F}$ and MLP^{var_dyn} in the case of DAODEN_full), we use the following algorithm:

Algorithm 2: Generate stochastic sequence

Result: A sequence $\mathbf{S}$ of length N, generated by the model $\{\mathcal{F}, MLP^{var\_dyn}\}$, starting form the initial condition $\mathbf{x}_0$.

Inputs: N, $\mathcal{F}$, MLP^{var_dyn}, $\mathbf{x}_0$;

$\mathbf{x} = \mathbf{x}_0$;

$\mathbf{S} = list()$;

t = 0;

while $t < N$ **do**

  $\mu = \mathcal{F}^1(\mathbf{x})$;

  $\mathbf{d}^{dyn} = MLP^{var\_dyn}(\mathbf{x})$;

  $\mathbf{x} \sim \mathcal{N}(\mu, \mathbf{d}^{dyn}\mathbf{I})$;

  $\mathbf{S}$.append($\mathbf{x}$);