Nonlin. Processes Geophys., 33, 401–424, 2026
https://doi.org/10.5194/npg-33-401-2026

Nonlinear Processes in Geophysics
Open Access
EGU

Elucidating the performance of data assimilation neural networks for chaotic dynamics

Marc Bocquet[1], **Tobias Sebastian Finn**[1], **Sibo Cheng**[1], **and Alban Farchi**[2]

[1]CEREA, ENPC, EDF R&D, Institut Polytechnique de Paris, Île-de-France, France
[2]ECMWF, Shinfield Park, Reading, UK

Correspondence: Marc Bocquet (marc.bocquet@enpc.fr)

Received: 19 January 2026 – Discussion started: 29 January 2026
Revised: 12 June 2026 – Accepted: 7 July 2026 – Published: 11 August 2026

Abstract. In supervised data assimilation machine learning emulation, the training data contain targets produced by an existing data assimilation scheme, such as analysis increments. By contrast, *data assimilation networks* were recently proposed to learn the analysis operator while they are embedded in the forecast–analysis cycle: their only targets are the true trajectory and the observations thereof. They are therefore trained to produce a stable and accurate sequential estimator, rather than to reproduce the output of a prescribed data assimilation algorithm. Conceptually more fundamental, yet computationally more challenging, such learned data assimilation scheme was shown to achieve accuracy comparable to that of the ensemble Kalman filter when applied to low-order chaotic dynamics. Strikingly, the same accuracy can be reached with a single state forecast instead of an ensemble, hence bypassing the need to explicitly represent forecast uncertainty.

In this study, we extend the investigation of such learned analysis operators beyond the preliminary experiments reported so far. First, we analyse the emergence of local patterns encoded in the operator, which accounts for the remarkable scalability of the approach to high-dimensional state spaces. Second, we assess the performance of the learned operators in stronger nonlinear regimes of the chaotic dynamics. We show that they can match the efficiency of the iterative ensemble Kalman filter, the baseline in this context, while avoiding the need for nonlinear iterative optimisation. Throughout the paper, we seek underlying reasons for the efficiency of the approach, drawing on insights from both machine learning and nonlinear data assimilation.

1 Introduction

Accurate prediction of geophysical flows relies on the continual correction of model trajectories using observations. This sequential data assimilation (DA) process is essential in high-dimensional chaotic systems, where errors amplify rapidly and model imperfections accumulate over time (Kalnay, 2003; Asch et al., 2016; Carrassi et al., 2018). In operational meteorology and oceanography, ensemble-based Kalman filters and ensemble variational methods provide reliable and well-understood frameworks for these updates, but they remain computationally demanding and rely on explicit representations of flow-dependent forecast error covariances.

Recent developments have suggested that part of this complexity can be replaced by learned analysis operators. In supervised DA machine learning emulation, the training dataset contains targets produced by an existing DA scheme, such as analysis increments (e.g., Härter and de Campos Velho, 2012; Cintra and de Campos Velho, 2018; Maddy et al., 2024). By contrast, one could learn the analysis operator while it is embedded in the forecast–analysis cycle: its only targets are the true trajectory and the observations thereof (McCabe and Brown, 2021). Such analysis operator is therefore trained to produce a stable and accurate sequential estimator, rather than to reproduce the output of a prescribed DA algorithm. Within this latter approach, termed *data assimilation networks* (DANs, Boudier et al., 2023), a surprising result has emerged: as demonstrated by Bocquet et al. (2024), a learned analysis operator can match the accuracy of a well-tuned ensemble Kalman filter (EnKF) even when it uses only single forecast trajectories, without any ensemble.

This result challenges the long-standing assumption that explicit ensemble representations are indispensable to estimate flow-dependent uncertainties in chaotic systems.

Explaining this phenomenon is a central question for DA methodology. Initial investigations suggest that the learned operator implicitly reconstructs aspects of the analysis error covariances normally diagnosed from an ensemble, effectively uncovering key uncertainty directions from the forecast state alone (Bocquet et al., 2024). This behaviour is consistent with viewing the full DA cycle as a random dynamical system, for which generalised forms of the multiplicative ergodic theorem (Oseledec, 1968) offer a state-dependent structure linking model trajectories to dominant error-growth directions. At the same time, numerical experiments show that the neural network is likely to identify local patterns rather than memorising global states, which explains why its performance scales to larger systems and remains robust across different model dimensions.

This paper deepens the investigation of these mechanisms. In Sect. 2, we dive into the methodology of DANs, and recall the main results and questions raised in Bocquet et al. (2024). In Sect. 3, building on a more thorough analysis of the performance dependence on the batch size, dataset length, and the number of assimilation cycles used during backpropagation, we study how the operator behaves when interpreted as a diagnostic of uncertainty, and we propose methods to expose and interpret the local structures it extracts. In Sect. 4, we test the limits of the approach in more strongly nonlinear regimes, where the iterative ensemble Kalman filter (Sakov et al., 2012) and its generalisations often serve as the most accurate baselines. We show that the learned operator can reach comparable performance without requiring an ensemble or a nonlinear optimisation, and we offer a data assimilation-based interpretation for this ability. Section 5 presents our conclusions. Supporting numerical and mathematical results are collated in the appendices of this paper.

Throughout the paper, we will illustrate our results with the Lorenz-96 model (L96, Lorenz and Emanuel, 1998). More broadly, our goal is not only to document the performance of the learned analysis operators but also to clarify the mechanisms that underlie them, thereby contributing to the growing theoretical understanding of how deep learning and sequential data assimilation interact in chaotic geophysical systems (Cheng et al., 2023).

2 Theory and methods

In this section, we provide a deeper description of the problem, its context, and its mathematical formulation.

2.1 Sequential data assimilation for chaotic dynamics

Mathematically, data assimilation (DA), and in particular filtering algorithms, are meant to accurately estimate the state vector $\mathbf{x}_k^{\mathrm{t}} \in \mathbf{E}_{\mathrm{x}} \triangleq \mathbb{R}^{N_{\mathrm{x}}}$ of a physical system, where "t" stands for truth, at times t_k for $k = 1, \ldots, K$ along a trajectory of the dynamical system. These states are evolved according to

$$\mathbf{x}_{k+1}^{\mathrm{t}} = \mathcal{M}\left(\mathbf{x}_k^{\mathrm{t}}\right), \tag{1a}$$

where $\mathcal{M}$ is the integrated model over $t_{k+1} - t_k = \Delta_t$. The dynamical system $\mathcal{M}$ is assumed to be chaotic, such as for most geofluids, which is a prime incentive for frequently updating our knowledge of the system. It is furthermore assumed ergodic and autonomous, i.e. does not explicitly depend on time. Furthermore, the physical system is observed through

$$\mathbf{y}_k = \mathcal{H}_k\left(\mathbf{x}_k^{\mathrm{t}}\right) + \boldsymbol{\varepsilon}_k, \quad \boldsymbol{\varepsilon}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k), \tag{1b}$$

where $\mathbf{y}_k \in \mathbf{E}_{\mathrm{y}}^k \triangleq \mathbb{R}^{N_{\mathrm{y},k}}$ is the observation vector at t_k obtained from the hidden state $\mathbf{x}_k^{\mathrm{t}}$ via an observation operator $\mathcal{H}_k$, and perturbed by a white-in-time Gaussian noise $\boldsymbol{\varepsilon}_k$ of mean $\mathbf{0}$ and covariance matrix $\mathbf{R}_k$. This very common but simplified formulation of the filtering problem with additive Gaussian noise is sufficient for the goals of this paper.

A filtering DA scheme estimates the hidden state $\mathbf{x}_k^{\mathrm{t}}$ at t_k from the observations available up to that time. The analysis $\mathbf{x}_k^{\mathrm{a}}$ is a point estimator, such as the posterior mean or maximum a posteriori estimate, of the conditional probability density function $p(\mathbf{x}_k^{\mathrm{t}} | \mathbf{y}_k, \mathbf{y}_{k-1}, \ldots, \mathbf{y}_1)$. A *sequential* filtering DA scheme infers $\mathbf{x}_k^{\mathrm{a}}$ from $\mathbf{y}_k$ and from background information about the state at t_{k-1} (possibly brought forward to t_k using $\mathcal{M}$).

2.2 Learning the data assimilation analysis

2.2.1 Training scheme

The approach developed in Bocquet et al. (2024), subsequently referred to as Boc24, in the wake of McCabe and Brown (2021) and Boudier et al. (2023) is summarised in the following since it is the foundation of the present paper. The analysis step of the DA scheme is assumed to be given by the (incremental) analysis operator $\mathbf{a}_{\boldsymbol{\theta}}$, typically a (deep) neural network, which depends on a set of weights and biases, stacked in the $\boldsymbol{\theta}$ vector, and which is defined, at time t_k, through

$$\mathbf{x}_k^{\mathrm{a}} = \mathbf{x}_k^{\mathrm{f}} + \mathbf{a}_{\boldsymbol{\theta}}\left(\mathbf{x}_k^{\mathrm{f}}, \mathbf{H}_k^{\intercal} \mathbf{R}_k^{-1} \boldsymbol{\delta}_k\right), \tag{2a}$$

$$\boldsymbol{\delta}_k \triangleq \mathbf{y}_k - \mathcal{H}_k\left(\mathbf{x}_k^{\mathrm{f}}\right), \tag{2b}$$

where $\boldsymbol{\delta}_k$ is the innovation, $\mathbf{x}_k^{\mathrm{a}}$ is the analysis state mentioned previously, and $\mathbf{x}_k^{\mathrm{f}}$ is the forecast state, to be defined shortly. $\mathbf{H}_k$ is the tangent linear operator of $\mathcal{H}_k$. Should $\mathbf{a}_{\boldsymbol{\theta}}$ be a function of $\boldsymbol{\delta}_k$ rather than $\mathbf{H}_k^{\intercal} \mathbf{R}_k^{-1} \boldsymbol{\delta}_k$, it could only handle static observation configurations, and would require to be retrained whenever that configuration changes. Although not the focus of this paper, this important issue is circumvented here by using the mapping from observation to state space $\mathbf{H}_k^{\intercal} \mathbf{R}_k^{-1}$,

such that both inputs of $\mathbf{a}_{\boldsymbol{\theta}}$ are in the same static state space $\mathbf{E}_{\mathrm{x}}$. For compactness, we define the projected innovation

$$\boldsymbol{\zeta}_k \triangleq \mathbf{H}_k^{\intercal}\mathbf{R}_k^{-1}\boldsymbol{\delta}_k = \mathbf{H}_k^{\intercal}\mathbf{R}_k^{-1}\left(\mathbf{y}_k - \mathbf{H}_k\left(\mathbf{x}_k^{\mathrm{f}}\right)\right). \tag{2c}$$

The DAN analysis operator thus receives two fields in state space, $\mathbf{x}_k^{\mathrm{f}}$ and $\boldsymbol{\zeta}_k$.

In the DA forecast step, the analysis state is propagated to the subsequent date through

$$\mathbf{x}_{k+1}^{\mathrm{f}} = \mathcal{M}\left(\mathbf{x}_k^{\mathrm{a}}\right). \tag{3}$$

The operator $\mathbf{a}_{\boldsymbol{\theta}}$ is trained by comparing the analysis states to the true states, through the loss

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{r=1}^{N_{\mathrm{r}}}\sum_{k=1}^{N_{\mathrm{c}}}\left\|\mathbf{x}_k^{\mathrm{t},r} - \mathbf{x}_k^{\mathrm{a},r}(\boldsymbol{\theta})\right\|^2, \tag{4}$$

where $\mathbf{x}_k^{\mathrm{t},r}$ are the state vectors of the true trajectory, indexed by the time index k and a trajectory index r as N_{r} of them are processed concurrently in the training of $\mathbf{a}_{\boldsymbol{\theta}}$.

The N_{c} parameter counts the number of cycles of each DA run. It can potentially be infinite since trajectories can be generated online as the training progresses. The implementation of similar losses is detailed in McCabe and Brown (2021), Boudier et al. (2023), and Sect. II of Boc24. This notably requires to truncate backpropagation in time (Tang and Glass, 2018) which restricts the dependence on $\boldsymbol{\theta}$ over $N_{\mathrm{iter}} \ll N_{\mathrm{c}}$ cycles only so as to reduce the computational cost and the excessive GPU memory (VRAM) requirement. Alternative losses based on probability density functions (pdf) diagnostics are possible and discussed in, e.g., Boudier et al. (2023). We have shown that a semi-supervised loss where the training dataset reduces to the sparse and noisy observations is also possible by generalising a proposal of McCabe and Brown (2021), but is out of the scope of this paper.

2.2.2 Accuracy of the discovered data assimilation scheme

Note that an ensemble variant of the update Eq. (2) was first considered in McCabe and Brown (2021), Boudier et al. (2023), and later by Boc24. Experimenting with the same low-order chaotic model, they found an accuracy of the learned DA scheme close to that of a well-tuned EnKF, which is per se very promising. In those experiments, DAN is built on an ensemble of analyses and forecasts. However, Boc24 demonstrated that this accuracy remains unchanged when these ensembles are reduced to a single state. This is a very surprising result since it is expected (and verified in the L96 context) that the EnKF-like methods critically relying on an ensemble representing the *errors of the day*, have a significant edge over other sequential DA methods that leverage a single forecast state such as 3D-Var and 4D-Var (see Bocquet and Sakov, 2013, for a quantitative comparison with the same model). This explains why numerical weather prediction (NWP) centres operating 4D-Var actually rely on an ensemble of such 4D-Var, a technique called EDA (see, e.g., Chap. 7 in Asch et al., 2016; Bannister, 2017, and references within), or use information from a concurrent EnKF (Buehner et al., 2015). That is why achieving the accuracy of a well-tuned EnKF with a single forecast state should have far-reaching implications on the mechanisms and designs of DA algorithms for chaotic dynamics, and warrants investigating the reason for such feat.

2.3 Investigating the reason for this efficiency

To unveil the mechanisms leveraged by the learned analysis operator to achieve this accuracy, Boc24 performed a linearisation of the operator $\mathbf{a}_{\boldsymbol{\theta}}$ in the projected innovation $\boldsymbol{\zeta}$:

$$\mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\zeta}) \approx \mathbf{P}^{\mathrm{a}}(\mathbf{x})\boldsymbol{\zeta}, \tag{5}$$

where $\mathbf{P}^{\mathrm{a}}(\mathbf{x})$ is the resulting linear operator that acts on $\boldsymbol{\zeta}$. Comparing with the linearisation of the classical Kalman update, it must coincide with the analysis error covariance matrix which could be associated by formal analogy to the learned analysis operator $\mathbf{a}_{\boldsymbol{\theta}}$, hence the notation. This covariance matrix was numerically obtained in Boc24 through a linear regression in between a large ensemble of $\boldsymbol{\zeta}$ samples (corresponding to the projected innovations $\mathbf{H}^{\intercal}\mathbf{R}^{-1}\boldsymbol{\delta}$) and $\mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\zeta})$ outputs. It was found that this error covariance matrix is remarkably close to that of a well-tuned EnKF, especially for its dominant eigenvectors (which carry most of the uncertainty in the analysis).

Hence, the performance of DAN, as reported in Boc24, must be to a large extent due to its ability to infer flow-dependent effective analysis error covariances from the forecast state, as revealed by the local dependence of $\mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x}^{\mathrm{f}}, \boldsymbol{\zeta})$ on the projected innovation $\boldsymbol{\zeta}$. Such an ability is pivotal for maintaining the accuracy of sequential DA over time. In the present paper, we will sometimes reason in terms of the forecast error covariance matrix $\mathbf{P}^{\mathrm{f}}$ in place of $\mathbf{P}^{\mathrm{a}}$, since the flow dependence is more naturally encoded in $\mathbf{P}^{\mathrm{f}}$, while $\mathbf{P}^{\mathrm{a}}$ is obtained after conditioning on the observations. Moreover, in the linear/Gaussian case, these two covariance matrices are straightforwardly related by

$$\left(\mathbf{P}^{\mathrm{a}}\right)^{-1} = \left(\mathbf{P}^{\mathrm{f}}\right)^{-1} + \mathbf{H}^{\intercal}\mathbf{R}^{-1}\mathbf{H}, \tag{6}$$

assuming invertibility of the error covariance matrices. Beyond the linear/Gaussian case, perturbing the innovations and applying DAN can generate an empirical analysis ensemble; propagating this ensemble with the model $\mathcal{M}$ then provides a way to diagnose the corresponding forecast error covariances.

Other alternatives to ensemble forecasting to estimate the flow-dependent error statistics, such as deriving the dynamics of the statistical moments of the errors (Pannekoucke et

al., 2016, 2018), or estimating these dynamics through machine learning (Pannekoucke and Fablet, 2020; Sacco et al., 2024; Lu, 2025) are either numerically very costly or in their infancy.

Boc24 conjectured that DAN's ability to assess the flow-dependent error statistics can be fundamentally explained by the existence of an ad-hoc *multiplicative ergodic theorem* (MET). From the seminal MET result by Oseledec (1968), we know that, for an autonomous ergodic dynamical system such as $\mathcal{M}$, there exists, for almost every state on the attractor, a measurable mapping from the state to the corresponding Oseledets subspaces. Generalising, one can consider the whole sequential DA process as a dynamical system on its own (Carrassi et al., 2008). Such DA process is not autonomous because it indirectly depends on the truth trajectory and the time-dependent observation operators. Moreover, it is a random process, since stochasticity is injected via the noisy observations. It turns out that generalised variants of the MET for non-autonomous random dynamics are possible (Arnold, 1998; Chekroun et al., 2011; Flandoli and Tonello, 2021; Ghil and Sciamarella, 2023) and are potentially applicable to such sequential DA process. Hence, Boc24 conjectured that $\mathbf{a}_{\boldsymbol{\theta}}$ can exploit such state-dependent information to infer an effective analysis error covariance matrix, together with how to process this information and combine it with the innovation.

To explain the efficiency of $\mathbf{a}_{\boldsymbol{\theta}}$, one may suggest that the neural network memorises global configurations of the forecast state, with very limited ability to generalise. On the contrary, Boc24 showed via indirect scaling experiments that $\mathbf{a}_{\boldsymbol{\theta}}$ learns to identify local patterns (i.e. with a limited range in space), which was made easier by the architecture of $\mathbf{a}_{\boldsymbol{\theta}}$ being a residual convolutional neural network. Indeed, when the dimension N_{x} of the L96 state space is increased, and new $\mathbf{a}_{\boldsymbol{\theta}}$ operators are learned but with a fixed number of weights and biases of the backbone architecture, the accuracy remains that of a well tuned EnKF of matching dimension. Yet, in the large N_{x} limit, $\mathbf{a}_{\boldsymbol{\theta}}$ with the same number of degrees of freedom should not be able to memorise increasingly numerous global patterns. Hence, $\mathbf{a}_{\boldsymbol{\theta}}$ must extract local patterns, and a limited number of them. This is further supported by learning $\mathbf{a}_{\boldsymbol{\theta}}$ with the original L96 dimension, $N_{\mathrm{x}} = 40$, but applying it to L96 with significantly different N_{x} in successful DA runs (this is allowed by the convolutional architecture which does not explicitly depend on N_{x}), performing on par with a well tuned EnKF. Hence, any local pattern learned in the case $N_{\mathrm{x}} = 40$, must still be spotted by $\mathbf{a}_{\boldsymbol{\theta}}$ where $N_{\mathrm{x}} \neq 40$. This outcome is consistent with the existence of such local patterns, since L96 is an extensive model when N_{x} is increased, with the number of nonlinear interacting waves in the model being proportional to N_{x}. This neat scalability has limitations. If increasing the dimension changed the local statistics or local dynamical balances of the model, for example through a substantial change of spectral slope or local instability mechanisms, a DAN trained at the smaller dimension should not be expected to generalise without fine-tuning, additional training, or architectural adaptation.

3 Exploration of data assimilation networks

With the previously established context in mind, we now explore what DANs ($\mathbf{a}_{\boldsymbol{\theta}}$) learn in mildly nonlinear regimes. The neural network associated to $\mathbf{a}_{\boldsymbol{\theta}}$ implements Eq. (2). The architecture for $\mathbf{a}_{\boldsymbol{\theta}}$ we choose in the present paper is the same as the one reported in Boc24, i.e. a simple residual convolutional neural network that accounts well for the spatial homogeneity of L96 (and hence its statistical stationarity). For the sake of self-sufficiency, Appendix A describes this architecture. All the training tasks are carried out over a training dataset with a minimisation of the loss controlled by computing the loss over a validation dataset.

3.1 Numerical setup

The numerical experiments of the present paper are performed on L96, a chaotic model abundantly used for benchmarking new sequential data assimilation algorithms. As a reminder, L96 represents a mid-latitude zonal circle of the global atmosphere. It is governed by $N_{\mathrm{x}} = 40$ ordinary differential equations:

$$\frac{\mathrm{d}x_n}{\mathrm{d}t} = (x_{n+1} - x_{n-2})x_{n-1} - x_n + F\,, \tag{7}$$

where $F = 8$, and with cyclic boundary conditions. The resulting model is chaotic with 13 positive and 1 neutral Lyapunov exponents. Its Lyapunov time, defined as the inverse of the first Lyapunov exponent, is about 0.60, which corresponds to 3 d of a typical weather forecasting model (Lorenz and Emanuel, 1998).

Although the results are generalisable to sparse observations, the observation operator will mostly be chosen to be the identity $\mathbf{H}_k \overset{\Delta}{=} \mathbf{I}_{\mathrm{x}}$. The observations are read off the true state and perturbed with an unbiased white-in-time Gaussian additive noise of covariance matrix $\mathbf{R}_k \overset{\Delta}{=} \mathbf{I}_{\mathrm{x}}$ following Eq. (1b). The time-step between observation batches and updates is set to $\Delta_t = t_{k+1} - t_k = 0.05$ for all experiments with the exception of Sect. 4.1. This will be our reference DA setup. Examples of alternative sparseness and noise levels are given in Boc24, but here as well when relevant.

All the $\mathbf{a}_{\boldsymbol{\theta}}$ operators learned in this configuration are subsequently evaluated with a time-averaged analysis root mean square error (RMSE) whose mean square is averaged over the N_{x} variables. This RMSE is assessed over a dataset of independent test trajectories, similarly to traditional DA twin experiments. For brevity, this score computed for each trained DAN scheme will simply be called *test RMSE* of the DAN scheme in the rest of this paper.

In this configuration, running a well-tuned EnKF in a twin experiment and comparing its analysis to the truth yield a test

RMSE between 0.18 and 0.20 depending on the ensemble size N_e and whether localisation is used or not. By contrast, a basic but well-tuned 3D-Var or a reasonably short window basic but well tuned 4D-Var yields a test RMSE of about 0.40. They largely underperform the EnKF because they fail to capture the errors of day (Bocquet and Sakov, 2013; Fillion et al., 2018).

In order to be able to perform a large number of training experiments on a limited number of GPUs and limited VRAM, we carried out a sensitivity study on the batch size, the size of the datasets, and the backpropagation truncation, beyond the restricted set of experiments reported in Boc24. Since the results are technical and mostly of practical interest, they are reported in Appendix B. They were nonetheless instrumental in obtaining the main numerical results of this paper.

3.2 Linearisation in the innovations

In this section, we discuss the relevance of expanding $\mathbf{a}_{\boldsymbol{\theta}}$ linearly in the innovations, as in Eq. (5). We recall that the focus is on the analysis step of the DAN process, where $N_e = 1$, i.e. a single forecast state is propagated in between updates. Under the additional Gaussian and quasi-linear assumptions used in this subsection, a *close to optimal* analysis associated with the Kalman update is the posterior mean which coincides with the maximum a posteriori of the conditional pdf, hence by the minimum of the cost function associated to the analysis. In the following, the observation operator is assumed linear (or linearised) for simplicity, i.e. $\mathcal{H} \triangleq \mathbf{H}$.

3.2.1 A Gaussian standpoint on the analysis

Here, we further assume that the background errors are Gaussian. Nonetheless, as opposed to a basic 3D-Var, the background error covariance matrix depends on the forecast state. Hence, the typical analysis cost function associated to the analysis at any given time step has the form:

$$\mathcal{J}(\mathbf{x}) = \frac{1}{2}\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2_{\mathbf{R}^{-1}} + \frac{1}{2}\left\|\mathbf{x} - \mathbf{x}^{\mathrm{f}}\right\|^2_{(\mathbf{P}^{\mathrm{f}}(\mathbf{x}^{\mathrm{f}}))^{-1}}. \tag{8}$$

As a consequence, $\mathcal{J}$ is quadratic in $\mathbf{x}$, strictly convex, and its minimum argument is (Daley, 1991)

$$\begin{aligned}\mathbf{x}^{\mathrm{a}} &= \mathbf{x}^{\mathrm{f}} + \left[\mathbf{H}^{\intercal}\mathbf{R}^{-1}\mathbf{H} + \left(\mathbf{P}^{\mathrm{f}}(\mathbf{x}^{\mathrm{f}})\right)^{-1}\right]^{-1}\mathbf{H}^{\intercal}\mathbf{R}^{-1}\left(\mathbf{y} - \mathbf{H}\mathbf{x}^{\mathrm{f}}\right)\\ &= \mathbf{x}^{\mathrm{f}} + \mathbf{P}^{\mathrm{a}}(\mathbf{x}^{\mathrm{f}})\boldsymbol{\zeta},\end{aligned} \tag{9}$$

where the projected innovation $\boldsymbol{\zeta}$ has been defined by Eq. (2c). Equation (9) matches the linearisation Eq. (5). The purpose of this derivation is to show that, under Gaussian/quasi-linear assumptions, DAN is likely to solve the optimisation problem Eq. (8), that its update follows Eq. (9), and that its Jacobian with respect to $\boldsymbol{\zeta}$ plays the role of an effective analysis error covariance matrix $\mathbf{P}^{\mathrm{a}}(\mathbf{x}^{\mathrm{f}})$. Whether this local interpretation is sufficient is then tested numerically.

3.2.2 Numerical evidence

To numerically test whether such a linearisation is a good approximation for $\mathbf{a}_{\boldsymbol{\theta}}$, we created a modified DAN which explicitly learns an effective map $\mathbf{x}^{\mathrm{f}} \longmapsto \mathbf{P}^{\mathrm{a}}(\mathbf{x}^{\mathrm{f}})$ and linearly combines it with the projected innovations, with the goal to strictly follow the update equation Eq. (9). Details on our scalable implementation of Eq. (9), for such a DAN linear in the projected innovations (from now on called linear-in-$\boldsymbol{\zeta}$ DAN), can be found in Appendix C. One must keep in mind that this linear-in-$\boldsymbol{\zeta}$ DAN is numerically inefficient since it requires to build a representation of the covariance matrix $\mathbf{P}^{\mathrm{a}}$ before being applied to the projected innovations, an operation which is likely to be achieved with our standard DAN without ever computing $\mathbf{P}^{\mathrm{a}}$ explicitly.

This *deconstruction* of $\mathbf{a}_{\boldsymbol{\theta}}$ directly connects with the heuristic developed by Sacco et al. (2024) and Sakov (2025), where the mapping $\mathbf{x}^{\mathrm{f}} \longmapsto \mathbf{P}^{\mathrm{f}}(\mathbf{x}^{\mathrm{f}})$ is learned and then successfully used within a classical EnKF. Numerically, the linear-in-$\boldsymbol{\zeta}$ DAN scheme turns out to be as accurate as a well-tuned EnKF with $N_e = 40$ in the reference setup ($\Delta_t = 0.05$ in particular), with a test RMSE of ~ 0.19, which is in line with the results by Sacco et al. (2024) and Sakov (2025). Hence, we can claim that the good performance of DAN *in this mildly nonlinear regime* importantly relies on the (implicit) estimation of the mapping $\mathbf{x}^{\mathrm{f}} \longmapsto \mathbf{P}^{\mathrm{f}}(\mathbf{x}^{\mathrm{f}})$, and subsequently $\mathbf{x}^{\mathrm{f}} \longmapsto \mathbf{P}^{\mathrm{a}}(\mathbf{x}^{\mathrm{f}})$.

3.2.3 On the importance of the $\mathbf{x}^{\mathrm{f}} \longmapsto \mathbf{P}^{\mathrm{f}}(\mathbf{x}^{\mathrm{f}})$ map

Let us focus on Sakov (2025), whose algorithmic constructions targeted at estimating $\mathbf{P}^{\mathrm{f}}(\mathbf{x}^{\mathrm{f}})$ are especially transparent and whose numerical tests were carried out in the reference setup described in Sect. 3.1. Their Algorithm A1 proceeds as follows: (i) the state $\mathbf{x}^{\mathrm{f}}$ is backtracked by T time steps; (ii) the tangent linear model, evaluated along the resulting state trajectory from t_{-T} to t_0, is applied to a matrix of state perturbations $\varepsilon\mathbf{I}_{\mathrm{x}}$; and (iii) the resulting perturbations at time t_0 are used to estimate $\mathbf{P}^{\mathrm{f}}(\mathbf{x}^{\mathrm{f}})$. This procedure is closely related to the Assimilation in the Unstable Space (AUS) methods (Palatella et al., 2013; Carrassi et al., 2022), since for sufficiently large T the output of the tangent linear model at t_0 provides a square-root factor of the backward Lyapunov vectors. But it is already sufficient to yield a competitive analysis relying on the errors of the day estimated through $\mathbf{P}^{\mathrm{f}}(\mathbf{x}^{\mathrm{f}})$ which notably relies on a single state.

Algorithm A2 of Sakov (2025) also backtracks the state $\mathbf{x}^{\mathrm{f}}$ by T time steps but differs in the subsequent step: instead of propagating perturbations, it applies a (square-root) Kalman filter to an initial covariance matrix $\varepsilon^2\mathbf{I}_{\mathrm{x}}$ from t_{-T} to t_0, yielding a more refined estimate of $\mathbf{P}^{\mathrm{f}}(\mathbf{x}^{\mathrm{f}})$ at t_0. This yields an even

more accurate DA algorithm on par with a well tuned EnKF, again relying on $\mathbf{P}^{\mathrm{f}}(\mathbf{x}^{\mathrm{f}})$ assessed from a single state.

Bocquet et al. (2017) and Bocquet and Carrassi (2017) have laid the theoretical grounds to understand why these principled demonstrations are successful. Bocquet et al. (2017) focused on a degenerate Kalman filter, equivalent to an EnKF in Gaussian and quasi-linear conditions. In this setting, the covariance evolution is formally decoupled from the state update, with the important caveat that the forecast error covariance $\mathbf{P}^{\mathrm{f}}$ depends on the state $\mathbf{x}^{\mathrm{f}}$ at time t_0. Indeed, Bocquet et al. (2017) showed that, asymptotically (i.e. for large T), $\mathbf{P}^{\mathrm{f}}$ depends on the system dynamics only through the Lyapunov vectors. By the MET, the corresponding Oseledets subspaces are measurable functions of the state $\mathbf{x}^{\mathrm{f}}$ at t_0 (for almost every state on the attractor). Hence, following Bocquet et al. (2017), Algorithm A2 by Sakov (2025) can be directly interpreted as the single state-dependent covariance propagation of a degenerate Kalman filter. In the case of nonlinear dynamics, enforcing the dependence of the error covariance $\mathbf{P}^{\mathrm{f}}$ on $\mathbf{x}^{\mathrm{f}}$ is even more beneficial since the covariance evolution now explicitly depends on $\mathbf{x}^{\mathrm{f}}$.

These theoretical results and proof experiments provide a rationale for the existence of a map $\mathbf{x}^{\mathrm{f}} \longmapsto \mathbf{P}^{\mathrm{f}}(\mathbf{x}^{\mathrm{f}})$ and its importance in estimating the errors-of-the-day in DA schemes based on a single forecast state $N_{\mathrm{e}} = 1$.

3.3 Patterns

In this section, we visualise a footprint of the local patterns leveraged by DAN to make its inferences. To that end, we study the dependence of $\nabla_{\boldsymbol{\zeta}} \mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\zeta})_{|\boldsymbol{\zeta}=\mathbf{0}} \approx \mathbf{P}^{\mathrm{a}}(\mathbf{x})$ on $\mathbf{x}$, i.e. the forecast state. This can be seen as a linear-in-$\boldsymbol{\zeta}$ sensitivity analysis of $\mathbf{a}_{\boldsymbol{\theta}}$ with respect to its inputs.

3.3.1 Mean marginal gain tensor

Specifically, the sensitivity analysis will focus on Jacobians of the analysis operator with respect to $\mathbf{x}$:

$$\boldsymbol{\Gamma}(\mathbf{x}, \boldsymbol{\zeta}) \triangleq \nabla_{\mathbf{x}} \nabla_{\boldsymbol{\zeta}} \mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\zeta}), \tag{10a}$$

$$\boldsymbol{\Gamma}(\mathbf{x}) \triangleq \boldsymbol{\Gamma}(\mathbf{x}, \mathbf{0}) = \nabla_{\mathbf{x}} \nabla_{\boldsymbol{\zeta}} \mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\zeta})_{|\boldsymbol{\zeta}=\mathbf{0}}. \tag{10b}$$

Since $\boldsymbol{\Gamma}(\mathbf{x})$ is a proxy to $\nabla_{\mathbf{x}} \mathbf{P}^{\mathrm{a}}(\mathbf{x})$ as per Eq. (5), $\boldsymbol{\Gamma}(\mathbf{x})$ can be interpreted as the *marginal* variation of $\mathbf{P}^{\mathrm{a}}(\mathbf{x})$ when the forecast state $\mathbf{x}$ is perturbed. Under Gaussianity and linearity and assuming $\mathbf{H}^{\intercal}\mathbf{R}^{-1} = \mathbf{I}_{\mathrm{x}}$, $\mathbf{P}^{\mathrm{a}}(\mathbf{x})$ coincides with the Kalman gain. That is why we will call $\boldsymbol{\Gamma}(\mathbf{x})$ the *marginal gain tensor*, or simply *marginal gain*. Mathematically, this is a *3-tensor field*, i.e. a map from the state space $\mathbf{E}_{\mathrm{x}}$ to $\mathbf{E}_{\mathrm{x}}^3$. Componentwise, using a conventional placement of indices, its definition reads

$$[\boldsymbol{\Gamma}(\mathbf{x})]_{ijl} \triangleq \left(\partial_{x_l} \partial_{\zeta_j} a_{\boldsymbol{\theta},i}(\mathbf{x}, \boldsymbol{\zeta})\right)_{|\boldsymbol{\zeta}=\mathbf{0}}. \tag{11}$$

To mitigate the complexity in studying the map $\mathbf{x} \longmapsto \boldsymbol{\Gamma}(\mathbf{x})$, we introduce the *mean marginal gain*, $\overline{\boldsymbol{\Gamma}}$. This 3-tensor is defined as the average of $\boldsymbol{\Gamma}(\mathbf{x})$ over the K states of a long trajectory $\mathcal{T}_{\mathbf{x}} = \left\{\mathbf{x}_k^{\mathrm{t}}\right\}_{k=1,\ldots,K} \in \mathbf{E}_{\mathrm{x}}^K$ of the $\mathcal{M}$–based ergodic chaotic dynamics:

$$\overline{\boldsymbol{\Gamma}} \triangleq \langle \boldsymbol{\Gamma}(\mathbf{x}) \rangle_{\mathbf{x} \in \mathcal{T}_{\mathbf{x}}} \underset{K \to \infty}{=} \int \mathrm{d}\mathbf{x}\, \pi(\mathbf{x}) \boldsymbol{\Gamma}(\mathbf{x}) = \mathbb{E}_{\mathbf{x} \sim \pi} \left[\boldsymbol{\Gamma}(\mathbf{x})\right], \tag{12}$$

where π is the invariant distribution of the ergodic chaotic dynamics.

As recalled in Sect. 2.3, $\nabla_{\boldsymbol{\zeta}} \mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\zeta})_{|\boldsymbol{\zeta}=\mathbf{0}} \approx \mathbf{P}^{\mathrm{a}}(\mathbf{x})$ was heuristically estimated in Boc24 using samples of $\boldsymbol{\zeta}$ followed by a linear regression. This estimation through a regression can be formally justified using the following argument. We wish to average $\nabla_{\boldsymbol{\zeta}} \mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\zeta})$ over the pdf $\rho(\boldsymbol{\zeta})$ of the projected innovations $\boldsymbol{\zeta}$, which is assumed to be a Gaussian with a positive-definite covariance matrix $\boldsymbol{\Sigma}_\rho$. Then it can be shown that

$$\mathbb{E}_{\boldsymbol{\zeta} \sim \rho}\left[\nabla_{\boldsymbol{\zeta}} \mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\zeta})\right] = \int \mathrm{d}\boldsymbol{\zeta}\, \rho(\boldsymbol{\zeta}) \nabla_{\boldsymbol{\zeta}} \mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\zeta}) \tag{13a}$$

$$= \boldsymbol{\Sigma}_\rho^{-1} \mathrm{Cov}_{\boldsymbol{\zeta} \sim \rho}\left[\boldsymbol{\zeta}, \mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\zeta})\right], \tag{13b}$$

where $\mathrm{Cov}[\cdot, \cdot]$ is the covariance operator, and $\boldsymbol{\Sigma}_\rho^{-1}$ operates on the first tensor factor. A proof is given in Appendix D, along with a generalisation to the case where $\boldsymbol{\Sigma}_\rho$ is only semi positive-definite, which is made necessary because the set $\left\{\boldsymbol{\zeta}_k = \mathbf{H}_k^{\intercal}\mathbf{R}_k^{-1}\boldsymbol{\delta}_k\right\}_k$ may only span a subspace of $\mathbf{E}_{\mathrm{x}}$ if $\mathbf{H}_k$ is not injective. This result is none other than Stein's lemma (Liu, 1994) applied to the Gaussian approximation of ρ and $\mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x}, \cdot)$. It aligns with its original use in ensemble DA (Stordal et al., 2016; Raanes et al., 2019; Fillion et al., 2020) and with the connection established by Lemma 2 of Agarwal et al. (2021) between $\boldsymbol{\Sigma}_\rho^{-1}\mathrm{Cov}_{\boldsymbol{\zeta} \sim \rho}\left[\boldsymbol{\zeta}, \mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\zeta})\right]$, which is a perturbations-based estimator (Ribeiro et al., 2016), and a SmoothGrad estimator (Smilkov et al., 2017). In the limit where $\boldsymbol{\zeta}$, as a random vector, is concentrated around $\mathbf{0}$ and ρ can be approximated as Gaussian, which corresponds to the most common *weak assimilation* regime where the information content of the innovation is small compared to that of the background, we have

$$\boldsymbol{\Sigma}_\rho^{-1}\mathrm{Cov}_{\boldsymbol{\zeta} \sim \rho}\left[\boldsymbol{\zeta}, \mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\zeta})\right] = \int \mathrm{d}\boldsymbol{\zeta}\, \rho(\boldsymbol{\zeta}) \nabla_{\boldsymbol{\zeta}} \mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\zeta}) \tag{14a}$$

$$\approx \nabla_{\boldsymbol{\zeta}} \mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\zeta})_{|\boldsymbol{\zeta}=\mathbf{0}}, \tag{14b}$$

which, with Eq. (13a), relates the integral form $\mathbb{E}_{\boldsymbol{\zeta} \sim \rho}[\nabla_{\boldsymbol{\zeta}} \mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\zeta})]$ to the gradient form $\nabla_{\boldsymbol{\zeta}} \mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\zeta})_{|\boldsymbol{\zeta}=\mathbf{0}}$ of the sensitivity of $\mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\zeta})$ with respect to $\boldsymbol{\zeta}$.

Building on this correspondance, we can connect the following mean marginal gain

$$\widetilde{\boldsymbol{\Gamma}} \triangleq \mathbb{E}_{\mathbf{x} \sim \pi, \boldsymbol{\zeta} \sim \rho}[\boldsymbol{\Gamma}(\mathbf{x}, \boldsymbol{\zeta})] = \int \mathrm{d}\mathbf{x}\mathrm{d}\boldsymbol{\zeta}\, \pi(\mathbf{x}) \rho(\boldsymbol{\zeta}) \boldsymbol{\Gamma}(\mathbf{x}, \boldsymbol{\zeta}), \tag{15}$$

defined from Eq. (10a) and more closely related to the DA process, to the mean marginal gain $\overline{\mathbf{\Gamma}}$ as defined in Eq. (12):

$$\widetilde{\mathbf{\Gamma}} = \mathbb{E}_{\mathbf{x}\sim\pi,\boldsymbol{\zeta}\sim\rho}\left[\mathbf{\Gamma}(\mathbf{x},\boldsymbol{\zeta})\right] \tag{16a}$$
$$= \mathbb{E}_{\mathbf{x}\sim\pi}\left[\nabla_{\mathbf{x}}\mathbb{E}_{\boldsymbol{\zeta}\sim\rho}\left[\nabla_{\boldsymbol{\zeta}}\mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x},\boldsymbol{\zeta})\right]\right] \tag{16b}$$
$$\approx \mathbb{E}_{\mathbf{x}\sim\pi}\left[\nabla_{\mathbf{x}}\nabla_{\boldsymbol{\zeta}}\mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x},\boldsymbol{\zeta})_{|\boldsymbol{\zeta}=\mathbf{0}}\right] \tag{16c}$$
$$= \mathbb{E}_{\mathbf{x}\sim\pi}\left[\mathbf{\Gamma}(\mathbf{x})\right] \tag{16d}$$
$$= \overline{\mathbf{\Gamma}}. \tag{16e}$$

However, the *empirical mean* of $\mathbf{\Gamma}(\mathbf{x},\boldsymbol{\zeta})$ from which to evaluate $\widetilde{\mathbf{\Gamma}}$, and denoted $\widehat{\mathbf{\Gamma}}$, and which is the numerical estimation of the sensitivity out of a long enough DA run, may differ from both theoretical means $\widetilde{\mathbf{\Gamma}}$ and $\overline{\mathbf{\Gamma}}$. That is why we show in Appendix E how $\widehat{\mathbf{\Gamma}}$ approximates $\widetilde{\mathbf{\Gamma}}$.

Note that Eq. (16a) is a simple point estimator, which formalises that the averaged Jacobian $\mathbb{E}_{\boldsymbol{\zeta}}[\nabla_{\boldsymbol{\zeta}}\mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x},\boldsymbol{\zeta})]$ is well approximated by $\nabla_{\boldsymbol{\zeta}}\mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x},\boldsymbol{\zeta})|_{\boldsymbol{\zeta}=\mathbf{0}}$ when the distribution of projected innovations is concentrated near zero. The Stein lemma, which provides a regression-based estimator of this average, is not required per se to ascertain Eq. (16a), but offers an alternative expression which we leverage in Appendix H.

3.3.2 Invariance and equivariance

An important way to reduce the computational cost and complexity of interpreting the mean marginal gain $\widetilde{\mathbf{\Gamma}}$ is to exploit the symmetries of the DA problem, when applicable. If the model variables are observed homogeneously and homoscedastically, then the probability laws defining the DA problem are invariant under these symmetries, and the corresponding analysis operator is equivariant. As a result, the components of $\widetilde{\mathbf{\Gamma}}$ related by symmetry are redundant, so that its interpretation can be restricted to a reduced set of representative components.

For instance, in the one-dimensional periodic L96 model with homogeneous and homoscedastic observations, a cyclic translation of the state variables induces the same cyclic translation of the observations, innovations, and analysis increments. The probability laws defining the DA problem are therefore invariant under the translation group, while the corresponding analysis operator is equivariant. Consequently, if the trained DAN preserves this symmetry, any trajectory-averaged sensitivity tensor must have a circulant structure: the information associated with different grid points is identical up to a cyclic shift. The group-theoretical argument below provides a formal justification for this circulant averaging.

Let us generically denote $\mathcal{G}$ such group of symmetries for the DA process, which are assumed to be isometries. It is chosen to be the maximal group for which both the dynamics and the observation process are equivariant. Appendix F gives the formal definitions for this group of symmetries.

The action of $g \in \mathcal{G}$ on $\mathbf{\Gamma}(\mathbf{x},\boldsymbol{\zeta})$ is denoted $g\odot\mathbf{\Gamma}(\mathbf{x},\boldsymbol{\zeta})$. It is a tensor field for $\mathcal{G}$, in the sense that it is *equivariant* under the action of this symmetry group following the transformation rule, $\forall g \in \mathcal{G}, \forall \mathbf{x} \in \mathbf{E}_{\mathrm{x}}, \forall \boldsymbol{\zeta} \in \mathbf{E}_{\mathrm{x}}$:

$$g \odot \mathbf{\Gamma}(\mathbf{x},\boldsymbol{\zeta}) \triangleq \mathbf{\Gamma}(g\circ\mathbf{x}, g\circ\boldsymbol{\zeta}) = g \otimes g^{\intercal} \otimes g^{\intercal} \circ \mathbf{\Gamma}(\mathbf{x},\boldsymbol{\zeta}), \tag{17}$$

where the ordering of the three tensorial factors follows the convention of Eq. (11). A proof of the equivariance Eq. (17) is proposed in Appendix F.

Invoking the ergodicity of the dynamics, any symmetry of $\mathcal{G}$ leaves the invariant distribution of the dynamics π, and the projected innovation distribution ρ, unchanged, $\forall g \in \mathcal{G}, \forall \mathbf{x} \in \mathbf{E}_{\mathrm{x}}, \forall \boldsymbol{\zeta} \in \mathbf{E}_{\mathrm{x}}$:

$$\pi(g\circ\mathbf{x}) = \pi(\mathbf{x}), \quad \rho(g\circ\boldsymbol{\zeta}) = \rho(\boldsymbol{\zeta}), \tag{18}$$

where $g\circ(\cdot)$ denotes the action of g on a field.

Leveraging the equivariance of the marginal gain and the symmetries of the invariant distribution, we now demonstrate the invariance of the mean marginal gain $\widetilde{\mathbf{\Gamma}}$; for $g \in \mathcal{G}$, we have

$$g \odot \widetilde{\mathbf{\Gamma}} = \int \mathrm{d}\mathbf{x}\mathrm{d}\boldsymbol{\zeta}\, \pi(\mathbf{x})\rho(\boldsymbol{\zeta}) g \odot \mathbf{\Gamma}(\mathbf{x},\boldsymbol{\zeta}) \tag{19a}$$
$$= \int \mathrm{d}\mathbf{x}\mathrm{d}\boldsymbol{\zeta}\, \pi(\mathbf{x})\rho(\boldsymbol{\zeta}) \mathbf{\Gamma}(g\circ\mathbf{x}, g\circ\boldsymbol{\zeta}) \tag{19b}$$
$$= \int \mathrm{d}(g^{-1}\circ\mathbf{x})\mathrm{d}(g^{-1}\circ\boldsymbol{\zeta}) \times \pi(g^{-1}\circ\mathbf{x})\rho(g^{-1}\circ\boldsymbol{\zeta})\mathbf{\Gamma}(\mathbf{x},\boldsymbol{\zeta}) \tag{19c}$$
$$= \int \mathrm{d}\mathbf{x}\mathrm{d}\boldsymbol{\zeta}\, \pi(\mathbf{x})\rho(\boldsymbol{\zeta})\mathbf{\Gamma}(\mathbf{x},\boldsymbol{\zeta}) = \widetilde{\mathbf{\Gamma}}, \tag{19d}$$

where a change of variables was carried out from Eq. (19a) to Eq. (19c). The invariance of π and ρ under $\mathcal{G}$, and the fact that the determinant of the Jacobian of g^{-1} is 1 since g^{-1} is an isometry were utilised from Eq. (19c) to Eq. (19d). We finally conclude:

$$\forall g \in \mathcal{G}: \quad g \odot \widetilde{\mathbf{\Gamma}} = \widetilde{\mathbf{\Gamma}}. \tag{20}$$

The same result can be obtained for $\overline{\mathbf{\Gamma}}$, with a simpler derivation only involving the invariant distribution π of the underlying dynamics. However, the symmetry group $\overline{\mathbf{\Gamma}}$ must be the same as that of $\widetilde{\mathbf{\Gamma}}$, and not the potentially larger group associated to π:

$$\forall g \in \mathcal{G}: \quad g \odot \overline{\mathbf{\Gamma}} = \overline{\mathbf{\Gamma}}. \tag{21}$$

As a consequence, in the rest of this section, we assume that the results apply equally to either $\widetilde{\mathbf{\Gamma}}$ or $\overline{\mathbf{\Gamma}}$.

3.3.3 Numerical illustrations

Leveraging the equivariance induced by the translational invariance of the L96 model, for a fixed reference site r, we define the *slice* matrix $\overline{\mathbf{\Omega}}^{(r)} \in \mathbb{R}^{N_{\mathrm{x}}\times N_{\mathrm{x}}}$ component-wise by

$$\overline{\Omega}^{(r)}_{ij} \triangleq \overline{\Gamma}_{ijr}. \tag{22}$$

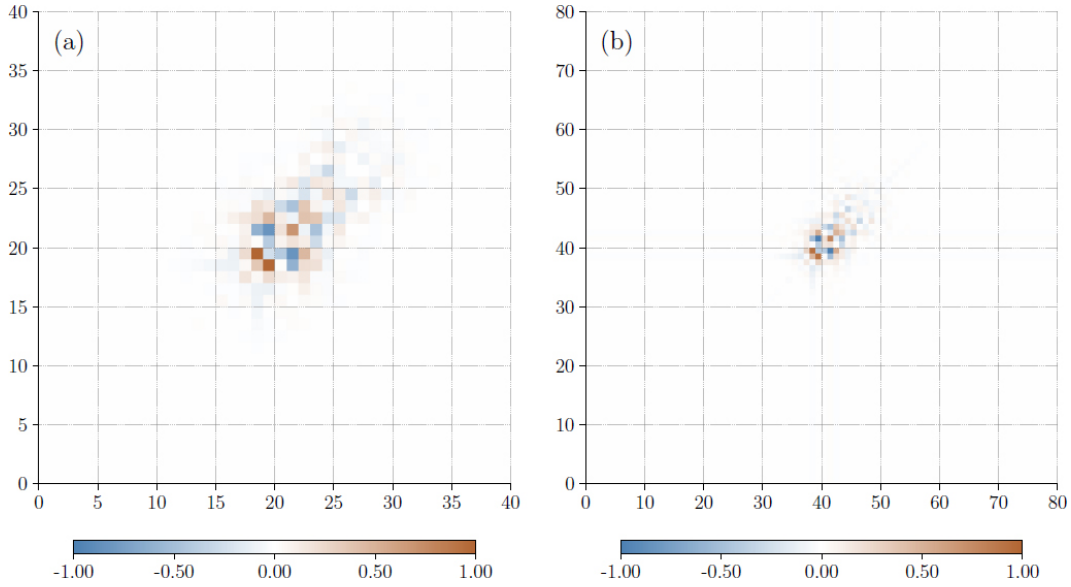

Figure 1. Plot of the normalised mean marginal gain $\overline{\Omega}^{(r)}_{ij}/\max|\overline{\Omega}^{(r)}_{ij}|$ in the fully observed DA configuration of the L96 model, obtained with a model size of $N_\mathrm{x} = 40$ (panel **a**) and $N_\mathrm{x} = 80$ (panel **b**).

The index r is not an additional free index of $\overline{\mathbf{\Omega}}^{(r)}$; it labels the chosen slice of the averaged 3-tensor. For visual convenience, r is chosen to be in the middle of the domain $r = \lfloor N_\mathrm{x}/2 \rfloor$ in the L96 case. The simple but tedious details justifying Eq. (22) are given in Appendix G. How to numerically compute this $\overline{\mathbf{\Omega}}^{(r)}$ matrix is then discussed in Appendix H.

To start with, we choose the fully observed DA setup $\mathbf{H} = \mathbf{R} = \mathbf{I}_\mathrm{x}$, which applies to both the training of $\mathbf{a}_{\boldsymbol{\theta}}$ and the subsequent DA tests. The computation of $\overline{\mathbf{\Omega}}^{(r)}$ is carried out through the composite approach (see Appendix H). The results are shown in Fig. 1.

The dominant values of $|\overline{\mathbf{\Omega}}^{(r)}|$ form a pattern. They are concentrated in the vicinity of the perturbed variable of the forecast state, which makes them local. To further qualify the local patterns, we also trained $\mathbf{a}_{\boldsymbol{\theta}}$ on a L96 model but with $N_\mathrm{x} = 80$, i.e. twice the size of the standard L96 model, while all other parameters either related to the dynamics or the DA experiments, remain the same. Then, $\overline{\mathbf{\Omega}}^{(r)}$ is similarly computed and plotted in Fig. 1. As expected, the same pattern emerges with the same spatial extension. This further supports the EnKF-like performance of $\mathbf{a}_{\boldsymbol{\theta}}$ trained with $N_\mathrm{x} = 40$ when tested with $N_\mathrm{x} = 80$ (as recalled in Sect. 2.3).

3.3.4 Patterns with non-trivial observation operators

We carried out the exact same experiments described above, but now with three different observation configurations instead of the full observation setup. The corresponding $\overline{\mathbf{\Omega}}^{(r)}$ are plotted in Fig. 2. Panel (a) corresponds to the fully observed configuration for reference. The first configuration (panel b) corresponds to the observation of every other site: $[\mathbf{H}]_{j\,i} = \delta_{i,2j}$ for $1 \leq i \leq N_\mathrm{x}$ and $1 \leq j \leq \lfloor N_\mathrm{x}/2 \rfloor$. The second (panel c) corresponds, at each time step t_k, to an observation at N_y^k random but distinct sites, where N_y^k is uniformly drawn at each time step in between 0 and N_x (bounds included). The third configuration (panel d) is the same as the second but with N_y^k uniformly drawn at each time step in between $\lfloor N_\mathrm{x}/2 \rfloor$ and N_x. It is remarkable that the same local pattern emerges in all configurations, even when every other site is never observed. However, the magnitude of the sensitivities (values of the patterns) changes depending on the information balance in the analysis and hence in the gain magnitude. Finally, note that $\overline{\mathbf{\Omega}}^{(r)}$ only represents an average pattern. It is possible to exhibit a family of modal patterns but it would go beyond the aim of the present paper.

4 Data assimilation networks in stronger nonlinear regimes

In Sect. 3, we made several contributions to the understanding of DAN. The numerical experiments were set in a mildly nonlinear regime of L96 where $\Delta_t = 0.05$ in between analyses, known to correspond to 6 h of a synoptic meteorological model (Lorenz and Emanuel, 1998), and a forcing F set to 8. We now examine the ability of the method to learn efficient analysis schemes under stronger nonlinear conditions. While we exclude cases in which the nonlinearity is so severe that it induces implicit or explicit recurrent multi-modal priors, we do consider regimes that exhibit substantial departures from mild nonlinearity.

Probing such mildly to strongly nonlinear regimes can be achieved by less frequent observation, typically increasing

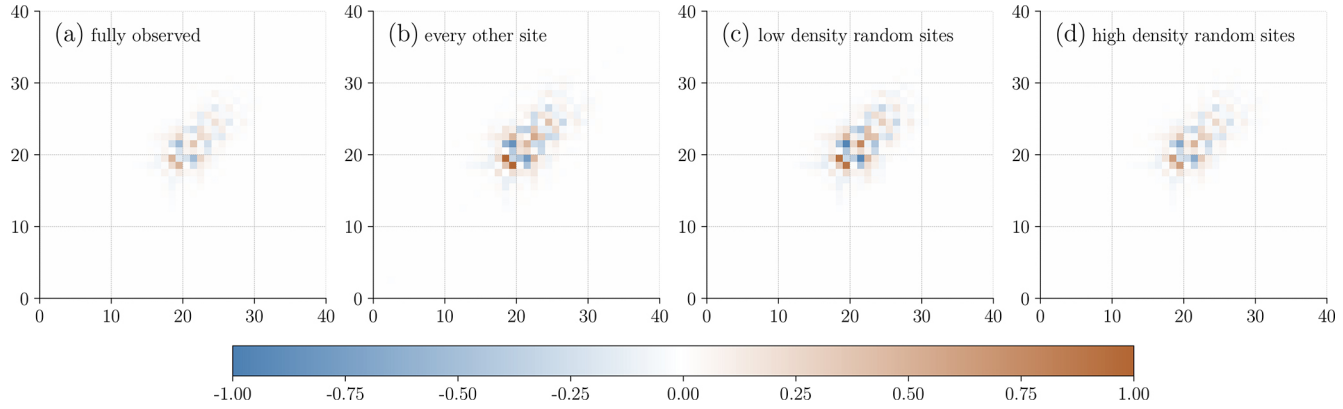

Figure 2. Plot of the mean marginal gain $\overline{\Omega}_{ij}^{(r)}$ normalised by the maximum of its absolute value over all entries and over the four panels, in several sparse observation DA configuration of the L96 model.

the update time-step Δ_t. The dimensional analysis of Appendix 1 in Bocquet and Carrassi (2017) shows that varying Δ_t is indeed relevant to achieve such objective for the L96 model versus, e.g., increasing the observation error amplitude. The forcing F can also be varied to that end. However, it rather stands as a signature of the magnitude of the instability of the dynamics (e.g., a covariant function of the Kaplan-Yorke dimension, a measure of the fractal dimension of the dynamics attractor) rather than the signature of the deviation from Gaussianity.

4.1 Performance as a function of the update time-step

Increasing Δ_t to multiples of 0.05 is the experimental design chosen in Sakov et al. (2012) and Bocquet and Sakov (2012) to evaluate the performance of the iterative Ensemble Kalman Filter (IEnKF). As the ensemble variant of the iterative Kalman filter (Wishner et al., 1969; Jazwinski, 1970), the IEnKF still stands, to our best knowledge, as the most accurate scalable DA method in mildly to strongly nonlinear conditions (Bocquet and Sakov, 2013). It is hence a hard-to-beat baseline for learning an advanced DA analysis scheme in such conditions. The remarkable performance of the IEnKF stems from its ensemble-variational formulation obtained from Bayesian first principles: an ensemble is used to construct a time-dependent prior, and the analysis is performed through a nonlinear iterative optimisation (Bocquet and Sakov, 2014).

The IEnKF can be made even more accurate (both for smoothing and filtering) by choosing longer DA windows, yielding the iterative ensemble Kalman smoother (IEnKS, Bocquet and Sakov, 2014; Raanes et al., 2019). However, it works over longer DA windows, as opposed to the DAN implemented here, which would bias the comparison. Actually, we have developed (recurrent) variants of DAN that work on extended DA windows similarly to the IEnKS, but reporting on them is beyond the goals of this paper.

We choose a simple common observational configuration for the many DA methods we intend to compare. All sites are observed through $\mathbf{H}_k \overset{\Delta}{=} \mathbf{I}_{\mathrm{x}}$ and $\mathbf{R}_k \overset{\Delta}{=} \mathbf{I}_{\mathrm{x}}$, a well documented setup in the literature. In this configuration, any useful, but not necessarily accurate, DA method must exhibit a test RMSE slightly below 1. We consider the following DA methods:

- A well tuned EnKF with an ensemble of size $N_{\mathrm{e}} = N_{\mathrm{x}} = 40$. Optimal multiplicative inflation is addressed through the finite-size EnKF (Bocquet, 2011; Bocquet et al., 2015). This first contender is meant to illustrate the progressive failure of the EnKF in stronger nonlinear conditions.
- A well tuned IEnKF with an ensemble of size $N_{\mathrm{e}} = N_{\mathrm{x}} = 40$, whose optimal multiplicative inflation is addressed through the finite-size IEnKF (Bocquet and Sakov, 2012). This is our hard baseline.
- A *baseline* learned DAN scheme using the neural network and the training parameters values set in Appendix B.
- A *boosted* learned DAN scheme using $N_{\mathrm{f}} = 80$, $N_{\mathrm{r}} = 2^{19}$, $N_{\mathrm{iter}} = 32$, $S_{\mathrm{b}} = 512$, which is parameter and data intensive, and hence much more time-consuming to train on a single GPU, while its inference remains cheap.
- A learned DAN scheme where the explicit dependence on the forecast state is discarded, while the dependence on the projected innovations is maintained. We expect the resulting DA method to perform similarly to a well tuned 3D-Var, see Boc24.
- A learned DAN scheme where the activation functions are all linear, such that $\mathbf{a}_{\boldsymbol{\theta}}$ is linear in its inputs. Like the previous approach, we expect the resulting DA method to perform similarly to a well tuned 3D-Var, see Boc24.

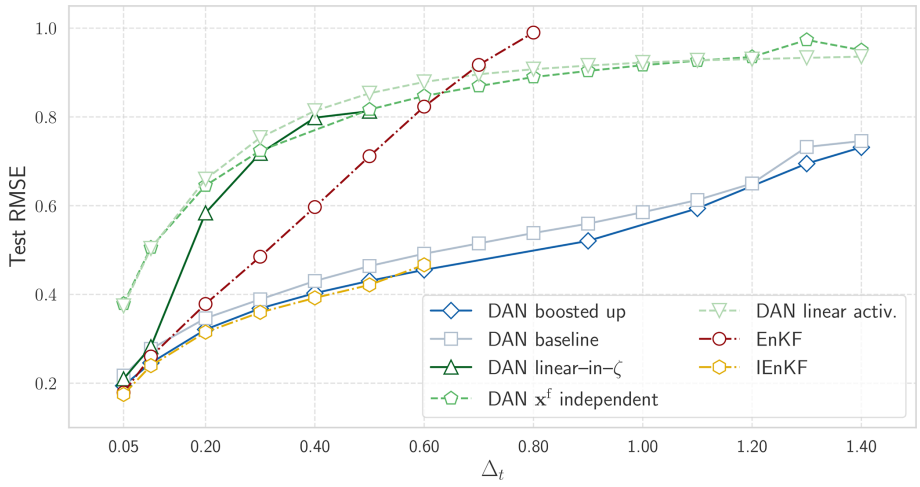

Figure 3. Test RMSEs of DA methods as a function of the update time-step Δ_t in between analyses. See text for details.

To avoid early divergences in the training when $\Delta_t \gg 0.05$, all the DAN schemes benefited from a modified Eq. (2):

$$\mathbf{x}^{\mathrm{a}}_k = \mathbf{x}^{\mathrm{f}}_k + \alpha \mathbf{H}_k^{\intercal} \mathbf{R}_k^{-1} \boldsymbol{\delta}_k + \mathbf{a}_{\boldsymbol{\theta}}\left(\mathbf{x}^{\mathrm{f}}_k, \mathbf{H}_k^{\intercal} \mathbf{R}_k^{-1} \boldsymbol{\delta}_k\right), \tag{23}$$

where α is a trainable scalar. Note that this formulation is mathematically equivalent to the original Eq. (2). This is only meant to explicitly enforce the solution $\mathbf{x}^{\mathrm{a}}_k = \mathbf{y}_k$ when $\mathbf{H}_k = \mathbf{I}_{\mathrm{x}}$, which guides the training in its first few epochs.

The test RMSEs of those DA methods, over a long DA run, are displayed in Fig. 3 as a function of Δ_t. Note that $\Delta_t = 0.60$ is already very significantly nonlinear as it corresponds to the Lyapunov time of L96, i.e. the time horizon beyond which the DA system becomes significantly non-Gaussian.

As expected, the two degraded DAN operators severely underperform the other DA schemes, which leverage non-static priors, with a test RMSE that ranges from 0.38 for $\Delta_t = 0.05$ to an asymptotic RMSE below 1 for much larger Δ_t. This is consistent with the findings of Boc24 and mirrors the performance of a 3D-Var with static background covariance matrix.

The EnKF offers a very good performance in the mildly nonlinear regime $\Delta_t \approx 0.05$ but gradually degrades as Δ_t is increased. Moreover, beyond $\Delta_t = 0.80$, the EnKF becomes uninformative and its test RMSE is not reported. As already shown by Sakov et al. (2012), the IEnKF offers significantly better performance, from a marginal improvement over the EnKF at $\Delta_t = 0.05$ that gets more and more significant as Δ_t is increased. Note that, beyond $\Delta_t = 0.60$, the IEnKF is trickier to stabilise and hence its performance is not reported.

Remarkably, the boosted DAN operator achieves performance very similar to the IEnKF, but can still be learned for much larger Δ_t, and remains informative with still a very significant edge over the static prior methods. Again, this is achieved without the use of an ensemble but of a single forecast state. Moreover, the learned DAN does not explicitly resort to a nonlinear (Gauss-Newton) iterative minimisation, in contrast to the IEnKF. This aspect of such DAN is reminiscent of approaches meant to learn a solver for a variational DA problem (Fablet et al., 2021; Frerix et al., 2021; Lafon et al., 2023; Filoche et al., 2023; Keller and Potthast, 2024). The baseline DAN operator is slightly less performing but follows the same trend. We could have evaluated the methods for even larger $\Delta_t > 1.40$; however, $\Delta_t = 1.20$ already stands for twice the Lyapunov time, which is equivalent to 6 d in the L96 correspondence.

We suspect that the test RMSEs should primarily be a function of $N_{\mathrm{iter}}\Delta_t$ rather than just Δ_t, accounting for the forgetful effect associated to the chaotic dynamics. Hence, to reach the same test RMSEs, N_{iter} could be roughly chosen inversely proportional to Δ_t. To test this hypothesis, we compare the test RMSE on a large number of trained DANs, varying Δ_t (including for $\Delta_t \leq 0.05$) and N_{iter} with results shown in Fig. 4. Leveraging the findings of Appendix B and the use of smaller batches to mitigate the numerical cost, we have chosen for their training, $N_{\mathrm{r}} = 2^{16} = 65\,536$ and $S_{\mathrm{b}} = 2^7 = 128$. The test RMSEs are shown in Fig. 4. The results corroborate the intuition with the weaker and weaker dependence on N_{iter} of the performance when Δ_t is increased. Conversely, a much larger N_{iter} is required when Δ_t gets very small ($\Delta_t \approx 0.01$).

Hence, DAN is numerically more difficult to train when $\Delta_t < 0.05$ requiring larger N_{iter}, which corresponds to quasi-linear regimes where DAN is nonetheless of limited interest.

4.2 Performance as a function of the energy forcing

The forcing F injects and removes energy from the L96 dynamics and feeds their instability. Besides Δ_t, this is another tunable parameter of the nonlinearity of the dynamics. There is a wealth of dynamical phenomenology of the L96 dynamics for a wide range of F (see, e.g., Barone et al., 2025, and references therein). For the range of F we focus on here, in-

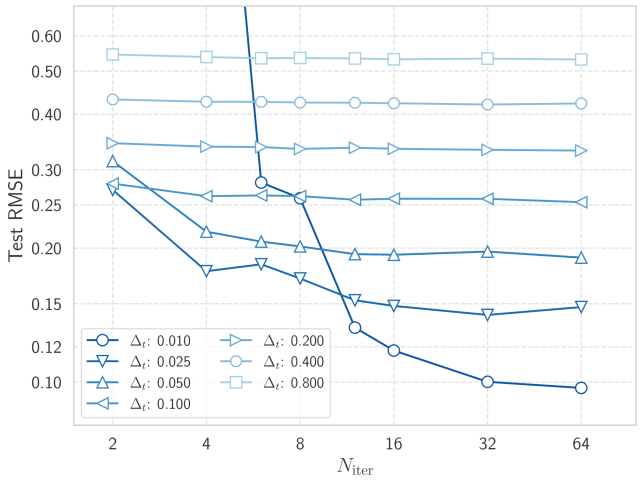

Figure 4. Test RMSEs of DANs as a function of both Δ_t and N_{iter}.

stabilities progressively develop in between $1 \lesssim F \lesssim 4$, while chaos fully sets in beyond $F \gtrsim 4$ (van Kekem and Sterk, 2018). The number of Lyapunov exponents, and similarly the Kaplan-Yorke dimension, increases monotonically from 4 to about 30 where it saturates (Karimi and Paul, 2010).

We choose the same setup as in the previous experiments and consider the following DA methods:

- A well tuned EnKF with an ensemble of size $N_e = N_x = 40$. Optimal multiplicative inflation is addressed through the finite-size EnKF (Bocquet, 2011; Bocquet et al., 2015), in its Dirac-Jeffreys variant (Bocquet et al., 2015) required to handle the weakly nonlinear regime ($4 \lesssim F \lesssim 8$).
- A well tuned IEnKF with an ensemble of size $N_e = N_x = 40$, whose optimal multiplicative inflation is addressed through the finite-size IEnKF (Bocquet and Sakov, 2012) in its Dirac-Jeffreys variant. This is our hard baseline.
- A (baseline) learned DAN scheme using the neural network and training parameters as defined in Appendix B.

The test RMSEs are displayed in Fig. 5. The IEnKF has a slight edge over the EnKF with larger F with the increasing Kaplan-Yorke dimension. Hence, the DA system does not deviate much from Gaussianity as F increases, rather, it is mainly the magnitude of the instabilities that changes. The DAN achieves a performance in between that of the EnKF and that of the IEnKF, which is patent for large F. Note that it turns out trickier to train DANs for F getting close to 4. There, the dynamics become more and more laminar and exhibit almost periodic waves, with patterns that, although simpler, are very different from those learned in the regimes explored so far.

4.3 Reasons for the efficiency of data assimilation networks

In the light of the previous numerical results, we discuss the reasons why DAN can be as accurate as the IEnKF, without an ensemble, without any experimental tuning, and even in mildly to strongly nonlinear conditions.

We have already shown in Sect. 3.2 that in the mildly nonlinear regime, i.e. $\Delta_t \approx 0.05$, the success of DAN mainly resides in its implicit estimation of the map $\mathbf{x}^f \longmapsto \mathbf{P}^f(\mathbf{x}^f)$, in line with the results by Sacco et al. (2024) and Sakov (2025), and closely related to the map $\mathbf{x}^f \longmapsto \mathbf{P}^a(\mathbf{x}^f)$. As shown in Sect. 3.2.1, this amounts to assume a non-static but Gaussian prior in the analysis: this is equivalent to having $\left\|\mathbf{x}-\mathbf{x}^f\right\|^2_{(\mathbf{P}^f(\mathbf{x}^f))^{-1}}$ as the background term in the analysis cost function. Whether this mechanism is sufficient to ensure the performance of DAN when $\Delta_t \geq 0.05$ is doubtful.

4.3.1 The linear-in-$\boldsymbol{\zeta}$ data assimilation network beyond mild nonlinearity

To address this question, let us assess the linear-in-$\boldsymbol{\zeta}$ DAN, see Sect. 3.2, in stronger nonlinear regimes $\Delta_t \geq 0.05$. We could anticipate that it accounts well for the errors of the day, but that it may not be able to handle stronger nonlinearity/non-Gaussianity, e.g., if $\Delta_t \gg 0.05$, because of the implied Gaussian prior. It should hence match the EnKF, rather than the IEnKF.

Let us check that hypothesis numerically. The test RMSEs of this specific DAN scheme, which mirrors Eq. (5), are reported in Fig. 3, as the linear-in-$\boldsymbol{\zeta}$ DAN. Let us first remark that, in accordance with the claims of Sect. 3.2, it performs as well as the EnKF for $\Delta_t = 0.05$. However, as Δ_t increases, its performance significantly degrades compared to the EnKF, not to mention the IEnKF. Even though it relies on an estimate of the $\mathbf{x}^f \longmapsto \mathbf{P}^a(\mathbf{x}^f)$ map, its underlying Gaussian assumption penalises it beyond the mildly nonlinear regime, as expected.

4.3.2 Deviation of the data assimilation network prior from Gaussianity

We conclude that the full DAN scheme not only exploits the flow-dependent information represented by the $\mathbf{x}^f \longmapsto \mathbf{P}^f(\mathbf{x}^f)$ map, but also learns an effective nonlinear analysis response consistent with a non-Gaussian prior, which is more informative than the Gaussian prior associated to the analysis cost function background term $\left\|\mathbf{x}-\mathbf{x}^f\right\|^2_{(\mathbf{P}^f(\mathbf{x}^f))^{-1}}$. As shown, both abilities are required for DAN to perform so well in the extended range of mildly to strongly nonlinear regimes.

We investigate the deviation of $\mathbf{a}_{\boldsymbol{\theta}}$ from the presumed quasi-linearity in $\boldsymbol{\zeta}$ by defining the scalar function

$$r(\lambda) = \mathbb{E}_{\mathbf{x},\boldsymbol{\zeta}}\left[\frac{\left\|\mathbf{P}^a(\mathbf{x})^{-1}\left(\mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x},\lambda\boldsymbol{\zeta})-\mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{0})\right)\right\|}{\left\|\mathbf{P}^a(\mathbf{x})^{-1}\left(\mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x},\boldsymbol{\zeta})-\mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{0})\right)\right\|}\right], \tag{24}$$

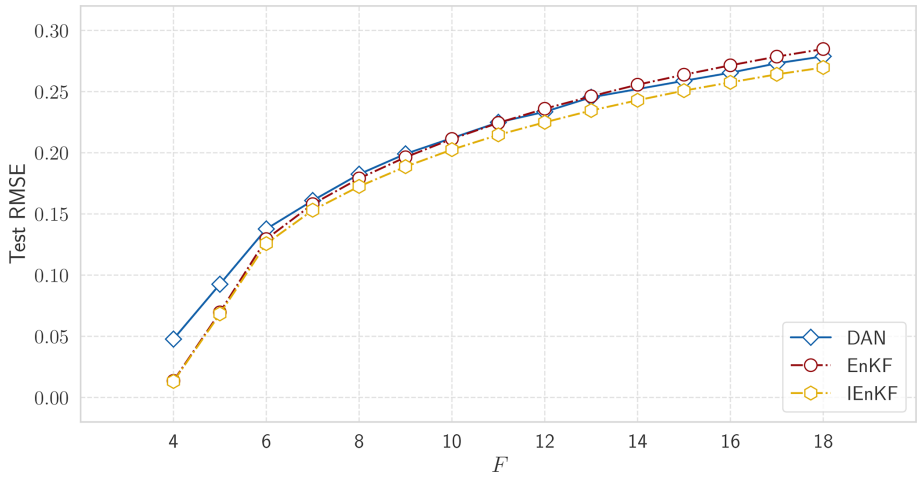

Figure 5. Test RMSEs of DA methods as a function of the forcing F. See text for details.

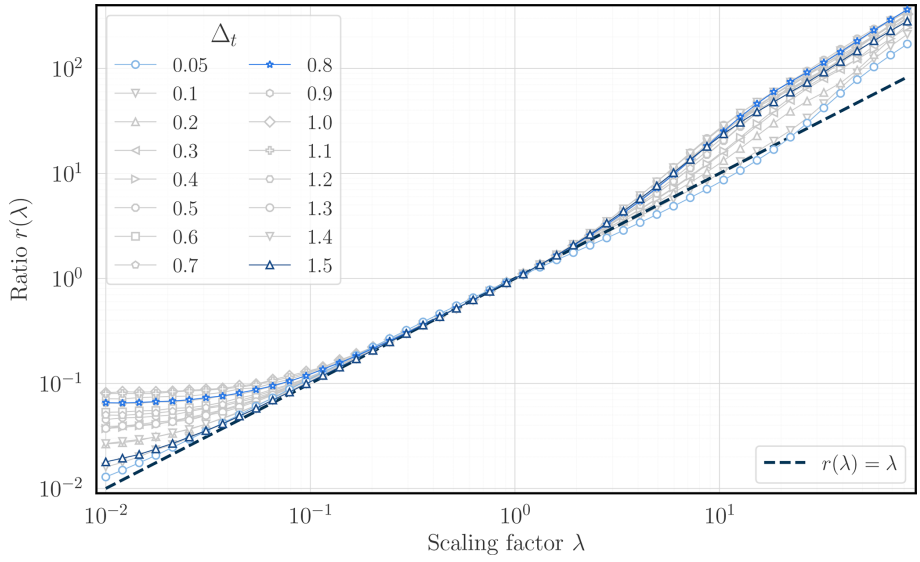

Figure 6. Scaling of the standardised analysis increments of $\mathbf{a}_{\boldsymbol{\theta}}$ in the projected innovations, for a large range of Δ_t values.

where $\|\cdot\|$ is the Euclidean norm. It depends on a dimensionless scale parameter $\lambda > 0$ and measures deviations from the linearity in $\boldsymbol{\zeta}$. In Eq. (24), $\mathbf{P}^{\mathrm{a}}(\mathbf{x})^{-1}$ is meant to standardise the deviations from linearity. Indeed, in the quasi-linear regime, we have

$$\mathbf{P}^{\mathrm{a}}(\mathbf{x})^{-1}\left(\mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x}, \lambda\boldsymbol{\zeta}) - \mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{0})\right) \approx \lambda\boldsymbol{\zeta}, \tag{25}$$

such that $r(\lambda) \simeq \lambda$ should hold. The ratio $r(\lambda)$ is estimated using perturbations as in Boc24: $\mathbf{x}$ and $\boldsymbol{\zeta}$ are sampled from the forecast state $\mathbf{x}^{\mathrm{f}}$ and the projected innovations $\boldsymbol{\zeta} = \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y} - \mathbf{H}\mathbf{x}^{\mathrm{f}})$ that are obtained from a long trajectory.

The deviation from $r(\lambda) \simeq \lambda$ is demonstrated in Fig. 6 with $\lambda \longmapsto r(\lambda)$ plotted for several values of Δ_t in the range $[0.05, 1.5]$. These curves should be appreciated knowing the range of values taken by the projected innovations in a long DA run for the selected values of Δ_t which, for each Δ_t, points to the relevant range of λ values to consider and which mainly contributes to the computation of $r(\lambda)$. This is shown in Fig. 7 in the form of histograms of those values. Given the range of innovation values and the logarithmic scale of Fig. 6, the most relevant part of the scaling behaviour sits in the range $\lambda \in [1, 10]$. In this range of the scaling, as seen in Fig. 6, the more nonlinear the DA run, the steeper $\lambda \longmapsto r(\lambda)$: the innovation impact is stronger than the one expected in the linear regime, especially for large magnitude innovations.

5 Discussion and conclusions

In this paper, we have continued to explore the potential of learning sequential DA operators with neural networks for tracking chaotic dynamical systems, following on the initial

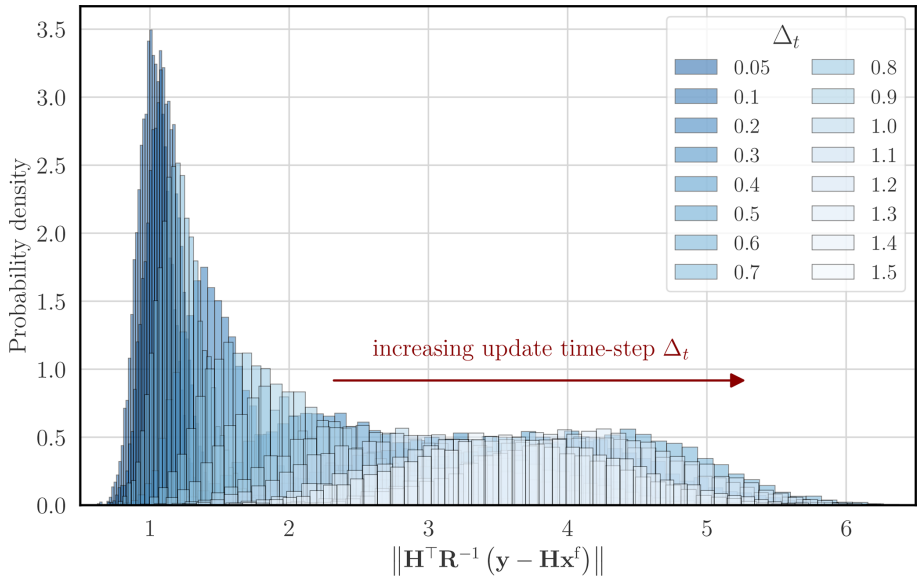

Figure 7. Histograms of the projected innovations' norm, for a large range of Δ_t values.

results and conclusions of Boc24 who built on the concepts of data assimilation networks (DANs) introduced by McCabe and Brown (2021) and Boudier et al. (2023). Assuming the dynamics to be known, the focus is on learning the analysis. Compared to learning the analysis from a dataset of inputs and outputs of DA runs, the training is numerically challenging since the analysis operator is learned through several DA cycles and from trajectories of the dynamics and its observations only.

The resulting DANs are robust, in the sense that they do not require inflation or any other correction and regularisation. They can operate without an ensemble and only use the forecast state as prior information. And yet, they are as accurate as a well tuned EnKF in mildly nonlinear regime, and as accurate as a well tuned IEnKF from mildly to strongly nonlinear conditions.

5.1 Abilities of data assimilation networks

We have previously shown that, to achieve this level of performance, DAN must implicitly learn a map $\mathbf{x}^f \longmapsto \mathbf{P}^a(\mathbf{x}^f)$, at least in the mildly nonlinear regime. Its existence is supported by a multiplicative ergodic theorem applied to the entire DA process viewed as a dynamical system. The network must implicitly identify spatial local patterns in $\mathbf{x}^f$ in order to internally represent components of $\mathbf{P}^f(\mathbf{x}^f)$ or $\mathbf{P}^a(\mathbf{x}^f)$, a property that makes the learned DA method scalable.

In this paper, we further examined the reliance of DANs on the map $\mathbf{x}^f \longmapsto \mathbf{P}^a(\mathbf{x}^f)$ by constructing an ad hoc learnable DAN that explicitly incorporates this mapping. For the L96 dynamics, we identified an average characteristic local pattern by leveraging both the invariant distribution of the dynamics and the translational symmetry of the model. We also showed that DAN can learn an effective nonlinear analysis response consistent with non-Gaussian prior information inferred from $\mathbf{x}^f$, and that this nonlinear analysis is necessary for DAN to match the performance of the IEnKF under more stringent nonlinear conditions. In this regime, we demonstrated that implicit or explicit knowledge of the mapping alone is insufficient. Thus, both mechanisms must operate within DAN, paralleling IEnKF's critical reliance on its ensemble for flow-dependent error estimation and on its Gauss-Newton iterative solver to probe departures of the dynamics from linearity.

5.2 Implication for the end-to-end processors in numerical weather prediction

Taken in the context of past DA literature, these results are somewhat surprising. For a long time, it was believed that an ensemble was essential for estimating flow-dependent errors. Moreover, the proper construction of non-Gaussian priors has remained a long-standing challenge, whereas even a simple DAN can rapidly learn an effective nonlinear analysis response consistent with such non-Gaussian information. These findings should inform the (re-)design of future DA algorithms.

These considerations have implications for what DAN-like operators are capable of achieving. For example, recently developed end-to-end atmospheric processors (Allen et al., 2025; Boucher et al., 2025) that ingest observations and predict future observations, implicitly construct their own internal representation of the system and repeatedly compare this latent state to newly assimilated observations. In doing so, they implicitly learn an analysis operator in their latent space. Lean et al. (2025) concluded that their proces-

sor, GraphDOP, must be able to learn not only a climatological background but also dynamical priors, a result that may seem surprising given that GraphDOP does not rely on any explicit background information. However, in light of our results, GraphDOP can be interpreted as learning a mapping from the 12 h observational snapshot used as input to the processor to a latent representation that contains information analogous to underlying error covariances. Consequently, it must be able to construct its own background, incorporating advanced error statistics that go beyond a mere climatology. This interpretation is consistent with the diagnostic study of Laloyaux et al. (2025), who applied classical DA tools, including forecast-sensitivity observation-impact diagnostics, to GraphDOP. Their results suggest that GraphDOP learns physically meaningful spatial relationships and latent Earth-system representations from observations. While this does not show that GraphDOP explicitly estimates a forecast-error covariance, it supports the broader view that end-to-end observation-driven processors can develop DA-like internal representations without being supplied with an explicit background state.

This challenges the view that the accuracy of such end-to-end processors is fundamentally limited by the absence of a background (such as a forecast ensemble or climatological information) in their inputs. While an explicit background representation would certainly provide additional information, the extent of the achievable performance gains remains a subtler question.

5.3 Ensembles, uncertainty quantification, and multiplicative ergodic theorem

Our results show that, in the perfect-model filtering setting considered here, an explicit forecast ensemble is not strictly required for a learned analysis operator to extract the flow-dependent information needed to achieve EnKF-like point-estimation accuracy. This conclusion should not be interpreted as a dismissal of ensemble representations in general. Ensembles remain essential for probabilistic forecasting, uncertainty quantification, the diagnosis of model and observation errors, smoothing, risk-sensitive applications, and regimes in which the posterior distribution is strongly non-Gaussian or multi-modal.

Moreover, the deterministic analysis operator $\mathbf{a}_{\boldsymbol{\theta}}$ could itself be used as a building block for ensemble generation. For instance, applying $\mathbf{a}_{\boldsymbol{\theta}}$ to perturbed innovations would produce a set of analysis states, in a way that is consistent with the local, innovation-dependent interpretation used in, e.g., Appendix I to obtain the linear regression approximation of $\mathbf{a}_{\boldsymbol{\theta}}$ with respect to $\boldsymbol{\zeta}$. Such an analysis ensemble could then be propagated by the forecast model to estimate forecast uncertainty. We have not pursued this probabilistic use of DAN in the present study, since our focus was on point-estimation accuracy, but it represents a natural direction for future work.

At a more theoretical level, the success of DAN may be facilitated by the existence, in the present setting, of a sufficiently regular and local dependence of the relevant flow-dependent analysis correction on the forecast state. This should not be taken for granted in general. For example, the multiplicative ergodic theorem ensures that covariant Lyapunov subspaces are defined as measurable functions of the state, for almost every state on the attractor, but it does not guarantee that this state-to-Oseledets-splitting map is continuous, local, or easily learnable from finite data. Other applications beyond DA, such as learning covariant Lyapunov vectors directly from the state, may therefore involve maps with poorer regularity or locality, making them substantially more difficult to learn and possibly less scalable. This question is left for future work.

Appendix A: Neural network architecture

As described in Boc24, we choose for $\mathbf{a}_{\boldsymbol{\theta}}$ a basic residual convolutional neural network (CNN) architecture. A schematic of the CNN architecture is displayed in Fig. A1. It begins with an initial convolution that takes $N_\mathrm{e}+1$ channels as inputs and, with N_f filters, outputs N_f channels. This initial layer is followed by N_b residual blocks. Each one of these blocks is a succession of N_sb sub-blocks. Each subblock is made of: (i) a convolutional layer with N_f channels as inputs, which has N_f filters and a kernel size f_v for each of its filter, (ii) a batch normalisation layer, and (iii) an activation function chosen to be Mish (Misra, 2019). The CNN ends with a final convolutional layer that takes N_f channels as inputs and, with N_e filters, outputs N_e channels. The kernel size of the initial and final channels is f_v. Hence, the internal state of the CNN consists of N_f copies of the latent space which we simply choose to be isomorphic to the state space $\mathbb{R}^{N_\mathrm{x}}$. In this paper, only $N_\mathrm{e}=1$ is used. For all experiments, the internal architecture parameter values are $N_\mathrm{b}=5$, $N_\mathrm{sb}=5$, $f_\mathrm{v}=5$.

Appendix B: Sensitivity to the batch size, the size of the datasets, and the backpropagation truncation

The key parameters in the design of $\mathbf{a}_{\boldsymbol{\theta}}$ and especially in its training are:

- the number of channels/filters N_f processed by the convolutional neural network (see Appendix A). This essentially gives away the complexity of the neural network, how many features it can identify and process.
- the number of trajectory N_r processed in parallel. The larger N_r, the more DA runs the neural network can learn from and be made robust against. 10 % of these are reserved for validation.
- the number of cycles N_iter through which the gradient is computed. This corresponds to the limit imposed

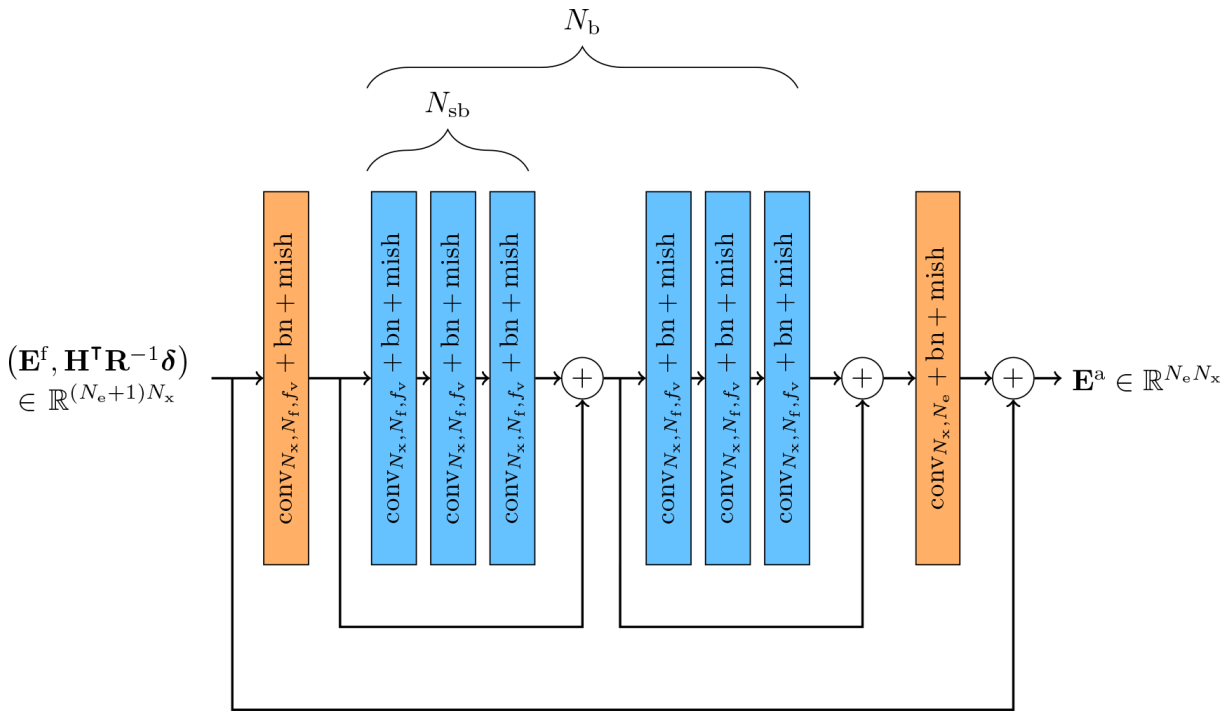

Figure A1. Architecture of the residual convolutional network where, for this specific illustration, $N_\mathrm{b} = 2$, $N_\mathrm{sb} = 3$. $\mathrm{conv}_{N_1,N_2,f}$ is a generic one-dimensional convolutional layer of dimension N_1, with N_2 filters of kernel size f. See text for more details.

by the truncated backpropagation through time. The larger $N_\mathrm{iter} \ll N_\mathrm{c}$, the better the information transmission from one cycle to the next should be learned, but the more costly and less accurate the gradients.

- the size of the batch S_b. Hence, the number of steps in each epoch is about $N_\mathrm{r}/S_\mathrm{b}$ (training and validation).

The training parameters N_r, N_iter, and to a lesser extend N_c are the key dimensions specifying how the data are fed to the training schedule as schematised in Fig. B1.

The dependence of the performance of $\mathbf{a}_{\boldsymbol{\theta}}$ on those parameters was studied in Boc24. The values $N_\mathrm{iter} = 16$, $N_\mathrm{r} = 2^{18}$, $S_\mathrm{b} = 2048$ were chosen for the $\mathbf{a}_{\boldsymbol{\theta}}$ hyperparameters of the reference setup, as a compromise between training speed and accuracy of the resulting $\mathbf{a}_{\boldsymbol{\theta}}$. However, the dependence on N_iter and S_b was barely reported and discussed, so that we focus on them in what follows.

Using the reference DA setup, we compute the test RMSE of the DAN schemes as a function of the truncation number N_iter. The results are shown in Fig. B2. As expected, $N_\mathrm{iter} \leq 5$ prevents DAN from learning an efficient prior that relies on the errors of the day. It is however remarkable that the improvement in the test RMSE as N_iter increases is noticeable up to about $N_\mathrm{iter} \simeq 40$, which corresponds to about 6 times the doubling time of the L96 model in the reference setup.

We have also experimented with the batch size S_b much more thoroughly than in Boc24. We found that using smaller batches is beneficial to the performance of the trained $\mathbf{a}_{\boldsymbol{\theta}}$. This may also enable reducing the number of trajectories N_r in the dataset. Nonetheless, we empirically found that the number of steps, i.e. $\sim N_\mathrm{r}/S_\mathrm{b}$ in each epoch still needs to remain large to achieve high accuracy. Experimenting, we learned a large set of $\mathbf{a}_{\boldsymbol{\theta}}$ operators with N_r in the range 2^{10}–2^{18}, while S_b is chosen in the range 2^3–2^8. The corresponding test RMSEs are plotted as a function of either S_b, as shown in Fig. B3a, or N_r as shown in Fig. B3b.

For this specific configuration a scaling law can be numerically estimated. The following Ansatz is assumed:

$$\mathrm{RMSE}^{1/\gamma} = f\left(\frac{N_\mathrm{r}^\alpha}{S_\mathrm{b}^\beta}\right), \tag{B1}$$

where f is a polynomial of order 2. This Ansatz (3 polynomial coefficients and 3 exponents) is fitted to the test RMSE results. The fit relevance can be visualised through the plot of f^γ in Fig. B4. The fitted exponents are $\gamma = 1.435$, $\alpha = 0.103$, and $\beta = 0.060$. What really matters is the ratio $\rho = \beta/\alpha = 0.58$, since the test RMSE turns out to show a strong dependence on $N_\mathrm{r}/S_\mathrm{b}^\rho$.

In practice, the interest of using smaller batches S_b and hence smaller N_r must be weighted against the overheads created by the dataset pipeline but also by the transfer of the batches from RAM to VRAM. It is however possible to run several training experiments in parallel over the same GPU with smaller batches. Hence, the efficiency of using mini-batches is very dependent on the accelerator device(s) and

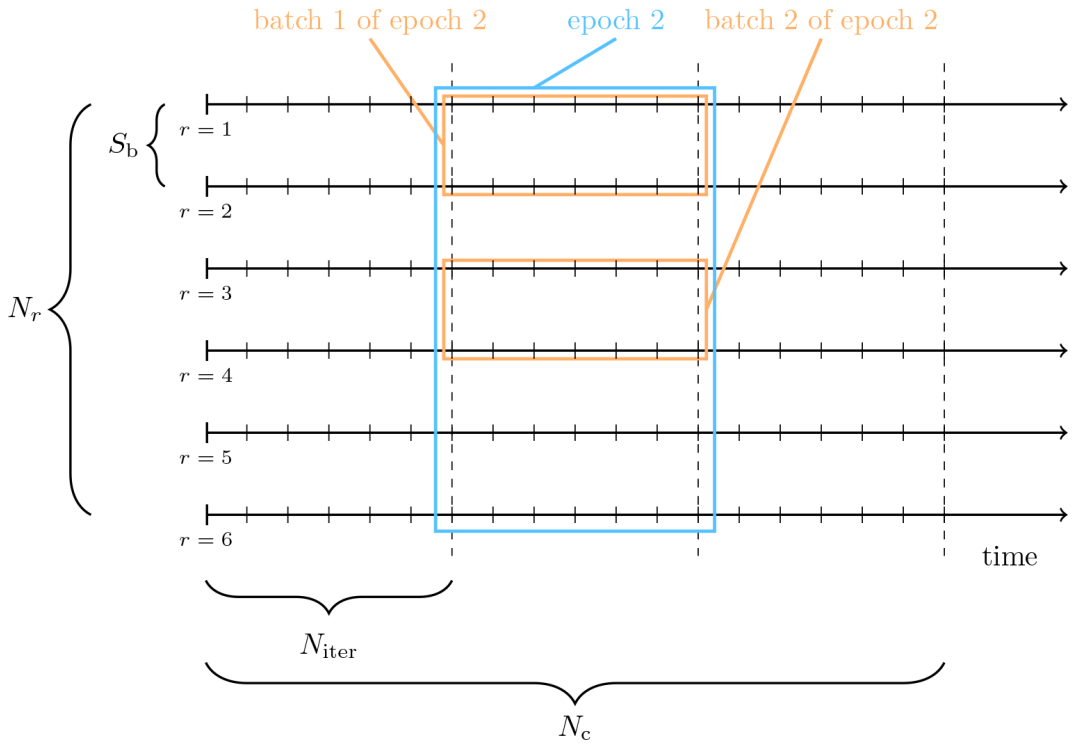

Figure B1. This schematic describes how data are fed to the training scheme; it must be read from left to right (time arrow). For instance, in epoch 2, N_{iter}-long segments of N_r trajectories feed the training algorithm. The segments are organised and provided in batches of size S_b.

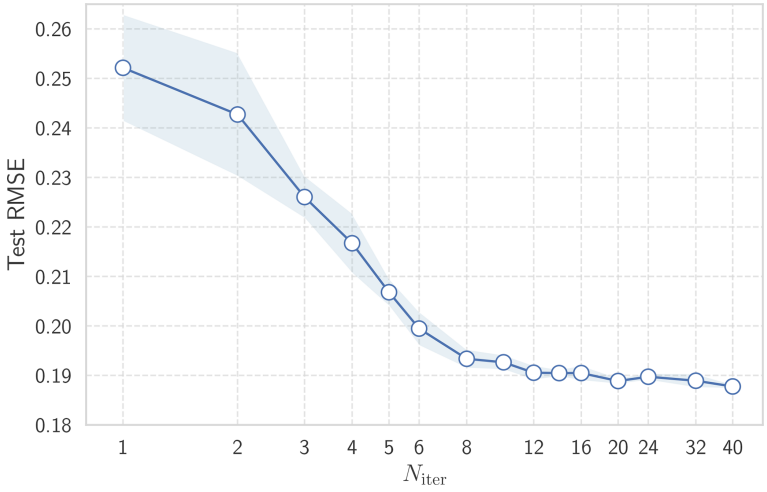

Figure B2. Test RMSE of DAN as a function of the truncation cycles number $N_{iter} = 1, \ldots, 40$ in the truncated backpropagation. For each value of N_c, an ensemble of 5 $\mathbf{a}_{\boldsymbol{\theta}}$ operators is learned. Plus and minus one standard deviation of the RMSE are displayed as shades around the RMSE curve.

the type of experiment to conduct. Note that, when training $\mathbf{a}_{\boldsymbol{\theta}}$ on much higher dimensional systems, relying on mini-batches may become mandatory so as to fit into VRAM.

Finally, the scaling law Eq. (B1) can be leveraged to reduce both N_r and S_b and still be able to extrapolate to values yielding better test RMSEs, using the dependence on N_r/S_b^{ρ}. This scaling could nonetheless change with a different DA setup.

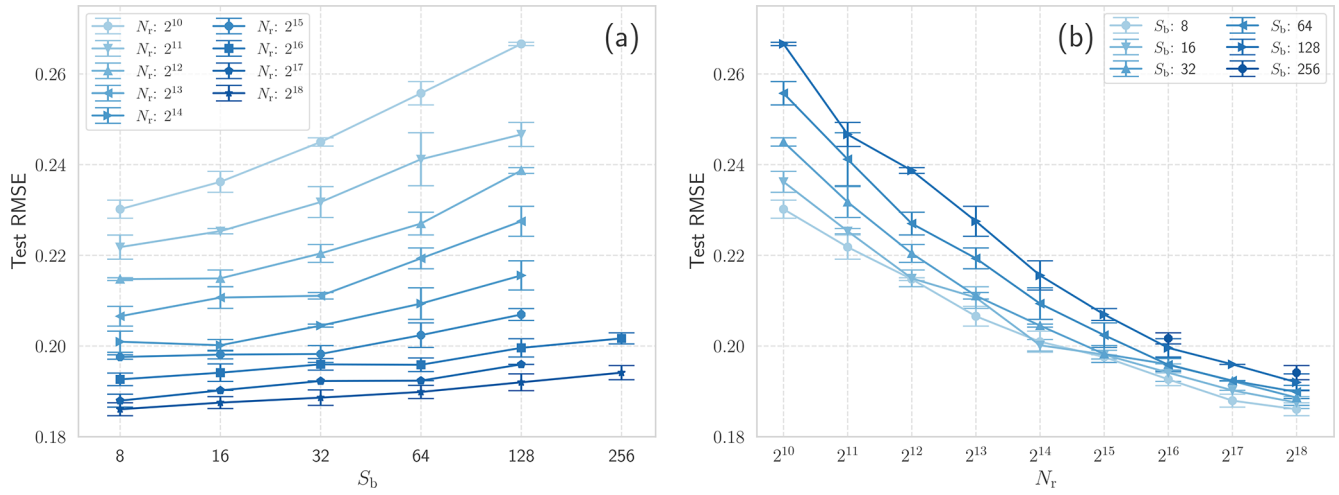

Figure B3. Test RMSEs of DAN as a function of the batch size S_b for a selection of N_r values (panel **a**) and as a function of the number of dataset trajectories N_r for a selection of S_b values (panel **b**). For each pair (S_b, N_r), an ensemble of 3 $\mathbf{a}_{\boldsymbol{\theta}}$ operators is learned, from which error bars (plus or minus the standard deviation) are estimated and added to the curve plots.

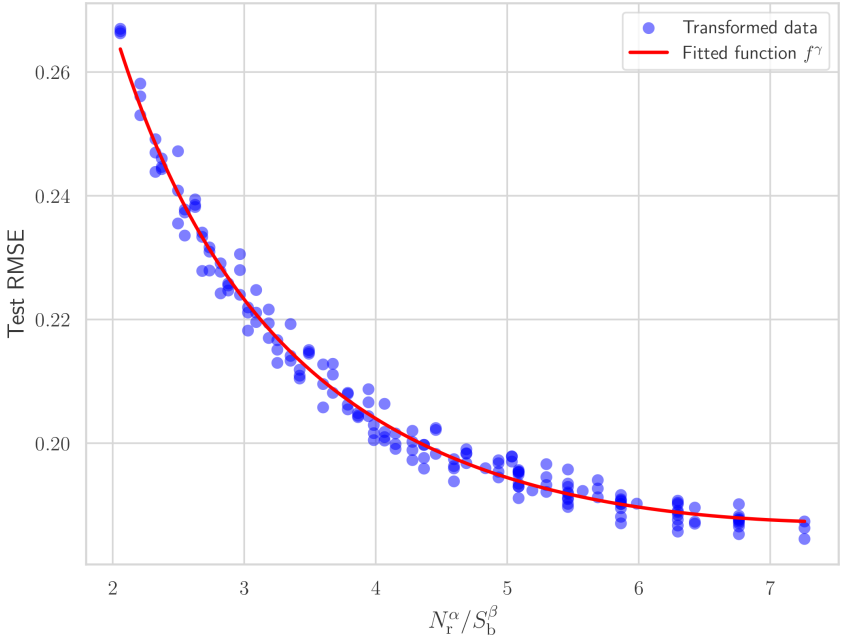

Figure B4. Checking the relevance of the functional regression by plotting the fit function f^γ with respect to N_r^α / S_b^β (curve), and of the test RMSE results (dots).

Appendix C: Neural network implementation of the effective $\mathbf{x}^f \longmapsto \mathbf{P}^a(\mathbf{x}^f)$ map

Here, we detail how to build the linear-in-$\boldsymbol{\zeta}$ DAN meant to enforce Eq. (9) in Sect. 3.2. Implementing the map $(\mathbf{x}^f, \boldsymbol{\zeta}) \longmapsto \mathbf{P}^a(\mathbf{x}^f) \cdot \boldsymbol{\zeta}$ as a neural network is non-trivial if we wish to make it scalable. First, $\mathbf{P}^a(\mathbf{x}^f)$ is symmetric positive definite; to enforce such constraint, the usual most efficient approach is to write: $\mathbf{P}^a(\mathbf{x}^f) = \mathbf{X}^a(\mathbf{x}^f)\mathbf{X}^a(\mathbf{x}^f)^\intercal$, where $\mathbf{X}^a(\mathbf{x}^f)$ is a matrix of anomalies which may be easier to interpret than $\mathbf{P}^a(\mathbf{x}^f)$. The first difficulty towards scalability is the fact that $\mathbf{P}^a(\mathbf{x}^f)$, or $\mathbf{X}^a(\mathbf{x}^f)$, is of size $N_x \times N_x$, which is considered non-scalable. However, if $\mathbf{P}^a(\mathbf{x}^f)$ can be approximated as low-rank, then $\mathbf{P}^a(\mathbf{x}^f) \approx \mathbf{X}^a(\mathbf{x}^f)\mathbf{X}^a(\mathbf{x}^f)^\intercal$ with $\mathbf{X}^a(\mathbf{x}^f)$ of size $N_x \times N_r$, with N_r remaining small enough when N_x is increased. Unfortunately, $\mathbf{P}^a(\mathbf{x}^f)$ should not realistically be considered low-rank. However, one can exploit the locality of the covariances and write $\mathbf{P}^a(\mathbf{x}^f) \approx \boldsymbol{\rho} \circ \left(\mathbf{X}^a(\mathbf{x}^f)\mathbf{X}^a(\mathbf{x}^f)^\intercal\right)$, where $\boldsymbol{\rho}$ is the localisation correlation matrix and $\circ$ is the Schur/Hadamard product. In addition to making $\mathbf{P}^a(\mathbf{x}^f)$ full rank in spite of a

manageable number of parameters in $\mathbf{X}^{\mathrm{a}}(\mathbf{x}^{\mathrm{f}})$, it also tapers spurious correlations that could be learned in the course of the training. Indeed, the implementation of the localisation significantly accelerates the training. This is reminiscent of the proposal by Bocquet and Farchi (2019) to estimate $\mathbf{X}^{\mathrm{a}}$ through a loss involving a Schur product with $\boldsymbol{\rho}$.

Furthermore, given $\mathbf{P}^{\mathrm{a}} = \boldsymbol{\rho} \circ (\mathbf{X}^{\mathrm{a}}(\mathbf{X}^{\mathrm{a}})^{\intercal})$ and the projected innovation $\boldsymbol{\zeta}$, the implementation of such mapping can be efficiently coded using (see, e.g., Desroziers et al., 2014)

$$\mathbf{P}^{\mathrm{a}} \cdot \boldsymbol{\zeta} = \boldsymbol{\rho} \circ \left(\mathbf{X}^{\mathrm{a}}\left(\mathbf{X}^{\mathrm{a}}\right)^{\intercal}\right) \cdot \boldsymbol{\zeta} = \sum_{l=1}^{N_{\mathrm{r}}} \mathbf{X}^{\mathrm{a}}_l \circ \left(\boldsymbol{\rho} \cdot \left(\mathbf{X}^{\mathrm{a}}_l \circ \boldsymbol{\zeta}\right)\right), \tag{C1}$$

where the $\mathbf{X}^{\mathrm{a}}_l$ are the columns of $\mathbf{X}^{\mathrm{a}}$, and $\cdot$ denotes the usual matrix/vector multiplication. The matrix multiplication by the localisation matrix $\boldsymbol{\rho}$ is scalable since $\boldsymbol{\rho}$ is assumed to be a banded matrix. With L96 in mind, i.e. in a one-dimensional context, with a localisation matrix support of (band-)width N_{l}, the numerical complexity of $\mathbf{P}^{\mathrm{a}} \cdot \boldsymbol{\zeta}$ is $N_{\mathrm{r}} N_{\mathrm{x}}(2 + N_{\mathrm{l}})$.

If localisation is not useful and $\mathbf{P}^{\mathrm{a}}(\mathbf{x}^{\mathrm{f}})$ is low-rank, then it is easy to implement $\mathbf{P}^{\mathrm{a}} \cdot \boldsymbol{\zeta} = \mathbf{X}^{\mathrm{a}} \cdot ((\mathbf{X}^{\mathrm{a}})^{\intercal} \cdot \boldsymbol{\zeta})$, which is reminiscent of the very popular machine learning *attention* mechanism. The numerical complexity is then $2N_{\mathrm{r}}N_{\mathrm{x}}$.

An alternative linear-in-$\boldsymbol{\zeta}$ DAN is to implement an *hypernetwork* which would map $\mathbf{x}^{\mathrm{f}}$ to a set of weights and biases of a linear neural network that would then be applied to $\boldsymbol{\zeta}$. Naively, the number of weights and biases could scale like N_{x}^2. However, we can instead map $\mathbf{x}^{\mathrm{f}}$ to weights and biases of a sequence of convolutional neural networks that we later apply to $\boldsymbol{\zeta}$. Yet, we did not test this more sophisticated construction since the former approach is scalable and successful.

Appendix D: Stein lemma for $\nabla_{\zeta} \mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\zeta})$

Here, we offer a proof of the Stein lemma in the degenerate case where the Gaussian density is singular within the embedding space. This is useful with sparse observations resulting in $\boldsymbol{\zeta}$ confined within a subspace of $\mathbf{E}_{\mathrm{x}}$. This subsumes the full-rank case. Let us assume that ρ is the pdf defined over $\mathbf{E}_{\mathrm{x}}$ of the Gaussian random vector $\boldsymbol{\zeta}$, whose mean is $\overline{\boldsymbol{\zeta}}$ and whose covariance matrix is $\boldsymbol{\Sigma}_{\rho}$ which is assumed positive semi-definite. Hence, the support of $\boldsymbol{\zeta}$ may be singular in $\mathbf{E}_{\mathrm{x}}$. That is why we resort to the singular value decomposition $\boldsymbol{\Sigma}_{\rho} = \mathbf{U}\boldsymbol{\Sigma}_{\lambda}\mathbf{U}^{\intercal}$, where $\mathbf{U}$ is an orthonormal (though not necessarily orthogonal) matrix such that $\mathbf{U}^{\intercal}\mathbf{U} = \mathbf{I}_{\mathrm{x}}$, and $\boldsymbol{\Sigma}_{\lambda}$ is a positive definite diagonal matrix of rank lower or equal to N_{x}. We can parametrise the random vector $\boldsymbol{\zeta}$ by

$$\boldsymbol{\zeta}(\boldsymbol{\omega}) = \overline{\boldsymbol{\zeta}} + \mathbf{U}\boldsymbol{\omega}, \tag{D1}$$

where the random vector $\boldsymbol{\omega}$ has $\lambda(\boldsymbol{\omega}) = n(\mathbf{0}, \boldsymbol{\Sigma}_{\lambda})$ for Gaussian pdf. Then, denoting $f(\boldsymbol{\zeta}) \triangleq \mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\zeta})$ for brevity, we have

$$\begin{aligned}
&\mathrm{Cov}_{\boldsymbol{\zeta}\sim\rho}\left[\boldsymbol{\zeta}, f(\boldsymbol{\zeta})\right] \\
&\quad = \mathrm{Cov}_{\boldsymbol{\omega}\sim\lambda}\left[\boldsymbol{\zeta}(\boldsymbol{\omega}), f(\boldsymbol{\zeta}(\boldsymbol{\omega}))\right] && \text{(D2a)}\\
&\quad = \int \mathrm{d}\boldsymbol{\omega}\lambda(\boldsymbol{\omega})(\boldsymbol{\zeta}(\boldsymbol{\omega}) - \overline{\boldsymbol{\zeta}}) \otimes \left(f(\boldsymbol{\zeta}(\boldsymbol{\omega})) - f(\overline{\boldsymbol{\zeta}})\right) && \text{(D2b)}\\
&\quad = \int \mathrm{d}\boldsymbol{\omega}\lambda(\boldsymbol{\omega})(\mathbf{U}\boldsymbol{\omega}) \otimes f(\boldsymbol{\zeta}(\boldsymbol{\omega})) && \text{(D2c)}\\
&\quad = \mathbf{U}\boldsymbol{\Sigma}_{\lambda} \int \mathrm{d}\boldsymbol{\omega}\lambda(\boldsymbol{\omega})\left(\boldsymbol{\Sigma}_{\lambda}^{-1}\boldsymbol{\omega}\right) \otimes f(\boldsymbol{\zeta}(\boldsymbol{\omega})) && \text{(D2d)}\\
&\quad = -\mathbf{U}\boldsymbol{\Sigma}_{\lambda} \int \mathrm{d}\boldsymbol{\omega}\lambda(\boldsymbol{\omega})\nabla_{\boldsymbol{\omega}} \ln\lambda(\boldsymbol{\omega}) \otimes f(\boldsymbol{\zeta}(\boldsymbol{\omega})) && \text{(D2e)}\\
&\quad = -\mathbf{U}\boldsymbol{\Sigma}_{\lambda} \int \mathrm{d}\boldsymbol{\omega}\nabla_{\boldsymbol{\omega}}\lambda(\boldsymbol{\omega}) \otimes f(\boldsymbol{\zeta}(\boldsymbol{\omega})) && \text{(D2f)}\\
&\quad = \mathbf{U}\boldsymbol{\Sigma}_{\lambda} \int \mathrm{d}\boldsymbol{\omega}\lambda(\boldsymbol{\omega})\nabla_{\boldsymbol{\omega}}\left(f(\boldsymbol{\zeta}(\boldsymbol{\omega}))\right) && \text{(D2g)}\\
&\quad = \mathbf{U}\boldsymbol{\Sigma}_{\lambda}\mathbf{U}^{\intercal} \int \mathrm{d}\boldsymbol{\omega}\lambda(\boldsymbol{\omega})(\nabla_{\boldsymbol{\zeta}} f)(\boldsymbol{\zeta}(\boldsymbol{\omega})) && \text{(D2h)}\\
&\quad = \boldsymbol{\Sigma}_{\rho}\mathbb{E}_{\boldsymbol{\omega}\sim\lambda}\left[(\nabla_{\boldsymbol{\zeta}} f)(\boldsymbol{\zeta}(\boldsymbol{\omega}))\right] && \text{(D2i)}\\
&\quad = \boldsymbol{\Sigma}_{\rho}\mathbb{E}_{\boldsymbol{\zeta}\sim\rho}\left[\nabla_{\boldsymbol{\zeta}} f\right]. && \text{(D2j)}
\end{aligned}$$

Hence, we conclude:

$$\mathbb{E}_{\boldsymbol{\zeta}\sim\rho}\left[\nabla_{\boldsymbol{\zeta}} f\right] = \boldsymbol{\Sigma}_{\rho}^{\dagger}\mathrm{Cov}_{\boldsymbol{\zeta}\sim\rho}\left[\boldsymbol{\zeta}, f(\boldsymbol{\zeta})\right], \tag{D3}$$

where $(\cdot)^{\dagger}$ is the Moore-Penrose inverse operator, which comes with the regularisation choice to taper $\mathbb{E}_{\boldsymbol{\zeta}\sim\rho}\left[\nabla_{\boldsymbol{\zeta}} f\right]$ outside of the range of $\boldsymbol{\zeta}$. Applied to $f(\boldsymbol{\zeta}) = \mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\zeta})$, this yields:

$$\mathbb{E}_{\boldsymbol{\zeta}\sim\rho}\left[\nabla_{\boldsymbol{\zeta}} \mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\zeta})\right] = \boldsymbol{\Sigma}_{\rho}^{\dagger}\mathrm{Cov}_{\boldsymbol{\zeta}\sim\rho}\left[\boldsymbol{\zeta}, \mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\zeta})\right], \tag{D4a}$$

which, if $\boldsymbol{\Sigma}_{\rho}$ is full rank, can be written (usual Stein lemma)

$$\mathbb{E}_{\boldsymbol{\zeta}\sim\rho}\left[\nabla_{\boldsymbol{\zeta}} \mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\zeta})\right] = \boldsymbol{\Sigma}_{\rho}^{-1}\mathrm{Cov}_{\boldsymbol{\zeta}\sim\rho}\left[\boldsymbol{\zeta}, \mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\zeta})\right]. \tag{D4b}$$

Appendix E: Comparing the empirical and theoretical mean marginal gains

In Sect. 3.3, we showed that the theoretical mean marginal gain $\overline{\boldsymbol{\Gamma}}$ is expected to be a good approximation of the less simple theoretical mean

$$\widetilde{\boldsymbol{\Gamma}} = \int \mathrm{d}\mathbf{x}\mathrm{d}\boldsymbol{\zeta}\, \pi(\mathbf{x})\rho(\boldsymbol{\zeta})\nabla_{\mathbf{x}}\nabla_{\boldsymbol{\zeta}}\mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\zeta}). \tag{E1}$$

The latter can now be related to the empirical mean of $\boldsymbol{\Gamma}$ over a long DA run, denoted $\widehat{\boldsymbol{\Gamma}}$, and which is obtained from numerical experiments. This is meant to ensure that a pattern emerging from our approximation of $\overline{\boldsymbol{\Gamma}}$, is nonetheless consistent with those learned through $\mathbf{a}_{\boldsymbol{\theta}}$ over the training DA dataset. Hence, given a long trajectory of true states and projected innovations $\mathcal{T}_{\mathbf{x},\boldsymbol{\zeta}} = \left\{\left(\mathbf{x}^{\mathrm{t}}_k, \boldsymbol{\zeta}_k\right)\right\}_{k=1,\ldots,K} \subset (\mathbf{E}_{\mathrm{x}} \times \mathbf{E}_{\mathrm{x}})^K$, the empirical sensitivity associated to $\widetilde{\boldsymbol{\Gamma}}$, and hence $\overline{\boldsymbol{\Gamma}}$, should be

$$\widehat{\mathbf{\Gamma}} \overset{\Delta}{=} \langle \mathbf{\Gamma}(\mathbf{x}^{\mathrm{t}}, \boldsymbol{\zeta}) \rangle_{(\mathbf{x}^{\mathrm{t}}, \boldsymbol{\zeta}) \in \mathcal{T}_{\mathbf{x},\boldsymbol{\zeta}}} \tag{E2a}$$

$$\underset{K \to \infty}{=} \int \mathrm{d}\mathbf{x}^{\mathrm{t}} \mathrm{d}\boldsymbol{\zeta}\; p(\mathbf{x}^{\mathrm{t}}, \boldsymbol{\zeta}) \mathbf{\Gamma}(\mathbf{x}^{\mathrm{t}}, \boldsymbol{\zeta}), \tag{E2b}$$

where $p(\mathbf{x}^{\mathrm{t}}, \boldsymbol{\zeta})$ is the joint distribution of $\mathbf{x}^{\mathrm{t}}$ and $\boldsymbol{\zeta}$. However, $\boldsymbol{\zeta}_k$ not only depends on $\mathbf{x}_k^{\mathrm{t}}$ but also on the forecast $\mathbf{x}_k^{\mathrm{f}}$, so that:

$$\widehat{\mathbf{\Gamma}} = \int \mathrm{d}\mathbf{x}^{\mathrm{t}} \mathrm{d}\mathbf{x}^{\mathrm{f}} \mathrm{d}\boldsymbol{\zeta}\; p(\mathbf{x}^{\mathrm{t}}, \mathbf{x}^{\mathrm{f}}, \boldsymbol{\zeta}) \mathbf{\Gamma}(\mathbf{x}^{\mathrm{t}}, \boldsymbol{\zeta}), \tag{E3}$$

or, introducing the forecast error $\mathbf{e}^{\mathrm{f}} \overset{\Delta}{=} \mathbf{x}^{\mathrm{f}} - \mathbf{x}^{\mathrm{t}}$,

$$\widehat{\mathbf{\Gamma}} = \int \mathrm{d}\mathbf{x}^{\mathrm{t}} \mathrm{d}\mathbf{e}^{\mathrm{f}} \mathrm{d}\boldsymbol{\zeta}\; p(\mathbf{x}^{\mathrm{t}}, \mathbf{e}^{\mathrm{f}}, \boldsymbol{\zeta}) \mathbf{\Gamma}(\mathbf{x}^{\mathrm{t}} + \mathbf{e}^{\mathrm{f}}, \boldsymbol{\zeta}). \tag{E4}$$

By marginalising over $\mathbf{x}^{\mathrm{t}}$, we have

$$\widehat{\mathbf{\Gamma}} = \int \mathrm{d}\mathbf{x}^{\mathrm{t}} \pi(\mathbf{x}^{\mathrm{t}}) \int \mathrm{d}\mathbf{e}^{\mathrm{f}} \mathrm{d}\boldsymbol{\zeta}\; p(\mathbf{e}^{\mathrm{f}}, \boldsymbol{\zeta} | \mathbf{x}^{\mathrm{t}}) \mathbf{\Gamma}(\mathbf{x}^{\mathrm{t}} + \mathbf{e}^{\mathrm{f}}, \boldsymbol{\zeta}) \tag{E5a}$$

$$= \int \mathrm{d}\mathbf{x}^{\mathrm{t}} \pi(\mathbf{x}^{\mathrm{t}}) \int \mathrm{d}\mathbf{e}^{\mathrm{f}} \mathrm{d}\boldsymbol{\zeta}\; p(\boldsymbol{\zeta} | \mathbf{e}^{\mathrm{f}}) p(\mathbf{e}^{\mathrm{f}} | \mathbf{x}^{\mathrm{t}}) \mathbf{\Gamma}(\mathbf{x}^{\mathrm{t}} + \mathbf{e}^{\mathrm{f}}, \boldsymbol{\zeta}) \tag{E5b}$$

$$\approx \int \mathrm{d}\mathbf{x}^{\mathrm{t}} \pi(\mathbf{x}^{\mathrm{t}}) \int \mathrm{d}\boldsymbol{\zeta} \left[p(\boldsymbol{\zeta} | \mathbf{e}^{\mathrm{f}}) \mathbf{\Gamma}(\mathbf{x}^{\mathrm{t}} + \mathbf{e}^{\mathrm{f}}, \boldsymbol{\zeta}) \right]_{\mathbf{e}^{\mathrm{f}} = \mathbf{0}} \tag{E5c}$$

$$= \int \mathrm{d}\mathbf{x}^{\mathrm{t}} \pi(\mathbf{x}^{\mathrm{t}}) \int \mathrm{d}\boldsymbol{\zeta}\; \rho(\boldsymbol{\zeta}) \mathbf{\Gamma}(\mathbf{x}^{\mathrm{t}}, \boldsymbol{\zeta}) = \widetilde{\mathbf{\Gamma}}. \tag{E5d}$$

From Eq. (E5a) to Eq. (E5b), we use $p(\mathbf{e}^{\mathrm{f}}, \boldsymbol{\zeta} | \mathbf{x}^{\mathrm{t}}) = p(\boldsymbol{\zeta} | \mathbf{e}^{\mathrm{f}}) p(\mathbf{e}^{\mathrm{f}} | \mathbf{x}^{\mathrm{t}})$ since the full dependence of $\boldsymbol{\zeta}$ on $\mathbf{x}^{\mathrm{t}}$ is in $\mathbf{e}^{\mathrm{f}}$. From Eq. (E5b) to Eq. (E5c), we assume that the forecast error norm is small compared to the norm of $\mathbf{x}^{\mathrm{t}}$, which should indeed be the case in the weak assimilation regime. From Eq. (E5c) to Eq. (E5d), we approximate $p(\boldsymbol{\zeta} | \mathbf{e}^{\mathrm{f}} = \mathbf{0})$ by the marginal $\rho(\boldsymbol{\zeta})$. This points to the (reasonable) approximations made when identifying $\widetilde{\mathbf{\Gamma}}$, Eq. (E1), with the empirical marginal gain $\widehat{\mathbf{\Gamma}}$, $\langle \mathbf{\Gamma}(\mathbf{x}, \boldsymbol{\zeta}) \rangle_{(\mathbf{x},\boldsymbol{\zeta}) \in \mathcal{T}_{\mathbf{x},\boldsymbol{\zeta}}}$, emerging from a long DA run.

Moreover, a formal expression for $\rho(\boldsymbol{\zeta} | \mathbf{e}^{\mathrm{f}})$ which appeared in the previous derivation is

$$p(\boldsymbol{\zeta} | \mathbf{e}^{\mathrm{f}}) = \int \mathrm{d}\boldsymbol{\varepsilon} \mathrm{d}\mathbf{H} \mathrm{d}\mathbf{R}\; p(\boldsymbol{\varepsilon}, \mathbf{H}, \mathbf{R} | \mathbf{e}^{\mathrm{f}}) \times \delta\big(\boldsymbol{\zeta} - \mathbf{H}^{\intercal} \mathbf{R}^{-1} (\boldsymbol{\varepsilon} - \mathbf{H}\mathbf{e}^{\mathrm{f}})\big), \tag{E6}$$

where $p(\boldsymbol{\varepsilon}, \mathbf{H}, \mathbf{R} | \mathbf{e}^{\mathrm{f}})$ is the pdf of the joint distribution for the observation error $\boldsymbol{\varepsilon}$, the observation operator $\mathbf{H}$, and the observation error covariance matrix $\mathbf{R}$, given the forecast error $\mathbf{e}^{\mathrm{f}}$. Marginalising over $\mathbf{e}^{\mathrm{f}}$, we obtain

$$\rho(\boldsymbol{\zeta}) = \int \mathrm{d}\mathbf{e}^{\mathrm{f}}\; p(\boldsymbol{\zeta} | \mathbf{e}^{\mathrm{f}}) p(\mathbf{e}^{\mathrm{f}}) \tag{E7a}$$

$$= \int \mathrm{d}\mathbf{e}^{\mathrm{f}} \mathrm{d}\boldsymbol{\varepsilon} \mathrm{d}\mathbf{H} \mathrm{d}\mathbf{R}\; p(\mathbf{e}^{\mathrm{f}}, \boldsymbol{\varepsilon}, \mathbf{H}, \mathbf{R}) \times \delta\big(\boldsymbol{\zeta} - \mathbf{H}^{\intercal} \mathbf{R}^{-1} (\boldsymbol{\varepsilon} - \mathbf{H}\mathbf{e}^{\mathrm{f}})\big). \tag{E7b}$$

This expression is helpful to formally investigate the symmetries of the distribution of $\boldsymbol{\zeta}$, for which Eq. (18) applies.

Appendix F: Proof of the equivariance of the marginal gain tensor

We wish to prove the equivariance of the marginal gain tensor, i.e. that for all $\mathbf{x}$ and $\boldsymbol{\zeta}$, $\mathbf{\Gamma}(g \circ \mathbf{x}, g \circ \boldsymbol{\zeta}) = g \otimes g^{\intercal} \otimes g^{\intercal} \circ \mathbf{\Gamma}(\mathbf{x}, \boldsymbol{\zeta})$, under the action of an isometry $g \in \mathcal{G}$. The action of g on either state vector $\mathbf{x}$ or $\boldsymbol{\zeta}$ is represented by an orthogonal matrix $\mathbf{G}$: $g \circ \mathbf{x} = \mathbf{G}\mathbf{x}$, $g \circ \boldsymbol{\zeta} = \mathbf{G}\boldsymbol{\zeta}$.

We say that g preserves the fibres (preimages) of $\mathcal{H}_k$ if, for all $\mathbf{x}, \mathbf{x}'$ such that $\mathcal{H}_k(\mathbf{x}) = \mathcal{H}_k(\mathbf{x}')$ one has $\mathcal{H}_k(g \circ \mathbf{x}) = \mathcal{H}_k(g \circ \mathbf{x}')$. An isometry g_{y} defined over the observation space of $\mathcal{H}_k$ can then be associated to such g through $g_{\mathrm{y}} \circ \mathcal{H}_k(g^{\intercal} \circ (\cdot)) = \mathcal{H}_k(\cdot)$. The action of such induced isometry g_{y} is represented by the orthogonal matrix $\mathbf{G}_{\mathrm{y}}$: $g_{\mathrm{y}} \circ \mathbf{y} = \mathbf{G}_{\mathrm{y}}\mathbf{y}$. Because $\mathbf{G}$ and $\mathbf{G}_{\mathrm{y}}$ are orthogonal, one has $\mathbf{G}^{-1} = \mathbf{G}^{\intercal}$ and $\mathbf{G}_{\mathrm{y}}^{-1} = \mathbf{G}_{\mathrm{y}}^{\intercal}$.

With these definitions in hand, we consider a maximal group of isometries $\mathcal{G}$ defined over the state space such that

- $\mathcal{G}$ preserves the fibres of all $\mathcal{H}_k$, inducing a group of isometries $\mathcal{G}_{\mathrm{y}}$ (possibly for each k). $\mathcal{G}_{\mathrm{y}}$ is isomorphic to $\mathcal{G}$,
- the associated $\mathcal{G}_{\mathrm{y}}$ satisfies $\mathbf{G}_{\mathrm{y}} \mathbf{R}_k \mathbf{G}_{\mathrm{y}}^{\intercal} = \mathbf{R}_k$, which accounts for any heteroscedasticity of the observation error statistics,
- $\mathcal{G}$ commutes with the autonomous dynamics, i.e. $\mathbf{G}\mathcal{M}(\mathbf{G}^{\intercal}(\cdot)) = \mathcal{M}(\cdot)$.

A consequence of the equivariance with respect to $\mathcal{H}_k$, i.e. $\mathbf{G}_{\mathrm{y}} \mathcal{H}_k(\mathbf{G}^{\intercal}(\cdot)) = \mathcal{H}_k(\cdot)$, is $\mathbf{G}_{\mathrm{y}} \mathbf{H}_k \mathbf{G}^{\intercal} = \mathbf{H}_k$, readily obtained by differentiation.[1] Coming back to the proof of the equivariance we first consider the optimisation problem that defines $(\mathbf{x}, \boldsymbol{\zeta}) \longmapsto \mathbf{a}_{\boldsymbol{\theta}}(g \circ \mathbf{x}, g \circ \boldsymbol{\zeta})$ for any g in such $\mathcal{G}$:

$$\mathcal{L}\big(\boldsymbol{\theta} \big| \{\mathbf{x}_k^{\mathrm{t}}, \mathbf{y}_k\}\big) = \sum_{k=1}^{K} \big\| \mathbf{x}_k^{\mathrm{t}} - \mathbf{x}_k^{\mathrm{a}}(\boldsymbol{\theta}) \big\|^2, \tag{F1a}$$

$$\mathbf{x}_k^{\mathrm{a}} = \mathbf{x}_k^{\mathrm{f}} + \mathbf{a}_{\boldsymbol{\theta}}\big(\mathbf{G}\mathbf{x}_k^{\mathrm{f}}, \mathbf{G}\boldsymbol{\zeta}_k\big), \tag{F1b}$$

$$\boldsymbol{\zeta}_k = \mathbf{H}_k^{\intercal} \mathbf{R}_k^{-1} \big(\mathbf{y}_k - \mathcal{H}_k\big(\mathbf{x}_k^{\mathrm{f}}\big)\big), \tag{F1c}$$

$$\mathbf{x}_{k+1}^{\mathrm{f}} = \mathcal{M}\big(\mathbf{x}_k^{\mathrm{a}}\big), \tag{F1d}$$

which is equivalent to

$$\mathcal{L}\big(\boldsymbol{\theta} \big| \{\mathbf{x}_k^{\mathrm{t}}, \mathbf{y}_k\}\big) = \sum_{k=1}^{K} \big\| \mathbf{G}\mathbf{x}_k^{\mathrm{t}} - \mathbf{G}\mathbf{x}_k^{\mathrm{a}}(\boldsymbol{\theta}) \big\|^2, \tag{F2a}$$

$$\mathbf{G}\mathbf{x}_k^{\mathrm{a}} = \mathbf{G}\mathbf{x}_k^{\mathrm{f}} + \mathbf{G}\mathbf{a}_{\boldsymbol{\theta}}\big(\mathbf{G}\mathbf{x}_k^{\mathrm{f}}, \mathbf{G}\boldsymbol{\zeta}_k\big), \tag{F2b}$$

$$\mathbf{G}\boldsymbol{\zeta}_k = \mathbf{G}\mathbf{H}_k^{\intercal} \mathbf{G}_{\mathrm{y}}^{\intercal} \mathbf{G}_{\mathrm{y}} \mathbf{R}_k^{-1} \mathbf{G}_{\mathrm{y}}^{\intercal} \mathbf{G}_{\mathrm{y}} \times \big(\mathbf{y}_k - \mathcal{H}_k\big(\mathbf{G}^{\intercal}\mathbf{G}\mathbf{x}_k^{\mathrm{f}}\big)\big), \tag{F2c}$$

$$\mathbf{G}\mathbf{x}_{k+1}^{\mathrm{f}} = \mathbf{G}\mathcal{M}\big(\mathbf{G}^{\intercal}\mathbf{G}\mathbf{x}_k^{\mathrm{a}}\big), \tag{F2d}$$

[1] A simplifying classical trick would be to consider that the physical system is fully observed at any time, with observation error variances that can take infinite values in the absence of observations, which would shift the observation equivariance constraint onto that of its statistics.

where Eq. (F2a) is obtained from Eq. (F1a) because $\mathbf{G}$ is orthogonal, and Eqs. (F2b, F2c, F2d) are obtained from a multiplication on the left by $\mathbf{G}$ and insertion of $\mathbf{G}^\intercal\mathbf{G} = \mathbf{I}_\mathrm{x}$ and $\mathbf{G}_\mathrm{y}^\intercal\mathbf{G}_\mathrm{y} = \mathbf{I}_\mathrm{y}$ in Eqs. (F1b, F1c, F1d). Hence, denoting $\tilde{\mathbf{x}}_k^\mathrm{a} = \mathbf{G}\mathbf{x}_k^\mathrm{a}$, $\tilde{\mathbf{x}}_k^\mathrm{f} = \mathbf{G}\mathbf{x}_k^\mathrm{f}$, $\tilde{\boldsymbol{\zeta}}_k = \mathbf{G}\boldsymbol{\zeta}_k$, the problem is reformulated as

$$\mathcal{L}\big(\boldsymbol{\theta}\big|\{\mathbf{x}_k^\mathrm{t}, \mathbf{y}_k\}\big) = \sum_{k=1}^{K} \big\|\mathbf{G}\mathbf{x}_k^\mathrm{t} - \tilde{\mathbf{x}}_k^\mathrm{a}(\boldsymbol{\theta})\big\|^2, \tag{F3a}$$

$$\tilde{\mathbf{x}}_k^\mathrm{a} = \tilde{\mathbf{x}}_k^\mathrm{f} + \mathbf{G}\mathbf{a}_{\boldsymbol{\theta}}\big(\tilde{\mathbf{x}}_k^\mathrm{f}, \tilde{\boldsymbol{\zeta}}_k\big), \tag{F3b}$$

$$\tilde{\boldsymbol{\zeta}}_k = \big(\mathbf{G}_\mathrm{y}\mathbf{H}_k\mathbf{G}^\intercal\big)^\intercal\big(\mathbf{G}_\mathrm{y}\mathbf{R}_k^{-1}\mathbf{G}_\mathrm{y}^\intercal\big) \times \big(\mathbf{G}_\mathrm{y}\mathbf{y}_k - \mathbf{G}_\mathrm{y}\mathcal{H}_k\big(\mathbf{G}^\intercal\tilde{\mathbf{x}}_k^\mathrm{f}\big)\big), \tag{F3c}$$

$$\tilde{\mathbf{x}}_{k+1}^\mathrm{f} = \mathbf{G}\mathcal{M}\big(\mathbf{G}^{-1}\tilde{\mathbf{x}}_k^\mathrm{a}\big). \tag{F3d}$$

Leveraging the assumptions on $\mathcal{G}$, we finally obtain

$$\mathcal{L}\big(\boldsymbol{\theta}\big|\{\mathbf{x}_k^\mathrm{t}, \mathbf{y}_k\}\big) = \sum_{k=1}^{K} \big\|\mathbf{G}\mathbf{x}_k^\mathrm{t} - \tilde{\mathbf{x}}_k^\mathrm{a}(\boldsymbol{\theta})\big\|^2, \tag{F4a}$$

$$\tilde{\mathbf{x}}_k^\mathrm{a} = \tilde{\mathbf{x}}_k^\mathrm{f} + \mathbf{G}\mathbf{a}_{\boldsymbol{\theta}}\big(\tilde{\mathbf{x}}_k^\mathrm{f}, \tilde{\boldsymbol{\zeta}}_k\big), \tag{F4b}$$

$$\tilde{\boldsymbol{\zeta}}_k = \mathbf{H}_k^\intercal\mathbf{R}_k^{-1}\big(\mathbf{G}_\mathrm{y}\mathbf{y}_k - \mathcal{H}_k\big(\tilde{\mathbf{x}}_k^\mathrm{f}\big)\big), \tag{F4c}$$

$$\tilde{\mathbf{x}}_{k+1}^\mathrm{f} = \mathcal{M}\big(\tilde{\mathbf{x}}_k^\mathrm{a}\big). \tag{F4d}$$

This shows that $\mathbf{a}_{\boldsymbol{\theta}}(\mathbf{G}\cdot, \mathbf{G}\cdot)$ is the solution of Eq. (F1) whose input is the dataset $\{\mathbf{x}_k^\mathrm{t}, \mathbf{y}_k\}$, while $\mathbf{G}\mathbf{a}_{\boldsymbol{\theta}}(\cdot,\cdot)$ is the solution of Eq. (F4) whose input is the dataset $\{\mathbf{G}\mathbf{x}_k^\mathrm{t}, \mathbf{G}_\mathrm{y}\mathbf{y}_k = \mathbf{G}_\mathrm{y}\mathcal{H}_k(\mathbf{x}_k^\mathrm{t}) + \mathbf{G}_\mathrm{y}\boldsymbol{\varepsilon}_k = \mathcal{H}_k(\mathbf{G}\mathbf{x}_k^\mathrm{t}) + \mathbf{G}_\mathrm{y}\boldsymbol{\varepsilon}_k\}$. Since both the invariant distribution of the dynamics and the distribution of the observations errors are invariant under $\mathcal{G}$, the datasets $\{\mathbf{x}_k^\mathrm{t}, \mathbf{y}_k\}$ and $\{\mathbf{G}\mathbf{x}_k^\mathrm{t}, \mathbf{G}_\mathrm{y}\mathbf{y}_k\}$ must asymptotically yield the same solution for a large enough number of samples K. This proves the equivariance, $\forall g \in \mathcal{G}, \forall \mathbf{x} \in \mathbf{E}_\mathrm{x}, \forall \boldsymbol{\zeta} \in \mathbf{E}_\mathrm{x}$:

$$\mathbf{a}_{\boldsymbol{\theta}}(g \circ \mathbf{x}, g \circ \boldsymbol{\zeta}) = g \circ \mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\zeta}). \tag{F5}$$

Then, taking the gradient with respect to $\mathbf{x}$ and $\boldsymbol{\zeta}$ of $\mathbf{a}_{\boldsymbol{\theta}}(g \circ \mathbf{x}, g \circ \boldsymbol{\zeta})$ yields a covariant action $g \otimes g$ onto the tensor factors for $\mathbf{x}$ and $\boldsymbol{\zeta}$, alternatively a contravariant action $g^\intercal \otimes g^\intercal$ on to the right-hand side of Eq. (F5), which proves the equivariance of $\boldsymbol{\Gamma}$, Eq. (17). The equivariance assumption on the observation operator is rather stringent. A weaker assumption is to assume that $\{\mathbf{G}_\mathrm{y}\mathcal{H}_k(\mathbf{G}^\intercal\mathbf{x}_k^\mathrm{f})\}_{k=1,\ldots,K}$ almost spans the same set as $\{\mathcal{H}_k(\mathbf{x}_k^\mathrm{f})\}_{k=1,\ldots,K}$ for $K \to \infty$. Then the optimisation problems Eq. (F1) and Eq. (F4) should almost coincide. This is for instance useful when one considers random observation operators for which operator instances have no specific symmetry, while their distribution does exhibit the symmetry, a case occurring in Sect. 3.3.4.

Appendix G: Sleek representation of the mean marginal gain

Assume that the states and projected innovations are defined as fields over a physical manifold $\mathcal{D}$, and further discretised at N_x collocation space points of $\mathcal{D}$ indexed by $i \in [\![1, \ldots, N_\mathrm{x}]\!]$. Hence, $\mathcal{G}$ is a discrete group of isometries. As a consequence, $g \in \mathcal{G}$ can be seen as a bijection of $[\![1, \ldots, N_\mathrm{x}]\!]$ and the action of $g \in \mathcal{G}$ on the fields $\mathbf{x}$ and $\boldsymbol{\zeta}$ reads

$$[g \circ \mathbf{x}]_i = [\mathbf{x}]_{g(i)}, \quad [g \circ \boldsymbol{\zeta}]_j = [\boldsymbol{\zeta}]_{g(j)}, \tag{G1}$$

respectively. Likewise, the action of $g \in \mathcal{G}$ on $\overline{\boldsymbol{\Gamma}}$ is, for all i, j, l:

$$\big[g \odot \overline{\boldsymbol{\Gamma}}\big]_{ijl} = \big[g \otimes g^\intercal \otimes g^\intercal \circ \overline{\boldsymbol{\Gamma}}\big]_{ijl} = \overline{\Gamma}_{g(i)\,g^\intercal(j)\,g^\intercal(l)}, \tag{G2}$$

so that Eq. (21) reads, for all i, j, l:

$$\overline{\Gamma}_{ijl} = \overline{\Gamma}_{g(i)\,g^\intercal(j)\,g^\intercal(l)}. \tag{G3}$$

Let us choose one of the collocation points in $\mathcal{D}$ with index $r \in [\![1, \ldots, N_\mathrm{x}]\!]$. With the above assumptions, the orbit of the site indexed by r under the action of $\mathcal{G}$ is $[\![1, \ldots, N_\mathrm{x}]\!]$. We can then define a 2-tensor $\overline{\boldsymbol{\Omega}}^{(r)}$ componentwise by, for all i, j:

$$\overline{\Omega}_{ij}^{(r)} = \overline{\Gamma}_{ijr}. \tag{G4}$$

For all $l \in [\![1, \ldots, N_\mathrm{x}]\!]$, we can pick at least one $g_l^r \in \mathcal{G}$ such that $g_l^r(r) = l$ and let us denote its inverse by g_r^l which coincides with its adjoint $g^\intercal$ and satisfies $g_r^l(l) = r$ in particular. Hence, we have from Eq. (21) and for all i, j, l:

$$\overline{\Gamma}_{ijl} = \overline{\Gamma}_{g_l^r(i)\,g_r^l(j)\,g_r^l(l)} = \overline{\Gamma}_{g_l^r(i)\,g_r^l(j)\,r} = \overline{\Omega}^{(r)}_{g_l^r(i)\,g_r^l(j)}. \tag{G5}$$

As a consequence of the symmetry and Eq. (G5), the 3-tensor $\overline{\boldsymbol{\Gamma}}$ can be entirely specified by the 2-tensor $\overline{\boldsymbol{\Omega}}^{(r)}$, where r, a reference site index, is arbitrarily chosen. In the case where $\mathcal{D}$ is one-dimensional (as for L96), $\overline{\boldsymbol{\Omega}}^{(r)}$ is a matrix, hence depictable and more easily interpretable.

Appendix H: Numerical computation of the mean marginal gain

The mean marginal gain can be computed from states of a trajectory $\mathcal{T}_\mathbf{x}$ of the ergodic dynamics, and the ability to evaluate $\mathbf{x} \longmapsto \boldsymbol{\Gamma}(\mathbf{x})$ as defined by Eq. (10b). The trajectory should be long enough so that its states adequately sample the invariant distribution π. From Eq. (12), we hence have the empirical estimator:

$$\overline{\boldsymbol{\Gamma}} = \langle \boldsymbol{\Gamma}(\mathbf{x}) \rangle_{\mathbf{x} \in \mathcal{T}_\mathbf{x}} = \frac{1}{K} \sum_{k=1}^{K} \boldsymbol{\Gamma}(\mathbf{x}_k). \tag{H1}$$

As a result, the computational complexity of the mean marginal gain is proportional to K, but may be significantly alleviated by the presence of symmetries as discussed before. Such symmetries must make both π and ρ invariant even though the definition of $\overline{\boldsymbol{\Gamma}}$ only implicitly depends on ρ.

We now turn to the estimation of the marginal gain $\boldsymbol{\Gamma}(\mathbf{x})$. Its computation can be achieved through several routes with distinct numerical complexities which, as approximations,

may not be equivalent and may lead to mildly differing results.

The first way to compute $\mathbf{\Gamma}(\mathbf{x})$ is through automatic differentiation. As a second-order sensitivity of $\mathbf{a}_{\boldsymbol{\theta}}$ with respect to $\mathbf{x}$ and $\boldsymbol{\zeta}$, $\mathbf{\Gamma}(\mathbf{x}) = \nabla_{\mathbf{x}}\nabla_{\boldsymbol{\zeta}}\mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x},\boldsymbol{\zeta})_{|\boldsymbol{\zeta}=\mathbf{0}}$ requires taking the Jacobian of $\mathbf{a}_{\boldsymbol{\theta}}$ twice. Hence, such computation through either JAX, PyTorch or TensorFlow, can be prohibitive, with a substantial need for GPU memory. On an NVIDIA RTX5000 Ada GPU with 32 GB of memory, using the JAX-inspired PyTorch `torch.func` module (https://pytorch.org/docs/stable/func.html, last access: 7 August 2026), we found it to be achievable with the L96 model, difficult with a Kuramoto-Sivashinsky model (Kuramoto and Tsuzuki, 1976; Sivashinsky, 1977), but prohibitive with a single-layer QG model on the sphere. Hence, it is likely to be impractical with high-dimensional models.

Note that the mean Eq. (H1) can be computed through updates whenever a new $\mathbf{\Gamma}(\mathbf{x}_k)$ is computed, preventing the need to store them. Moreover, when exploiting symmetries of $\mathcal{G}$, the intermediate tensor

$$\Omega^{(r)}_{ij}(\mathbf{x}_k) = \frac{1}{N_{\mathrm{x}}}\sum_{l=1}^{N_{\mathrm{x}}} \Gamma_{g^l_r(i)\, g^r_l(j) l}(\mathbf{x}_k), \tag{H2a}$$

can be computed, and will contribute to the computation of $\overline{\mathbf{\Omega}}^{(r)}$ through the update of

$$\overline{\mathbf{\Omega}}^{(r)} = \langle \mathbf{\Omega}^{(r)}(\mathbf{x})\rangle_{\mathbf{x}\in\mathcal{T}_{\mathbf{x}}} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{\Omega}^{(r)}(\mathbf{x}_k). \tag{H2b}$$

In Boc24, either the gain $\mathbf{K}(\mathbf{x})$ or $\mathbf{P}^{\mathrm{a}}(\mathbf{x})$ were obtained by generating an ensemble of perturbations to feed a regression. The same idea can be used for $\mathbf{\Gamma}(\mathbf{x})$, once again assuming a quasi-linear behaviour of $\mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x},\boldsymbol{\zeta})$ and $\nabla_{\mathbf{x}}\mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x},\boldsymbol{\zeta})$ as functions of $\boldsymbol{\zeta}$. From the results in Sect. 3.3.1 and Appendix E, we infer that

$$\begin{aligned}
\mathbf{\Gamma}(\mathbf{x}) &= \nabla_{\mathbf{x}}\nabla_{\boldsymbol{\zeta}}\mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x},\boldsymbol{\zeta})_{|\boldsymbol{\zeta}=\mathbf{0}} && \text{(H3a)}\\
&\approx \mathbb{E}_{\boldsymbol{\zeta}\sim\rho}\left[\nabla_{\mathbf{x}}\nabla_{\boldsymbol{\zeta}}\mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x},\boldsymbol{\zeta})\right] && \text{(H3b)}\\
&\approx \mathbf{\Sigma}_{\rho}^{\dagger}\mathrm{Cov}_{\boldsymbol{\zeta}\sim\rho}\left[\boldsymbol{\zeta}, \nabla_{\mathbf{x}}\mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x},\boldsymbol{\zeta})\right], && \text{(H3c)}
\end{aligned}$$

which tells that a sampling approach can be applied to $\nabla_{\mathbf{x}}\mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x},\cdot)$. Assuming $\mathbf{a}_{\boldsymbol{\theta}}$ is regular enough, the gradients with respect to $\mathbf{x}$ and $\boldsymbol{\zeta}$ commute and we also have $\mathbf{\Gamma}(\mathbf{x}) \approx \nabla_{\mathbf{x}}\mathbb{E}_{\boldsymbol{\zeta}\sim\rho}\left[\nabla_{\boldsymbol{\zeta}}\mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x},\boldsymbol{\zeta})\right]$, although this could considerably complexify backpropagation if automatic differentiation is used to handle $\nabla_{\mathbf{x}}$.

Automatic differentiation is hence used only once for the computation of the Jacobian $\nabla_{\mathbf{x}}\mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x},\cdot)$, as opposed to the full automatic differentiation approach. Hence, $\mathbf{\Gamma}(\mathbf{x})$ can be computed using a *composite* Monte Carlo/differentiation approach. The details of the subsequent regression are reported in Appendix I.

Appendix I: Regression for the composite mean marginal gain

An ensemble of N_{p} perturbations $\partial\mathbf{a}_p$ generated from N_{p} samples $\boldsymbol{\zeta}_p \sim N(\mathbf{0}, \mathbf{\Xi})$ for $p = 1, \ldots, N_{\mathrm{p}}$ should first be computed,

$$\left[\partial\mathbf{a}_p\right]_{il} = \left[\partial_{x_l}\mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x},\boldsymbol{\zeta}_p)\right]_i, \tag{I1}$$

via an ensemble of first-order Jacobians. In the best linear unbiased estimator framework, the covariance matrix $\mathbf{\Xi}$ should roughly match $\mathbf{H}^{\intercal}\mathbf{R}^{-1}(\mathbf{R}+\mathbf{H}\mathbf{P}^{\mathrm{f}}\mathbf{H}^{\intercal})\mathbf{R}^{-1}\mathbf{H}$ where $\mathbf{P}^{\mathrm{f}}$ and $\mathbf{R}+\mathbf{H}\mathbf{P}^{\mathrm{f}}\mathbf{H}^{\intercal}$ are the forecast error and innovation covariance matrices, respectively. Hence, in the weak assimilation regime, we can use the approximation $\mathbf{\Xi} \approx \mathbf{H}^{\intercal}\mathbf{R}^{-1}\mathbf{H}$, which should be regarded as a scale for the perturbations anyway, and generate samples with $\boldsymbol{\zeta} = \mathbf{H}^{\intercal}\mathbf{R}^{-\frac{1}{2}}\boldsymbol{\xi}$, where $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{\mathrm{y}})$. Introducing for $p = 1, \ldots, N_{\mathrm{p}}$, the recentred samples

$$\partial\mathbf{a}'_p = \partial\mathbf{a}_p - N_{\mathrm{p}}^{-1}\sum_{p=1}^{N_{\mathrm{p}}}\partial\mathbf{a}_p, \quad \boldsymbol{\zeta}'_p = \boldsymbol{\zeta}_p - N_{\mathrm{p}}^{-1}\sum_{p=1}^{N_{\mathrm{p}}}\boldsymbol{\zeta}_p, \tag{I2}$$

we have from the definitions $\mathbf{C} \triangleq \mathrm{Cov}_{\boldsymbol{\zeta}\sim\rho}\left[\boldsymbol{\zeta}, \nabla_{\mathbf{x}}\mathbf{a}_{\boldsymbol{\theta}}(\mathbf{x},\boldsymbol{\zeta})\right]$ and $\mathbf{D} \triangleq \mathbf{\Sigma}_{\rho}$:

$$[\mathbf{C}]_{ijl} \approx \sum_{p=1}^{N_{\mathrm{p}}}\left[\boldsymbol{\zeta}'_p\right]_j\left[\partial\mathbf{a}'_p\right]_{il}, \quad [\mathbf{D}]_{jl} \approx \sum_{p=1}^{N_{\mathrm{p}}}\left[\boldsymbol{\zeta}'_p\right]_j\left[\boldsymbol{\zeta}'_p\right]_l, \tag{I3}$$

which, from Eq. (H3a), yields

$$[\mathbf{\Gamma}]_{ijl} = \sum_{m=1}^{N_{\mathrm{x}}}[\mathbf{C}]_{iml}\left[\mathbf{D}^{-1}\right]_{mj}. \tag{I4}$$

Obviously, in a high-dimensional context, the approach would necessitate reduction methods such as Lanczos vectors or (randomised) singular value decompositions, and the generation of the ensemble would require a massive vectorisation on GPUs. $\mathbf{\Omega}_r$ can then be computed by averaging $\mathbf{\Gamma}$ using, e.g., Eqs. (H2). Moreover, it is not difficult to show that $\mathbf{\Omega}_r$, as defined by Eq. (H2a), can alternatively be obtained by first averaging over $\mathbf{C}$ and $\mathbf{D}$ before performing the inversion of the regression, that is:

$$\mathbf{\Omega}_r = \overline{\mathbf{C}}\,\overline{\mathbf{D}}^{-1}, \tag{I5a}$$

$$\left[\overline{\mathbf{C}}\right]_{ijl} = \frac{1}{N_{\mathrm{x}}}\sum_{s=1}^{N_{\mathrm{x}}}[\mathbf{C}]_{g^i_s(i)\, g^s_l(j)\, s}, \tag{I5b}$$

$$\left[\overline{\mathbf{D}}\right]_{ij} = \frac{1}{N_{\mathrm{x}}}\sum_{s=1}^{N_{\mathrm{x}}}[\mathbf{D}]_{g^j_s(i)\, s}. \tag{I5c}$$

Either way, the composite approach may turn out numerically cheaper than the full differentiation approach.

Code and data availability. The core source code for training and testing one-dimensional DANs is publicly available under https://github.com/cerea-daml/Dan1D and Zenodo (https://doi.org/10.5281/zenodo.21427793, Bocquet, 202026).

Author contributions. MB performed the mathematical, algorithmic, and numerical analysis. MB and TSF discussed the implications of the results. All the authors worked on the structure of the manuscript. All the authors reviewed and edited the manuscript.

Competing interests. The contact author has declared that none of the authors has any competing interests.

Disclaimer. Publisher's note: Copernicus Publications remains neutral with regard to jurisdictional claims made in the text, published maps, institutional affiliations, or any other geographical representation in this paper. The authors bear the ultimate responsibility for providing appropriate place names. Views expressed in the text are those of the authors and do not necessarily reflect the views of the publisher.

Acknowledgements. The authors thank an anonymous Reviewer and Patrick N. Raanes for their sharp and inspiring comments and suggestions which benefited the paper. AI was used to polish the English of a limited number of paragraphs of the manuscript, and for a thorough check of notation consistency. CEREA is a member of Institut Pierre-Simon Laplace (IPSL).

Financial support. This project – Learning efficient Data Assimilation from Artificial Intelligence (DAbyAI) – has been supported by NVIDIA and their Academic Grant Program through the grant of two RTX 6000 Ada GPUs that were intensively used in the numerical experiments of this work. This paper is also a contribution to the DRUIDS project, supported by France 2030 PEPR Maths-Vives, grant ANR-24-EXMA-0002. Tobias S. Finn acknowledges the support of the project SASIP (grant no. G-24-66154) funded by Schmidt Sciences – a philanthropic initiative that seeks to improve societal outcomes through the development of emerging science and technologies, and the support of France 2030 PEPR Maths-Vives, grant no. ANR-24-EXMA-0001, project Climaths/-GenClimEx. Sibo Cheng acknowledges the support of the French Agence Nationale de la Recherche (ANR) under reference ANR-22-CPJ2-0143-01.

Review statement. This paper was edited by Natale Alberto Carrassi and reviewed by Patrick N. Raanes and one anonymous referee.

References

Agarwal, S., Jabbari, S., Agarwal, C., Upadhyay, S., Wu, S., and Lakkaraju, H.: Towards the Unification and Robustness of Perturbation and Gradient Based Explanations, in: Proceedings of the 38th International Conference on Machine Learning, vol. 139, edited by Meila, M. and Zhang, T., Proceedings of Machine Learning Research, PMLR, 110–119, https://proceedings.mlr.press/v139/agarwal21c.html (last access: 7 August 2026), 2021.

Allen, A., Markou, S., Tebbutt, W., Requeima, J., Bruinsma, W. P., Andersson, T. R., Herzog, M., Lane, N. D., Chantry, M., Hosking, J. S., and Turner, R. E.: End-to-end data-driven weather prediction, Nature, https://doi.org/10.1038/s41586-025-08897-0, 2025.

Arnold, L.: Random Dynamical Systems, Springer Berlin, Heidelberg, https://doi.org/10.1007/978-3-662-12878-7, 1998.

Asch, M., Bocquet, M., and Nodet, M.: Data Assimilation: Methods, Algorithms, and Applications, Fundamentals of Algorithms, SIAM, Philadelphia, https://doi.org/10.1137/1.9781611974546, 2016.

Bannister, R. N.: A review of operational methods of variational and ensemble-variational data assimilation, Q. J. Roy. Meteor. Soc., 143, 607–633, https://doi.org/10.1002/qj.2982, 2017.

Barone, A., Carrassi, A., Savary, T., Demaeyer, J., and Vannitsem, S.: Structural origins and real-time predictors of intermittency, Chaos, 35, 103119, https://doi.org/10.1063/5.0287572, 2025.

Bocquet, M.: Ensemble Kalman filtering without the intrinsic need for inflation, Nonlin. Processes Geophys., 18, 735–750, https://doi.org/10.5194/npg-18-735-2011, 2011.

Bocquet, M.: Dan1D: One-Dimensional Data Assimilation Network [Computer software], Zenodo [code], https://doi.org/10.5281/zenodo.21427793, 2026 (code also available at: https://github.com/cerea-daml/Dan1D, last access: 7 August 2026).

Bocquet, M. and Carrassi, A.: Four-dimensional ensemble variational data assimilation and the unstable subspace, Tellus A, 69, 1304504, https://doi.org/10.1080/16000870.2017.1304504, 2017.

Bocquet, M. and Farchi, A.: On the consistency of the perturbation update of local ensemble square root Kalman filters, Tellus A, 71, 1–21, https://doi.org/10.1080/16000870.2019.1613142, 2019.

Bocquet, M. and Sakov, P.: Combining inflation-free and iterative ensemble Kalman filters for strongly nonlinear systems, Nonlin. Processes Geophys., 19, 383–399, https://doi.org/10.5194/npg-19-383-2012, 2012.

Bocquet, M. and Sakov, P.: Joint state and parameter estimation with an iterative ensemble Kalman smoother, Nonlin. Processes Geophys., 20, 803–818, https://doi.org/10.5194/npg-20-803-2013, 2013.

Bocquet, M. and Sakov, P.: An iterative ensemble Kalman smoother, Q. J. Roy. Meteor. Soc., 140, 1521–1535, https://doi.org/10.1002/qj.2236, 2014.

Bocquet, M., Raanes, P. N., and Hannart, A.: Expanding the validity of the ensemble Kalman filter without the intrinsic need for inflation, Nonlin. Processes Geophys., 22, 645–662, https://doi.org/10.5194/npg-22-645-2015, 2015.

Bocquet, M., Gurumoorthy, K. S., Apte, A., Carrassi, A., Grudzien, C., and Jones, C. K. R. T.: Degenerate Kalman filter error covariances and their convergence onto the unsta-

ble subspace, SIAM/ASA J. Uncertain. Quantif., 5, 304–333, https://doi.org/10.1137/16M1068712, 2017.

Bocquet, M., Farchi, A., Finn, T. S., Durand, C., Cheng, S., Chen, Y., Pasmans, I., and Carrassi, A.: Accurate deep learning-based filtering for chaotic dynamics by identifying instabilities without an ensemble, Chaos, 34, 091104, https://doi.org/10.1063/5.0230837, 2024.

Boucher, E., Alexe, M., Lean, P., Pinnington, E., Lang, S., Laloyaux, P., Zampieri, L., de Rosnay, P., Bormann, N., and McNally, A.: Learning Coupled Earth System Dynamics with GraphDOP, arXiv [preprint], https://doi.org/10.48550/arXiv.2510.20416, 2025.

Boudier, P., Fillion, A., Gratton, S., Gürol, S., and Zhang, S.: Data Assimilation Networks, J. Adv. Model. Earth Sy., 15, e2022MS003353, https://doi.org/10.1029/2022MS003353, 2023.

Buehner, M., McTaggart-Cowan, R., Beaulne, A., Charette, C., Garand, L., Heilliette, S., Lapalme, E., Laroche, S., Macpherson, S. R., Morneau, J., and Zadra, A.: Implementation of Deterministic Weather Forecasting Systems based on Ensemble-Variational Data Assimilation at Environment Canada. Part I: The Global System, Mon. Weather Rev., 143, 2532–2559, https://doi.org/10.1175/MWR-D-14-00354.1, 2015.

Carrassi, A., Ghil, M., Trevisan, A., and Uboldi, F.: Data assimilation as a nonlinear dynamical systems problem: Stability and convergence of the prediction-assimilation system, Chaos, 18, 023112, https://doi.org/10.1063/1.2909862, 2008.

Carrassi, A., Bocquet, M., Bertino, L., and Evensen, G.: Data Assimilation in the Geosciences: An overview on methods, issues, and perspectives, WIREs Clim. Change, 9, e535, https://doi.org/10.1002/wcc.535, 2018.

Carrassi, A., Bocquet, M., Demaeyer, J., Gruzien, C., Raanes, P. N., and Vannitsem, S.: Data assimilation for chaotic dynamics, in: Data Assimilation for Atmospheric, Oceanic and Hydrologic Applications, vol. IV, edited by: Seon, K. P. and Liang, X., Springer International Publishing, Cham, 1–42, https://doi.org/10.1007/978-3-030-77722-7_1, 2022.

Chekroun, M. D., Simonnet, E., and Ghil, M.: Stochastic climate dynamics: Random attractors and time-dependent invariant measures, Physica D, 240, 1685–1700, https://doi.org/10.1016/j.physd.2011.06.005, 2011.

Cheng, S., Quilodran-Casas, C., Ouala, S., Farchi, A., Liu, C., Tandeo, P., Fablet, R., Lucor, D., Iooss, B., Brajard, J., Xiao, D., Janjic, T., Ding, W., Guo, Y., Carrassi, A., Bocquet, M., and Arcucci, R.: Machine learning with data assimilation and uncertainty quantification for dynamical systems: a review, IEEE/CAA J. Autom. Sin., 10, 1361–1387, https://doi.org/10.1109/JAS.2023.123537, 2023.

Cintra, R. S. and de Campos Velho, H. F.: Data assimilation by artificial neural networks for an atmospheric general circulation model, Chap. 17, IntechOpen, 265–286, https://doi.org/10.5772/intechopen.70791, 2018.

Daley, R.: Atmospheric Data Analysis, Cambridge University Press, New York, ISBN 978-0-521-38215-1, 1991.

Desroziers, G., Camino, J.-T., and Berre, L.: 4DEnVar: link with 4D state formulation of variational assimilation and different possible implementations, Q. J. Roy. Meteor. Soc., 140, 2097–2110, https://doi.org/10.1002/qj.2325, 2014.

Fablet, R., Chapron, B., Drumetz, L., Mémin, E., Pannekoucke, O., and Rousseau, F.: Learning Variational Data Assimilation Models and Solvers, J. Adv. Model. Earth Sy., 13, e2021MS002572, https://doi.org/10.1029/2021MS002572, 2021.

Fillion, A., Bocquet, M., and Gratton, S.: Quasi-static ensemble variational data assimilation: a theoretical and numerical study with the iterative ensemble Kalman smoother, Nonlin. Processes Geophys., 25, 315–334, https://doi.org/10.5194/npg-25-315-2018, 2018.

Fillion, A., Bocquet, M., Gratton, S., Gürol, S., and Sakov, P.: An iterative ensemble Kalman smoother in presence of additive model error, SIAM/ASA J. Uncertain. Quantif., 8, 198–228, https://doi.org/10.1137/19M1244147, 2020.

Filoche, A., Brajard, J., Charantonis, A., and Béréziat, D.: Learning 4DVAR Inversion Directly from Observations, in: Computational Science – ICCS 2023, edited by: Mikyška, J., de Mulatier, C., Paszynski, M., Krzhizhanovskaya, V. V., Dongarra, J. J., and Sloot, P. M., Springer Nature Switzerland, Cham, 414–421, https://doi.org/10.1007/978-3-031-36027-5_32, 2023.

Flandoli, F. and Tonello, E.: An introduction to random dynamical systems for climate, https://pagine.dm.unipi.it/flandoli/Part1bis.pdf (last access: 7 August 2026), 2021.

Frerix, T., Kochkov, D., Smith, J., Cremers, D., Brenner, M., and Hoyer, S.: Variational Data Assimilation with a Learned Inverse Observation Operator, in: Proceedings of the 38th International Conference on Machine Learning, edited by Meila, M. and Zhang, T., vol. 139 of Proceedings of Machine Learning Research, PMLR, 3449–3458, https://proceedings.mlr.press/v139/frerix21a.html (last access: 7 August 2026), 2021.

Ghil, M. and Sciamarella, D.: Review article: Dynamical systems, algebraic topology and the climate sciences, Nonlin. Processes Geophys., 30, 399–434, https://doi.org/10.5194/npg-30-399-2023, 2023.

Härter, T. P. and de Campos Velho, H. F.: Data Assimilation Procedure by Recurrent Neural Network, Eng. Appl. Comp. Fluid, 6, 224–233, https://doi.org/10.1080/19942060.2012.11015417, 2012.

Jazwinski, A. H.: Stochastic Processes and Filtering Theory, Academic Press, New York, ISBN 978-0-12-381550-7, https://doi.org/10.1016/S0076-5392(09)60368-4, 1970.

Kalnay, E.: Atmospheric Modeling, Data Assimilation and Predictability, Cambridge University Press, Cambridge, ISBN 978-0-521-79629-3, https://doi.org/10.1017/CBO9780511802270, 2003.

Karimi, A. and Paul, M. R.: Extensive chaos in the Lorenz-96 model, Chaos, 20, 043105, https://doi.org/10.1063/1.3496397, 2010.

Keller, J. D. and Potthast, R.: AI-based data assimilation: Learning the functional of analysis estimation, arXiv [preprint], https://doi.org/10.48550/arXiv.2406.00390, 2024.

Kuramoto, Y. and Tsuzuki, T.: Persistent propagation of concentration waves in dissipative media far from thermal equilibrium, Prog. Theor. Phys., 55, 356–369, https://doi.org/10.1143/PTP.55.356, 1976.

Lafon, N., Fablet, R., and Naveau, P.: Uncertainty Quantification When Learning Dynamical Models and Solvers With Variational Methods, J. Adv. Model. Earth Sy., 15, e2022MS003446, https://doi.org/10.1029/2022MS003446, 2023.

Laloyaux, P., Alexe, M., Boucher, E., Lean, P., Pinnington, E., Lang, S., Necker, T., and McNally, A.: Using data assimilation tools to dissect GraphDOP, arXiv [preprint], https://doi.org/10.48550/arXiv.2510.27388, 2025.

Lean, P., Alexe, M., Boucher, E., Pinnington, E., Lang, S., Laloyaux, P., Bormann, N., and McNally, A.: Learning from nature: insights into GraphDOP's representations of the Earth System, arXiv [preprint], https://doi.org/10.48550/arXiv.2508.18018, 2025.

Liu, J. S.: Siegel's formula via Stein's identities, Stat. Probabil. Lett., 21, 247–251, https://doi.org/10.1016/0167-7152(94)90121-X, 1994.

Lorenz, E. N. and Emanuel, K. A.: Optimal sites for supplementary weather observations: simulation with a small model, J. Atmos. Sci., 55, 399–414, https://doi.org/10.1175/1520-0469(1998)055<0399:OSFSWO>2.0.CO;2, 1998.

Lu, F.: U-Net Kalman Filter (UNetKF): An Example of Machine Learning-Assisted Data Assimilation, J. Adv. Model. Earth Sy., 17, e2023MS003979, https://doi.org/10.1029/2023MS003979, 2025.

Maddy, E. S., Boukabara, S. A., and Iturbide-Sanchez, F.: Assessing the Feasibility of an NWP Satellite Data Assimilation System Entirely Based on AI Techniques, IEEE J. Sel. Top. Appl., 17, 9828–9845, https://doi.org/10.1109/JSTARS.2024.3397078, 2024.

McCabe, M. and Brown, J.: Learning to Assimilate in Chaotic Dynamical Systems, in: Advances in Neural Information Processing Systems, vol. 34, edited by: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., Curran Associates, Inc., 12237–12250, https://proceedings.neurips.cc/paper_files/paper/2021/file/65cc2c8205a05d7379fa3a6386f710e1-Paper.pdf (last access: 7 August 2026), 2021.

Misra, D.: Mish: A Self Regularized Non-Monotonic Neural Activation Function, arXiv [preprint], https://doi.org/10.48550/arXiv.1908.08681, 2019.

Oseledec, V. I.: A multiplicative ergodic theorem. Ljapunov characteristic numbers for dynamical systems, Trans. Moscow Math Soc., 19, 197–231, 1968.

Palatella, L., Carrassi, A., and Trevisan, A.: Lyapunov vectors and assimilation in the unstable subspace: theory and applications, J. Phys. A-Math. Theor., 46, 254020, https://doi.org/10.1088/1751-8113/46/25/254020, 2013.

Pannekoucke, O. and Fablet, R.: PDE-NetGen 1.0: from symbolic partial differential equation (PDE) representations of physical processes to trainable neural network representations, Geosci. Model Dev., 13, 3373–3382, https://doi.org/10.5194/gmd-13-3373-2020, 2020.

Pannekoucke, O., Ricci, S., Barthelemy, S., Ménard, R., and Thual, O.: Parametric Kalman filter for chemical transport model, Tellus A, 68, 31457, https://doi.org/10.3402/tellusa.v68.31547, 2016.

Pannekoucke, O., Bocquet, M., and Ménard, R.: Parametric covariance dynamics for the nonlinear diffusive Burgers equation, Nonlin. Processes Geophys., 25, 481–495, https://doi.org/10.5194/npg-25-481-2018, 2018.

Raanes, P. N., Stordal, A. S., and Evensen, G.: Revising the stochastic iterative ensemble smoother, Nonlin. Processes Geophys., 26, 325–338, https://doi.org/10.5194/npg-26-325-2019, 2019.

Ribeiro, M. T., Singh, S., and Guestrin, C.: Why Should I Trust You?: Explaining the Predictions of Any Classifier, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, Association for Computing Machinery, New York, NY, USA, 1135–1144, https://doi.org/10.1145/2939672.2939778, 2016.

Sacco, M. A., Pulido, M., Ruiz, J. J., and Tandeo, P.: On-line machine-learning forecast uncertainty estimation for sequential data assimilation, Q. J. Roy. Meteor. Soc., 150, 2937–2954, https://doi.org/10.1002/qj.4743, 2024.

Sakov, P.: On building the state error covariance from a state estimate, arXiv [preprint], https://doi.org/10.48550/arXiv.2411.14809, 2025.

Sakov, P., Oliver, D. S., and Bertino, L.: An iterative EnKF for strongly nonlinear systems, Mon. Weather Rev., 140, 1988–2004, https://doi.org/10.1175/MWR-D-11-00176.1, 2012.

Sivashinsky, G. I.: Nonlinear analysis of hydrodynamic instability in laminar flames-I. Derivation of basic equations, Acta Astronaut., 4, 1177–1206, https://doi.org/10.1016/0094-5765(77)90096-0, 1977.

Smilkov, D., Thorat, N., Kim, B., Viégas, F. B., and Wattenberg, M.: SmoothGrad: removing noise by adding noise, CoRR, arXiv [preprint], https://doi.org/10.48550/arXiv.1706.03825, 2017.

Stordal, A. S., Szklarz, S. P., and Leeuwenburgh, O.: A Theoretical look at Ensemble-Based Optimization in Reservoir Management, Math. Geosci., 48, 399–417, https://doi.org/10.1007/s11004-015-9598-6, 2016.

Tang, H. and Glass, J.: On Training Recurrent Networks with Truncated Backpropagation Through time in Speech Recognition, in: 2018 IEEE Spoken Language Technology Workshop (SLT), 48–55, https://doi.org/10.1109/SLT.2018.8639517, 2018.

van Kekem, D. L. and Sterk, A. E.: Travelling waves and their bifurcations in the Lorenz-96 model, Physica D, 367, 38–60, https://doi.org/10.1016/j.physd.2017.11.008, 2018.

Wishner, R. P., Tabaczynski, J. A., and Athans, M.: A Comparison of Three Non-Linear Filters, Automatica, 5, 487–496, https://doi.org/10.1016/0005-1098(69)90110-1, 1969.