Using Differentiable Physics for Self-Supervised Assimilation of Chaotic Dynamical Systems

Michael McCabe
University of Colorado Boulder
michael.mccabe@colorado.edu

Jed Brown
University of Colorado Boulder
jed@jedbrown.org

Abstract

We propose a deep learning based data assimilation framework which we call *Amortized Assimilation* for state estimation in high-dimensional chaotic dynamical systems. Amortized assimilators utilize differentiable simulation of physics-derived system dynamics to enable end-to-end physics-aware gradient based training of denoising neural networks which update a simulated system state based on noisy observations. These hybrid models are able to learn to assimilate complex input distributions while maintaining a computable test-time update step in an entirely self-supervised manner using only sequences of noisy observations without loss of accuracy over training with ground truth targets. Numerical experiments demonstrate that amortized assimilators compare favorably with widely used data assimilation methods across common benchmark tasks.

1 Introduction

.

Data assimilation methods are widely used in the geosciences to reconcile noisy observations with dynamical models derived from prior scientific knowledge in applications where the accuracy of simulation-driven forecasts depends heavily on the accuracy of initial condition estimates [1, 2]. Historically, these methods have fallen into two major categories: efficient Bayesian filters which treat assimilation as statistical inference or flexible variational methods which view it as nonlinear optimization [3, 4]. We present a framework for learning nonlinear, non-Gaussian assimilators which we call *Amortized Assimilation*. Our method inherits the objective flexibility of variational approaches but amortizes the optimization procedure by training a neural network to directly update the state estimate based on both observed and simulated data in a manner akin to Bayesian filters.

Amortized assimilators combine deep learning-based state estimation with differentiable numerical models derived from physical knowledge. We show that incorporating differentiable simulation into training enables a novel self-supervised training approach which uses only noisy observations without loss of accuracy over fully-supervised training using ground truth target information that is rarely available in practice. Numerical results across a set of experiments on chaotic system benchmarks show that assimilators learned in this way can greatly improve the accuracy of state estimates in the challenging low ensemble size and high noise regimes where existing methods struggle.

Related Work Deep learning for data assimilation is a growing area, though one limited by the challenges we address in this work. Some early approaches have used reanalysis data produced from traditional assimilation to attempt supervised training of assimilators [5, 6]. Other efforts exploring the use of deep learning in data assimilation have pursued an alternative hybridization strategy to our approach where neural networks are used to learn dynamics while standard methods are used for assimilation [7–9]. Outside of data assimilation, a significant body of work has explored deep learning parameterizations of Bayesian filters [10–14], and particle filters [15–18].

Workshop on Differentiable Vision, Graphics, and Physics in Machine Learning at NeurIPS 2020.

2 Amortized Assimilation

The EnKF and Ensemble-based Filtering Let $x_t \in \mathbb{R}^n$ be the true state vector at time t for a dynamical system that evolves in time according to $\dot{x}_t = g(x_t)$ where $g : \mathbb{R}^n \to \mathbb{R}^n$ is a deterministic Lipschitz time-invariant function which admits a unique solution to the initial value problem. At a discrete set of points in time $T = \{0, \tau, 2\tau, \dots\}$ we observe a sequence of measurements $Y_T = \{y_t \mid 0 \leq t \leq T\}$. Each y_t represents a partial observation of the true state generated by an unbiased observation operator H_t acting on the system state with the form $y_t = H_t(x_t) + \eta_t$ where η_t is a Gaussian random variable with diagonal covariance $\sigma_t^2 I$. X_T denotes the corresponding set of true state values at each observation time. $\Phi_g : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$ is the flow that maps initial conditions x_t forward in time under dynamics g.

In this work, we focus on the filtering problem where the goal is to estimate $p(x_t|Y_t)$. Filtering is a vital tool for estimating initial conditions in simulation-based forecasting. Filtering algorithms are typically split into forecast and analysis steps. The forecast step under deterministic dynamics reduces to computing the pushforward of $p(x_{t-1}|Y_{t-1})$ by dynamics g. In the analysis step, the forecast is refined by new observations to estimate $p(x_t|Y_t)$. We denote the analysis and forecast estimates of x_t by $\hat{x}_t^a$ and $\hat{x}_t^\Phi$ respectively. The same syntax is maintained for analysis and forecast estimates of other properties of interest.

One of the most widely used methods for assimilation in large scale chaotic systems is the Ensemble Kalman Filter (EnKF) [19]. The EnKF is a sequential Monte Carlo [20] extension of the classical Kalman filter [21] which permits nonlinear dynamics while maintaining Gaussian assumptions. Instead of attempting to exactly compute the evolution of a probability density function under nonlinear dynamics, ensemble methods maintain an empirical approximation to the target distribution in the form of a set of samples or particles $\hat{x}_{i,t}$ whose evolution is simulated numerically. The EnKF differs from particle filter methods in that new observations are assimilated by applying the closed form Kalman filter update to each ensemble member where the empirical covariance of the ensemble $\hat{P}_t^\Phi$ is used in place of an exact covariance P_t.

The amortized assimilators proposed in this work are based on a reinterpretation of the EnKF update equations that is described in the supplementary materials. Like the EnKF, we employ an ensemble approach to uncertainty quantification, but to relax the Gaussian assumptions of the EnKF, we learn a data-driven update $f_\theta(\hat{x}_{i,t}^\Phi, \hat{P}_t^\Phi, y_t)$ in the form of a neural network parameterized by weights θ which are optimized to minimize the self-supervised loss we introduce in the next section.

Self-Supervised Assimilation

Recent work in computer vision has demonstrated the effectiveness of self-supervised learning for image denoising [22, 23]. Batson and Royer [23] introduced the $\mathcal{J}$-invariance framework and showed that for a denoising task over noisy feature set $y \in \mathbb{R}^n$ which acts as an unbiased estimator of a true signal $x \in \mathbb{R}^n$ and can be partitioned into subsets with uncorrelated noise, the $\mathcal{J}$-invariant function f which minimizes the self-supervised squared error $||f(y) - y||^2$ is in fact an optimal denoiser under the supervised squared error $||f(y) - x||^2$. We extend this framework to the dynamical systems setting to develop a self-supervised training method for data assimilation.

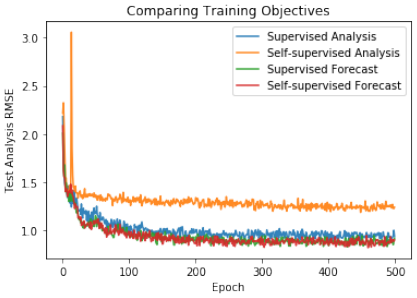

Figure 1: Comparing test-time supervised analysis loss on Lorenz 96 system across training objectives.

Consider the time-unrolled filtering setting. Recall that Y_T denotes the set of unbiased noisy realizations of X_T at all observation times up to and including time T. Sequential filters effectively partition Y_T into two subsets of features, $Y_{T-1} = \{y_i \mid 0 \leq i < T\}$ and $Y_{T-1}^C = \{y_T\}$. Any function which utilizes only Y_{T-1} to predict the values of Y_{T-1}^C is inherently $\mathcal{J}$-invariant. In fact, this is true for any arbitrary partition which acts analogously to a "donut" filter in image denoising, but the sequential nature of the problem limits the practical usefulness of other choices of partition.

Training a neural network to directly predict x_T from Y_{T-1} over large time steps is an enormously difficult task in chaotic systems, but for many problems in physics and engineering, robust physics-

derived models of system dynamics already exist. To take advantage of these numerical models, we want the assimilator to estimate x_{T-1} from Y_{T-1}. Unfortunately, x_{T-1} is rarely accessible in real-world settings. Additionally, a self-supervised objective comparing the analysis estimate $\hat{x}_t^a$ to y_{T-1} would be minimized by an identity mapping of the most recent observation and provide no denoising benefit. We address this issue by utilizing differentiable simulation to train with the following objective:

$$\mathcal{L}(\theta) = \frac{1}{T-1} \sum_{t=1}^{T-1} \left|\left| \frac{1}{m} \sum_{i=1}^{m} \left(H_{t+1}(f_\Phi(\hat{x}_{i,t}^\Phi, P_t^\Phi, y_t)) \right) - y_{t+1} \right|\right|^2 \tag{1}$$

where m is the ensemble size, $f_\Phi = \Phi_g(\cdot, \tau) \circ f_\theta$, and f_θ is our neural network parameterized by weights θ. The loss is computed by simulating the analysis ensemble members $\hat{x}_{i,t}^a$ resulting from the assimilation of a given observation y_t forward in time until the next available observation was recorded, then taking synthetic observations from the simulated ensemble members $\hat{y}_{i,t+1} = H_{t+1}(\hat{x}_{i,t+1}^\Phi)$ from which the mean synthetic observation is compared to the true observation.

When dynamics g admit a unique solution to the initial value problem, the flow $\Phi_g(\cdot, \tau)$ restricted to a fixed τ is injective and $\Phi_g(\hat{x}_t^a, \tau) = x_{t+1}$ if and only if $\hat{x}_t^a = x_t$. An f_θ which minimizes the self-supervised forecast objective therefore also minimizes the supervised squared error $||\hat{x}_t^a - x_t||^2$.

Figure 1 compares the test analysis loss for the partially observed Lorenz 96 system across training types and targets. The test error of the models trained with self-supervised and supervised forecast targets are indistinguishable while the model trained with a self-supervised analysis target is significantly less accurate.

Sequential Dropout Ensembles Ensemble-based uncertainty estimates are an elegant solution to the problem of computing the evolution of uncertainty under nonlinear dynamics, but without a probabilistic interpretation of the assimilator evaluating the analysis uncertainty is non-trivial. We address this by combining ensemble estimation with MC Dropout [24] which interprets the output from a dropout regularized neural network as samples from a variational approximation of $p(x_t|\hat{x}_t^\Phi, y_t)$. Defining $q(x_t|\hat{x}_t^\Phi, y_t, \theta)$ to be our variational density conditioned on model parameters θ. We would like to marginalize out the uncertainty over θ which is represented by variational distribution $q(\theta)$:

$$q(x_t|\hat{x}_t^\Phi, y_t) = \int p(\hat{x}_t|\hat{x}_t^\Phi, y_t, \theta) q(\theta)\, d\theta \tag{2}$$

In our case, $\hat{x}_t^\Phi$ is not a point estimate but rather a distribution over input states. We need to further marginalize out $\hat{x}_t^\Phi$ to get:

$$q(x_t|y_t) = \int \int p(\hat{x}_t|\hat{x}_t^\Phi, y_t, \theta) q(\theta) p(\hat{x}_t^\Phi)\, d\theta\, d\hat{x}_t^\Phi \tag{3}$$

This remains intractable; however, we have access to samples from the forecast distribution in the form of our ensemble members which reduces sampling from the analysis distribution into sequential sampling where we take MC Dropout samples using each of the forecast distribution samples as input.

Ensemble Amortized Filters The Ensemble Amortized Filter (EnAF) and Convolutional Ensemble Amortized Filter (ConvEnAF) are amortized assimilators which use the tools developed in the preceding sections to learn sequential filters from data. Amortized assimilators are trained as an unrolled compute graph where a set of potentially independent recurrent cells (corresponding to different observation types) are connected by the forward simulation of physics-based dynamics. Here, we use a non-standard memory-augmented recurrent cell mostly closely related to Recurrent Highway networks [25] using non-saturating memory [26]. Denoting cell memory by $c_{i,t}$, the update for our amortized filter block is as follows:

$$\begin{aligned}
\hat{x}_{i,t}^\Phi &= \Phi_g(\hat{x}_{i,t-1}^a, \tau) & \hat{P}_t^\Phi &= Cov(\hat{x}_{i,t}^\Phi) \\
z_{x_i}, z_{c_i} &= f_L(\hat{x}_{i,t}^\Phi, \hat{P}_t^\Phi, y_t, c_{i,t-1}) & \lambda_{x_i}, \lambda_{c_i} &= f_N(\hat{x}_{i,t}^\Phi, \hat{P}_t^\Phi, y_t, c_{i,t-1}) \\
\hat{x}_{i,t}^a &= \lambda_{x_i} \odot \hat{x}_{i,t}^\Phi + z_{x_i} & c_{i,t} &= \lambda_{c_i} \odot c_{i,t-1} + z_{c_i}
\end{aligned} \tag{4}$$

where f_L and f_N are arbitrary neural networks with dropout. f_L uses a linear output layer while f_N has an elementwise sigmoid activation on the output layer. The resulting update rules resemble a stable AR-1 process. We use only the diagonal of the full covariance matrix to improve scalability.

3 Experiments

We evaluate performance on three benchmark systems. Loss is reported as the time-averaged RMSE between the mean analysis state estimate and the true state. Each system is evaluated across a variety of ensemble sizes (denoted by m) and isotropic Gaussian observation noise levels (whose standard deviation is denoted by σ).

		m=5		*m = 10*		*m=15*		*m=20*	
σ	*Name*	*Full*	*Partial*	*Full*	*Partial*	*Full*	*Partial*	*Full*	*Partial*
1.0	LETKF	3.74	–	**0.29**	1.99	**0.28**	0.87	**0.28**	**0.69**
	4DVar	0.85	1.41	0.85	1.41	0.85	1.41	0.85	1.41
	iEnKF	–	–	3.55	–	2.55	2.96	0.27	0.89
	EnAF	.41	.88	0.40	0.85	0.41	0.83	0.39	0.81
	ConvEnAF	**.33**	**.73**	0.32	**0.71**	0.31	**0.70**	0.31	**0.68**
2.5	LETKF	3.52	3.81	1.05	2.23	**0.80**	2.07	0.81	1.91
	4DVar	1.65	3.13	1.65	3.13	1.65	3.13	1.65	3.13
	iEnKF	4.36	–	3.77	3.95	2.36	3.55	0.88	2.48
	EnAF	0.94	1.87	0.94	1.88	0.91	1.85	0.89	1.81
	ConvEnAF	**0.82**	**1.64**	**0.80**	**1.62**	**0.80**	**1.60**	**0.78**	**1.59**

Table 1: L96 RMSE.

We compare performance against a set of widely used filtering methods for data assimilation implemented in the Python DAPPER library [27]. These methods include 4DVar [28], the Local Ensemble Transform Kalman Filter (LETKF) [29], and the Iterative Ensemble Kalman Filter (iEnKF) [30]. Please see appendix B for full experiment settings.

Lorenz 96: Lorenz 96 [31] is a system of coupled differential equations with the following governing equations:

$$\dot{x}_i = (x_{i+1} - x_{i-2})x_{i-1} - x_i + F \tag{5}$$

The system is defined to have periodic boundary conditions $x_{K+1} = x_1$, $x_0 = x_K$, and $x_{-1} = x_{K-1}$ where K is the number of system dimensions. We set the number of dimensions to 40 and the forcing value to $F = 8$. Results are reported in Table 1. Partial observability is tested by observing only a rotating subset consisting of every fourth state variable.

Kuramoto-Sivashinsky: The KS equation [32] is a fourth-order partial differential equation known to exhibit chaotic behavior. In one spatial dimension, the governing equation can be written as:

$$u_t + u_x + u_{xxxx} + uu_x = 0 \tag{6}$$

with periodic boundary conditions. Results are reported in Table 2.

σ	*Name*	*m=5*	*m=10*	*m=15*	*m=20*
1.0	LETKF	1.411	**0.132**	**0.131**	**0.131**
	4DVar	0.444	0.444	0.444	0.444
	iEnKF	1.446	0.927	0.431	0.384
	ConvEnAF	**0.141**	0.138	0.136	0.136
2.5	LETKF	1.396	0.566	0.450	0.473
	4DVar	0.941	0.941	0.941	0.941
	iEnKF	1.573	1.185	0.859	0.615
	ConvEnAF	**0.370**	**0.365**	**0.363**	**0.362**

Table 2: KS RMSE.

Model Misspecification The two-level Lorenz system [31] introduces an additional coupling between observable variables x_t and a set of small scale unobservable variables v_t:

$$\begin{aligned} \dot{x}_i &= (x_{i+1} - x_{i-2})x_{i-1} - x_i + F - \frac{hc}{b} \sum_{j=J(i-1)+1} v_j \\ \dot{v}_j &= -cbv_{j+1}(v_{j+2} - v_j - 1) - cv_j + \frac{hc}{b} x_{\lfloor j-1/J \rfloor + 1} \end{aligned} \tag{7}$$

The forcing is set to $F = 8$ and the coupling parameters are set to $h = 1$, $c = 10$, $b = 10$ for consistency with the single level model. We include two levels of misspecification. First, we use the two-level Lorenz 96 model configured with $K = 40$ large scale variables and $J = 32$ small scale variables per large scale as the true model and the one-level system described in equation 5 as the forecast model. For the second, we train with an observation error $\sigma = 1$ but test with observation error $\sigma = 2.5$. Results are reported in Table 3.

σ	*Name*	*m=5*	*m=10*	*m=15*	*m=20*
1.0	LETKF	0.927	0.674	0.671	0.672
	4DVar	0.659	0.659	0.659	0.659
	iEnKF	2.551	1.290	1.180	1.082
	ConvEnAF	**0.530**	**0.514**	**0.509**	**0.507**
2.5	LETKF	1.677	1.601	1.627	1.637
	4DVar	1.430	1.430	1.430	1.430
	iEnKF	2.831	2.011	2.085	2.253
	ConvEnAF	**0.733**	**0.714**	**0.710**	**0.705**

Table 3: Model misspecification RMSE.

4 Conclusion

We presented a new framework for assimilation of chaotic dynamical systems that addresses several of the major challenges inhibiting the use of deep learning for data assimilation. Our hybrid deep learning-numerical simulation model enables the use of a self-supervised objective for learning in domains without ground truth data while addressing the need for effective uncertainty quantification.

References

[1] Stephen G. Penny and Thomas M. Hamill. Coupled data assimilation for integrated earth system analysis and prediction. *Bulletin of the American Meteorological Society*, 98(7): ES169–ES172, 2017. doi: 10.1175/BAMS-D-17-0036.1. URL https://doi.org/10.1175/BAMS-D-17-0036.1.

[2] Alberto Carrassi, Marc Bocquet, Laurent Bertino, and Geir Evensen. Data assimilation in the geosciences: An overview of methods, issues, and perspectives, 2018. ISSN 17577799.

[3] Mark Asch, Marc Bocquet, and Maëlle Nodet. *Data assimilation: methods, algorithms, and applications*, volume 11. SIAM, 2016.

[4] Kody Law, Andrew Stuart, and Konstantinos Zygalakis. *Data assimilation: A mathematical introduction*. 2015. ISBN 9783319203256. doi: 10.1007/978-3-319-20325-6.

[5] Fabrício P. Härter and Haroldo Fraga de Campos Velho. New approach to applying neural network in nonlinear dynamic model. *Applied Mathematical Modelling*, 32(12):2621–2633, dec 2008. ISSN 0307-904X. doi: 10.1016/J.APM.2007.09.006. URL https://www.sciencedirect.com/science/article/pii/S0307904X07002296{#}bib17.

[6] Rosângela Cintra, Haroldo Campos Velho, and Steven Cocke. Tracking the model: Data assimilation by artificial neural network. pages 403–410, 2016. doi: 10.1109/IJCNN.2016.7727227.

[7] Julien Brajard, Alberto Carassi, Marc Bocquet, and Laurent Bertino. Combining data assimilation and machine learning to emulate a dynamical model from sparse and noisy observations: a case study with the lorenz 96 model, 2020.

[8] Said Ouala, Ronan Fablet, Cédric Herzet, Bertrand Chapron, Ananda Pascual, Fabrice Collard, and Lucile Gaultier. Neural network based Kalman filters for the spatio-temporal interpolation of satellite-derived sea surface temperature. *Remote Sensing*, 2018. ISSN 20724292. doi: 10.3390/rs10121864.

[9] Marc Bocquet, Julien Brajard, Alberto Carrassi, and Laurent Bertino. Data assimilation as a learning tool to infer ordinary differential equation representations of dynamical models. *Nonlinear Processes in Geophysics*, 26(3):143–162, 2019.

[10] Tuomas Haarnoja, Anurag Ajay, Sergey Levine, and Pieter Abbeel. Backprop KF: Learning discriminative deterministic state estimators. In *Advances in Neural Information Processing Systems*, 2016.

[11] Huseyin Coskun, Felix Achilles, Robert Dipietro, Nassir Navab, and Federico Tombari. Long Short-Term Memory Kalman Filters: Recurrent Neural Estimators for Pose Regularization. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017. ISBN 9781538610329. doi: 10.1109/ICCV.2017.589.

[12] Maximilian Karl, Maximilian Soelch, Justin Bayer, and Patrick Van der Smagt. Deep variational bayes filters: Unsupervised learning of state space models from raw data. *arXiv preprint arXiv:1605.06432*, 2016.

[13] Rahul G. Krishnan, Uri Shalit, and David Sontag. Deep kalman filters, 2015.

[14] Syama Sundar Rangapuram, Matthias W Seeger, Jan Gasthaus, Lorenzo Stella, Yuyang Wang, and Tim Januschowski. Deep State Space Models for Time Series Forecasting. In S Bengio, H Wallach, H Larochelle, K Grauman, N Cesa-Bianchi, and R Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 7785–7794. Curran Associates, Inc., 2018. URL http://papers.nips.cc/paper/8004-deep-state-space-models-for-time-series-forecasting.pdf.

[15] Xiao Ma, Peter Karkus, David Hsu, and Wee Sun Lee. Particle filter recurrent neural networks, 2019.

[16] Shixiang Gu, Zoubin Ghahramani, and Richard E. Turner. Neural adaptive sequential Monte Carlo. In *Advances in Neural Information Processing Systems*, 2015.

[17] Tuan Anh Le, Maximilian Igl, Tom Rainforth, Tom Jin, and Frank Wood. Auto-encoding sequential Monte Carlo. In *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, 2018.

[18] Hongyuan Mei, Guanghui Qin, and Jason Eisner. Imputing missing events in continuous-time event streams. In *36th International Conference on Machine Learning, ICML 2019*, 2019. ISBN 9781510886988.

[19] G. Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *Journal of Geophysical Research*, 1994. ISSN 01480227. doi: 10.1029/94jc00572.

[20] Arnaud Doucet, Nando Freitas, and Neil Gordon. An Introduction to Sequential Monte Carlo Methods. In *Sequential Monte Carlo Methods in Practice*. 2001. doi: 10.1007/978-1-4757-3437-9_1.

[21] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering, Transactions of the ASME*, 1960. ISSN 1528901X. doi: 10.1115/1.3662552.

[22] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2Noise: Learning image restoration without clean data. In *35th International Conference on Machine Learning, ICML 2018*, 2018. ISBN 9781510867963.

[23] Joshua Batson and Loic Royer. Noise2Self: Blind Denoising by Self-Supervision, 2019.

[24] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *33rd International Conference on Machine Learning, ICML 2016*, 2016. ISBN 9781510829008.

[25] Julian Georg Zilly, Rupesh Kumar Srivastava, Jan Koutnik, and Jürgen Schmidhuber. Recurrent highway networks. In *34th International Conference on Machine Learning, ICML 2017*, 2017. ISBN 9781510855144.

[26] Sarath Chandar, Chinnadhurai Sankar, Eugene Vorontsov, Samira Ebrahimi Kahou, and Yoshua Bengio. Towards non-saturating recurrent units for modelling long-term dependencies, 2019.

[27] Patrick N Raanes and Others. nansencenter/DAPPER: Version 0.8, 2018. URL `https://doi.org/10.5281/zenodo.2029296`.

[28] F Rabier, H Järvinen, E Klinker, J.-F. Mahfouf, and A Simmons. The ECMWF operational implementation of four-dimensional variational assimilation. I: Experimental results with simplified physics. *Quarterly Journal of the Royal Meteorological Society*, 126(564):1143–1170, 2000. doi: 10.1002/qj.49712656415. URL `https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/qj.49712656415`.

[29] M. Bocquet. Ensemble Kalman filtering without the intrinsic need for inflation. *Nonlinear Processes in Geophysics*, 2011. ISSN 10235809. doi: 10.5194/npg-18-735-2011.

[30] Pavel Sakov, Dean S. Oliver, and Laurent Bertino. An iterative EnKF for strongly nonlinear systems. *Monthly Weather Review*, 2012. ISSN 00270644. doi: 10.1175/MWR-D-11-00176.1.

[31] Edward N. Lorenz. Predictability-a problem partly solved. In *Predictability of Weather and Climate*. 2006. ISBN 9780511617652. doi: 10.1017/CBO9780511617652.004.

[32] Yoshiki Kuramoto. Diffusion-Induced Chaos in Reaction Systems. *Progress of Theoretical Physics Supplement*, 1978. ISSN 0375-9687. doi: 10.1143/ptps.64.346.