Online algorithms for learning data-driven models of chaotic dynamics

Marc Bocquet[†], Quentin Malartic[†], Alban Farchi[†],
Massimo Bonavita[‡], Patrick Laloyaux[‡], Marcin Chrust[‡],

[†]CEREA, École des Ponts and EDF R&D, Île-De-France, France
[‡]ECMWF, Reading, United Kingdom.

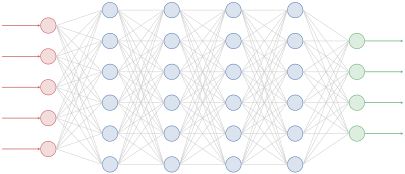
Outline

Machine learning for NWP with dense and perfect observations

- A typical (supervised) machine learning problem: given observations $\mathbf{y}_k$ of a system, derive a surrogate model of that system.

$$\mathcal{J}(\mathbf{p}) = \sum_{k=1}^{N_\mathrm{t}} \left\| \mathbf{y}_{k+1} - \mathcal{M}(\mathbf{p}, \mathbf{y}_k) \right\|^2 .$$

- $\mathcal{M}$ depends on a set of coefficients $\mathbf{p}$ (*e.g.*, the weights and biases of a neural network).



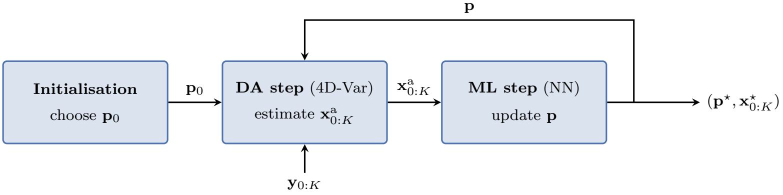
- This requires dense and perfect observations of the system. In NWP, observations are usually sparse and noisy: we need data assimilation!

Machine learning for NWP with sparse and noisy observations

- A rigorous Bayesian formalism for this problem:[1]

$$\mathcal{J}(\mathbf{p}, \mathbf{x}_0, \ldots, \mathbf{x}_{N_\mathrm{t}}) = \frac{1}{2} \sum_{k=0}^{N_\mathrm{t}} \left\| \mathbf{y}_k - \mathcal{H}_k(\mathbf{x}_k) \right\|^2_{\mathbf{R}_k^{-1}} + \frac{1}{2} \sum_{k=0}^{N_\mathrm{t}-1} \left\| \mathbf{x}_{k+1} - \mathcal{M}(\mathbf{p}, \mathbf{x}_k) \right\|^2_{\mathbf{Q}_k^{-1}} .$$

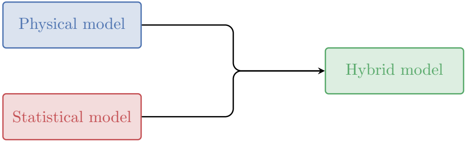
- This resembles a typical weak-constraint 4D-Var cost function!
- DA is used to estimate the state and then ML is used to estimate the model.



[1] [Bocquet et al. 2019; Bocquet et al. 2020; Brajard et al. 2020] in the wake of [Hsieh et al. 1998; Abarbanel et al. 2018]

Machine learning for NWP: learning model error

- Even though NWP models are not perfect, they are already quite good!
- Instead of building a surrogate model from scratch, we use the DA-ML framework to build a hybrid surrogate model, with a physical part and a statistical part:[2]



- In practice, the statistical part is trained to learn the error of the physical model.
- In general, it is easier to train a correction model than a full model: we can use smaller NNs and less training data.

[2] [Farchi et al. 2021; Brajard et al. 2021].

Outline

Typical architecture of a physical model

- The model is defined by a set of ODEs or PDEs which define the tendencies:

$$\frac{\partial \mathbf{x}}{\partial t} = \phi(\mathbf{x}). \tag{1}$$

- A numerical scheme is used to integrate the tendencies from time t to $t + \delta t$ (*e.g.*, Runge–Kutta):

$$\mathbf{x}(t + \delta t) = \mathcal{F}\Big(\mathbf{x}(t)\Big). \tag{2}$$

- Several integration steps are composed to define the resolvent from one analysis (or window) to the next:

$$\mathcal{M} : \mathbf{x}_k \mapsto \mathbf{x}_{k+1} = \mathcal{F} \circ \cdots \circ \mathcal{F}(\mathbf{x}_k) \tag{3}$$

Resolvent correction $\mathcal{M}$

- Physical model and of NN are independent.
- NN must predict the analysis increments.
- Resulting hybrid model not suited for short-term predictions.
- For DA, need to assume linear growth of errors in time to rescale correction.

Tendency correction ϕ

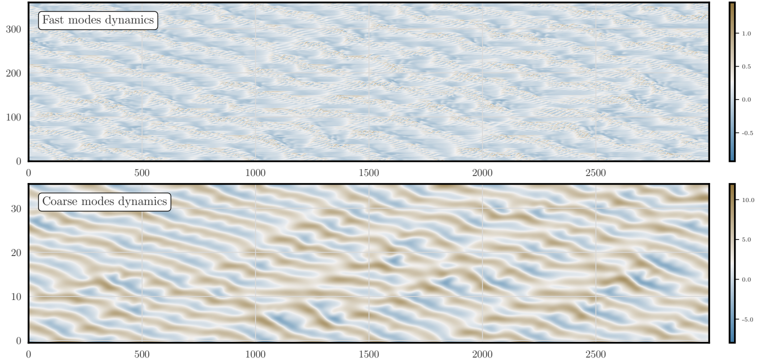
- Physical model and NN are entangled.
- Need TL of physical model to train NN!
- Resulting hybrid model suited for any prediction.
- Can be used as is for DA.

Two-scale Lorenz model (L05III)

- The two-scale Lorenz model (L05III) model: 36 slow & 360 fast variables, with equations:

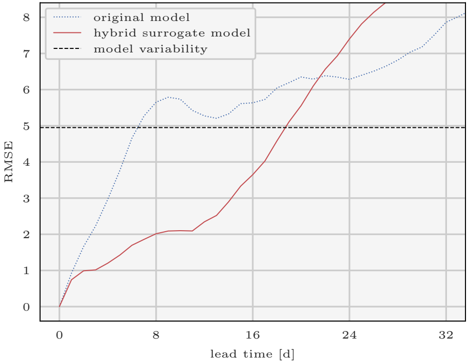
$$\frac{\mathrm{d}x_n}{\mathrm{d}t} = \psi_n^+(\mathbf{x}) + F - h\frac{c}{b}\sum_{m=0}^{9} u_{m+10n},$$

$$\frac{\mathrm{d}u_m}{\mathrm{d}t} = \frac{c}{b}\psi_m^-(b\mathbf{u}) + h\frac{c}{b}x_{m/10}, \quad \text{with} \quad \psi_n^\pm(\mathbf{x}) = x_{n\mp1}(x_{n\pm1} - x_{n\mp2}) - x_n,$$



Numerical illustration with the two-scale Lorenz system

- The non-corrected model is the one-scale Lorenz system.
- Noisy observations are assimilated using strong-constrained 4D-Var.
- Simple CNNs are trained using the 4D-Var analysis.



Data assimilation score

Model	Analysis RMSE
No correction	0.31
Resolvent correction	0.28
Tendency correction	0.24
True model	0.22

- The tendencies corr. is more accurate than the resolvent corr., even with smaller NNs and less training data.
- The tendencies corr. benefits from the interaction with the physical model.
- The resolvent corr. is highly penalised (in DA) by the assumption of linear growth of errors.

Online model error correction

- So far, the model error has been learnt offline: the ML (or training) step first requires a long analysis trajectory.
- We now investigate the possibility to perform online learning, *i.e.* improving the correction as new observations become available.
- To do this, we use the formalism of DA to estimate both the state and the NN parameters:

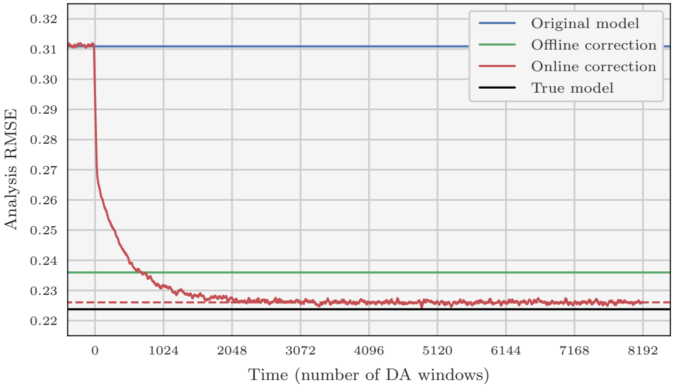
$$\mathcal{J}(\mathbf{p},\mathbf{x}) = \frac{1}{2}\left\|\mathbf{x}-\mathbf{x}^{\mathrm{b}}\right\|^2_{\mathbf{B}_{\mathrm{x}}^{-1}} + \frac{1}{2}\left\|\mathbf{p}-\mathbf{p}^{\mathrm{b}}\right\|^2_{\mathbf{B}_{\mathrm{p}}^{-1}} + \frac{1}{2}\sum_{k=0}^{L}\left\|\mathbf{y}_k-\mathcal{H}_k\circ\mathcal{M}^k(\mathbf{p},\mathbf{x})\right\|^2_{\mathbf{R}_k^{-1}}.$$

- For simplicity, we have neglected potential cross-covariance between state and NN parameters in the prior.
- Information is flowing from one window to the next using the prior for the state $\mathbf{x}^{\mathrm{b}}$ and for the NN parameters $\mathbf{p}^{\mathrm{b}}$.

⟶ This approach is very similar to classical parameter estimation in DA, and it can be seen as a NN formulation of weak-constraint 4D-Var.

Numerical illustration with the same two-scale Lorenz system

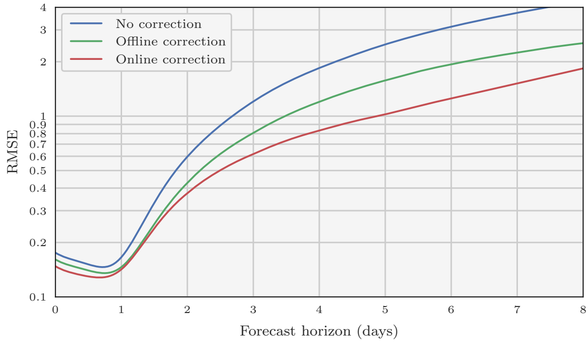
- We use the tendency correction approach, with the same simple CNN as before, and still using 4D-Var.



- The online correction steadily improves the model.
- At some point, the online correction gets more accurate than the offline correction.
- Eventually, the improvement saturates. The analysis error is similar to that obtained with the true model!

Online learning: towards an operational implementation with OOPS

- Development of a fortran NN library to interact with the fortran implementation of the forecast model.
- Interfacing the NN library with OOPS to estimate the NN parameters with DA.
- Simplifications of the NN correction:
  - the correction is additive, and added after each integration step (close to tendency correction);
  - the correction is computed independently for each atmospheric column[3].
  - the correction is computed at the start of the DA window and not updated during the window;
  - in practice, it requires only small adjustments to the current WC 4D-Var already implemented.
- Demonstration with OOPS-QG with promising results, implementation with OOPS-IFS in progress.



[3] [Bonavita et al. 2020]

Outline

Online learning with a LEnKF: Augmented state vector

▶ So far, learning was based on variational techniques using all available data. Can one design a sequential (online) ensemble scheme that progressively updates both the state and the model as data are collected?

▶ In the following, we make the assumptions:
(i) *autonomous* and *local* dynamics,
(ii) *homogeneous* dynamics or *heterogeneous* dynamics, or *mixed* dynamics.

▶ Parameters of the model:

$$\mathbf{p} \in \mathbb{R}^{N_\mathrm{p}} \text{ [global parameters]}, \quad \mathbf{q} \in \mathbb{R}^{N_\mathrm{q}} \text{ [local parameters]}.$$

▶ Augmented state formalism [Jazwinski 1970; Ruiz et al. 2013]:

$$\mathbf{z} = \begin{bmatrix} \mathbf{x} \\ \mathbf{p} \\ \mathbf{q} \end{bmatrix} \in \mathbb{R}^{N_\mathrm{z}}, \qquad \text{with} \quad N_\mathrm{z} = N_\mathrm{x} + N_\mathrm{p} + N_\mathrm{q}.$$

▶ Just a more ambitious parameter estimation problem!?
Yes! But we have to fill in several critical gaps of the parameter-estimation-via-EnKF literature.

Online learning with a LEnKF: The problems

- We use the augmented state formalism with local ensemble Kalman filters (EnKFs): LEnSRF and LETKF, which are keys for scalability.

- Adequacy and inadequacy between the main LEnKF classes and the estimation of local and global parameters:

Table: Adequacy (green) and inadequacy (red) between LEnKF types and the estimation of local, global and mixed parameters. CL refers to covariance localisation and DL refers to domain localisation.

LEnKF type	Global parameters	Local parameters	Mixed set of parameters
LEnSRF (CL)	well suited localisation in parameter space?	suited numerically costly	unclear solution proposed here
LETKF (DL)	only approximate[4] solution proposed here	well suited	unclear solution proposed here

- Beware that nonlocal observations require CL!

[4] [Aksoy et al. 2006]

Online learning with a LEnKF: The solutions

Table: Summary of the EnKF-ML family of algorithms

Inference problem	Dom. Local. local obs. only	Cov. Local. numerically costly	Dom. + Cov. Local.
State	LETKF [Hunt et al. 2007]	LEnSRF [Whitaker et al. 2002]	L^2EnSRF [Farchi et al. 2019]
State + global param.	LETKF-ML [Bocquet et al. 2021] new implementation[5]	LEnSRF-ML [Bocquet et al. 2021] new implementation	L^2EnSRF-ML not discussed
State + global & local param.	LETKF-HML new algorithm	LEnSRF-HML new algorithm	L^2EnSRF-HML new algorithm

Main results

New EnKF update formula and new LEnSRF/LETKF algorithms with parameter estimation:
global parameters $\longrightarrow$ LETKF-ML, LEnSRF-ML, L^2EnSRF-ML,
global and local parameters $\longrightarrow$ LETKF-HML, LEnSRF-HML, and L^2EnSRF-HML.

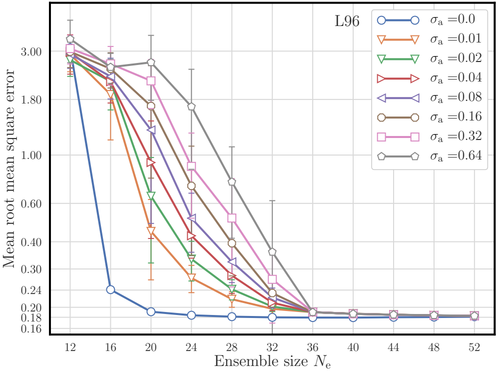
[5] new implementations and new algorithms: [Malartic et al. 2022]

Focus on the augmented dynamics and its unstable subspace

- **Augmented dynamics** (model persistence or Brownian motion):

$$\begin{bmatrix} \mathbf{x}_k \\ \mathbf{p}_k \end{bmatrix} \mapsto \begin{bmatrix} \mathbf{F}^k(\mathbf{x}_k, \mathbf{p}_k) \\ \mathbf{p}_k \end{bmatrix}$$

- Assuming (i) N_0 is the dimension of the *unstable neutral subspace* of the reference dynamics, (ii) N_e is the size of the ensemble, then, in order for the augmented global EnKF (EnKF-ML) to be stable, we must have: $N_\mathrm{e} \gtrapprox N_0 + N_\mathrm{p} + 1.$



Focus on the LEnSRF-ML update and global parameters

- Covariance localisation in the augmented space:

$$\mathbf{B}_{\mathsf{xx}} = \boldsymbol{\rho}_{\mathsf{xx}} \circ \left[\mathbf{X}^{\mathsf{f}}_{\mathsf{x}} \left(\mathbf{X}^{\mathsf{f}}_{\mathsf{x}}\right)^{\top}\right], \qquad \mathbf{B}_{\mathsf{px}} = \boldsymbol{\rho}_{\mathsf{px}} \circ \left[\mathbf{X}^{\mathsf{f}}_{\mathsf{p}} \left(\mathbf{X}^{\mathsf{f}}_{\mathsf{x}}\right)^{\top}\right] = \mathbf{B}^{\top}_{\mathsf{xp}}, \qquad \mathbf{B}_{\mathsf{pp}} = \boldsymbol{\rho}_{\mathsf{pp}} \circ \left[\mathbf{X}^{\mathsf{f}}_{\mathsf{p}} \left(\mathbf{X}^{\mathsf{f}}_{\mathsf{p}}\right)^{\top}\right].$$

- The localisation matrix $\boldsymbol{\rho}_{\mathsf{xx}}$ almost certainly makes $\mathbf{B}_{\mathsf{xx}}$ positive definite.

- The localisation matrix $\boldsymbol{\rho}_{\mathsf{px}}$ has to be uniform with respect to space because the parameters are global. This yields[6]:

$$\boldsymbol{\rho} = \begin{bmatrix} \boldsymbol{\rho}_{\mathsf{xx}} & \mathbf{1}_{\mathsf{x}}\boldsymbol{\zeta}^{\top}_{\mathsf{p}} \\ \boldsymbol{\zeta}_{\mathsf{p}}\mathbf{1}^{\top}_{\mathsf{x}} & \boldsymbol{\rho}_{\mathsf{pp}} \end{bmatrix}, \tag{4}$$

where $\boldsymbol{\zeta}_{\mathsf{p}} \in \mathbf{R}^{N_{\mathsf{p}}}$ is a vector of tapering coefficients.

- The positive definitness of $\boldsymbol{\rho}$ generates constraints on $\boldsymbol{\zeta}_{\mathsf{p}}$. A sufficient condition for positive definitness of $\boldsymbol{\rho}$ is:

$$\|\boldsymbol{\zeta}_{\mathsf{p}}\| \leq \sqrt{\frac{\lambda^{\min}_{\mathsf{p}}\lambda^{\min}_{\mathsf{x}}}{N_{\mathsf{x}}}}, \tag{5}$$

where $\lambda^{\min}_{\mathsf{p}}, \lambda^{\min}_{\mathsf{x}}$ are the smallest eigenvalues of $\boldsymbol{\rho}_{\mathsf{pp}}, \boldsymbol{\rho}_{\mathsf{xx}}$, respectively.

[6] [Ruckstuhl et al. 2018; Bocquet et al. 2021; Malartic et al. 2022]

Numerical illustration on the inhomogeneous Lorenz96 model (L96i)

▶ We use the LEnKF-HML on the L96i model, i.e. with unknown dynamics (global parameters) and unknown inhomogeneous forcings (40 local parameters).

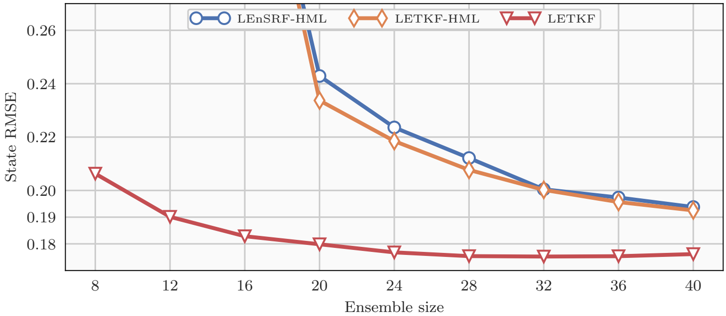


Figure: Time-averaged state analysis RMSE as a function of the ensemble size with the LEnSRF-HML (in blue) and the LETKF-HML (in yellow). For reference, the red line shows the scores obtained with the LETKF when the model is known.

Numerical illustration on the multi-layer L96 model (mL96)

▶ The mL96 model[7] is a vertical stack of $N_\mathsf{v} = 32$ coupled (atmospheric) layers, each layer being a L96 model with $N_\mathsf{h} = 40$ variables. The total state dimension is hence $N_\mathsf{x} = N_\mathsf{h} \times N_\mathsf{v} = 1280$, and the model's equations are :

$$\frac{\mathrm{d}x_{v,h}}{\mathrm{d}t} = (x_{v,h+1} - x_{v,h-2})x_{v,h-1} - x_{v,h} + F_{v,h} + \Gamma_{v+1,h} - \Gamma_{v,h}, \tag{6}$$

where $x_{v,h}$ is the h-th horizontal variable of the v-th vertical layer.

▶ The h index applies periodically in $\{1, \dots, N_\mathsf{h}\}$. The forcing term F is inhomogeneous; it is set constant over each layer and decreases from $F_{1,h} = 8$ for the bottom layer to $F_{N_\mathsf{v},h} = 4$ for the top layer.

▶ The last two terms correspond to the vertical coupling between adjacent layers, with

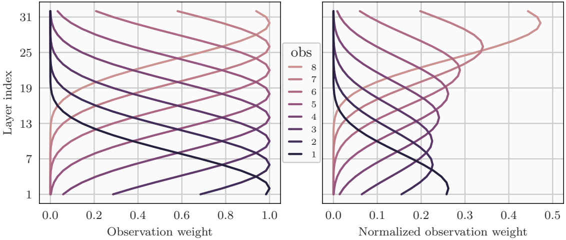
$$\Gamma_{v,h} \triangleq \begin{cases} x_{v,h} - x_{v-1,h} & \text{if } 2 \leq v \leq N_\mathsf{v}, \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

▶ We use the L^{2}EnSRF-HML on the observations of mL96, with unknown dynamics (global parameters) and unknown inhomogeneous forcings (local parameters).

[7] [Farchi et al. 2019]

Numerical illustration on the multi-layer Lorenz96 model

- Nonlocal radiance-like observations (averaging kernel for each of the 8 satellite channels without (left panel) and with (right panel) normalisation.)



- Numerical results (RMSEs):

Inference problem	N_0	Algorithm	Model	Loc.	N_e	state RMSE
1: $\mathbf{x}$	≈ 50	EnSRF	mL96		≥ 50	0.08
		L^2EnSRF	mL96	✓	≥ 10	0.08
2: $(\mathbf{x}, \mathbf{a}, \mathbf{f}_\mathsf{v}, \mathbf{f}_\mathsf{h})$	$\approx 50 + 88$	EnSRF-HML	sur $(\mathbf{a}, \mathbf{f}_\mathsf{v}, \mathbf{f}_\mathsf{h})$		≥ 140	0.11
		L^2EnSRF-HML	sur $(\mathbf{a}, \mathbf{f}_\mathsf{v}, \mathbf{f}_\mathsf{h})$	✓	50	0.12

Outline

Conclusions

- Main messages:
  - Bayesian DA view on state and model estimation.
    DA can address goals assigned to ML but with partial & noisy observations.
  - Online "WC" 4D-Var can be used to sequentially estimate both state and model.
  - Online EnKFs-ML can also be used to sequentially estimate both state and model.
  - Holes in the EnKF parameter estimation theory successfully filled in.
  - Successful on 1D and 2D low-order models (L96, L05III, L96i, mL96, OOPS QG).
  - Successful generalised algorithms that mix LA and DL, local and global paramters.
- In progress:
  - Application to the Marshall-Molteni 3-layer QG model on the sphere.
  - Application to the ERA5 and CMIP data (WeatherBench).
  - Application to the mighty ECMWF IFS.

References I

[1] H. D. I. Abarbanel, P. J. Rozdeba, and S. Shirman. "Machine Learning: Deepest Learning as Statistical Data Assimilation Problems". In: *Neural Computation* 30 (2018), pp. 2025–2055.

[2] A. Aksoy, F. Zhang, and J. Nielsen-Gammon. "Ensemble-based simultaneous state and parameter estimation in a two-dimensional sea-breeze model". In: *Mon. Wea. Rev.* 134 (2006), pp. 2951–2969.

[3] M. Bocquet, A. Farchi, and Q. Malartic. "Online learning of both state and dynamics using ensemble Kalman filters". In: *Foundations of Data Science* 3 (2021), pp. 305–330.

[4] M. Bocquet et al. "Bayesian inference of chaotic dynamics by merging data assimilation, machine learning and expectation-maximization". In: *Foundations of Data Science* 2 (2020), pp. 55–80.

[5] M. Bocquet et al. "Data assimilation as a learning tool to infer ordinary differential equation representations of dynamical models". In: *Nonlin. Processes Geophys.* 26 (2019), pp. 143–162.

[6] M. Bonavita and P. Laloyaux. "Machine Learning for Model Error Inference and Correction". In: *J. Adv. Model. Earth Syst.* 12 (2020), e2020MS002232.

[7] J. Brajard et al. "Combining data assimilation and machine learning to emulate a dynamical model from sparse and noisy observations: a case study with the Lorenz 96 model". In: *J. Comput. Sci.* 44 (2020), p. 101171.

[8] J. Brajard et al. "Combining data assimilation and machine learning to infer unresolved scale parametrisation". In: *Phil. Trans. R. Soc. A* 379 (2021), p. 20200086.

[9] A. Farchi and M. Bocquet. "On the efficiency of covariance localisation of the ensemble Kalman filter using augmented ensembles". In: *Front. Appl. Math. Stat.* 5 (2019), p. 3.

[10] A. Farchi et al. "Using machine learning to correct model error in data assimilation and forecast applications". In: *Q. J. R. Meteorol. Soc.* 147 (2021), pp. 3067–3084.

[11] W. W. Hsieh and B. Tang. "Applying Neural Network Models to Prediction and Data Analysis in Meteorology and Oceanography". In: *Bull. Amer. Meteor. Soc.* 79 (1998), pp. 1855–1870.

[12] B. R. Hunt, E. J. Kostelich, and I. Szunyogh. "Efficient data assimilation for spatiotemporal chaos: A local ensemble transform Kalman filter". In: *Physica D* 230 (2007), pp. 112–126.

[13] A. H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, New-York, 1970, p. 376.

[14] Q. Malartic, A. Farchi, and M. Bocquet. "Global and local parameter estimation using local ensemble Kalman filters: applications to online machine learning of chaotic dynamics". In: *Q. J. R. Meteorol. Soc.* 0 (2022). Accepted for publication, pp. 00–00.

References II

[15] Y. M. Ruckstuhl and T. Janjić. "Parameter and state estimation with ensemble Kalman filter based algorithms for convective-scale applications". In: *Q. J. R. Meteorol. Soc.* 144 (2018), pp. 826–841.

[16] J. J. Ruiz, M. Pulido, and T. Miyoshi. "Estimating model parameters with ensemble-based data assimilation: A Review". In: *J. Meteorol. Soc. Japan* 91 (2013), pp. 79–99.

[17] J. S. Whitaker and T. M. Hamill. "Ensemble Data Assimilation without Perturbed Observations". In: *Mon. Wea. Rev.* 130 (2002), pp. 1913–1924.