Data-driven reconstruction of chaotic dynamics using data assimilation and machine learning

Marc Bocquet[1], Julien Brajard[2,3], Alberto Carrassi[4,5] & Laurent Bertino[2]

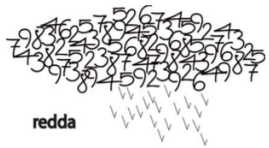
(1) CEREA, joint laboratory École des Ponts ParisTech and EDF R&D, Université Paris-Est

(2) Nansen Environmental and Remote Sensing Center

(3) Sorbonne University, CNRS-IRD-MNHN, LOCEAN

(4) Department of meteorology, University of Reading, United Kingdom

(5) Mathematical institute, University of Utrecht, The Netherlands



Outline

From model error to the absence of a model

▶ Data assimilation and model error

Numerical predictions in geophysics based on data assimilation crucially depends on both initial condition and model error [Magnusson et al., 2013]. There are methods developed to mitigate model error:

- additive stochastic noise [Trémolet, 2006; Raanes et al., 2015; Sakov et al. 2018]
- estimation of uncertain model parameters
- physically-driven stochastic perturbations [e.g., Buizza et al., 1999], stochastic subgrid parameterizations [e.g., Resseguier et al., 2017], inflation [e.g., Raanes et al., 2019]

▶ Data-driven forecast of a physical system

One step further: renounce physically-based models and use massive observation

- use data assimilation together with analogues [Lguensat et al., 2017]
- use diffusion maps for a spectral representation of datasets [e.g., Harlim, 2018]
- use neural networks (NNs), echo states networks, & deep learning [Park and Zhu 1994; Pathak, Lu, et al. 2017; Dueben and Bauer 2018; Vlachas et al. 2019] to represent the resolvent.

Building a surrogate model

- Learning the dynamics of a model from its output
  - more explicit (possibly with NNs) representations of the dynamics using specific regressors e.g., [Paduart et al. 2010; Brunton et al. 2016].
  - design NNs that mimic integration schemes [Wang and Lin 1998; Fablet et al. 2018; Long et al. 2018]

- Our goal
  - Use a data assimilation framework to infer both a surrogate model and the state trajectory within a time window over which the reference model is only partially & noisily observed.

Outline

ODE representation for the surrogate model

- Ordinary differential equations (ODEs) representation of the surrogate dynamics

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \boldsymbol{\Phi}_{\mathbf{A}}(\mathbf{x}), \qquad \boldsymbol{\Phi}_{\mathbf{A}}(\mathbf{x}) = \mathbf{A}\mathbf{r}(\mathbf{x}),$$

where

- $\mathbf{A}$ is a matrix of coefficients of size $N_\mathrm{x} \times N_\mathrm{p}$
- $\mathbf{r}(\mathbf{x})$ is a vector of nonlinear regressors of size N_p. For instance, for one-dimensional spatial systems and up to bilinear order:

$$\mathbf{r}(\mathbf{x}) = \left[1, \{x_n\}_{0 \leqslant n < N_\mathrm{x}}, \{x_n x_m\}_{0 \leqslant n \leqslant m < N_\mathrm{x}}\right].$$

  A priori, $N_\mathrm{p} = \binom{N_\mathrm{x}+1}{2} = \frac{1}{2}(N_\mathrm{x}+1)(N_\mathrm{x}+2)$ such regressors.

$\longrightarrow$ Intractable in high-dimension! (typically $N_\mathrm{x} \approx 10^6$ and beyond)

Assumptions and symmetries

▶ Locality

Locality of the physics: all multivariate monomials in the ODEs have variables x_n that belong to a stencil, i.e. a local arrangement of grid points around a given node.

- s_n is the stencil around node n, the pattern being the same for all nodes.
- the set of required monomials can therefore be reduced to (in 1D)

$$\mathbf{r}(\mathbf{x}) = \left[1, \{x_n\}_{0 \leqslant n < N_\mathrm{x}}, \{x_n x_m\}_{0 \leqslant n \leqslant m < N_\mathrm{x}, m \in s_n}\right].$$

  In 1D and with a stencil of size $2L+1$, there are $N_\mathrm{p} = 1 + N_\mathrm{x}(2+L)$ monomials.

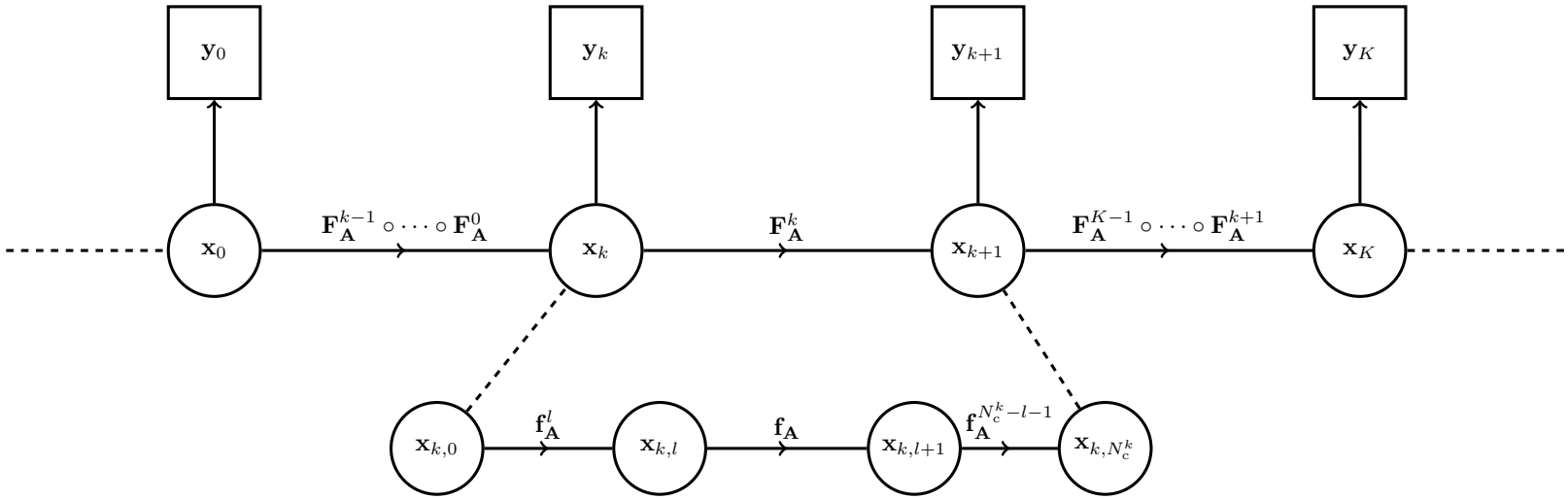
- $\mathbf{A}$ becomes sparse and can be squeezed into a dense rearrangement of $\mathbf{A}$. In 1D and with a stencil of size $2L+1$, the size of the dense $\mathbf{A}$ is

$$N_\mathrm{x} \times N_\mathrm{a} \quad \text{where} \quad N_\mathrm{a} = \sum_{l=L+1}^{2L+2} l = \frac{3}{2}(L+1)(L+2).$$

▶ Homogeneity

Moreover, we can additionally assume translational invariance. In that case $\mathbf{A}$ becomes a vector of size N_a.

Integration scheme and cycling



- Compositions of integration schemes:

$$\mathbf{x}_{k+1} = \mathbf{F}_\mathbf{A}^k(\mathbf{x}_k) \quad \text{where} \quad \mathbf{F}_\mathbf{A}^k \equiv \mathbf{f}_\mathbf{A}^{N_c^k} \equiv \underbrace{\mathbf{f}_\mathbf{A} \circ \ldots \circ \mathbf{f}_\mathbf{A}}_{N_c^k \text{ times}},$$

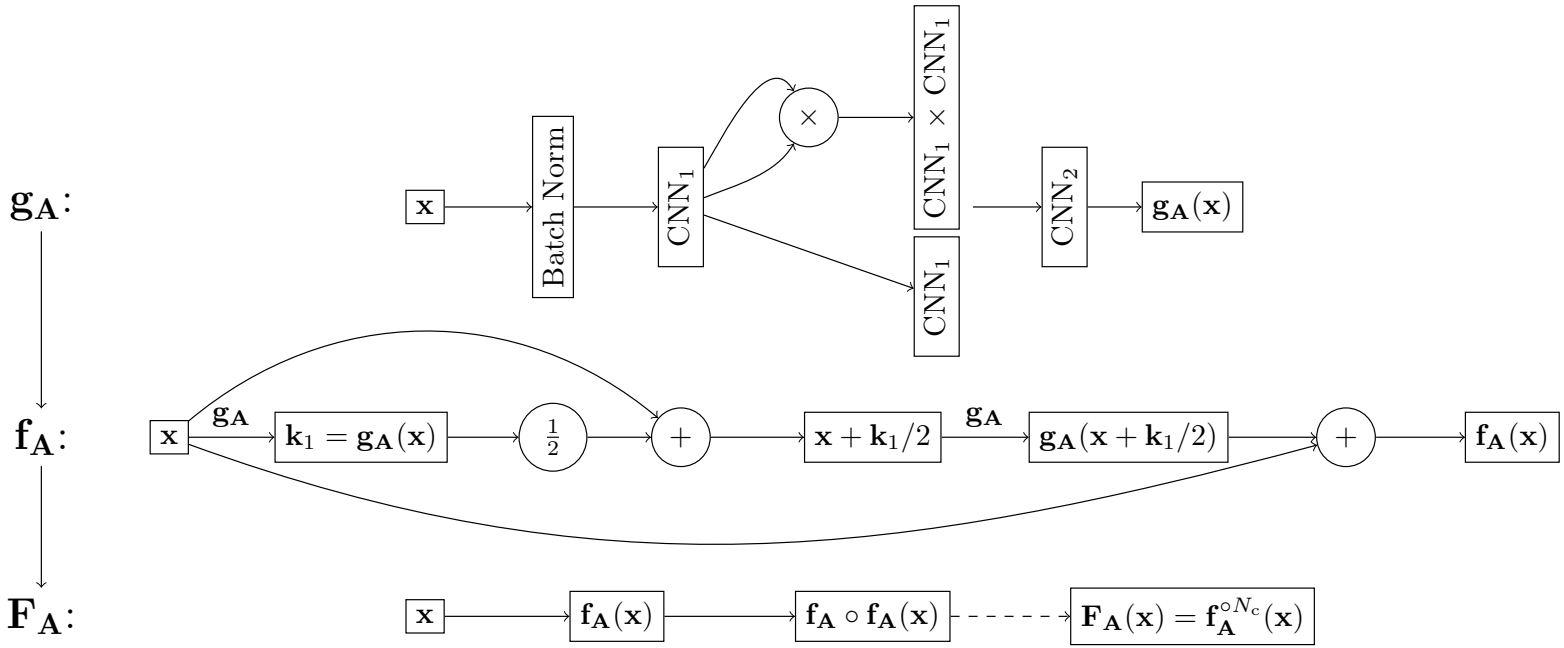
- Choosing a Runge-Kutta method as integration scheme:

$$\mathbf{f}_\mathbf{A}(\mathbf{x}) = \mathbf{x} + h \sum_{i=0}^{N_{\mathrm{RK}}-1} \beta_i \mathbf{k}_i, \qquad \mathbf{k}_i = \boldsymbol{\Phi}_\mathbf{A}\left(\mathbf{x} + h \sum_{j=0}^{i-1} \alpha_{i,j} \mathbf{k}_j\right).$$

Outline

Neural network models

- We tested many simple architectures, all following the structure of N_c explicit Runge-Kutta schemes, with linear or nonlinear activation functions:
  - The sleek algebraic representation above does not rely on ML libraries (TensorFlow, PyTorch, etc.). But it was also implemented as NNs using these tools.
  - Convolutional layers were used for local, homogeneous systems.
  - Locally connected convolutional layers were used for local, heterogeneous systems.



Outline

Bayesian analysis of the joint problem

- Bayesian view on state and model estimation:

$$p(\mathbf{A}, \mathbf{Q}_{1:K}, \mathbf{x}_{0:K} | \mathbf{y}_{0:K}, \mathbf{R}_{0:K}) = \frac{p(\mathbf{y}_{0:K} | \mathbf{x}_{0:K}, \mathbf{A}, \mathbf{Q}_{1:K}, \mathbf{R}_{0:K}) p(\mathbf{x}_{0:K} | \mathbf{A}, \mathbf{Q}_{1:K}) p(\mathbf{A}, \mathbf{Q}_{1:K})}{p(\mathbf{y}_{0:K}, \mathbf{R}_{0:K})}.$$

- Data assimilation cost function assuming Gaussian errors and Markovian dynamics:

$$\begin{aligned}\mathcal{J}(\mathbf{A}, \mathbf{x}_{0:K}, \mathbf{Q}_{1:K}) = & \frac{1}{2} \sum_{k=0}^{K} \left\{ \| \mathbf{y}_k - \mathbf{H}_k(\mathbf{x}_k) \|^2_{\mathbf{R}_k^{-1}} + \ln |\mathbf{R}_k| \right\} \\ & + \frac{1}{2} \sum_{k=1}^{K} \left\{ \left\| \mathbf{x}_k - \mathbf{F}_{\mathbf{A}}^{k-1}(\mathbf{x}_{k-1}) \right\|^2_{\mathbf{Q}_k^{-1}} + \ln |\mathbf{Q}_k| \right\} \\ & - \ln p(\mathbf{x}_0, \mathbf{A}, \mathbf{Q}_{1:K}).\end{aligned}$$

$\longrightarrow$ Allows to rigorously handle partial and noisy observations.

- Typical machine learning cost function with $\mathbf{H}_k = \mathbf{I}_k$ in the limit $\mathbf{R}_k \longrightarrow \mathbf{0}$:

$$\mathcal{J}(\mathbf{A}) \approx \frac{1}{2} \sum_{k=1}^{K} \left\| \mathbf{y}_k - \mathbf{F}_{\mathbf{A}}^{k-1}(\mathbf{y}_{k-1}) \right\|^2_{\mathbf{Q}_k^{-1}} - \ln p(\mathbf{y}_0, \mathbf{A}).$$

Similar outcome or improved upon [Hsieh and Tang 1998; Abarbanel et al. 2018].

Bayesian analysis of the joint problem

- If $\mathbf{Q}_{1:K}$ are known, we look for minima of

$$\mathcal{J}(\mathbf{A}, \mathbf{x}_{0:K} | \mathbf{Q}_{1:K}) = -\ln p(\mathbf{A}, \mathbf{x}_{0:K} | \mathbf{y}_{0:K}, \mathbf{R}_{0:K}, \mathbf{Q}_{1:K})$$

which is not as general as $\mathcal{J}(\mathbf{A}, \mathbf{x}_{0:K}, \mathbf{Q}_{1:K})$.

(1) - The optimization of $\mathcal{J}(\mathbf{A}, \mathbf{x}_{0:K} | \mathbf{Q}_{1:K})$ can be solved using a full variational approach.

  - In [Bocquet et al. 2019], $\mathcal{J}(\mathbf{A}, \mathbf{x}_{0:K} | \mathbf{Q}_{1:K})$ is optimized using a full weak-constraint 4D-Var where both $\mathbf{x}_{0:K}$ and $\mathbf{A}$ are control variables (assuming $\mathbf{Q}_{1:K}$ is known).

(2) - The optimization of $\mathcal{J}(\mathbf{A}, \mathbf{x}_{0:K} | \mathbf{Q}_{1:K})$ can be solved using a coordinate descent.

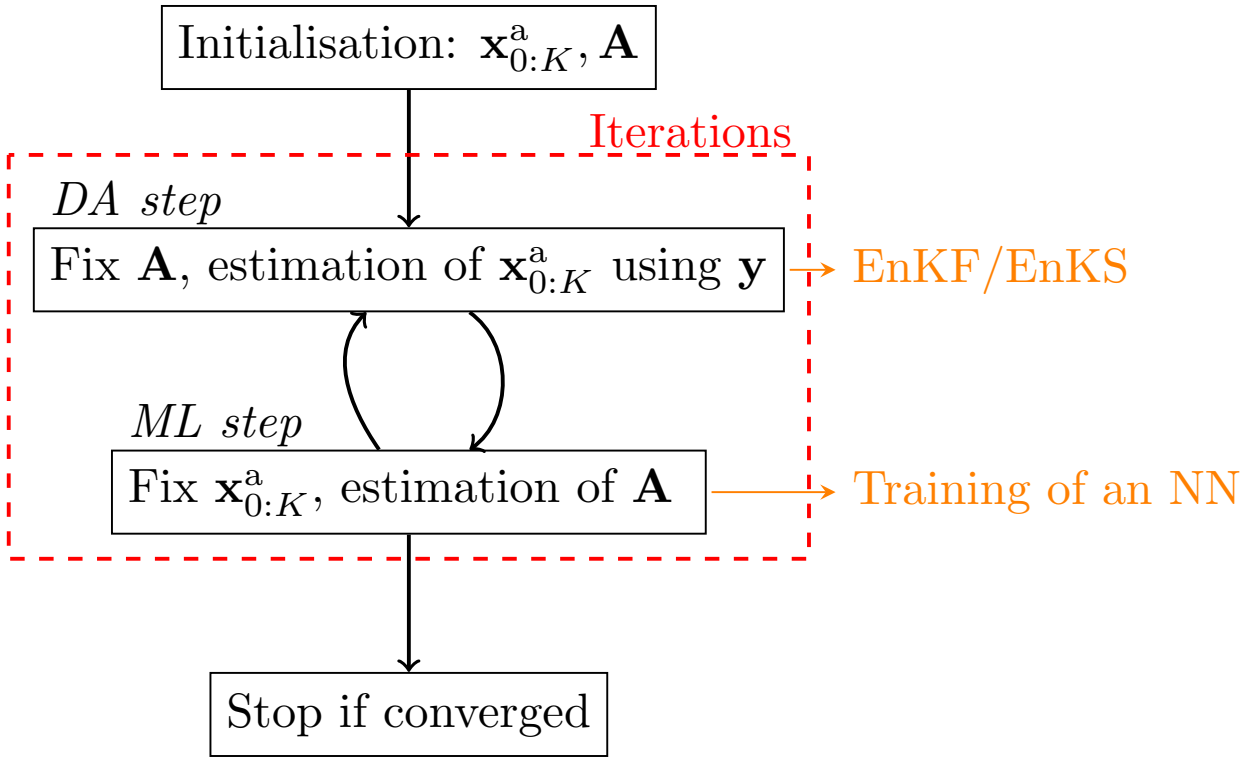
  - For $\mathcal{J}(\mathbf{A}, \mathbf{x}_{0:K} | \mathbf{Q}_{1:K})$: using a weak constraint 4D-Var for $\mathbf{x}_{0:K}$ and a variational optimization problem for $\mathbf{A}$ [Bocquet et al. 2019].

  - For $\mathcal{J}(\mathbf{A}, \mathbf{x}_{0:K} | \mathbf{Q}_{1:K})$: using an EnKF for $\mathbf{x}_{0:K}$ and a variational optimization problem for $\mathbf{A}$ [Brajard et al. 2020].

Bayesian analysis of the joint problem

- Coordinate descent of [Brajard et al. 2020].

Hybrid data assimilation and machine learning techniques.



- The coordinate descent algorithm is interpreted as an expectation-maximization (EM) algorithm by [Nguyen et al. 2019].

Bayesian analysis of the marginal problem

- Looking only for the dynamics and its model error:

$$p(\mathbf{A}, \mathbf{Q}_{1:K} | \mathbf{y}_{0:K}, \mathbf{R}_{0:K}) = \int \mathrm{d}\mathbf{x}_{0:K}\, p(\mathbf{A}, \mathbf{Q}_{1:K}, \mathbf{x}_{0:K} | \mathbf{y}_{0:K}, \mathbf{R}_{0:K})$$

yielding the loss function

$$\mathcal{J}(\mathbf{A}, \mathbf{Q}_{1:K}) = -\ln p(\mathbf{A}, \mathbf{Q}_{1:K} | \mathbf{y}_{0:K}, \mathbf{R}_{0:K}).$$

- A MAP solution (minimum of $\mathcal{J}$) is provided by the EM algorithm. Applying it for the reconstruction of a dynamical system has been suggested in [Ghahramani and Roweis 1999], using an extended Kalman smoother, or for the estimation of subgrid stochastic processes in [Pulido et al. 2018] using an ensemble Kalman smoother.

Reminder on the EM algorithm

- Goal of the EM method: find a local maximum over θ of:

$$p(\theta|\mathbf{y}) = \frac{p(\mathbf{y}|\theta)p(\theta)}{p(\mathbf{y})} = \frac{p(\theta)}{p(\mathbf{y})}\int \mathrm{d}\mathbf{x}\, p(\mathbf{y}|\mathbf{x},\theta)p(\mathbf{x}|\theta),$$

where expressions for $p(\mathbf{y}|\mathbf{x},\theta)$ and for $p(\mathbf{x}|\theta)$ are known, whereas the integral being intractable, an analytic expression for $p(\mathbf{y}|\theta)$ is not known.

- The algorithm principle of the EM method [Dempster et al. 1977] consists in iterating:
  - The expectation step: Given $\theta^{(j)}$, compute

$$\mathcal{L}(\theta|\theta^{(j)}) = \mathbb{E}_{\mathbf{x}|\mathbf{y},\theta^{(j)}}\left[\ln p(\mathbf{x},\mathbf{y},\theta)\right].$$

  - The maximization step: Look for a local maximum of $\mathcal{L}(\theta|\theta^{(j)})$ and set it to be

$$\theta^{(j+1)} = \mathrm{argmax}_\theta\, \mathcal{L}(\theta|\theta^{(j)}).$$

- Monte Carlo approximation of $\mathcal{L}(\theta|\theta^{(j)})$ [Wei and Tanner 1990]: A sample estimator is

$$\mathcal{L}(\theta|\theta^{(j)}) \approx \frac{1}{N_\mathrm{e}}\sum_{i=1}^{N_\mathrm{e}} \ln p(\mathbf{x}_i^{(j)},\mathbf{y},\theta), \qquad \text{with} \quad \mathbf{x}_i^{(j)} \sim \mathbf{x}|\mathbf{y},\theta^{(j)}.$$

Algorithm for the full solution of the marginal problem (1/2)

- The expectation step: EnKS over a long period $[t_0, t_K]$ which accounts for model error (SQRT-CORE scheme, [Raanes et al. 2015]). The output is $\mathbf{E}_{0:K}^{(j)} \in \mathbb{R}^{K \times N_\mathrm{x} \times N_\mathrm{e}}$.

- The maximization step: Minimize:

$$\begin{aligned}\mathcal{L}^{(j)}(\mathbf{A}, \mathbf{Q}) =& -\frac{1}{N_\mathrm{e}} \sum_{i=1}^{N_\mathrm{e}} \ln p(\mathbf{E}_i^{(j)}, \mathbf{y}_{0:K}, \mathbf{A}, \mathbf{Q}, \mathbf{R}_{0:K}) \\ =& \frac{1}{2N_\mathrm{e}} \sum_{i=1}^{N_\mathrm{e}} \sum_{k=1}^{K} \left\{ \left\| \mathbf{x}_{k,i}^{(j)} - \mathbf{F}_{\mathbf{A}}^{k-1}(\mathbf{x}_{k-1,i}^{(j)}) \right\|_{\mathbf{Q}^{-1}}^2 + \ln |\mathbf{Q}| \right\} \\ & - \ln p(\mathbf{x}_{0,i}^{(j)}, \mathbf{A}, \mathbf{Q}) + \cdots .\end{aligned}$$

Algorithm for the full solution of the marginal problem (2/2)

- The maximization step can be achieved by either a joint optimization on

$$\mathcal{L}^{(j)}(\mathbf{A}, \mathbf{Q})$$

or by a coordinate descent over $\mathbf{A}$ and $\mathbf{Q}$, which alternates (i) a minimization on

$$\mathcal{L}(\mathbf{A}, \mathbf{Q}^{(j,p)}) = \frac{1}{2N_e} \sum_{i=1}^{N_e} \sum_{k=1}^{K} \left\| \mathbf{x}_{k,i}^{(j)} - \mathbf{F}_{\mathbf{A}^{(j,p)}}^{k-1}(\mathbf{x}_{k-1,i}^{(j)}) \right\|_{\mathbf{Q}^{(j,p)-1}}^2,$$

yielding $\mathbf{A}^{(j,p)}$ and (ii)

$$\mathbf{Q}^{(j,p+1)} = \frac{1}{KN_e} \sum_{i=1}^{N_e} \sum_{k=1}^{K} \left(\mathbf{x}_{k,i}^{(j)} - \mathbf{F}_{\mathbf{A}^{(j,p)}}^{k-1}(\mathbf{x}_{k-1,i}^{(j)}) \right) \left(\mathbf{x}_{k,i}^{(j)} - \mathbf{F}_{\mathbf{A}^{(j,1)}}^{k-1}(\mathbf{x}_{k-1,i}^{(j)}) \right)^\top.$$

In practice, only one iteration of this coordinate descent (which is exact if $\mathbf{Q} = q\mathbf{I}_x$).

- Could be numerically very costly!

Algorithm for an approximate solution of the marginal problem

- The expectation step: EnKS over a long period $[t_0, t_K]$ which accounts for model error (SQRT-CORE scheme). The outputs are $\overline{\mathbf{x}}_{0:K}^{(j)}$ and $\mathbf{Q}^{(j+1)}$ computed online by accumulating over the time window.

- The maximization step: Minimize:

$$\begin{aligned}\mathcal{L}^{(j)}(\mathbf{A}, \mathbf{Q}^{(j+1)}) &= -\ln p(\overline{\mathbf{x}}_{0:K}^{(j)}, \mathbf{y}_{0:K}, \mathbf{A}, \mathbf{Q}^{(j+1)}, \mathbf{R}_{0:K}) \\ &= \frac{1}{2}\sum_{k=1}^{K}\left\{\left\|\overline{\mathbf{x}}_k^{(j)} - \mathbf{F}_{\mathbf{A}}^{k-1}(\overline{\mathbf{x}}_{k-1}^{(j)})\right\|^2_{\mathbf{Q}^{(j+1)^{-1}}} + \ln\left|\mathbf{Q}^{(j+1)}\right|\right\} \\ &\quad - \ln p(\overline{\mathbf{x}}_0^{(j)}, \mathbf{A}, \mathbf{Q}^{(j+1)}) + \cdots .\end{aligned}$$

Note the use of the ensemble mean instead of the ensemble.

- No iteration in the maximization step over $\mathbf{A}$ and $\mathbf{Q}$ (should be fine if $\mathbf{Q} = q\mathbf{I}_x$).

Non-informative hyperpriors on $\mathbf{Q}$ (Jeffreys')

- If $\mathbf{Q} = q\mathbf{I}_{\mathrm{x}}$:

$$\mathcal{L}(\mathbf{A}, \mathbf{Q}) = -\ln p(q) + \frac{Ks}{2q} + \frac{KN_{\mathrm{x}}}{2}\ln(q) + \dots$$

where

$$s = \frac{1}{KN_{\mathrm{e}}} \sum_{i=1}^{N_{\mathrm{e}}} \sum_{k=1}^{K} \left\| \mathbf{x}_{k,i} - \mathbf{F}_{\mathbf{A}}^{k-1}(\mathbf{x}_{k-1,i}) \right\|^2.$$

Minimizing on q yields for the maximization step:

$$q = \frac{K}{KN_{\mathrm{x}} + 2} s.$$

- General $\mathbf{Q}$:

$$\mathcal{L}(\mathbf{A}, \mathbf{Q}) = -\ln p(\mathbf{Q}) + \frac{K}{2}\mathrm{Tr}\left(\mathbf{S}\mathbf{Q}^{-1}\right) + \frac{\mathrm{K}}{2}\ln(|\mathbf{Q}|) + \dots$$

where

$$\mathbf{S} = \frac{1}{KN_{\mathrm{e}}} \sum_{i=1}^{N_{\mathrm{e}}} \sum_{k=1}^{K} \left(\mathbf{x}_{k,i} - \mathbf{F}_{\mathbf{A}}^{k-1}(\mathbf{x}_{k-1,i})\right) \left(\mathbf{x}_{k,i} - \mathbf{F}_{\mathbf{A}}^{k-1}(\mathbf{x}_{k-1,i})\right)^{\top}.$$

Minimizing on $\mathbf{Q}$ yields for the maximization step:

$$\mathbf{Q} = \frac{K}{K + N_{\mathrm{x}} + 1} \mathbf{S}.$$

Hyperpriors on **A**

▶ The design of the hyperprior for **A** is primarily driven by physical modeling and numerical stability [Bocquet et al. 2019].

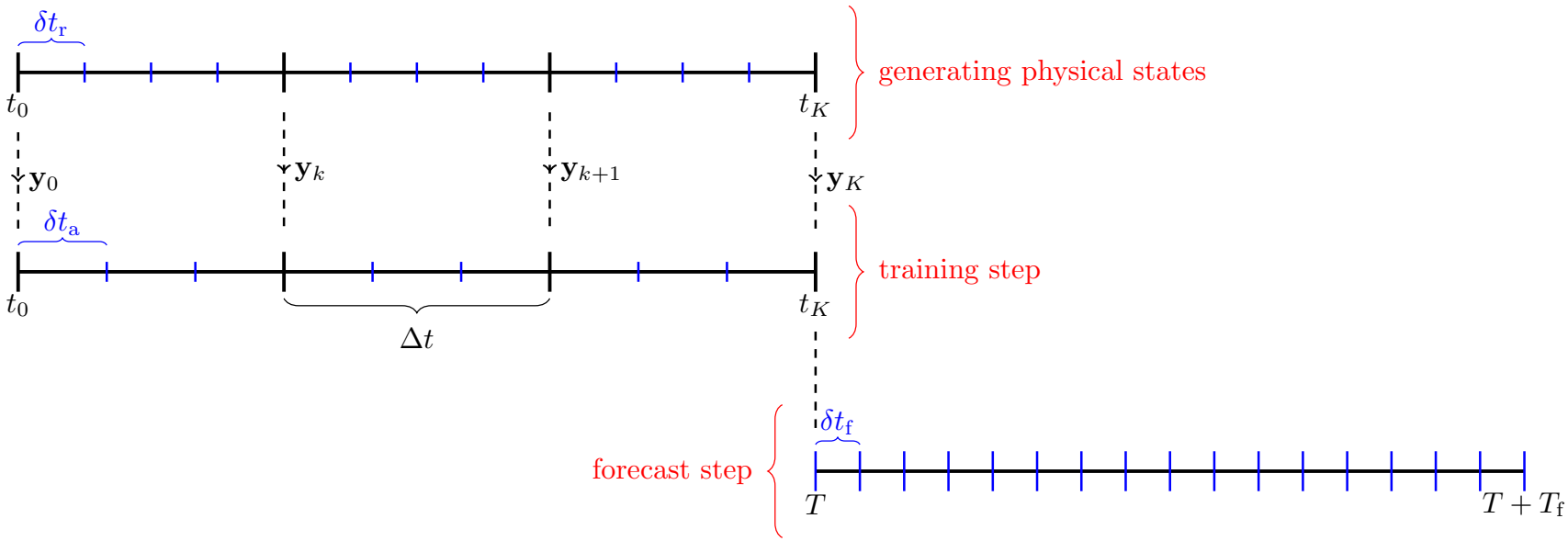
▶ Practically, an hyperprior for **A** could be implemented by adding a regularization term (typically L1 or L2 norm) on the coefficients of **A**, corresponding to specific prior statistical assumptions for **A**. We avoid such regularization here by mostly considering very long training windows, and because **A** is rather well constrained by locality and/or homogeneity.

▶ However, with higher dimensional physical models, larger **A**, deeper NN representations, and shorter training windows by comparison, methods used in machine learning and deep learning to regularize and avoid overfitting could be used, for instance dropouts and stochastic optimization techniques [LeCun et al. 2012].

Outline

Experiment plan

- The reference model, the surrogate model and the forecasting system



- Metrics of comparison:
  - Model: ODE coefficients norm $\|\mathbf{A}_\mathrm{a} - \mathbf{A}_\mathrm{r}\|_\infty$.
  - Forecast skill [FS]: Normalized RMSE (NRMSE) between the reference and the surrogate forecasts as a function of the lead time (averaged over many initial conditions).
  - Lyapunov spectrum [LS].
  - Power spectrum density [PSD].

Identifiable model and perfect observations

▶ Inferring the dynamics from dense & noiseless observations of identifiable models

- The Lorenz 63 model (L63, 3 variables):

$$\frac{\mathrm{d}x_0}{\mathrm{d}t} = \sigma(x_1 - x_0),$$
$$\frac{\mathrm{d}x_1}{\mathrm{d}t} = \rho x_0 - x_1 - x_0 x_2,$$
$$\frac{\mathrm{d}x_2}{\mathrm{d}t} = \rho x_0 x_1 - \beta x_2,$$

$\longrightarrow \|\mathbf{A}_\mathrm{a} - \mathbf{A}_\mathrm{r}\|_\infty \sim 10^{-13}$ close to perfect reconstruction at machine precision.

- The Lorenz 96 model (L96, 40 variables)

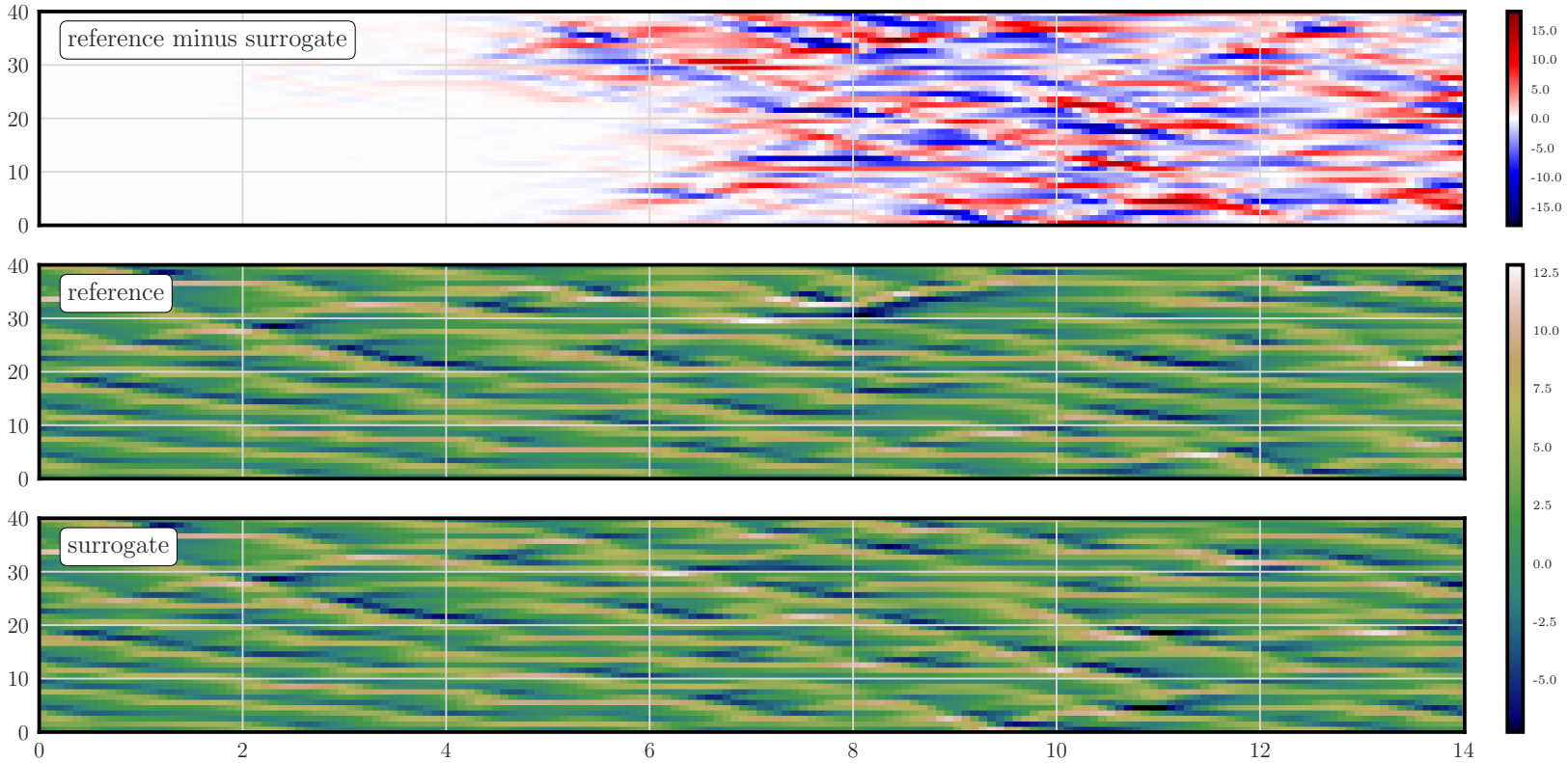
$$\frac{\mathrm{d}x_n}{\mathrm{d}t} = (x_{n+1} - x_{n-2})x_{n-1} - x_n + F,$$

$\longrightarrow \|\mathbf{A}_\mathrm{a} - \mathbf{A}_\mathrm{r}\|_\infty \sim 10^{-13}$ close to perfect reconstruction at machine precision.

Almost identifiable model and perfect observations

- Inferring the dynamics from dense & noiseless observations of a non-identifiable model

The Lorenz 96 model (40 variables). Surrogate model based on an RK2 scheme.
Analysis of the modeling depth as a function of N_c.

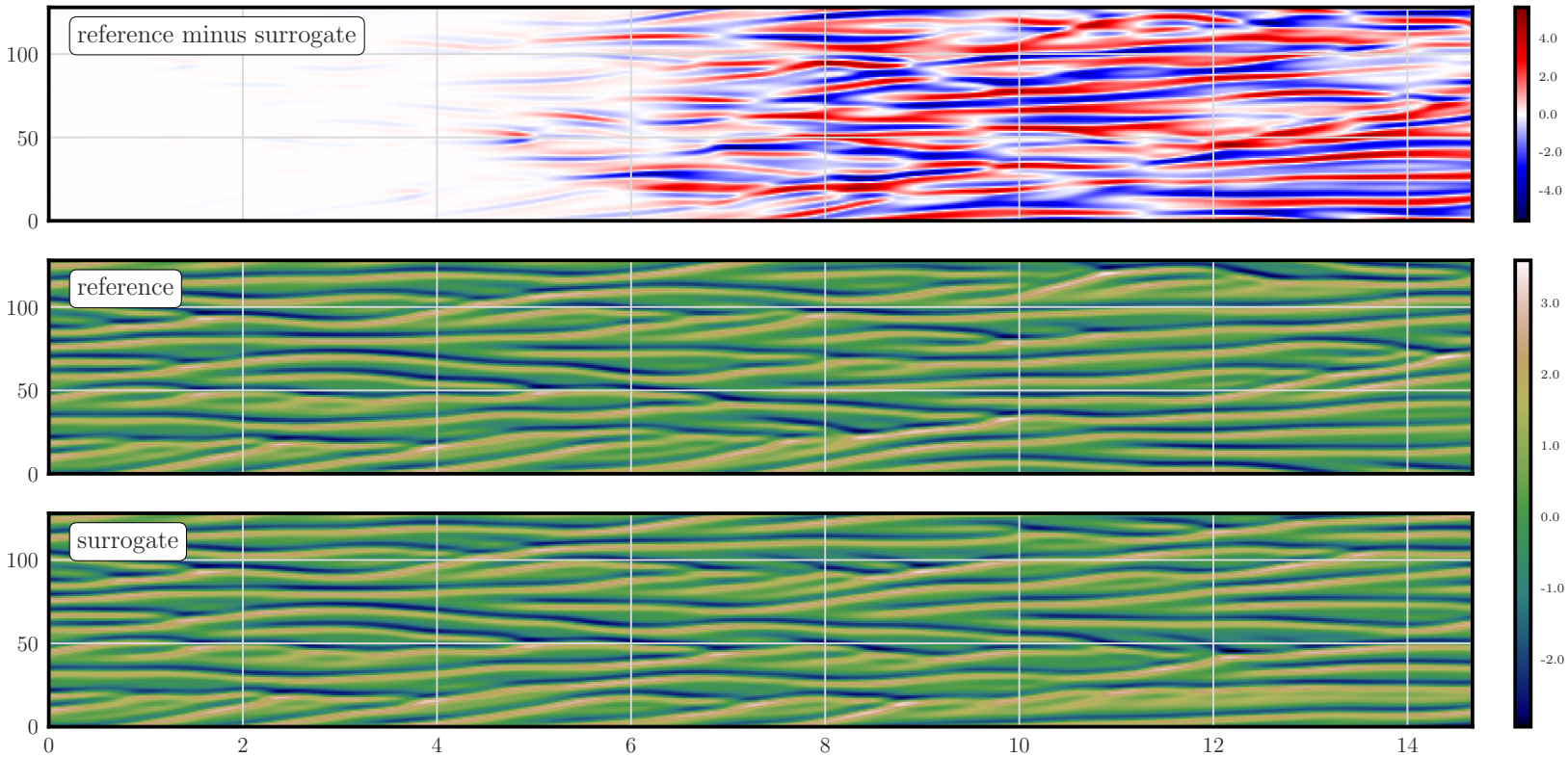


Not so identifiable model and perfect observations

- Inferring the dynamics from dense & noiseless observations of a non-identifiable model

The Kuramoto-Sivashinski (KS) model (continuous, 128 variables).

$$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} - \frac{\partial^2 u}{\partial x^2} - \frac{\partial^4 u}{\partial x^4},$$

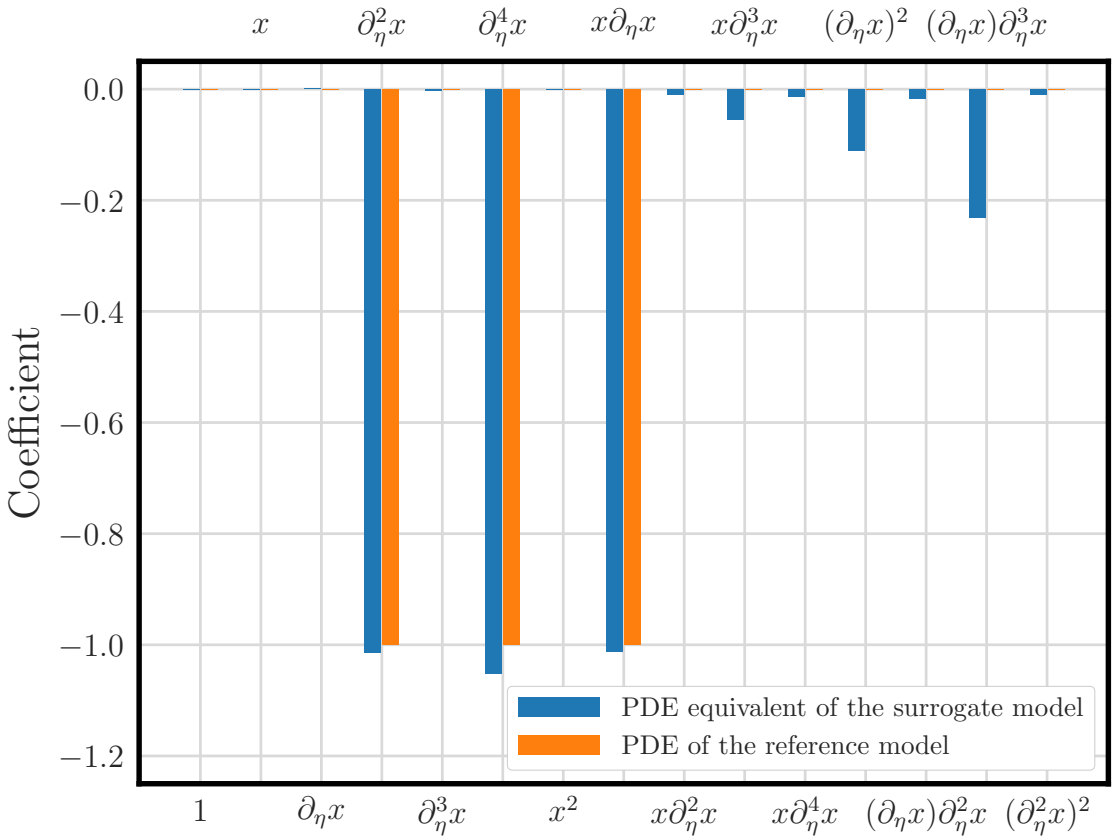


Not so identifiable model and perfect observations

▶ Inferring the dynamics from dense & noiseless observations of a non-identifiable model

The Kuramoto-Sivashinski (KS) model (continuous, 128 variables).

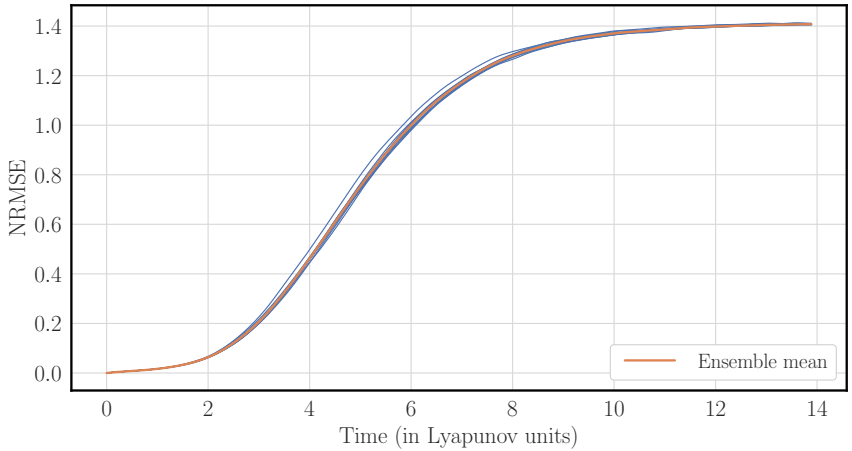
$$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} - \frac{\partial^2 u}{\partial x^2} - \frac{\partial^4 u}{\partial x^4},$$

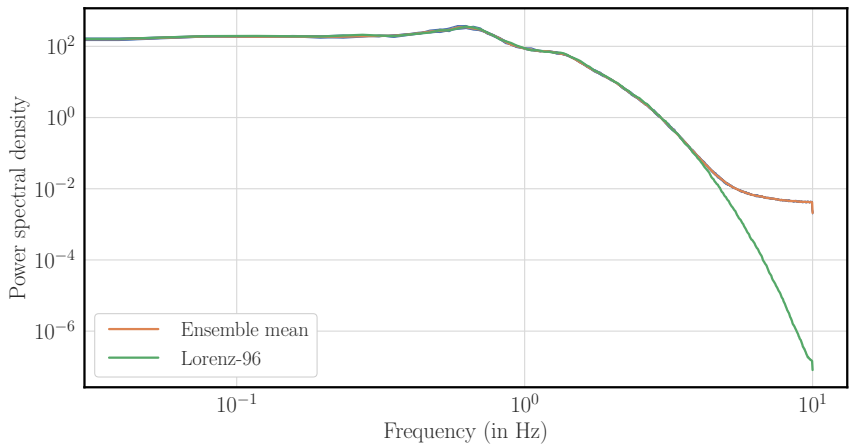


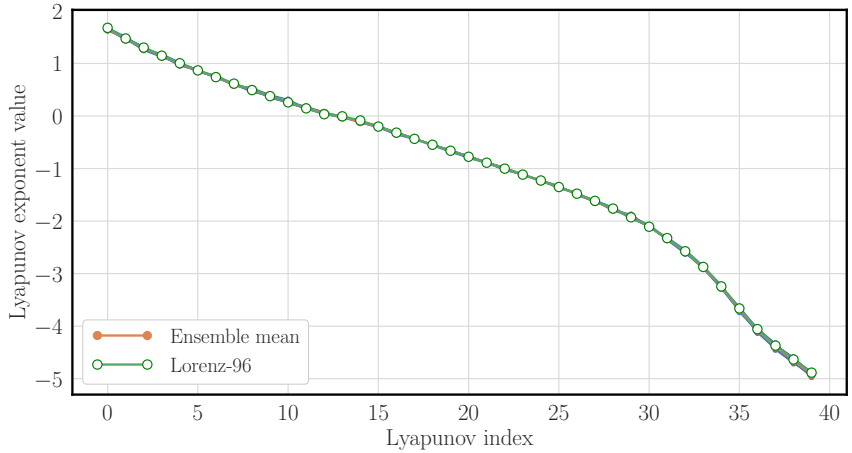
Almost identifiable model and imperfect observations

- Very good reconstruction of the long-term properties of the model (L96 model).

- Approximate scheme
- Fully observed
- Significantly noisy observations $\mathbf{R} = \mathbf{I}$
- Long window $K = 5000$, $\Delta t = 0.05$
- EnKS with $L = 4$
- 30 EM iterations





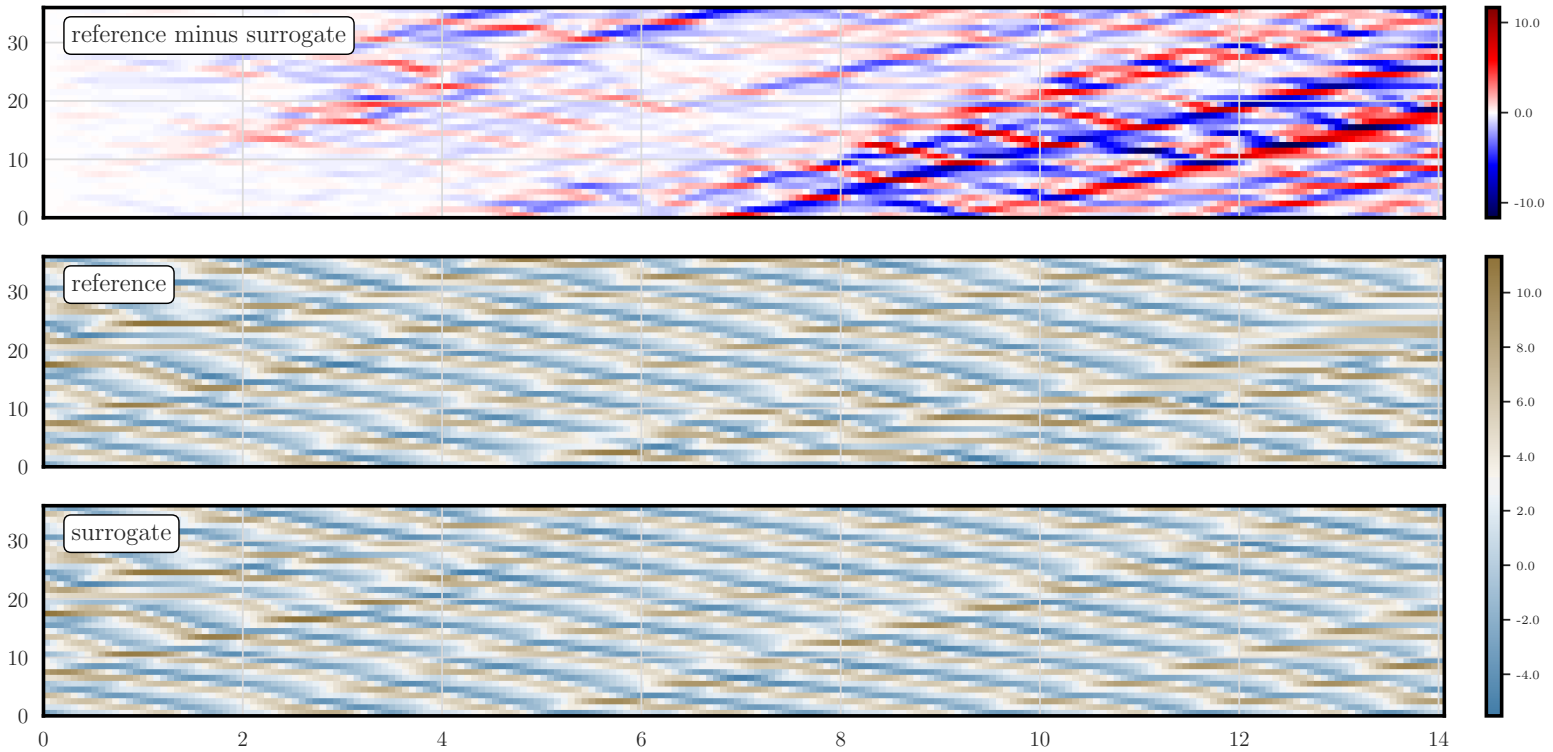


Non-identifiable model and imperfect observations

- The Lorenz 05III (two-scale) model (36 slow & 360 fast variables).

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

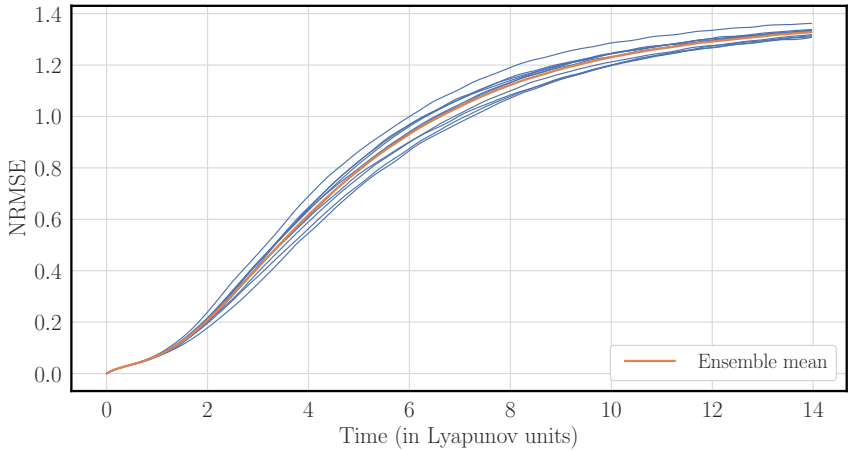
$$\frac{\mathrm{d}u_m}{\mathrm{d}t} = \frac{c}{b}\psi_m^-(b\mathbf{u}) + h\frac{c}{b}x_{m/10}, \quad \text{with} \quad \psi_n^\pm(\mathbf{x}) = x_{n\mp1}(x_{n\pm1} - x_{n\mp2}) - x_n,$$

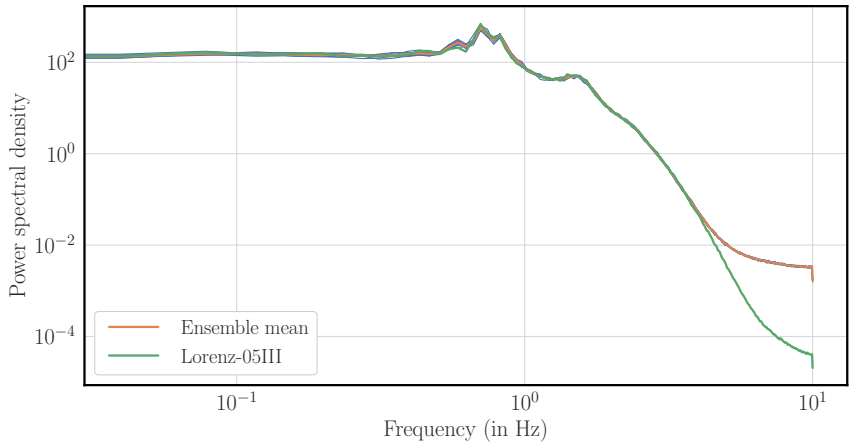


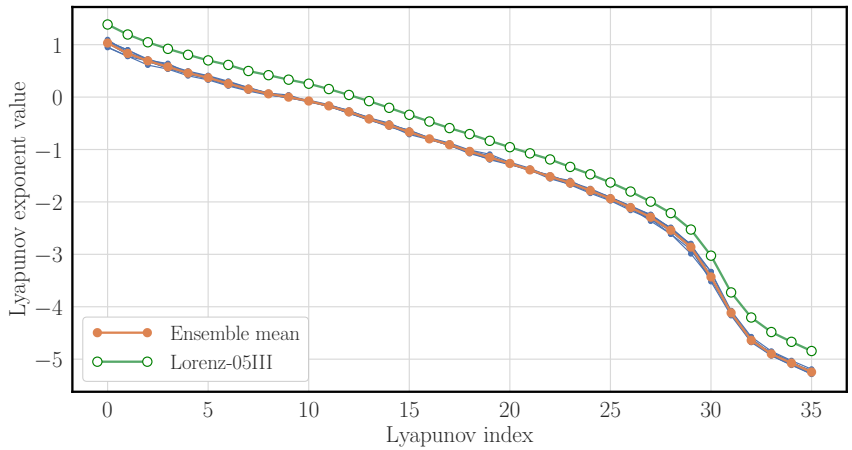
Non-identifiable model and imperfect observations

- Good reconstruction of the long-term properties of the model (L05III model).

- Approximate scheme
- Observation of the coarse modes only
- Significantly noisy observations $\mathbf{R} = \mathbf{I}$
- Long window $K = 5000$, $\Delta t = 0.05$
- EnKS with $L = 4$
- 30 EM iterations







Comparison of the full and approximate schemes

▶ Full scheme computationally much more demanding than the approximate scheme:
(i) Evaluation of the loss function N_e times more costly
(ii) Storage N_e times more demanding.

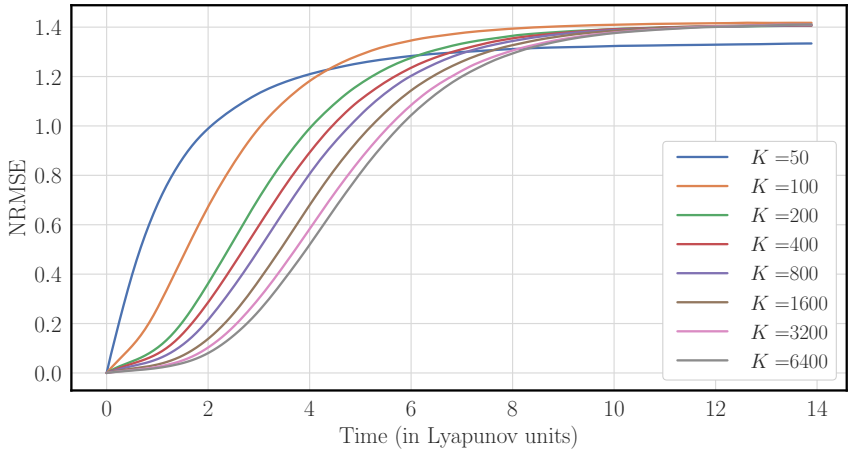
▶ The LS and PSD (long-term) are very close to each other. However, the FS of the approximately is better than that of the full scheme. Slight overfitting?

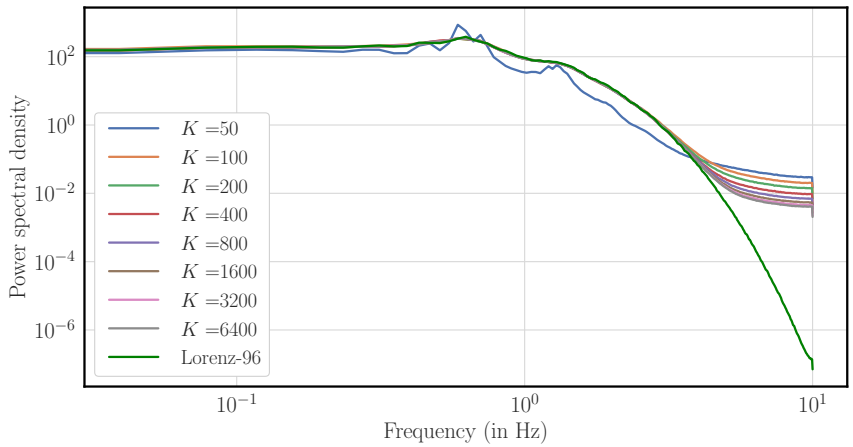
▶ Scalar indicators:

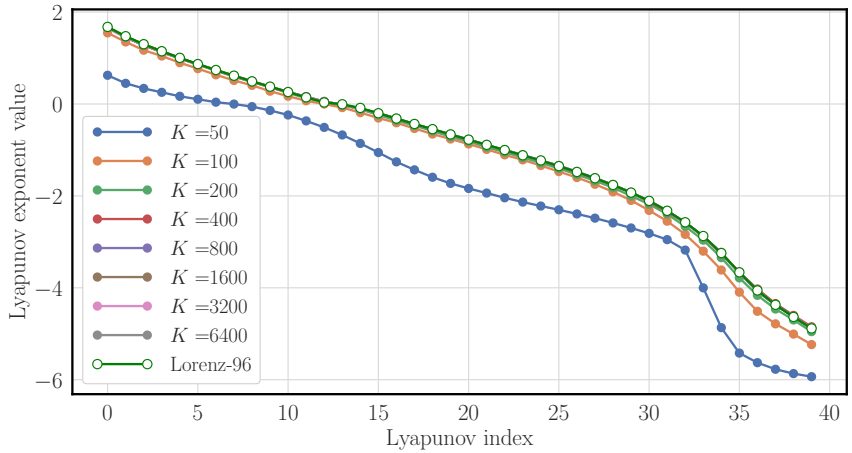
Model	Scheme	$\pi_{\frac{1}{2}}$	σ_q	λ_1
L96	Approximate	4.56 ± 0.06	0.08790 ± 210^{-5}	1.66 ± 0.02
L96	Full	4.24 ± 0.07	0.09152	1.66 ± 0.02
L05III	Approximate	4.06 ± 0.21	0.07720 ± 210^{-5}	1.03 ± 0.05
L05III	Full	3.97 ± 0.17	0.08024	1.03 ± 0.04

Dependence on the window length (L96)

- Approximate scheme
- Fully observed
- Significantly noisy observations $\mathbf{R} = \mathbf{I}$
- Variable window length, $\Delta t = 0.05$
- EnKS with $L = 4$
- 30 EM iterations

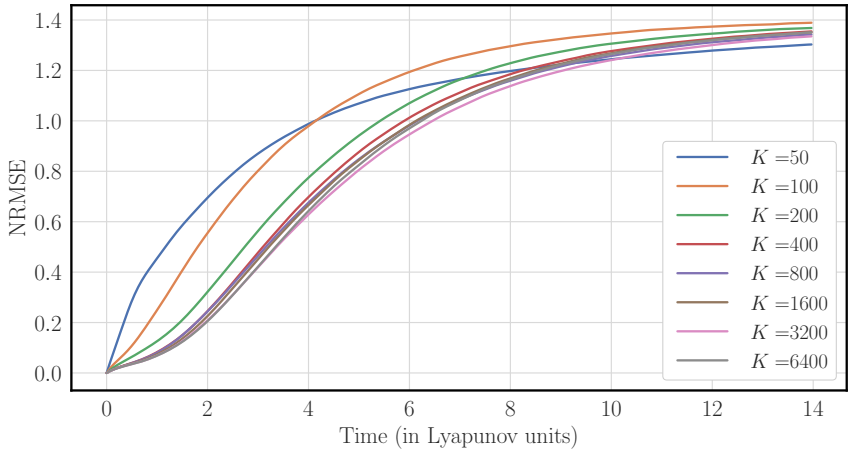


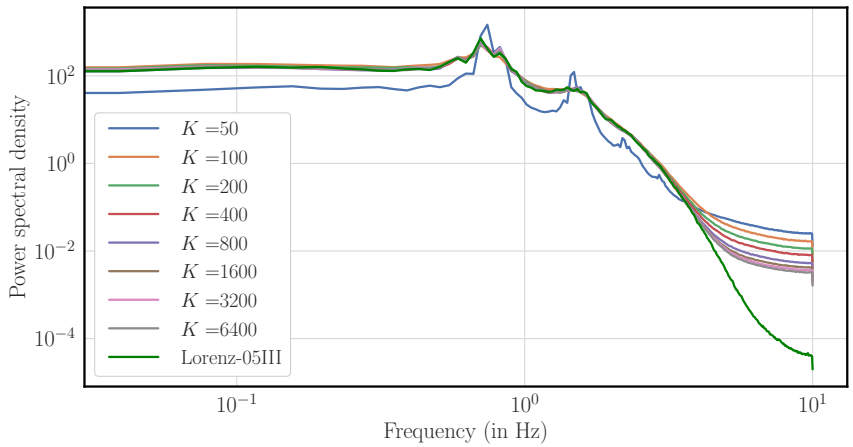


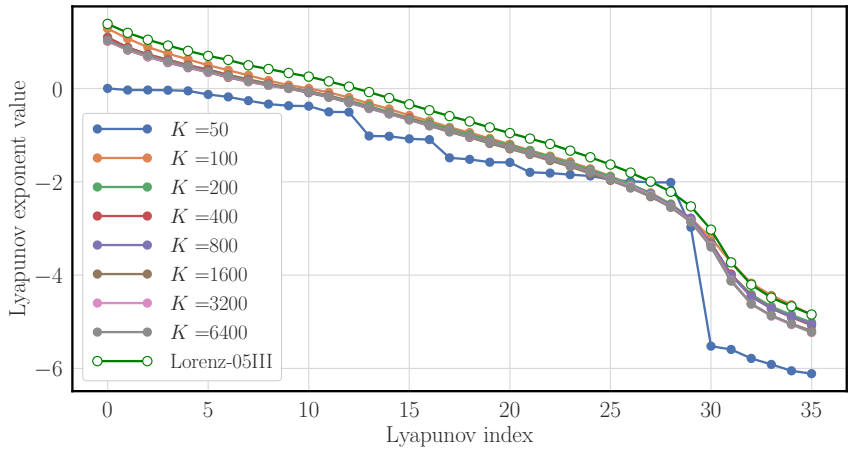


Dependence on the window length (L05III)

- Approximate scheme
- Observation of the coarse modes only
- Significantly noisy observations $\mathbf{R} = \mathbf{I}$
- Variable window length, $\Delta t = 0.05$
- EnKS with $L = 4$
- 30 EM iterations

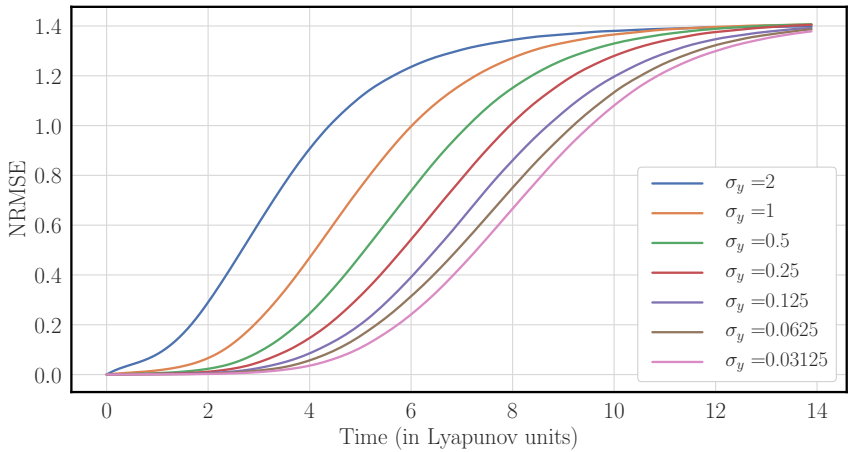


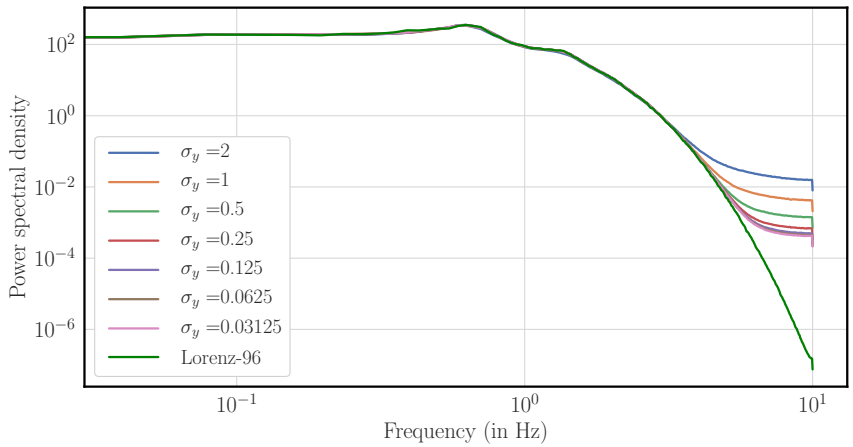


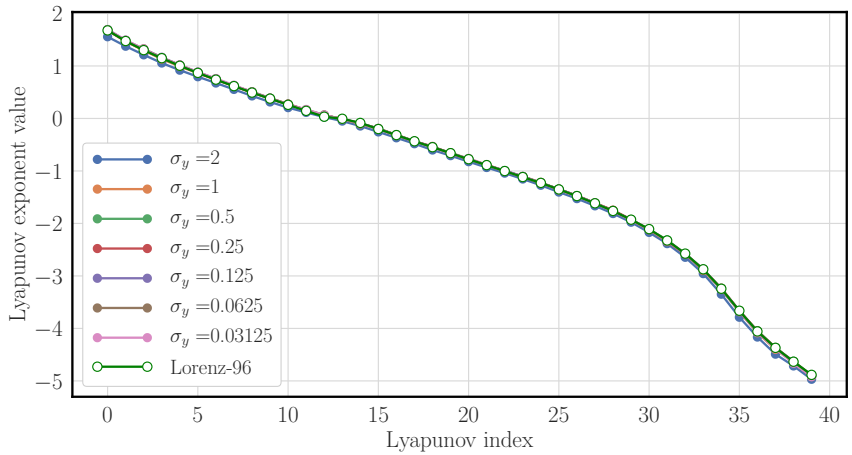


Dependence on the observation noise (L96)

- Approximate scheme
- Fully observed
- Variable observation variance $\mathbf{R} = \sigma_y^2 \mathbf{I}$
- $K = 5000$, $\Delta t = 0.05$
- EnKS with $L = 4$
- 30 EM iterations

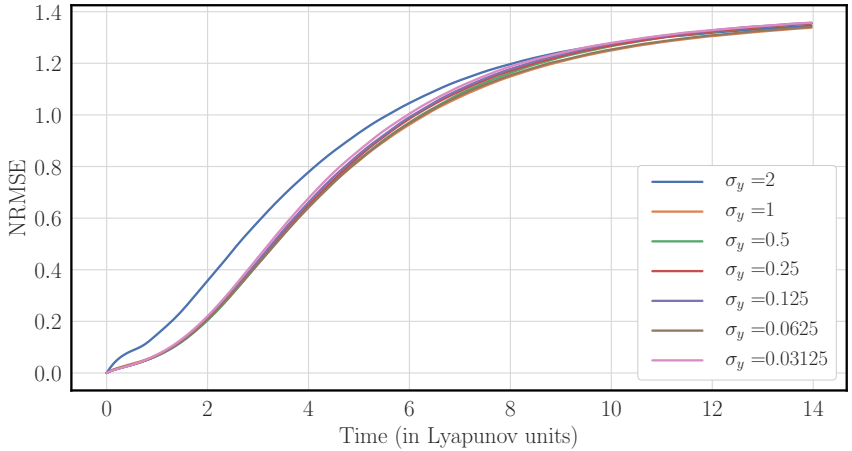


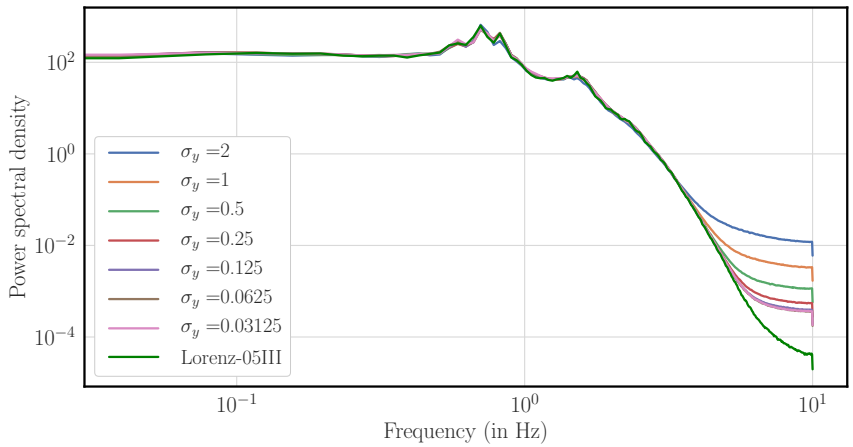


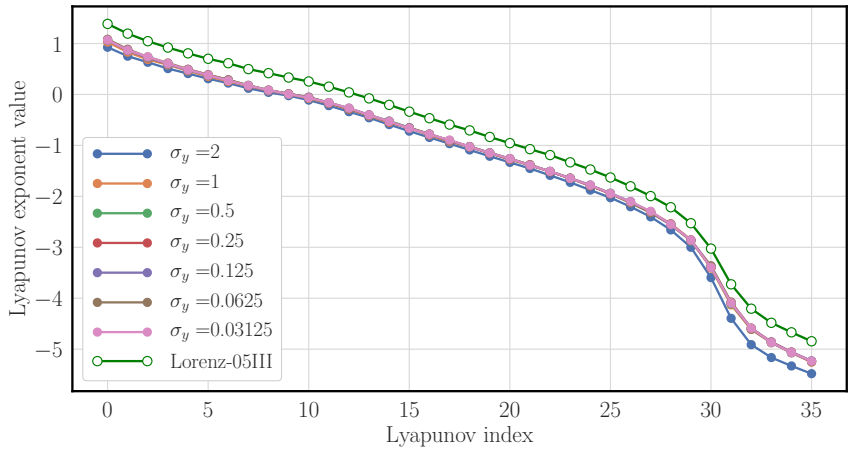


Dependence on the observation noise (L05III)

- Approximate scheme
- Observation of the coarse modes only
- Variable observation variance $\mathbf{R} = \sigma_y^2 \mathbf{I}$
- $K = 5000$, $\Delta t = 0.05$
- EnKS with $L = 4$
- 30 EM iterations

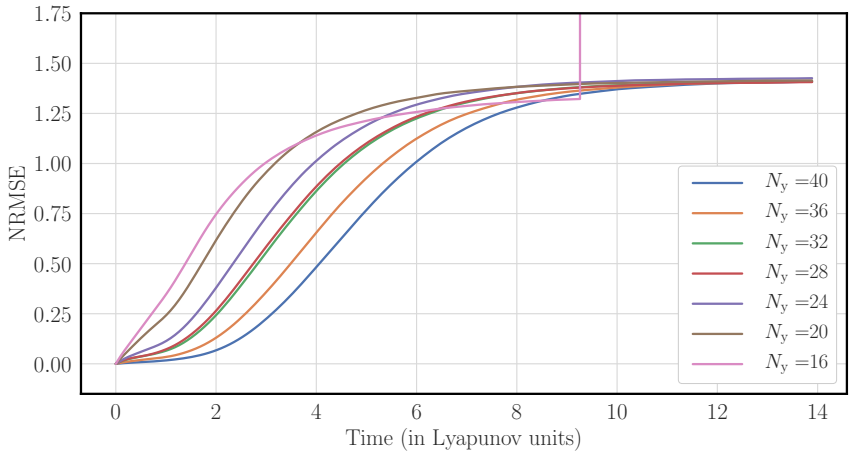


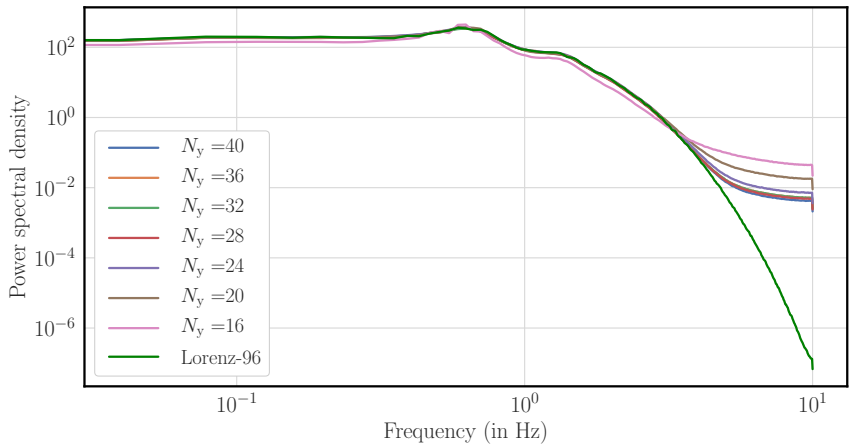


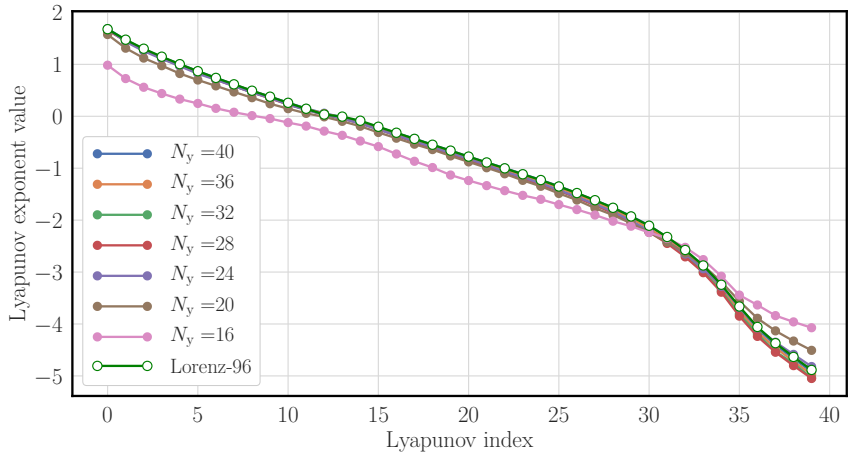


Dependence on the observation density (L96)

- Approximate scheme
- Variable observation density
- Significantly noisy observations $\mathbf{R} = \mathbf{I}$
- $K = 5000$, $\Delta t = 0.05$
- EnKS with $L = 4$
- 30 EM iterations

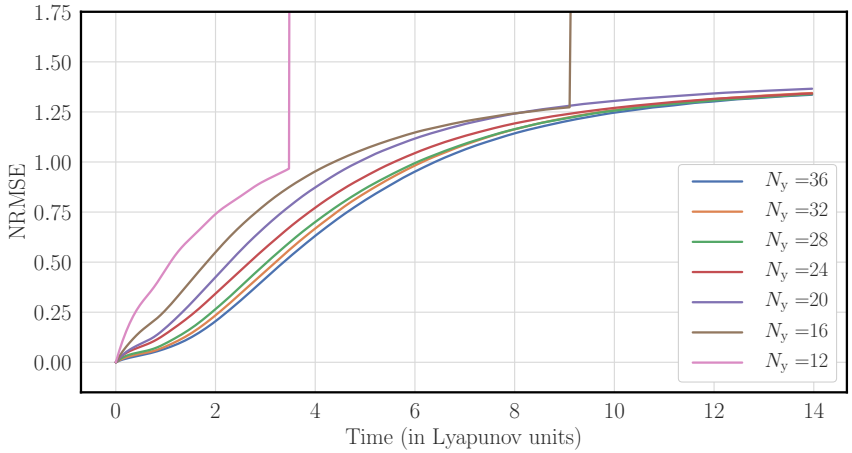


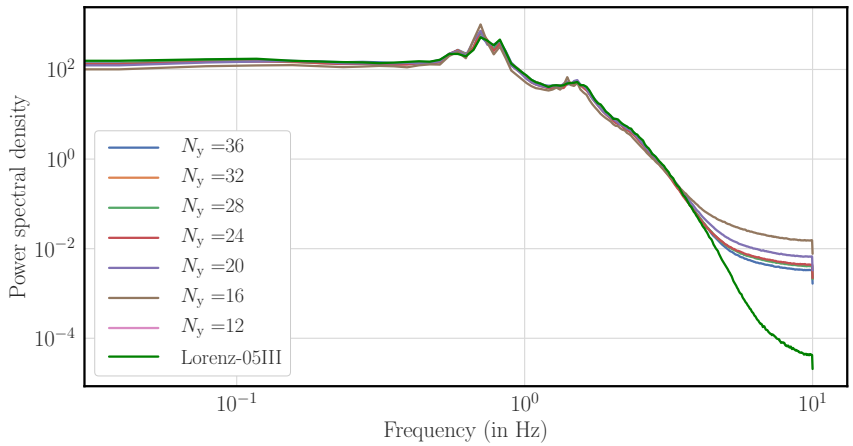


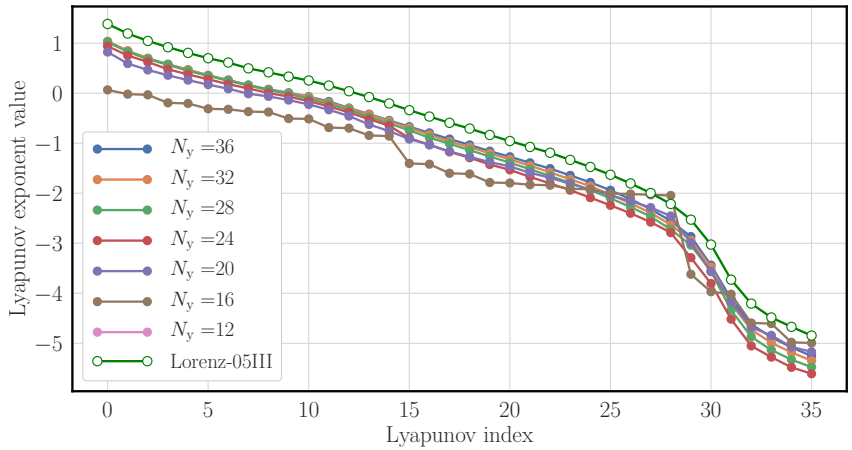


Dependence on the observation density (L05III)

- Approximate scheme
- Variable obs. of the coarse modes
- Significantly noisy observations $\mathbf{R} = \mathbf{I}$
- $K = 5000$, $\Delta t = 0.05$
- EnKS with $L = 4$
- 30 EM iterations







Outline

Conclusions

All results presented here are from [Bocquet et al. 2019; Brajard et al. 2020; Bocquet et al. 2020].

- Main messages:
  - Bayesian DA view on state and model estimation.
    DA can address goals assigned to ML but with partial & noisy observations.
  - Numerical costs of high-dimensional systems significantly reduced by locality and homogeneity assumptions.
  - The EM technique, full or approximate, is successful. Only coordinate minimization was shown to be successful so far in such context.
  - The method can handle very long training windows.
  - Successful on various 1D low-order models (L63, L96, KS, L05III) in presence of partial observation with significant noise.

- Open questions and technical hardships (non-exhaustive):
  - Non-autonomous dynamics?
  - Implicit integration schemes?
  - Online learning scheme?
  - More complex models?

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] M. Bocquet et al. "Bayesian inference of dynamics from partial and noisy observations using data assimilation and machine learning". In: (2020). eprint: arXiv:2001.06270.

[3] 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.

[4] 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: (2020). eprint: arXiv:2001.01520.

[5] S. L. Brunton, J. L. Proctor, and J. N. Kutz. "Discovering governing equations from data by sparse identification of nonlinear dynamical systems". In: *PNAS* (2016), p. 201517384.

[6] A. P. Dempster, N. M. Laird, and D. B. Rubin. "Maximum likelihood from incomplete data via the EM algorithm". In: *Journal of the Royal Statistical Society. Series B* (1977), pp. 1–38.

[7] P. D. Dueben and P. Bauer. "Challenges and design choices for global weather and climate models based on machine learning". In: *Geosci. Model Dev.* 11 (2018), pp. 3999–4009.

[8] R. Fablet, S. Ouala, and C. Herzet. "Bilinear residual neural network for the identification and forecasting of dynamical systems". In: *EUSIPCO 2018, European Signal Processing Conference*. Rome, Italy, 2018, pp. 1–5.

[9] Z. Ghahramani and S. T. Roweis. "Learning nonlinear dynamical systems using an EM algorithm". In: *Advances in neural information processing systems*. 1999, pp. 431–437.

[10] 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.

[11] Y. A. LeCun et al. "Efficient backprop". In: *Neural networks: Tricks of the trade*. Springer, 2012, pp. 9–48.

[12] Z. Long et al. "PDE-Net: Learning PDEs from Data". In: *Proceedings of the 35th International Conference on Machine Learning*. 2018.

[13] V. D. Nguyen et al. "EM-like Learning Chaotic Dynamics from Noisy and Partial Observations". In: *arXiv preprint arXiv:1903.10335* (2019).

[14] J. Paduart et al. "Identification of nonlinear systems using polynomial nonlinear state space models". In: *Automatica* 46 (2010), pp. 647–656.

[15] D. C. Park and Y. Zhu. "Bilinear recurrent neural network". In: *Neural Networks, 1994. IEEE World Congress on Computational Intelligence., 1994 IEEE International Conference on*. Vol. 3. 1994, pp. 1459–1464.

References II

[16] J. Pathak, B. Hunt, et al. "Model-Free Prediction of Large Spatiotemporally Chaotic Systems from Data: A Reservoir Computing Approach". In: *Phys. Rev. Lett.* 120 (2018), p. 024102.

[17] J. Pathak, Z. Lu, et al. "Using machine learning to replicate chaotic attractors and calculate Lyapunov exponents from data". In: *Chaos* 27 (2017), p. 121102.

[18] M. Pulido et al. "Stochastic parameterization identification using ensemble Kalman filtering combined with maximum likelihood methods". In: *Tellus A* 70 (2018), p. 1442099.

[19] P. N. Raanes, A. Carrassi, and L. Bertino. "Extending the square root method to account for additive forecast noise in ensemble methods". In: *Mon. Wea. Rev.* 143 (2015), pp. 3857–38730.

[20] P. R. Vlachas et al. "Forecasting of Spatio-temporal Chaotic Dynamics with Recurrent Neural Networks: a comparative study of Reservoir Computing and Backpropagation Algorithms". In: *arXiv preprint arXiv:1910.05266* (2019).

[21] Y.-J. Wang and C.-T. Lin. "Runge-Kutta neural network for identification of dynamical systems in high accuracy". In: *IEEE Transactions on Neural Networks* 9 (1998), pp. 294–307.

[22] G. C. G. Wei and M. A. Tanner. "A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms". In: *Journal of the American Statistical Association* 85 (1990), pp. 699–704.