Uncertainty Quantification When Learning Dynamical Models and Solvers With Variational Methods

Nicolas Lafon, Ronan Fablet, Philippe Naveau

JAMES | Journal of Advances in Modeling Earth Systems

RESEARCH ARTICLE
10.1029/2022MS003446

Uncertainty Quantification When Learning Dynamical Models and Solvers With Variational Methods

N. Lafon[1], R. Fablet[2], and P. Naveau[1]

[1]Laboratoire des Sciences du Climat et de l'Environnement, EstimR, IPSL-CNRS, CEA Saclay, Gif-sur-Yvette, France, [2]IMT Atlantique, UMR CNRS Lab-STICC, Brest, France

Key Points:

- We propose a neural approach to jointly solve data assimilation and uncertainty quantification problems using a variational Bayes formulation
- Our end-to-end neural architecture implements a learnable gradient-based solver for a Evidence Lower BOund criterion
- Studies conducted on Lorenz 63 dynamics and on river discharge from the Danube river network highlight the potential of our approach

Correspondence to:
N. Lafon,
nicolas.lafon@lsce.ipsl.fr

Citation:
Lafon, N., Fablet, R., & Naveau, P. (2023). Uncertainty quantification when learning dynamical models and solvers with variational methods. *Journal of Advances in Modeling Earth Systems*, *15*, e2022MS003446. https://doi.org/10.1029/2022MS003446

Received 11 OCT 2022
Accepted 27 SEP 2023

Author Contributions:
Conceptualization: N. Lafon, R. Fablet, P. Naveau
Investigation: N. Lafon, R. Fablet, P. Naveau
Methodology: N. Lafon, R. Fablet, P. Naveau
Software: N. Lafon, R. Fablet
Supervision: R. Fablet, P. Naveau
Writing – original draft: N. Lafon, R. Fablet, P. Naveau
Writing – review & editing: N. Lafon, R. Fablet, P. Naveau

Abstract In geosciences, data assimilation (DA) addresses the reconstruction of a hidden dynamical process given some observation data. DA is at the core of operational systems such as weather forecasting, operational oceanography and climate studies. Beyond the reconstruction of the mean or most likely state, the inference of the state posterior distribution remains a key challenge, that is, quantify uncertainties as well as to inform intrinsical stochastic variabilities. Indeed, DA schemes, such as variational DA and Kalman methods, can have difficulty in dealing with complex non-linear processes. A growing literature investigates the cross-fertilization of DA and machine learning. This study proposes an end-to-end neural scheme based on a variational Bayes inference formulation to jointly address DA and uncertainty quantification. It combines an Evidence Lower BOund variational cost to a trainable gradient-based solver to infer the state posterior probability distribution function given observation data. The inference of the posterior and the trainable solver are learnt jointly. We demonstrate the relevance of the proposed scheme for a Gaussian parameterization of the posterior and different case-study experiments, including Lorenz 63 dynamics and river flow measurements. A benchmark with respect to state-of-the-art schemes is provided.

Plain Language Summary The spatiotemporal reconstruction of a dynamical process from some observation data is at the core of a wide range of applications in geosciences. This is particularly true for weather forecasting, operational oceanography and climate studies. However, the reconstruction of a given dynamic and the prediction of future states must take into account the uncertainties that affect the system. Thus, the available observation measurements are only provided with a limited accuracy. Besides, the encoded physical equations that model the evolution of the system do not capture the full complexity of the real system. Finally, the numerical approximation generates a non-negligible error. For these reasons, it seems relevant to calculate a probability distribution of the state system rather than the most probable state. Using recent advances in machine learning techniques for inverse problems, we propose an algorithm that jointly learns a parametric distribution of the state, the dynamics governing the evolution of the parameters, and a solver. Experiments conducted on synthetic reference data sets, as well as on data sets describing environmental systems, validate our approach.

1. Introduction

The reconstruction and forecasting of dynamical systems from available observations are key challenges in earth sciences (see e.g., Welch & Bishop, 1995). These tasks have been classically addressed by data assimilation (DA) approaches, especially variational DA and ensemble Kalman schemes (see e.g., Evensen, 2009). DA methods have greatly improved over years, especially by accounting for model error, which is important when dealing with misrepresented physical processes (Machenhauer & Kirchner, 2000) and unresolved small-scale processes (Hamill & Whitaker, 2005) with respect to the space-time model resolution. Whereas Kalman-based ensemble methods (Evensen, 1994; Gordon et al., 1993) do take into account model error from the beginning, in a variational setting, operational systems based on 4D-Var (Rabier et al., 2000) moved from strong constraints assumptions (Le Dimet & Talagrand, 1986) to weak constraints one (see e.g., Trémolet, 2006) to reach this goal. In both approaches, estimating the model error in the form of a model error covariance matrix becomes crucial.

In many applications, such as risk assessment (see e.g., Mohsan et al., 2021), it is critical to evaluate the uncertainty in the state predicted by the DA method. This is the issue we focus on in this paper. This uncertainty quantification problem can be viewed as estimating the whole posterior distribution of the state given observations rather than focusing on the mean or mode of this posterior. However, standard variational methods do not directly

allow to estimate uncertainties of the predicted state and have to be specifically tuned to this purpose (Isaksen et al., 2010), while Kalman-based ensemble methods provide a Gaussian estimate of the posterior distribution of the state through a covariance matrix updated at each time step (see e.g., Evensen, 2003; Evensen & Van Leeuwen, 2000) which is relevant in Gaussian-linear case and typically fail in cases with strong non-linearity (Evensen et al., 2022). Particle filters (Gordon et al., 1993; Van Leeuwen, 2009) are the main methods for sampling the full posterior probability distribution function (pdf), but they suffer from curse of dimensionality when dealing with high-dimensional states (Snyder et al., 2008). This may prevent their application to real-world cases. As variational Bayes (VB) refers to the field of research dedicated to approximating the full posterior of latent variables of a Bayesian model given observation data (Blei & Jordan, 2006; Jordan et al., 1999), we note that assessing the uncertainties in the predicted state is indeed a VB problem. Inferring the posterior through a VB formulation often requires to maximize an Evidence Lower BOund (ELBO; see e.g., Hoffman et al., 2013). To this end, learning-based approaches appeared to be particularly relevant (Kingma & Welling, 2013).

Recently, a rich literature has emerged to apply machine learning (ML) paradigms to address DA issues. ML schemes are particularly efficient to solve complex and high-dimensional optimization problems and encounter numerous successes including image classification (Krizhevsky et al., 2012; Le, 2013), natural language processing (Otter et al., 2020), language translation (Sutskever et al., 2014) computational physics (Mohan et al., 2023; Raissi et al., 2019). Regarding DA, ML-based algorithms offer new means to learn the governing equations of the dynamics (Fablet et al., 2018; Long et al., 2018) and the associated flow operator (Bocquet et al., 2020; Scher & Messori, 2019), or model correction terms (Farchi et al., 2021), directly from model outputs. Some approaches are even designed to be used in a plug-and-play manner in state-of-the-art DA schemes (Fablet, Chapron, et al., 2021). When considering variational DA, trainable emulators of the adjoint operator of the dynamics (Nonnenmacher & Greenberg, 2021) or directly of the gradient-based DA solver (Fablet, Chapron, et al., 2021) emerged as appealing solutions. Similarly, recent studies have explored learning-based Kalman techniques (de Bézenac et al., 2020). The latter is particularly relevant to address uncertainty quantification. The underlying assumption of the existence of the linear-Gaussian latent space may however restrict their application in real-world case-studies. Generative adversarial networks also naturally arose as appealing ML tools to develop new ensemble DA schemes (Silva et al., 2023).

In this paper, we propose a ML-based approach to consistently approximate by a Gaussian distribution the posterior distribution of the state of a dynamical system given a set of observations. This involves estimating both the mean and the covariance parameters of the Gaussian distribution. Since we are producing probabilistic predictions, the standard mean square error (MSE) is not appropriate as a learning cost. Instead, we choose the logarithmic score as the learning function which is consistent with probabilistic predictions. Our approach relies on a training stage where both true states and observations are available. To circumvent the instability when minimizing the chosen learning function, we constrain our output parameters to be close to an optimum with respect to another cost derived from a VB inference formulation. We prove that the optimum of this cost should be a good first-guess of the minimum of our learning function. Our end-to-end architecture exploits a trainable surrogate representation of the dynamics and a trainable gradient-based solver. It can therefore be considered as an extension of Fablet, Chapron, et al. (2021) to estimate the covariance of the posterior in addition to the mean. To the best of our knowledge, this is the first study which combines a trainable solver for variational DA along with a VB formulation. We claim that our approach could be extended to broader families of posteriors than Gaussian.

This paper is structured as follows. Section 2 introduces necessary background on weak-constraint variational DA. Section 3 presents the proposed approach, based on ELBO maximization, and the associated end-to-end neural framework. Numerical experiments on Lorenz 63 dynamics and discharges on Danube river network are reported in Section 4. Finally, concluding remarks are provided in Section 5.

2. Background on Weak-Constraint Variational Formulation

DA relies on state-space formulation for some time-dependent state x and associated time-dependent observations y. Within a discretized setting, $x(t)$ and $y(t)$ are random vectors of respective dimension n and d with $n \geq d$ for each t. Given $x(t_0)$, the state-space formulation could be set as:

$$\begin{aligned} x(t) &= \mathcal{M}(x(t-\Delta t)) + \eta(t) \quad t \in \Omega_T = \{t_0 + \Delta t, \ldots, t_0 + N\Delta t\} \\ y(t) &= \mathcal{H}(x(t)) + \epsilon(t), \qquad t \in O_T \subset \Omega_T \end{aligned} \tag{1}$$

with $\mathcal{M}$ the dynamical model and $\mathcal{H}$ the observation operator. In the following, we improperly denote by x and y the concatenation of $x(t)$ and $y(t)$ on each t for which they exist. Random noise η and ϵ represent respectively the model error and the observation error. Assuming a zero-mean random noise η, the weak-constraint variational DA formulation (Sasaki, 1970) states the reconstruction or forecasting of x given y as the minimization of the following cost:

$$U_\phi(x, y) = \sum_{t_i \in O_T} \|\mathcal{H}(x(t_i)) - y(t_i)\|^2_{\mathbf{R}} + \sum_{t_i \in \Omega_T} \|x(t_i) - \phi(x)(t_i)\|^2_{\mathbf{Q}}, \tag{2}$$

where, to match notation of Fablet, Chapron, et al. (2021), we have defined ϕ as the following operator

$$\phi(x)(t) = \mathcal{M}(x(t - \Delta t)). \tag{3}$$

Note that in Equation 2, we deliberately omit the background term used to measure the distance to a given background state, which acts as a Tikhonov regularization term on the minimization issue. We made this choice because our approach does not require the explicit use of a background term in a cost function. On the right side of Equation 2, the first term represents the data fidelity term with respect to the observations, whereas the second one penalizes the discrepancy between the state and the underlying dynamics. The considered norms are Mahalanobis norm (see Appendices A and B) with respect to covariance matrices $\mathbf{R}$ and $\mathbf{Q}$, of respective shape $d \times d$ and $n \times n$. $\mathbf{R}$ is the observation error covariance matrix while $\mathbf{Q}$ is the model error covariance matrix. The estimation of these matrices is of paramount importance (see e.g., Tandeo et al., 2018; Trémolet, 2007) to correctly estimate x. Lag-innovation (Belanger, 1974), and Bayesian inference-based methods such as Stroud et al. (2018) and Tandeo et al. (2015) addressed the estimation of these matrices.

3. Proposed Approach

The minimization of the variational cost of Equation 2 allows to estimate the state x but not to approximate the whole posterior distribution $p(x|y)$. We propose a deep learning scheme which approximates the posterior by a Gaussian distribution. In Section 3.1, we derive a new cost, named stochastic variational cost, to estimate covariances in addition to the mean state. Then, based on the work of Fablet, Chapron, et al. (2021), we introduce a deep learning scheme in Section 3.2 that imposes its outputs to be close to a minimum of the stochastic variational cost. Our deep learning scheme consists of two elements, a neural solver of the stochastic variational cost, and a surrogate model over posterior parameters. Finally, in Section 3.3 we explain how both elements of our approach could be learned jointly from ground-truth data with respect to a logarithmic score. This score allows us to evaluate the quality of the approximation we make to the true posterior. In contrast to Kalman methods (Evensen & Van Leeuwen, 2000), our approach does not rely on the prior computation of the model error covariance matrix.

3.1. Deriving Stochastic Variational Cost Through Variational Bayes Formulation

We consider the state-space formulation of Equation 1. In the following, $\mathcal{H}$ is a linear operator such that $\mathcal{H}(x(t)) = \mathbf{H}x(t)$ with $\mathbf{H}$ a $d \times n$ matrix. VB inference (Kingma & Welling, 2013) relies on the approximation of the true posterior pdf $p(x|y)$ by a parametric target pdf $q(x|y)$. For any parametric target pdf, the log of the evidence, in this case the log probability of observations y, admits the following lower bound:

$$\log p(y) \geq \mathbb{E}_{x \sim q(\cdot|y)} \log\left(\frac{p(x, y)}{q(x|y)}\right),$$

with equality whenever $q(x|y) = p(x|y)$ for any x. This lower bound is called ELBO. We can equivalently rewrite this inequality:

$$\log p(y) \geq \mathbb{E}_{x \sim q(\cdot|y)} \log(p(y|x)) - D_{KL}(q(x|y)\|p(x)), \tag{4}$$

where D_{KL} denotes the Kullback-Leibler divergence which measures how two distributions differ from each other, and is given by:

$$D_{KL}(q\|p) = \mathbb{E}_{x \sim q} \log\left(\frac{q(x)}{p(x)}\right).$$

Maximizing the ELBO can then lead to a computationally-tractable maximization of a lower-bound of the likelihood $p(y)$ (Hoffman et al., 2013). Thus, VB inference consists in maximizing the ELBO with respect to q, so q approximates the posterior distribution.

Let us further assume a Gaussian parametrization for target pdf $q(x|y)$ and a Gaussian additive noise model for observation likelihood $p(y|x)$. In practice, we set

$$q(x|y) = \prod_{t_i \in \Omega_T} q^{(t_i)}(x(t_i)|y) \text{ with } q^{(t_i)}(x(t_i)|y) = \mathcal{N}(x(t_i)\ ;\ \mu(t_i), \mathbf{\Sigma}(t_i)),$$

and

$$p(y|x) = \prod_{t_i \in O_T} p(y(t_i)|x(t_i)) \text{ with } p(y(t_i)|x(t_i)) = \mathcal{N}(y(t_i)\ ;\ \mathbf{H}x(t_i), \mathbf{R}).$$

Following Appendix B, we then derive:

$$\mathbb{E}_{x\sim q(\cdot|y)} \log(p(y|x)) = -\frac{1}{2} \sum_{t_i \in O_T} \left(\mathrm{Tr}\left(\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} \mathbf{\Sigma}(t_i)\right) + \|\mathbf{H}\mu(t_i) - y(t_i)\|^2_{\mathbf{R}} \right), \tag{5}$$

up to a function of $\mathbf{R}$. Under the assumption that norm of the posterior covariances is significantly smaller than that of the observation covariance, this term reduces to $-\frac{1}{2} \sum_{t_i \in O_T} \|\mathbf{H}\mu(t_i) - y(t_i)\|^2_{\mathbf{R}}$.

With regards to the Kullback-Leibler divergence in ELBO expression of Equation 4, an analytic expression is only tractable for some specific priors. By analytic expression, we mean an expression built with well-known operations that lend themselves readily to calculation. For illustration purposes, let assume a Gaussian prior whose pdf satisfies $p(x) = \prod_{t_i \in \Omega_T} \mathcal{N}(x(t_i)\ ;\ m, \mathbf{S})$, then we can derive the following analytical expression:

$$-D_{KL}(q(x|y)\|p(x)) = -\frac{1}{2} \sum_{t_i \in \Omega_T} \left(\mathrm{Tr}\left(\mathbf{S}^{-1} \mathbf{\Sigma}(t_i)\right) + \|\mu(t_i) - m\|^2_S + \log\left(\frac{|\mathbf{S}|}{|\mathbf{\Sigma}(t_i)|} \right) \right). \tag{6}$$

In the general case, that is, without assuming any specific form for the prior, we can only state that $-D_{KL}(q(x|y)||p(x))$is a non-positive function of the approximate posterior parameters $\theta = \{\theta(t_i) = (\mu(t_i), \mathbf{\Sigma}(t_i)),\ t_i \in \Omega_t\}$. Let us call g this non-negative function. To match the generic formulation of the prior term in Equation 2, we consider the following form for $g(\theta)$:

$$g(\theta) = -\sum_{t_i \in \Omega_T} \|\Phi(\theta)(t_i) - \theta(t_i)\|^2, \tag{7}$$

where $\mathbf{\Phi}$ is an operator on time-series space.

This form is widely used in ML regularizing techniques experimented by Ryu et al. (2019); Venkatakrishnan et al. (2013) and referred to as plug-and-play methods for inverse problems. Besides, as detailed in Appendix C, we may note that Equation 6 may be rewritten in this form. Since the prior is left unspecified, $\mathbf{\Phi}$ is unknown, and we rely on an estimator $\tilde{\Phi}$ of $\mathbf{\Phi}$ to compute g. Overall, from the ELBO formulation, we infer the cost given by

$$U_{\mathbf{\Phi}}(\theta, y) = \sum_{t_i \in O_T} \|\mathbf{H}\mu(t_i) - y(t_i)\|_{\mathbf{R}} + \sum_{t_i \in \Omega_T} \|\tilde{\Phi}(\theta)(t_i) - \theta(t_i)\|^2. \tag{8}$$

As long as $\tilde{\Phi}$ is a valid approximation of $\mathbf{\Phi}$, the minimum of such a cost with respect to θ should be a good solution for the posterior approximation. Notice that Equation 8 can be viewed as a variational cost associated with an augmented state space formulation on the posterior parameters, which is why we call it stochastic variational cost.

3.2. Proposed Neural Architecture

Within a learning setting, the approximate posterior is parameterized by a set ω of weights and biases of a neural network (NN) framework, and is denoted $q_\omega(x|y)$. Additionally, let us give ourselves an initial state $\theta^{(0)}$ for the parameters of the posterior approximation, which depends on y. For example, we can choose as initial mean state the linear interpolation between available observations, and as initial covariance matrix the identity matrix. Then,

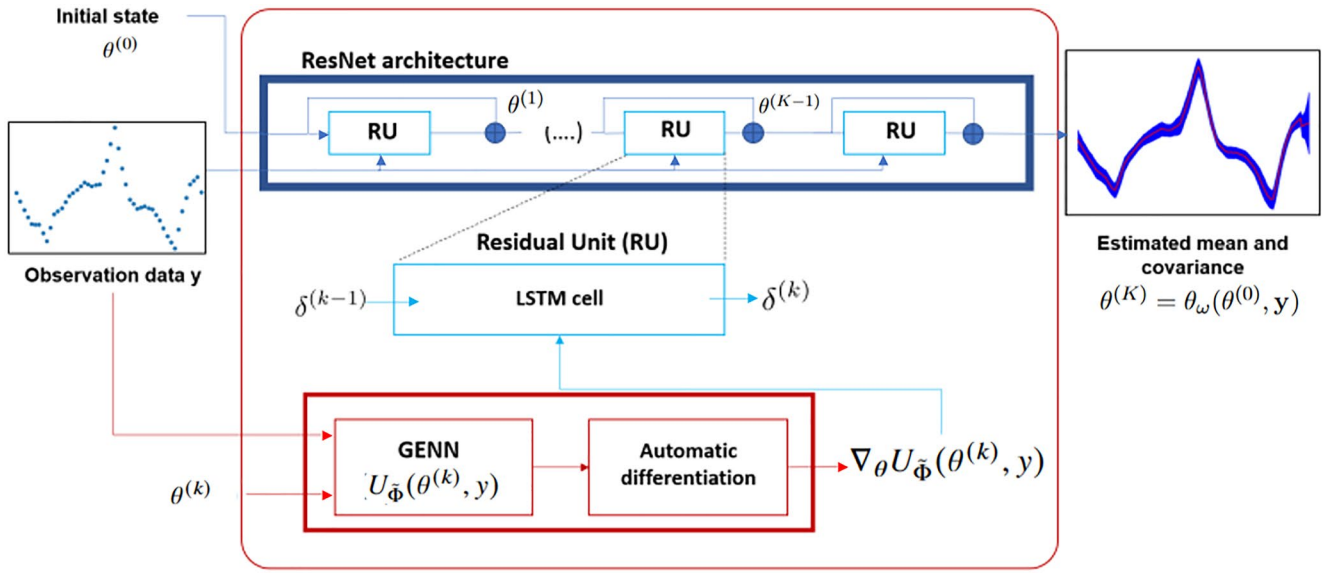

Figure 1. Proposed end-to-end architecture. Illustration comes from L63 experiment. Given a partial observation piece of data y and an initial pdf state $\theta^{(0)}$, the proposed network calculates the optimized parameters $\theta^{(K)}$ after K steps in the solver. On the right-hand side, red curve contains the mean state and the blue envelope is a rescaled visualization of the covariance. $\delta^{(k)}$ is the difference between the parameters at iteration step (k) and at iteration step $(k-1)$. GENN stands for Gibbs Energy NN and ResNet for residual network.

our approach takes as input the initial state $\theta^{(0)}$ and the observations y, and outputs the parameters of the target distribution. In our approach, θ, as defined in Section 3.1, is a function of ω, $\theta^{(0)}$ and y. We then write the output of our approach $\theta_\omega(\theta^{(0)}, y)$. Note that this implies that each $\mu(t_i)$ and $\mathbf{\Sigma}(t_i)$ are function of ω, $\theta^{(0)}$, and y. We make explicit the dependence on ω by noting in the following $\mu_\omega(t_i)$ and $\mathbf{\Sigma}_\omega(t_i)$. The set of parameters ω of the network are trained to optimize an inference score $\mathcal{S}(q_\omega(x|y), p(x|y))$, that we will detail in Section 3.3, which allows to estimate the proximity between the true posterior and its approximation by the target distribution.

The rest of this section is devoted to describing our architecture in Section 3.2.1 and the reasons why we chose it in Section 3.2.2.

3.2.1. Neural Set-Up

Our end-to-end approach is made of two key ingredients: a neural parameterization for the operator $\tilde{\Phi}$, and a trainable gradient-based solver of the stochastic variational cost defined in Equation 8. $\tilde{\Phi}$ is parameterized as a convolutional NN with specific constraints. The neural solver is a recurrent NN with stacked long short-term memory (LSTM) cells which implements a gradient-based solver for the targeted cost function. As our framework relies on two different components, remark that we can write $\omega = \{\omega_{\tilde{\Phi}}, \omega_s\}$ with $\omega_{\tilde{\Phi}}$ the NN parameters of $\tilde{\Phi}$ and ω_s the NN parameters of the solver. From a coding perspective, the proposed neural architecture was implemented using PyTorch framework. Figure 1 shows the working principle of our end-to-end architecture.

Architecture of $\tilde{\Phi}$. $\tilde{\Phi}$ is a convolutional NN with specific constraints, known as Gibbs Energy NN (Fablet, Beauchamp, et al., 2021). More precisely, we have $\tilde{\Phi}(\theta) = f_1 \circ f_2(\theta)$. f_2 is a convolutional layer where the central values of all convolution kernels are set to zero such that $f_2(\theta)(t_j)$ does not depend on $\theta(t_j)$. f_1 is a convolutional NN which composes a number of convolution layers with rectified linear unit activation, where the kernel size of all convolution layers is 1 along time and space dimensions. In the experiments, f_1 has 3 convolution layers.

Neural solver parametrization. The minimization with respect to θ of the stochastic variational cost (Equation 8) is performed by means of a neural solver. We use a residual NN architecture with LSTM blocks (Schmidhuber & Hochreiter, 1997). Each block is fed on one side with the increment between the estimated parameters at the entry of the block and the input parameters $\theta^{(0)}$, and on the other side by the gradient of the stochastic variational

cost with respect to θ applied on the current estimated parameters. This solver optimizes iteratively the estimated parameters. To be more explicit, after k iterations in the LSTM-based solver, the parameters are updated as follow:

$$\begin{cases} g^{(k+1)} = LSTM\left(\alpha \nabla_\theta U_{\tilde{\Phi}}\left(\theta^{(k)}, y\right), h^{(k)}, c^{(k)}\right), \\ \theta^{(k+1)} = \theta^{(k)} - \mathcal{L}\left(g^{(k+1)}\right), \end{cases}$$

with α a scalar parameter, $h^{(k)}$, $c^{(k)}$ internal states of the LSTM model and $\mathcal{L}$ a linear layer. The number of iterations in the LSTM-based solver has been tuned during experiments and optimal values are comprised between 10 and 20 iterations.

3.2.2. Motivation

Combination of $\tilde{\Phi}$ and the neural solver. Optimizing an inference score S can be very complex, so appropriately constraining the model is a fast and efficient solution to converge quickly to an optimum. We demonstrate in the developments of Section 3.1 that minimizing the cost of Equation 8 approximately equates to maximize the ELBO inference cost. The chosen architecture allows to constrain the model by making sure via the learned solver that the output $\theta_\omega(\theta^{(0)}, y)$ is close to a minimum of Equation 8. To summarize, we look for the best solution in the sense of inference among the suitable solutions in the sense of stochastic variational cost. The idea of a learned dynamical operator coupled with a learned neural solver was introduced in Fablet, Chapron, et al. (2021). As the formulation of Equation 8 is somehow similar to that considered in Fablet, Chapron, et al. (2021), we adapt their architecture to our case.

Choice of a Gibbs Energy NN for $\tilde{\Phi}$. From Equation 8, we note that the minimum of the stochastic variational cost with respect to $\tilde{\Phi}$ is reached whenever $\tilde{\Phi}$ is equal to the identity, whatever θ is. Letting $\tilde{\Phi}$ be equal to the identity suppresses the constraint corresponding to the second term on the right-hand side of Equation 8. Thus, $U_{\tilde{\Phi}}$ would become a function of $\mu(t_i)$ and y. Consequently, $\tilde{\Phi}$ would remain equal to Id, and covariance parameters would remain constant throughout the remainder of the training phase. This has to be prevent since we want to keep optimizing the covariance parameters during training. To this end, the Gibbs energy NN forces the convolutional NN to differ from the identity operator. Additionally, thanks to this constraint parametrization, $\tilde{\Phi}$ can be interpreted as a surrogate model over the mean and covariance parameters of the target distribution. Notice that other choices of NN representation could have been made, such as convolutional auto-encoder. For an intercomparison, we refer to Beauchamp et al. (2020).

Choice of a LSTM for the solver. NNs with LSTM cells belong to the class of recurrent NN. They are particularly suitable for processing sequential data. In our case, our working data is a sequence of time-space series $\theta^{(k)}$ obtained by gradient descent (see Equation 9). LSTM-based updates are the classical parameterization of learned solver schemes (see e.g., Andrychowicz et al., 2016; Hospedales et al., 2021).

3.3. Learning Setting

In our experimental setting, we have access during training stage to a data set of true states $\mathbf{x} = \{\mathbf{x}^{(i)}, 1 \leq i \leq m\}$, and corresponding observation data set $\mathbf{y} = \{\mathbf{y}^{(i)}, 1 \leq i \leq m\}$, with $\mathbf{x}^{(i)}$ and $\mathbf{y}^{(i)}$ realizations of the discretized setting given in Equation 1. The outputs of our approach $\theta_\omega(\theta^{(0)}, \mathbf{y}^{(i)})$ is composed of means and covariances denoted $\mu_\omega^{(i)}(t_j)$ and $\mathbf{\Sigma}_\omega^{(i)}(t_j)$ for $t_j \in \Omega_t$, where dependence on $\mathbf{y}^{(i)}$ is indicated by upper indices to keep the notation uncluttered. In this context, a commonly used method to evaluate the performed DA approach is the MSE. This criterion measures the distance in the mean square sense between the true state of the system and the average state predicted by the approach. In the case of our approach, this corresponds for a time series $\mathbf{x}^{(i)}$ to the following cost:

$$MSE\left(\mathbf{x}^{(i)}, \theta_\omega\left(\theta^{(0)}, \mathbf{y}^{(i)}\right)\right) = \frac{1}{N} \sum_{j=1}^{N} \|\mathbf{x}^{(i)}(t_j) - \mu_\omega^{(i)}(t_j)\|_2^2, \tag{9}$$

where $\|.\|_2$ is the Euclidean norm. The score of Equation 9 is denoted R-score in the following, which stands for reconstruction score.

However, this metric only allows us to compare the mean of the random vector $x|y$ with the mean of our approximated posterior. This is insufficient if we want to compare our posterior approximation with the whole true

posterior distribution. The right framework to assess statistical forecast is through proper scoring rule (Dawid & Musio, 2014; Gneiting & Raftery, 2007; Tsyplakov, 2013). A scoring rule is a function $S(q, \mathbf{x})$ of a pdf q and an outcome $\mathbf{x}$. By extension, we denote $S(q, p) = \mathbb{E}_{\mathbf{x}\sim p(x)} S(q, x)$. A scoring rule is, by definition, said to be proper if:

$$S(p, p) \geq S(q, p).$$

It is further strictly proper if the equality holds only for $q = p$.

Even if the distribution forecast depends on observations as in our approach, using a proper scoring rule is still consistent, as proved by Tsyplakov (2011) and Holzmann and Eulert (2014). In this context the logarithmic score defined by

$$S_{\log}(q, \mathbf{x}) = \log q(\mathbf{x}),$$

is a strictly proper scoring rule (Dawid & Musio, 2014). That is why we set our training objective L as the minimization of the opposite of the logarithmic score, which leads to:

$$\begin{aligned} \mathrm{L}\left(\mathbf{x}^{(i)}, \theta_\omega\left(\theta^{(0)}, \mathbf{y}^{(i)}\right)\right) &= -\frac{1}{N} S_{\log}\left(q_\omega\left(\cdot \mid \mathbf{y}^{(i)}\right),\ \mathbf{x}^{(i)}\right), \\ &= \frac{1}{2N} \sum_{j=1}^{N} \left(\|\mathbf{x}^{(i)}(t_j) - \mu_\omega^{(i)}(t_j)\|_{\mathbf{\Sigma}_\omega^{(i)}(t_j)} + \log|\mathbf{\Sigma}_\omega^{(i)}(t_j)| \right), \end{aligned} \tag{10}$$

where we have deliberately omitted the constant $\frac{n}{2}\log 2\pi$. We denote this criterion P-score for probabilistic score in the following. The P-score is also known as negative log-likelihood. Notice that the R-score and the P-score are proportional only when the covariance of the approximate posterior reduces to a constant scalar covariance matrix. The mean R-score and P-score over the whole data set $\mathbf{x}$ is given by averaging respectively Equations 9 and 10 over the m couples of true states and observations of the data sets $\mathbf{x}$ and $\mathbf{y}$.

The parameters ω of our network are optimized to minimize the P-score by the stochastic gradient descent Adam available in PyTorch. In our experimental learning setting, we set a batch size of 64 and a maximum number of 1,000 epochs. At predefined epochs, the learning rate is decreased. It ranges from 10^{-3} to 10^{-7}. The parameterization for which the P-score is the lowest on the validation data set is saved. We let the reader refer to the code available online (https://doi.org/10.5281/zenodo.7729564) for additional details on the implementation.

4. Numerical Experiments

To assess the relevance of the proposed approach, we consider two case-studies: namely, the Lorenz 63 dynamics and an application to a real data set corresponding to the monitoring of Danube river discharges. In the following, our approach will be referred to as 4D-VarnetSto. The baseline approach is the Ensemble Kalman Smoother and will be abbreviated as EnKS. The different approaches will be evaluated against two main criteria: the average P-score (Equation 10) and the average R-score (Equation 9) over the test data set.

4.1. L63 Dynamics

4.1.1. Standard L63 Dynamics

The Lorenz dynamics is a system made of the following ordinary differential equations (Lorenz, 1963):

$$\begin{aligned} \frac{dx_1}{dt} &= \sigma(x_2 - x_1), \\ \frac{dx_2}{dt} &= \rho x_1 - x_2 - x_1 x_3, \\ \frac{dx_3}{dt} &= x_1 x_2 - \beta x_3. \end{aligned} \tag{11}$$

We use the following parametrization: $\sigma = 8$, $\rho = 28$, and $\beta = \frac{8}{3}$. In this setup, the Lorenz 63 system has a chaotic dynamics. A fourth-order Runge-Kutta integration scheme (Butcher, 1996) with 0.01 time step enables us to simulate the time series. Figure 2a is a trajectory of this dynamics for 200 time steps.

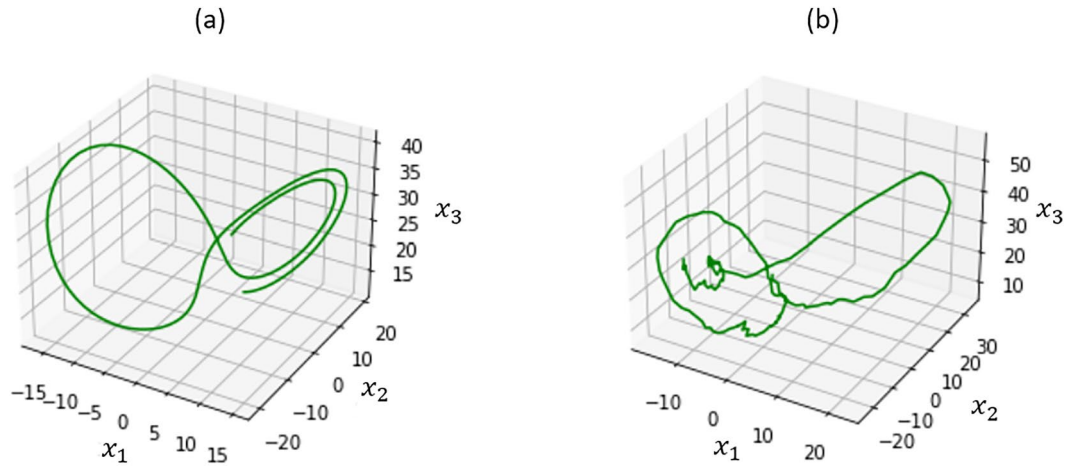

Figure 2. Evolution of Lorenz dynamics for (a) standard model (see Equation 11) and (b) stochastic model of Chapron et al. (2018) (Equation 12) for 200 time steps of 0.01 length each.

4.1.2. Stochastic L63 Dynamics

In order to introduce model noise in L63 dynamics, we use the stochastic framework designed by Chapron et al. (2018). It intends to mimic stochastic behavior in large-scale geophysical flow dynamics. The ordinary differential equation (Equation 11) becomes a stochastic differential equation:

$$\begin{aligned} dX_1 &= \left(\sigma(X_2 - X_1) - \frac{4}{2\Gamma}X_1\right)dt, \\ dX_2 &= \left(\rho X_1 - X_2 - X_1X_3 - \frac{4}{2\Gamma}\right)dt + \frac{\rho - X_3}{\Gamma^{\frac{1}{2}}}dB_t, \\ dX_3 &= \left(X_1X_2 - \beta X_3 - \frac{8}{2\Gamma}X_3\right)dt + \frac{X_2}{\Gamma^{\frac{1}{2}}}dB_t. \end{aligned} \tag{12}$$

dB_t is a white noise, formally the difference of a standard Brownian motion. Γ is the new parameter of our model which is fixed to 2 in our experiments. Note that if $\Gamma \to \infty$, we recover the original model. Figure 2b is a three-dimensional plot for a time series of this stochastic Lorenz 63 version. Adding the model noise strongly deteriorate the smoothness and the convergence to standard Lorenz attractor.

4.1.3. Training Setting and Results

For both dynamics, we consider a time series of 200,000 time steps. From this time series, we create a training set containing 10,000 sub-series of 200 time steps, and validation and test sets each consisting of 2,000 sub-series of 200 time steps. The sub-series overlap within a data set but do not overlap from one data set to another. Observations of the true state are made available solely for the first variable of the system, every 8 time steps, adding a white Gaussian observation noise of variance set to 2.

Including the parameters of the neural solver and those of $\tilde{\Phi}$, our network has roughly 19,000 parameters to learn. We train our NN in two stages. First, for each time series, the initial state $\theta^{(0)} = \{\mu^{(0)}\mathbf{\Sigma}^{(0)}\}$ is initialized as follows:

- $\mu^{(0)}$ is the linear interpolation between observations for its first variable and the mean of the observations for the other variables;
- $\mathbf{\Sigma}^{(0)}$ is the identity matrix.

We find a first optimum while constraining the estimated covariance matrix to be diagonal. In a second step, we start a new learning session to find a non-diagonal covariance matrix using the previously found optimum as initial state $\theta^{(0)}$. This two-step procedure aims to force the covariance matrix to be definite and positive during the training process. Imposing positive-definiteness directly on the whole output matrix is not an easy task while

Table 1
Scores of 4D-VarnetSto and Ensemble Kalman Smoother (EnKS) for L63 Simulations for Both Dynamics

Approach	Model noise	R-score	P-score
4D-VarnetSto with x_1 observed	No	1.35	−7.36
	Yes	10.53	−3.46
EnKS with x_1 observed	No	2.19	0.41
	Yes	17.32	15.26
EnKS with x_1 and x_2 observed	No	0.56	−4.25
	Yes	3.99	8.89
EnKS with all variables observed	No	0.24	−6.71
	Yes	2.81	10.21

Note. Model noise sets to "No" indicates standard dynamics (see Equation 11), "Yes" implies stochastic one (see Equation 12). Only the first variable is observed when performing 4D-VarnetSto. In EnKS experiments, from one to all variables are considered as observed. Two benchmark score are evaluated: the MSE of the reconstruction of the true state (R-score, see Equation 9), and the mean of the negative log-likelihood of the predicted parametric distribution applied in true state (P-score, see Equation 10).

in the diagonal covariance matrix case this is easy to enforce. Indeed, it only requires strictly positive values for the outputs on the diagonal, and zeros elsewhere. So first, we find an optimal diagonal covariance matrix, then we search for a complete covariance matrix by perturbing this optimum.

We compare our method with the EnKS of Evensen and Van Leeuwen (2000). In our experiment, the EnKS has 500 ensemble members and a time lag of 30 time units. No inflation is used. We have chosen a very large ensemble size because we want to be sure to correctly represent the approximation of the posterior made by the EnKS. Indeed, we mainly want to compare the quality of the approximation of the posterior made by the different approaches. For both dynamics, the EnKS is run through 20,000 time steps and evaluated on the last 15,000 time steps to be sure the calibration phase is over. Notice that in the stochastic dynamics case, the model error matrix of the EnKS is a diagonal matrix constant over time which coefficients are obtained by averaging the model error. Thus, in the stochastic case, we expect our approach to approximate the posterior far better than the EnKS as it does not rely on a imperfect model and an approximate model error matrix. Table 1 compiles the results for the appropriate scores. If the first variable is observed for both our approach and EnKS, the 4D-VarnetSto outperforms the EnKS in each score for both dynamics. By adding observed variables in EnKS experiment, the R-score and P-score decrease. For the standard dynamics, the R-score for the EnKS with at least two variables observed become lower than its value in the 4D-VarnetSto experiment, but the P-score stays above. This confirms that our posterior approximation is in any case better that the one proposed by the EnKS. As for the stochastic dynamics, the conclusion are rather similar. The R-score of our approach with one observed variable is better than the one of the EnKS. Again, regardless of the number of variables observed, the P-score is much lower using our approach than using the EnKS, and by even larger amounts than in the deterministic experiment. To conclude with the results of Table 1, we can state that in identical settings, our approach outperforms by far the EnKS in both criteria. Adding observed variables to the EnKS allows to obtain better R-score than our approach but the P-score stays above, which indicates that our approach is better suited for estimating the whole posterior than EnKS. As a side remark, our R-score is similar to the one reported by Fablet, Chapron, et al. (2021) (R-score of 1.34 in Table 1). This is a very good thing, as it indicates that adding complexity to their model does not deteriorate the quality of the state prediction.

Figure 3 compares estimated states (orange curve) and the associated 95% confidence interval (green area) with the real states (blue curve) defined by Equation 11 in the context of standard dynamics. Figure 4 presents the same elements for the stochastic dynamics defined by Equation 12. Both figures represent time series for which the attractor changes its wing. The change of wing is realized when the variables x_1 and x_2 simultaneously go from a maximum to a minimum or vice versa. In Figure 3, the mean state estimated by our approach (top three graphs) and the true state of the system are almost merged. Moreover, the area representing the uncertainty is also very thin but widens for a given variable when an extremum is reached. The uncertainty is slightly larger for the unobserved variables x_2 and x_3 than for the observed variable x_1. Comparatively, the state reconstructed by EnKS when only x_1 is observed (middle three graphs) coincides less well with the true state. The uncertainty is also larger, especially during the wing change (between $t = 50$ and 125). When the three variables are observed for the EnKS (bottom three graphs), the real state and the reconstructed state are difficult to distinguish, the area representing the uncertainty is very narrow and widens slightly during the wing change. In Figure 4, we first note that the EnKS with only x_1 observed performs poorly. It does not succeed in correctly representing the dynamics (middle three graphs). When observing the three variables for the EnKS (bottom three graphs), the estimated state becomes accurate. However, the true state curve is almost never contained within the confidence interval. This visually confirms the poor results obtained on the P-score and indicate that the posterior approximation is not accurate. On the contrary, we observe that the confidence interval estimated by our approach seems consistent (top three graphs). The true state curve is globally contained within a fairly narrow confidence interval.

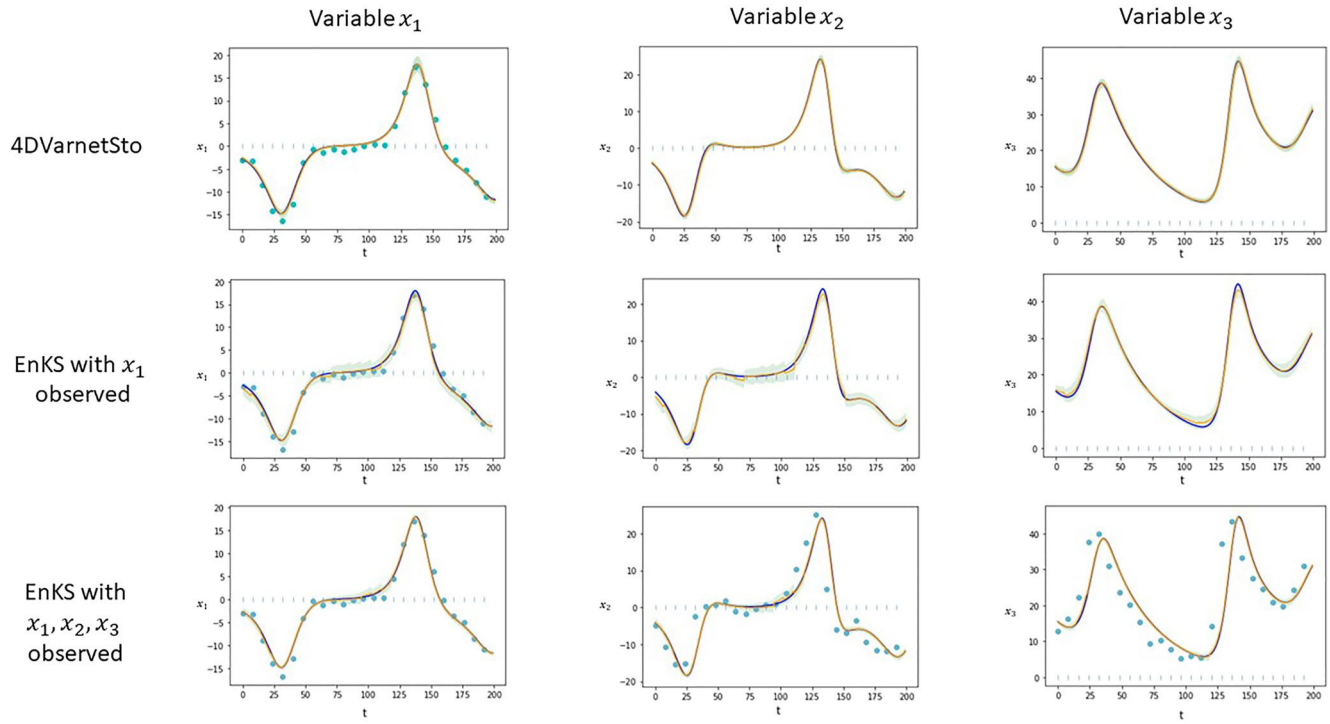

Figure 3. Experiments with standard Lorenz dynamics (Equation 11). For a set of observations (cyan dots) on given timesteps (light blue dashes on the time axis), the true state (blue curve) and estimated state (orange curve) are plotted for our approach and Ensemble Kalman Smoother (EnKS) with one or all variables observed. The estimated 95% confidence intervals are represented by the green area.

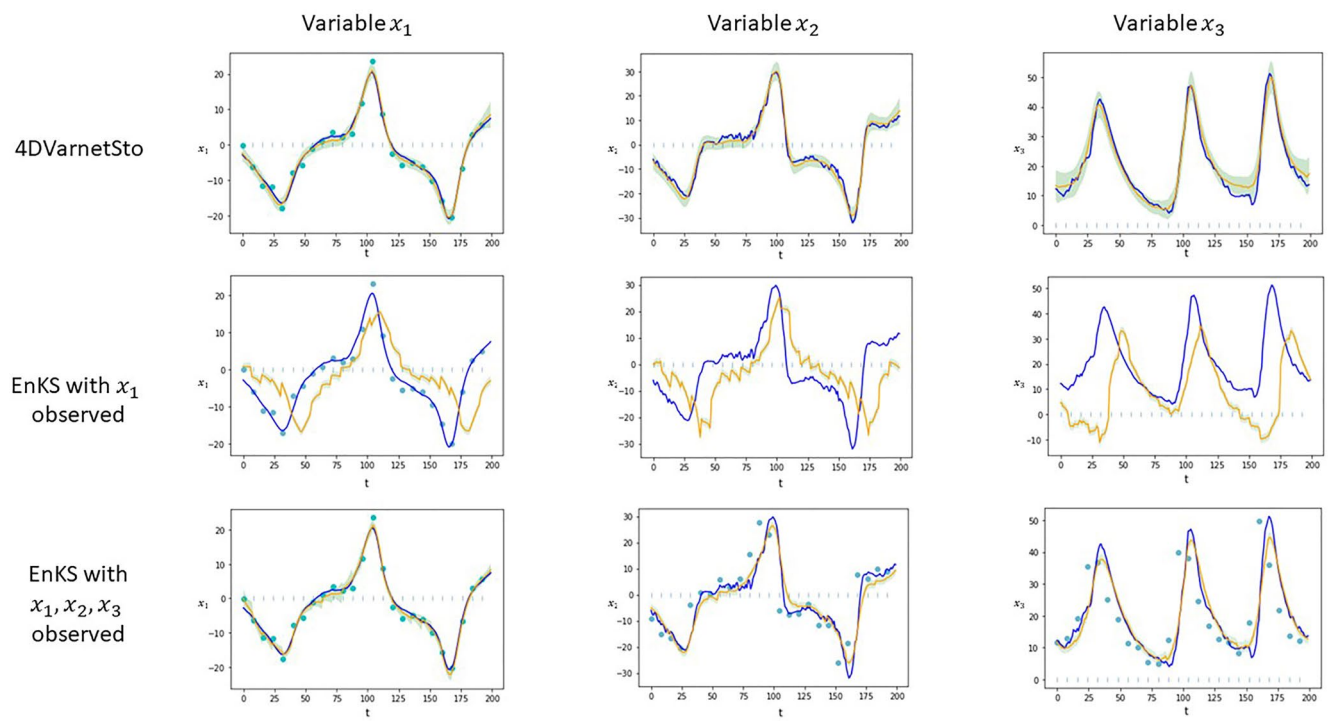

Figure 4. Experiments with the stochastic Lorenz dynamics of Chapron et al. (2018) (Equation 12). See Figure 3 for details.

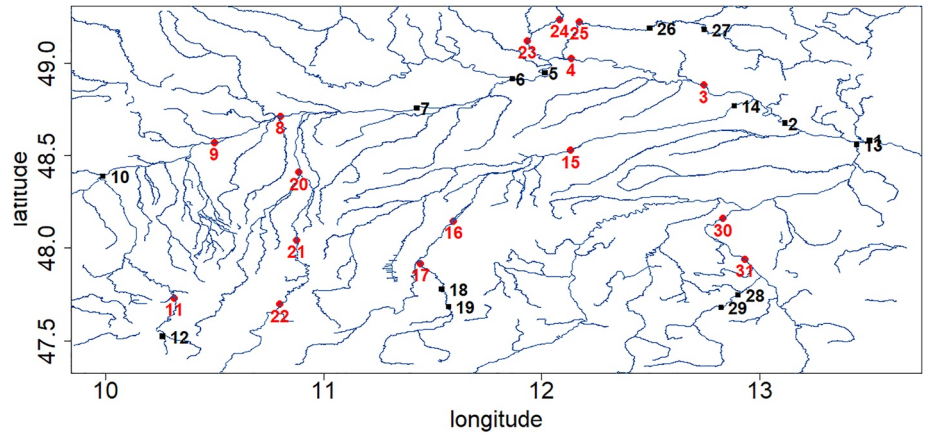

Figure 5. Topographic map of the upper Danube basin with the 31 gauging stations. A data set of 50 years of daily measurements is considered (from 1960 to 2010). In training setting, we assume that some stations are observed (red dots) and the other are unobserved (black squares). We further assume that the observed stations have available observations only once every 4 days.

4.2. Danube River Network for Discharge Measurements

The upper Danube basin is an European river network whose drainage basin covers a large part of Austria, Switzerland and of the south of Germany. Figure 5 shows the topography of the Danube basin as well as the locations of the 31 stations at which daily measurements of river discharge are available. Stations considered as observed or unobserved in our experiment are colored differently. The daily measurements series have lengths from 51 to 110 years. We restrict ourselves to the period 1960–2010 for which all stations have available measurements. This data set have been widely studied in the community of multivariate extremes (see for example Asadi et al., 2015; Mhalla et al., 2020).

This experiment with a real data set aims to meet several objectives. Learning an unknown dynamics and associated uncertainties is challenging. The data-driven models that can be learned lacks important variables (precipitation, snowmelt) to be highly reliable, and consequently encompass high error model. Thus, the ability of our approach to adapt to a high level of model error is studied. Finally, the approximation of the posterior made by our approach is compared to a Gaussian approximation which we call constant covariance approach. In this comparative approach, the mean state is estimated using the approach described in Fablet, Chapron, et al. (2021), and the covariance matrix is a diagonal matrix whose coefficients are constants and set as the variance of the error at each station.

In this experiment, we consider that the observed data correspond to the state of the system. It is equivalent to consider no observation noise, namely $\mathbf{y}^{(i)} = \mathbf{x}^{(i)}_{|O_T}$ for each i, where $\mathbf{x}^{(i)}_{|O_T}$ is the restriction of $\mathbf{x}^{(i)}$ to O_T. To avoid divergence of the P-score on the set of observations O_t, we modify the P-score slightly by redefining it as follows:

$$\mathrm{L}\left(\mathbf{x}^{(i)}, \theta_\omega\left(\theta^{(0)}, \mathbf{y}^{(i)}\right)\right) = \frac{1}{2N_t} \sum_{t_j \in \Omega_T \setminus O_T} \left(\|\mathbf{x}^{(i)}(t_j) - \mu^{(i)}_\omega(t_j)\|_{\Sigma^{(i)}_\omega(t_j)} + \log|\Sigma^{(i)}_\omega(t_j)| \right), \tag{13}$$

where N_t is the cardinal of $\Omega_T \setminus O_T$. Given the spatial dimension of the state, we limit ourselves to output diagonal covariance matrix. Consequently, our NN is trained using only the first step of the process described in Section 4.1.3. The initial state $\theta^{(0)}$ is also defined as described in this first step. In order to leave the stochastic variational cost defined, we set $\mathbf{R}$ to the identity in Equation 8. Using the criterion of Equation 13, half of the stations are considered to be observed every 4 days (see red locations in Figure 5). We consider time series of 48 consecutive days. For each time series, our goal is to estimate the mean and covariance of the approximate posterior distribution of flow on each day of the time series and at each station, including where observations are missing. The training data set comprises 9,999 time series of 48 days, validation and test set 1,749 each. To construct these data sets, we divided the 51 years of daily measurement into 550-day blocks. In each block, the first 350 days create 303 time series for the training data set. The 200 remaining days are divided in two and

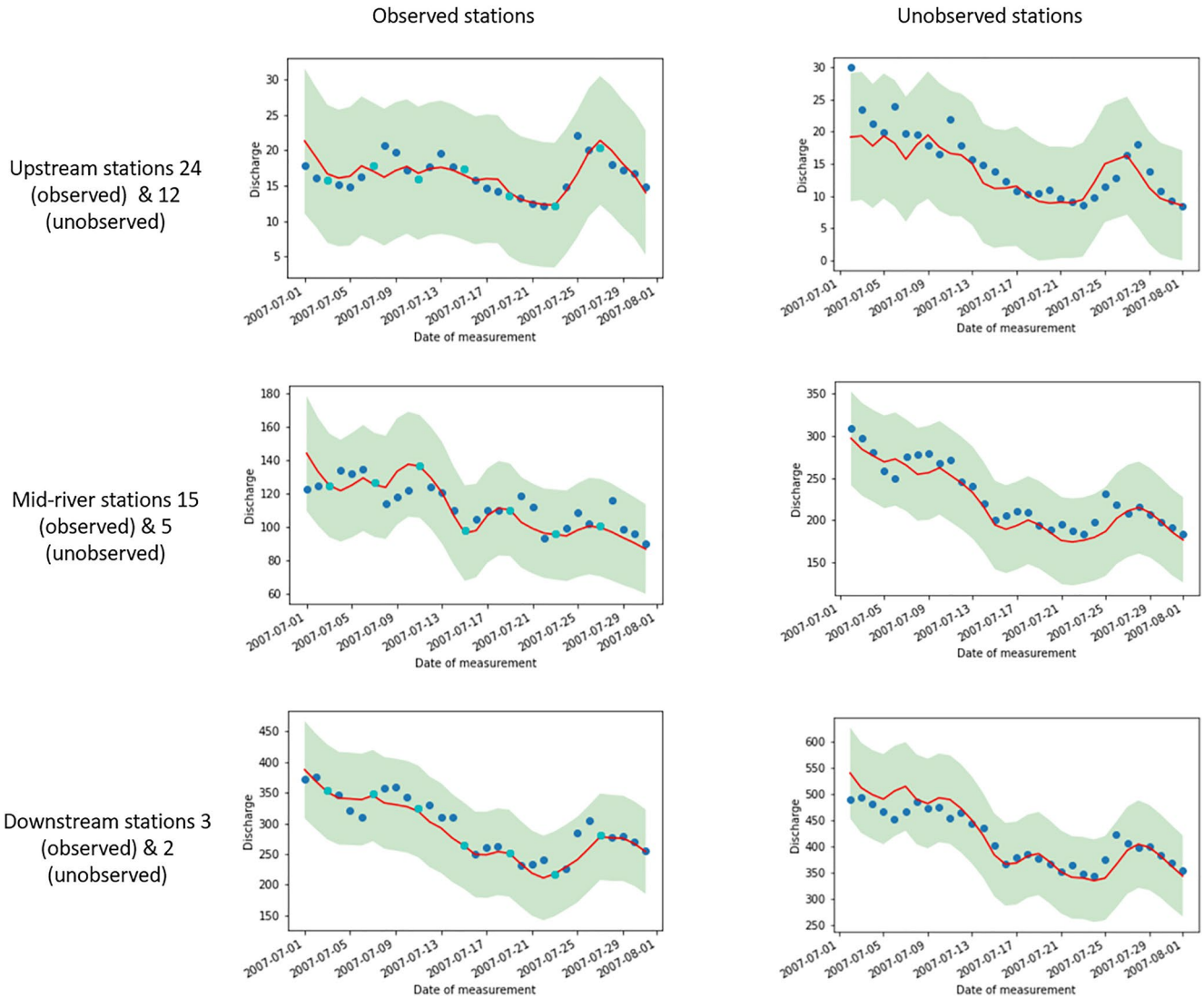

Figure 6. For a summer month (July 2007), we show the estimated discharge (red curve), the 95% confidence interval (green area) estimated by our method for observed and unobserved stations at different elevations. The daily measurements are also represented according to whether they are available (light blue dots) or unavailable(deeper blue) as inputs. The discharges are expressed in m^3/s.

create 53 time series for validation set and as many for the test set. Note that within a data set, time series are overlapping. Figures 6 and 7 show the estimated mean state (red curve), the confidence interval (green area) and the daily measurements (blue dots) for a summer and winter month, respectively. The stations are identical from one figure to another.

Seasonality plays an important role in discharge analysis, and here, we focus on the summer and winter seasons. In summer, flows are lower than in winter and subject to important variations in absolute value. This is linked essentially to snow or ice melts at altitude, as well as to episodes of heavy precipitation. For similar reasons, different station elevations, and thus different positions along the river system, were chosen. Stations upstream of the river system have lower flows than those downstream. Flows at upstream stations vary greatly depending on local weather and climate events.

The relative variance estimated by our approach is larger in Figure 6 than in Figure 7. This finding is consistent with the initial considerations about variances in summer and winter. The estimated variance is also more constant in summer than in winter. One can assume that the model error is such that it becomes difficult to detect patterns that would reduce the uncertainty. In winter, on the other hand, the estimated confidence interval varies significantly,

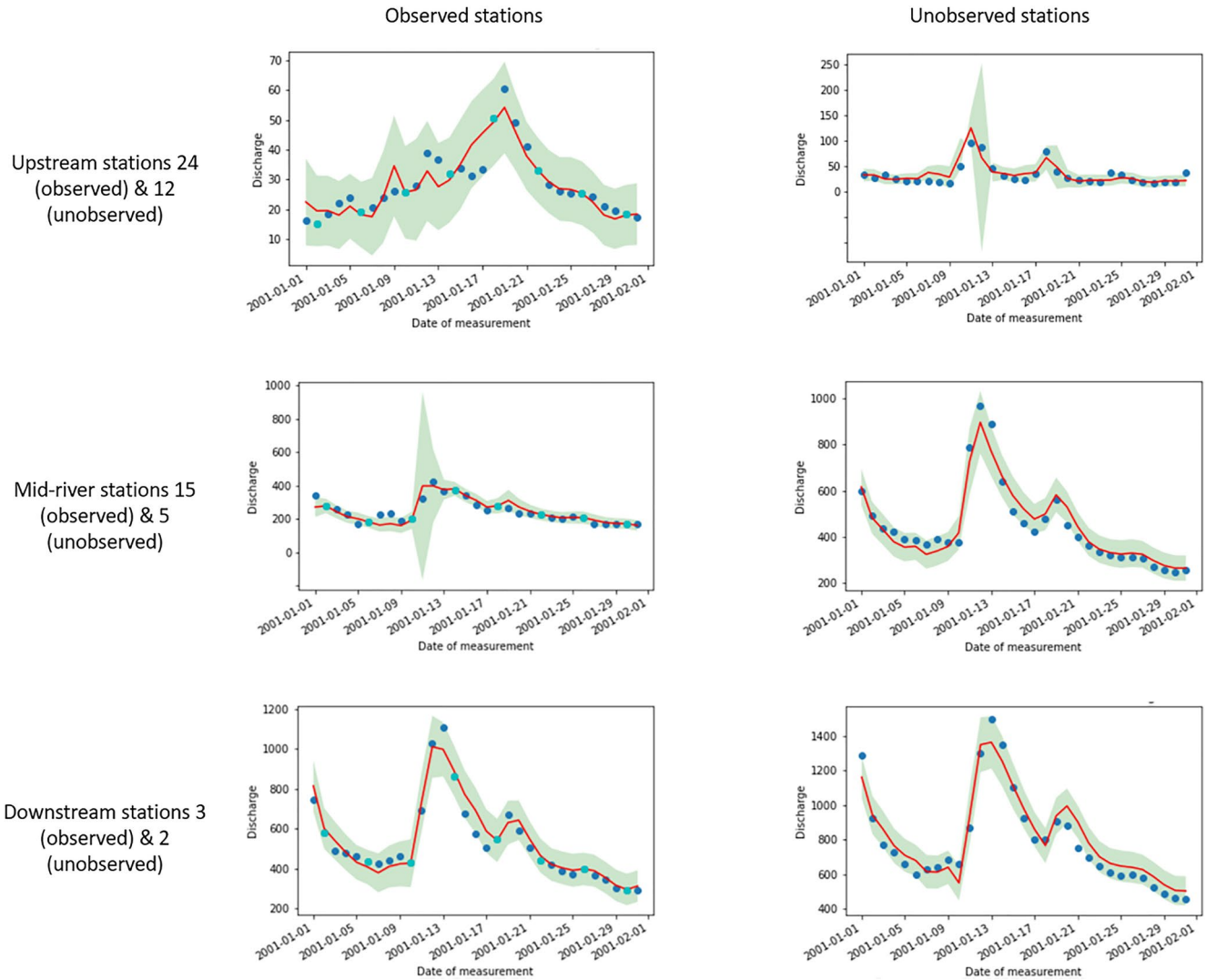

Figure 7. Winter month (January 2000) (see Figure 6 for details).

and seems to widen at the peaks reached by the flow. We notice that our predictions are sometimes biased for a large number of consecutive time steps. This is particularly true in Figure 7 where a negative bias between the observations and the predicted mean exist. It is visible for downstream and mid-river unobserved stations between 21 January 2001 and 1 February 2001. The presence of available observations drastically reduces the bias.

In order to compare our approach with the constant covariance approach, we average the P-score and R-score restricted to $\Omega_t \backslash O_t$ over the test data set, for both our approach and the comparative constant covariance approach. As the discharges at different stations have not the same order of magnitude, we rescaled the discharges at each stations to a time series with mean 0 and standard deviation sets to 1 before training both approaches. The scores for the rescaled discharges are given in Table 2. We find that estimating the covariance in addition to the mean state does not degrade the R-score. Indeed the R-score obtained by our approach and by the constant covariance approach are almost identical. Moreover, we significantly improve the P-score over constant covariance approach and we can infer that the variations of variances given by our approach allow a significant improvement of posterior approximation.

Table 2
Scores of 4D-VarnetSto and Constant Covariance Approach for Rescaled Danube River Discharges

Approach	R-score	P-score
4D-VarnetSto	3.4	−0.018
Constant covariance	3.38	1.05

Note. Two benchmark scores are evaluated: the R-score and P-score on unobserved time steps average on test data set.

5. Conclusion

Based on previous works which introduced an end-to-end learning framework for variational assimilation problems, we extend this approach to uncertainty quantification. Using a stochastic variational cost derived from an ELBO maximization with respect to a target Gaussian distribution, we have been able to find a Gaussian approximation of the pdf of the posterior. The learning framework comprises a neural-network representation of the dynamics of the parameters and a neural solver for the considered stochastic variational cost. Both solver and dynamics of the parameters are learnt jointly in a context of logarithmic score optimization. This joint learning process offers new perspectives for VB-based cost minimization in DA problems.

Lorenz 63 dynamics and discharges on Danube river networks have been studied. As regards the Lorenz dynamics, our approach captures well the dynamics and the uncertainty. When adding state-dependent model noise, we have been able to retrieve complex type of uncertainty structure. The experiments on the Danube river system provide a setting where the dynamics are unknown, and the data to estimate them incomplete. In this context, our approach allows us to calculate a consistent estimate of the flow, the associated dynamics and the uncertainties.

Our findings also underlines that beyond state-of-the-art results obtained for MSE of reconstruction, our approach is well-suited for logarithmic score. This is a real improvement over reference ensemble methods which suffer from limitations and require careful adaptation to achieve good performance on such score. This indicates that posterior approximation reached with our approach is more consistent than those provide by ensemble methods.

We claim that our approach could be applicable to problems of higher dimension thanks to the versatility of NNs, which could constitute interesting fields of application. Besides, future works will also focus on improving the accuracy of the upper quantile of the predicted distribution. A parameterization of the posterior by heavy tail distribution (see e.g., Resnick, 2007) could be an improvement track. Moreover, as discharges are positive values, a Gaussian parametrization is not ideal to infer uncertainties. More broadly, symmetrical distribution cannot consistently estimate large uncertainty in this problem as it could cover negative flow value. Extending our approach to non-symmetrical distribution would be of interest.

Finally, one limitation of our approach is the need for a data set of true states, which is generally not possible in practice. Thus, there is still significant room for further progress with respect to the application of such approach in operational settings.

Appendix A: Mahalanobis Norm

Given a vector z of dimension n and a positive-definite matrix $\mathbf{A}$ of dimension $n \times n$, the Mahalanobis norm of z is denoted $\|z\|_{\mathbf{A}}$ and is given by

$$\|z\|_{\mathbf{A}} = z^T \mathbf{A}^{-1} z.$$

Appendix B: Proof of Equation 5

We first state an important result. Let $p(x) = \mathcal{N}(x\ ;\ m, \mathbf{\Sigma})$ be the pdf of a multivariate Gaussian. For any matrix $\mathbf{A}$, we have (see Petersen & Pedersen, 2008, Section 8),

$$\mathbb{E}_{x\sim p}\left[x^T \mathbf{A} x\right] = \mathrm{Tr}(\mathbf{A\Sigma}) + m^T \mathbf{A} m. \tag{B1}$$

Let $q(x) = \mathcal{N}(x\ ;\ \mu, \mathbf{\Sigma})$ and $p(y|x) = \mathcal{N}(y\ ;\ \mathbf{H}x, \mathbf{R})$. Then, we have

$$\begin{aligned}
\mathbb{E}_{x\sim q} \log(p(y|x)) &= \mathbb{E}_{x\sim q}\left[\log\left(\frac{1}{\sqrt{(2\pi)^n |\mathbf{R}|}} \exp -\frac{1}{2}(Hx - y)^T \mathbf{R^{-1}} (Hx - y)\right)\right], \\
&= -\log\left(\sqrt{(2\pi)^n |\mathbf{R}|}\right) - \frac{1}{2}\mathbb{E}_{x\sim q}\left[(\mathbf{H}x - y)^T \mathbf{R^{-1}} (\mathbf{H}x - y)\right], \\
&= -\log\left(\sqrt{(2\pi)^n |\mathbf{R}|}\right) - \frac{1}{2} y^T \mathbf{R^{-1}} y + y^T \mathbf{R^{-1}} \mathbf{H}\mu - \frac{1}{2}\mathbb{E}_{x\sim q}\left[x^T \mathbf{H}^T \mathbf{R^{-1}} \mathbf{H} x\right].
\end{aligned}$$

From Equation B1, we obtain

$$\mathbb{E}_{x\sim q}\log(p(y|x)) = f(\mathbf{R}) - \frac{1}{2}y^T\mathbf{R}^{-1}y + y^T\mathbf{R}^{-1}\mathbf{H}\mu - \frac{1}{2}\left(\mu^T\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mu + \mathrm{Tr}\left(\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{\Sigma}\right)\right),$$
$$= f(\mathbf{R}) - \frac{1}{2}\left(\mathrm{Tr}\left(\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{\Sigma}\right) + (y-\mathbf{H}\mu)^T\mathbf{R}^{-1}(y-\mathbf{H}\mu)\right).$$

where $f(\mathbf{R}) = -\frac{1}{2}(d\log 2\pi + \log|\mathbf{R}|)$. Equation 5 follows.

Appendix C: Proof of Equation 7

We consider the following norm on the space spanned by $\theta = (\mu, \mathbf{\Sigma})$:

$$\|(\mu,\mathbf{\Sigma})\| = \|\mu\|_2 + \left(\mathrm{Tr}\left(\mathbf{\Sigma}^2\right)\right)^{\frac{1}{2}},$$

where $\|.\|_2$ is the Euclidean norm. Then, given

$$g(\theta) = -\frac{1}{2}\left(\mathrm{Tr}\left(\mathbf{S}^{-1}\Sigma\right) + \|\mu - m\|_S^2 + \log\left(\frac{|\mathbf{S}|}{|\mathbf{\Sigma}|}\right)\right),$$

we obtain $g(\theta) = -\|\Phi(\theta) - \theta\|^2$ if we consider the following expression for Φ:

$$\Phi(\mu,\ \mathbf{\Sigma}) = \left(\mathbf{L}(\mu - m) + \mu,\ \frac{1}{d}\left(\mathrm{Tr}\left(\mathbf{S}^{-1}\mathbf{\Sigma}\right) + \log\left(\frac{|\mathbf{S}|}{|\mathbf{\Sigma}|}\right)\right)\mathrm{Id} + \mathbf{\Sigma}\right),$$

with $\mathbf{L}$ such that $\mathbf{L}^2 = \mathbf{S}^{-1}(\mu - m)(\mu - m)^T\mathbf{S}^{-1}$.

Extending this result, it proves that Equation 6 can be written in the form of Equation 7.

Data Availability Statement

We provide our associated code available at https://doi.org/10.5281/zenodo.7729564 which comprises the generation of the synthetic data sets. Danube river network data set is make available by the Bavarian Environmental Agency at http://www.gkd.bayern.de.

Acknowledgments
This work was supported by the French Agence Nationale de la Recherche (ANR) under reference ANR-Melody (ANR-19-CE46-0011). Part of this work was supported by 80 PRIME CNRS-INSU, ANR-20-CE40-0025-01 (T-REX project), and the European H2020 XAIDA (Grant 101003469). Our warmest thanks go to the reviewers, who provided insightful comments throughout the revision process. They have greatly contributed to improving the paper.

References

Andrychowicz, M., Denil, M., Gomez, S., Hoffman, M. W., Pfau, D., Schaul, T., et al. (2016). Learning to learn by gradient descent by gradient descent. *Advances in Neural Information Processing Systems*, *29*, 3988–3996.

Asadi, P., Davison, A. C., & Engelke, S. (2015). Extremes on river networks. *Annals of Applied Statistics*, *9*(4), 2023–2050. https://doi.org/10.1214/15-aoas863

Beauchamp, M., Fablet, R., Ubelmann, C., Ballarotta, M., & Chapron, B. (2020). Intercomparison of data-driven and learning-based interpolations of along-track nadir and wide-swath SWOT altimetry observations. *Remote Sensing*, *12*(22), 3806. https://doi.org/10.3390/rs12223806

Belanger, P. R. (1974). Estimation of noise covariance matrices for a linear time-varying stochastic process. *Automatica*, *10*(3), 267–275. https://doi.org/10.1016/0005-1098(74)90037-5

Blei, D. M., & Jordan, M. I. (2006). *Variational inference for Dirichlet process mixtures*. Bayesian Analysis.

Bocquet, M., Brajard, J., Carrassi, A., & Bertino, L. (2020). Bayesian inference of chaotic dynamics by merging data assimilation, machine learning and expectation-maximization. *arXiv preprint arXiv:2001.06270*, *2*(1), 55–80. https://doi.org/10.3934/fods.2020004

Butcher, J. C. (1996). A history of Runge-Kutta methods. *Applied Numerical Mathematics*, *20*(3), 247–260. https://doi.org/10.1016/0168-9274(95)00108-5

Chapron, B., Dérian, P., Mémin, E., & Resseguier, V. (2018). Large-scale flows under location uncertainty: A consistent stochastic framework. *Quarterly Journal of the Royal Meteorological Society*, *144*(710), 251–260. https://doi.org/10.1002/qj.3198

Dawid, A. P., & Musio, M. (2014). Theory and applications of proper scoring rules. *Metron*, *72*(2), 169–183. https://doi.org/10.1007/s40300-014-0039-y

de Bézenac, E., Rangapuram, S. S., Benidis, K., Bohlke-Schneider, M., Kurle, R., Stella, L., et al. (2020). Normalizing Kalman filters for multivariate time series analysis. *Advances in Neural Information Processing Systems*, *33*, 2995–3007.

Evensen, G. (1994). Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *Journal of Geophysical Research*, *99*(C5), 10143–10162. https://doi.org/10.1029/94jc00572

Evensen, G. (2003). The ensemble Kalman filter: Theoretical formulation and practical implementation. *Ocean Dynamics*, *53*(4), 343–367. https://doi.org/10.1007/s10236-003-0036-9

Evensen, G. (2009). *Data assimilation: The ensemble Kalman filter* (Vol. 2). Springer.

Evensen, G., & Van Leeuwen, P. J. (2000). An ensemble Kalman smoother for nonlinear dynamics. *Monthly Weather Review*, *128*(6), 1852–1867. https://doi.org/10.1175/1520-0493(2000)128<1852:aeksfn>2.0.co;2

Evensen, G., Vossepoel, F. C., & van Leeuwen, P. J. (2022). *Data assimilation fundamentals: A unified formulation of the state and parameter estimation problem*. Springer Nature.

Fablet, R., Beauchamp, M., Drumetz, L., & Rousseau, F. (2021). Joint interpolation and representation learning for irregularly sampled satellite-derived geophysical fields. *Frontiers in Applied Mathematics and Statistics*, *7*, 655224. https://doi.org/10.3389/fams.2021.655224

Fablet, R., Chapron, B., Drumetz, L., Mémin, E., Pannekoucke, O., & Rousseau, F. (2021). Learning variational data assimilation models and solvers. *Journal of Advances in Modeling Earth Systems*, *13*(10), e2021MS002572. https://doi.org/10.1029/2021ms002572

Fablet, R., Ouala, S., & Herzet, C. (2018). Bilinear residual neural network for the identification and forecasting of geophysical dynamics. In *2018 26th European Signal Processing Conference (EUSIPCO)* (pp. 1477–1481). IEEE.

Farchi, A., Laloyaux, P., Bonavita, M., & Bocquet, M. (2021). Using machine learning to correct model error in data assimilation and forecast applications. *Quarterly Journal of the Royal Meteorological Society*, *147*(739), 3067–3084. https://doi.org/10.1002/qj.4116

Gneiting, T., & Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, *102*(477), 359–378. https://doi.org/10.1198/016214506000001437

Gordon, N. J., Salmond, D. J., & Smith, A. F. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F-Radar and Signal Processing*, *140*(2), 107–113. https://doi.org/10.1049/ip-f-2.1993.0015

Hamill, T. M., & Whitaker, J. S. (2005). Accounting for the error due to unresolved scales in ensemble data assimilation: A comparison of different approaches. *Monthly Weather Review*, *133*(11), 3132–3147. https://doi.org/10.1175/mwr3020.1

Hoffman, M. D., Blei, D. M., Wang, C., & Paisley, J. (2013). Stochastic variational inference. *Journal of Machine Learning Research*, *14*, 1303–1347.

Holzmann, H., & Eulert, M. (2014). The role of the information set for forecasting—With applications to risk management. *The Annals of Applied Statistics*, 595–621.

Hospedales, T., Antoniou, A., Micaelli, P., & Storkey, A. (2021). Meta-learning in neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(9), 5149–5169. https://doi.org/10.1109/tpami.2021.3079209

Isaksen, L., Bonavita, M., Buizza, R., Fisher, M., Haseler, J., Leutbecher, M., & Raynaud, L. (2010). *Ensemble of data assimilations at ECMWF* (Technical Memorandum 636). ECMWF.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., & Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine Learning*, *37*(2), 183–233. https://doi.org/10.1023/a:1007665907178

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, *25*.

Le, Q. V. (2013). Building high-level features using large scale unsupervised learning. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 8595–8598). IEEE.

Le Dimet, F.-X., & Talagrand, O. (1986). Variational algorithms for analysis and assimilation of meteorological observations: Theoretical aspects. *Tellus A: Dynamic Meteorology and Oceanography*, *38*(2), 97–110. https://doi.org/10.3402/tellusa.v38i2.11706

Long, Z., Lu, Y., Ma, X., & Dong, B. (2018). PDE-Net: Learning PDES from data. In *International Conference on Machine Learning* (pp. 3208–3216). PMLR.

Lorenz, E. N. (1963). Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, *20*(2), 130–141. https://doi.org/10.1175/1520-0469(1963)020<0130:dnf>2.0.co;2

Machenhauer, B., & Kirchner, I. (2000). Diagnosis of systematic initial tendency errors in the ECHAM AGCM using slow normal mode data assimilation of ECMWF reanalysis data. *CLIVAR Exchanges*, *5*(4), 9–10.

Mhalla, L., Chavez-Demoulin, V., & Dupuis, D. J. (2020). Causal mechanism of extreme river discharges in the upper Danube basin network. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, *69*(4), 741–764. https://doi.org/10.1111/rssc.12415

Mohan, A. T., Lubbers, N., Chertkov, M., & Livescu, D. (2023). Embedding hard physical constraints in neural network coarse-graining of three-dimensional turbulence. *Physical Review Fluids*, *8*(1), 014604.

Mohsan, M., Vardon, P. J., & Vossepoel, F. C. (2021). On the use of different constitutive models in data assimilation for slope stability. *Computers and Geotechnics*, *138*, 104332. https://doi.org/10.1016/j.compgeo.2021.104332

Nonnenmacher, M., & Greenberg, D. S. (2021). Deep emulators for differentiation, forecasting, and parametrization in Earth science simulators. *Journal of Advances in Modeling Earth Systems*, *13*(7), e2021MS002554. https://doi.org/10.1029/2021ms002554

Otter, D. W., Medina, J. R., & Kalita, J. K. (2020). A survey of the usages of deep learning for natural language processing. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(2), 604–624. https://doi.org/10.1109/tnnls.2020.2979670

Petersen, K. B., & Pedersen, M. S. (2008). The matrix cookbook. *Technical University of Denmark*, *7*(15), 510.

Rabier, F., Järvinen, H., Klinker, E., Mahfouf, J.-F., & Simmons, A. (2000). The ECMWF operational implementation of four-dimensional variational assimilation. I: Experimental results with simplified physics. *Quarterly Journal of the Royal Meteorological Society*, *126*(564), 1143–1170. https://doi.org/10.1002/qj.49712656415

Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, *378*, 686–707.

Resnick, S. I. (2007). *Heavy-tail phenomena: Probabilistic and statistical modeling*. Springer Science and Business Media.

Ryu, E., Liu, J., Wang, S., Chen, X., Wang, Z., & Yin, W. (2019). Plug-and-play methods provably converge with properly trained denoisers. In *International Conference on Machine Learning* (pp. 5546–5557). PMLR.

Sasaki, Y. (1970). Some basic formalisms in numerical variational analysis. *Monthly Weather Review*, *98*(12), 875–883. https://doi.org/10.1175/1520-0493(1970)098<0875:sbfinv>2.3.co;2

Scher, S., & Messori, G. (2019). Generalization properties of feed-forward neural networks trained on Lorenz systems. *Nonlinear Processes in Geophysics*, *26*(4), 381–399. https://doi.org/10.5194/npg-26-381-2019

Schmidhuber, J., & Hochreiter, S. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Silva, V. L. S., Heaney, C. E., & Pain, C. C. (2023). Generative network-based reduced-order model for prediction, data assimilation and uncertainty quantification. In *LatinX in AI Workshop at ICML 2023 (Regular Deadline)*.

Snyder, C., Bengtsson, T., Bickel, P., & Anderson, J. (2008). Obstacles to high-dimensional particle filtering. *Monthly Weather Review*, *136*(12), 4629–4640. https://doi.org/10.1175/2008mwr2529.1

Stroud, J. R., Katzfuss, M., & Wikle, C. K. (2018). A Bayesian adaptive ensemble Kalman filter for sequential state and parameter estimation. *Monthly Weather Review*, *146*(1), 373–386. https://doi.org/10.1175/mwr-d-16-0427.1

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems*, *27*, 3104–3112.

Tandeo, P., Ailliot, P., Bocquet, M., Carrassi, A., Miyoshi, T., Pulido, M., & Zhen, Y. (2018). Joint estimation of model and observation error covariance matrices in data assimilation: A review. Colloque national d'assimilation de données.

Tandeo, P., Pulido, M., & Lott, F. (2015). Offline parameter estimation using EnKF and maximum likelihood error covariance estimates: Application to a subgrid-scale orography parametrization. *Quarterly Journal of the Royal Meteorological Society*, *141*(687), 383–395. https://doi.org/10.1002/qj.2357

Trémolet, Y. (2006). Accounting for an imperfect model in 4D-Var. *Quarterly Journal of the Royal Meteorological Society: A Journal of the Atmospheric Sciences, Applied Meteorology and Physical Oceanography*, *132*(621), 2483–2504. https://doi.org/10.1256/qj.05.224

Trémolet, Y. (2007). Model-error estimation in 4D-Var. *Quarterly Journal of the Royal Meteorological Society: A Journal of the Atmospheric Sciences, Applied Meteorology and Physical Oceanography*, *133*(626), 1267–1280. https://doi.org/10.1002/qj.94

Tsyplakov, A. (2011). Evaluating density forecasts: A comment. University Library of Munich. Available at SSRN 1907799.

Tsyplakov, A. (2013). Evaluation of probabilistic forecasts: Proper scoring rules and moments. University Library of Munich. Available at SSRN 2236605.

Van Leeuwen, P. J. (2009). Particle filtering in geophysical systems. *Monthly Weather Review*, *137*(12), 4089–4114. https://doi.org/10.1175/2009mwr2835.1

Venkatakrishnan, S. V., Bouman, C. A., & Wohlberg, B. (2013). Plug-and-play priors for model based reconstruction. In *2013 IEEE Global Conference on Signal and Information Processing* (pp. 945–948). IEEE.

Welch, G., & Bishop, G. (1995). An introduction to the Kalman filter.