Learning to Assimilate in Chaotic Dynamical Systems

Anonymous Author(s)
Affiliation
Address
email

Abstract

The accuracy of simulation-based forecasting in chaotic systems is heavily dependent on high-quality estimates of the system state at the beginning of the forecast. Data assimilation methods are used to infer these initial conditions by systematically combining noisy, incomplete observations and numerical models of system dynamics to produce highly effective estimation schemes. We introduce a self-supervised framework, which we call *amortized assimilation*, for learning to assimilate in dynamical systems. Amortized assimilation combines deep learning-based denoising with differentiable simulation, using independent neural networks to assimilate specific observation types while connecting the gradient flow between these sub-tasks with differentiable simulation and shared recurrent memory. This hybrid architecture admits a self-supervised training objective which is minimized by an unbiased estimator of the true system state even in the presence of only noisy training data. Numerical experiments across several chaotic benchmark systems highlight the improved effectiveness of our approach compared to widely-used data assimilation methods.

1 Introduction

Forecasting in the geosciences is an initial value problem of enormous practical significance. In high value domains like numerical weather prediction [1], climate modeling [2], atmospheric chemistry [3], seismology [4], and others [5–7], forecasts are produced by estimating the current state of the dynamical system of interest and integrating that state forward in time using numerical models based on the discretization of differential equations. These numerical models are derived from physical principles and possess desirable extrapolation and convergence properties. Yet despite the efficacy of these models and the vast amount of compute power used in generating these forecasts, obtaining highly accurate predictions is non-trivial. Discretization introduces numerical errors and it is often too computationally expensive to directly simulate the system of interest at the resolution necessary to capture all relevant features.

Further complicating matters in geoscience applications is that many systems of interest are chaotic, meaning that small errors in the initial condition estimates can lead to significant forecasting errors over relatively small time frame [8]. This can be problematic when the initial condition for a forecast is current state of the earth, a large-scale actively evolving system that cannot be directly controlled. Modern numerical weather prediction (NWP) models utilize hundreds of millions of state variables scattered over a three-dimensional discretization of the earth's atmosphere [9]. To perfectly initialize a simulation, one would need to know the true value of all of these state variables simultaneously. This data is simply not available. In reality, what is available are noisy, partial measurements scattered non-uniformly in time and space.

The inverse problem of estimating the true state of a dynamical system from imperfect observations is the target of our work and of the broader field known as data assimilation. Data assimilation techniques produce state estimates by systematically combining noisy observations with the numerical

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

model [10, 9]. The data assimilation problem differs from other state estimation problems in that the numerical model acts both as an evolution equation and as a mechanism for transporting information from regions where observations are dense to regions where observations are sparse [11]. Thus for accurate estimates of the full system state, it is important to utilize both the model and observations.

One recent trend in deep learning research has been the push to develop neural network models to replace expensive, oft repeated processes. These methods incur a large upfront cost to train the network in exchange for a faster solution to the problem. These amortized methods were initially developed for statistical inference [12] but have also been used in the context of numerical simulation [13], and meta-learning [14]. Our work extends this approach to variational data assimilation [15].

We introduce a self-supervised framework for learning to assimilate. Our design incorporates objective flexibility of variational assimilation but amortizes the expense of solving the nonlinear optimization problem into the training of a neural network that then acts as a sequential filter. We develop the self-supervised training procedure by drawing connections from the assumptions inherent to modern data assimilation techniques to the assumptions made in recent self-supervised image denoising methods [16] and showing that through the use of differentiable simulation, we can extend these image denoising approaches to the dynamical systems settings.

Our theoretical contributions indicate that by minimizing our self-supervised objective, we also minimizes an upper bound on the supervised (noise-free) objective of interest. This upper bound is exact in the interpolation regime when training loss reaches zero. Furthermore, our experimental results demonstrate that amortized assimilation methods match or outperform conventional approaches across several benchmark systems while requiring a much smaller number of forward simulations of the dynamics to obtain competitive results which could enable higher resolution simulations.

2 Preliminaries

Problem Setting In this work, we consider a dynamical system with state variable $\mathbf{x}(t) \in K \subset \mathbb{R}^d$ where K is a compact subset of $\mathbb{R}^d$ and system dynamics defined by the differential equation:

$$\frac{d\mathbf{x}}{dt} = g(\mathbf{x}(t)) \tag{1}$$

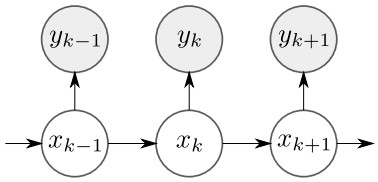


Figure 1: Observations are assumed to be generated from a Hidden Markov Model.

with g as a time-invariant, Lipschitz continuous map from $\mathbb{R}^d \to \mathbb{R}^d$. The state is observed at a sequence of K discrete time points $\{\tau_0, \tau_1, \ldots, \tau_K\}$ generating the time-series $\mathbf{y}_{0:K} = \{\mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_K\}$ where $0 \leq k \leq K$ is an index corresponding to the time point τ_k. These observations are imperfect representations of the system states $\mathbf{x}_{0:K} = \{\mathbf{x}_0, \mathbf{x}_1, \ldots \mathbf{x}_K\}$ generated from observation operators of the form:

$$\mathbf{y}_k = \mathcal{H}_k(\mathbf{x}_k) + \boldsymbol{\eta}_k \tag{2}$$

where $\mathcal{H}_k$ is an arbitrary potentially nonlinear function and $\boldsymbol{\eta}_k \sim \mathcal{N}(0, \sigma_k^2 \mathbf{I})$. This noise model is often referred to as a Hidden Markov Model (Figure 1). In the classical assimilation setting, the observation operators are assumed to be known *a priori*. For a problem like numerical weather prediction, these operators may represent measurements taken of certain atmospheric quantities at a particular measurement location.

Data assimilation is then the inverse problem of estimating the true trajectory $\mathbf{x}_{0:K}$ from our noisy observations $\mathbf{y}_{0:K}$ and our model of the system evolution:

$$\mathbf{x}(\tau_{k+1}) = \mathcal{M}_{k:k+1}(\mathbf{x}(\tau_k)) = \mathbf{x}(\tau_k) + \int_{\tau_k}^{\tau_{k+1}} g(\mathbf{x}(t))\, dt. \tag{3}$$

Sequential Filtering In the absence of ground truth, it is practical to present data assimilation from a statistical perspective. In particular, the quantity of interest that we are concerned with in this paper is the *filtering distribution* $p(\mathbf{x}_k \mid \mathbf{y}_{0:k})$. Efficient algorithms for estimating the filtering distribution

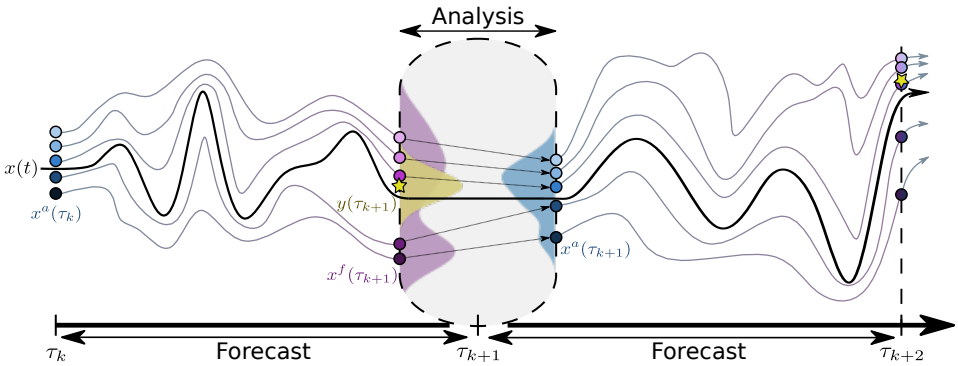


Figure 2: Ensemble filters model the time-evolution of uncertainty under nonlinear dynamics by maintaining a set of samples from the state distribution and simulating their trajectories forward in time. Observations are then used to refine these forecast estimates in what is called the analysis step.

are derived from exploiting dependence relationships in the generative model to produce the two step recursion:

$$\textbf{Forecast:} \qquad p(\mathbf{x}_k \mid \mathbf{y}_{0:k-1}) = \int_{\mathbb{R}^d} p(\mathbf{x}_k \mid \mathbf{x}_{k-1})\, d\mathbf{x}_{k-1} \tag{4a}$$

$$\textbf{Analysis:} \qquad p(\mathbf{x}_k \mid \mathbf{y}_{0:k}) \propto p(\mathbf{y}_k \mid \mathbf{x}_k) p(\mathbf{x}_k \mid \mathbf{y}_{0:k-1}) \tag{4b}$$

In subsequent sections, we will use $\mathbf{x}_k^f$ and $\mathbf{x}_k^a$ to refer to forecast and analysis estimates of $\mathbf{x}_k$ respectively.

In the case of linear dynamics with Gaussian uncertainties, the Kalman filter [17] provides both the forecast and analysis distributions exactly in closed form. However, the general case is significantly more complicated. Under nonlinear dynamics, computing the forecast distribution requires the solution of the Fokker-Planck partial differential equation [18]. While finding exact solution is computationally infeasible, one can produce what is called an ensemble estimate of the forecast distribution using Sequential Monte Carlo methods [19].

Ensemble filters (Figure 2) replace an exact representation of the uncertainty over states with an empirical approximation in the form of a small number of samples called particles or ensemble members. The forecast distribution can then be estimated by integrating the dynamics forward in time for each ensemble member independently. Allowing for full generality leads to particle filter approaches while enforcing Gaussian assumptions leads to the Ensemble Kalman filter (EnKF) [20]. The former is rarely used in data assimilation settings as particle filters can become degenerate in high dimensions [21] while the latter has been quite successful in practice. Like the classical Kalman filter, the EnKF assumes that both the state distribution and observation likelihoods are Gaussian. The latter is often reasonable, but the former is a rather strong assumption under nonlinear dynamics.

Ensemble methods possess an inherent trade-off in that larger ensembles provide better uncertainty estimates, but each ensemble member requires independently integrating the dynamics which can be significantly more expensive than the assimilation step itself. Bayesian filters rely on good covariance estimates to maintain stability. One of our goals is to develop an approach that maintains accuracy at smaller ensemble sizes.

Variational Assimilation The alternative to sequential filters is variational assimilation. Variational assimilation methods, like 4D-Var [22], directly solve for the initial conditions of a system by finding the state $\mathbf{x}_0$ which minimizes the negative log posterior of the posterior density using nonlinear optimization. This is expensive, but gradient-based optimization utilizing the adjoint of the numerical model relaxes the persistent Gaussian state assumptions necessary for tractable Bayesian filters. Furthermore, these methods can be augmented by auxilary objectives enforcing physical properties like conservation laws.

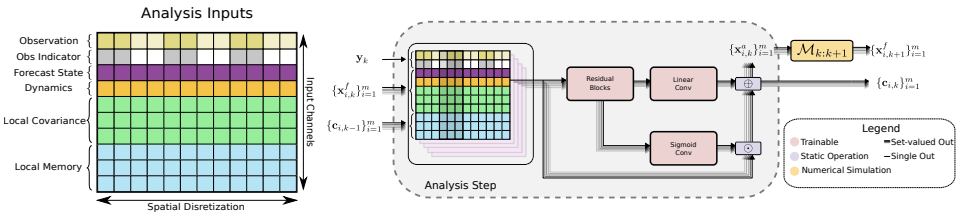


Figure 3: The AmEnF replaces the analysis step of a conventional ensemble filter with a neural network trained using historical data. Left: the model uses an augmented state which includes both standard inputs as well as recurrent memory. Right: the architecture used for a single analysis step.

3 Amortized Assimilation

3.1 Amortized Ensemble Filters

We present the more general amortized assimilation framework from the perspective of a specific architecture which we call the Amortized Ensemble Filter (AmEnF). The AmEnF (Figure 3) is an ensemble filter in that we maintain an ensemble of state estimates at all times which are used to estimate uncertainty over time. The forecast step is equivalent to that of the EnKF – the ensemble state estimates are integrated forward in time by the numerical model of system dynamics.

Our analysis step is derived from a similar perspective to Haarnoja et al. [23] in that we view sequential filtering as a particular recurrent prediction model which can be unrolled into a multi-step computation graph allowing us the train using backpropogation through time [24]. However, rather than just training observation operators as is done in [23], as we're operating in a domain where the Gaussian state assumptions of the EnKF are often violated, we attempt to fully replace the EnKF update equations with a learned function of the forecast value of the ensemble member $\mathbf{x}^f_{i,k}$, the ensemble covariance P^f_k, and the observation y_k. We further augment these features with the dynamics $g(\mathbf{x}^f_{i,k})$ and coordinate-wise recurrent memory $\mathbf{c}_{i,k-1}$ resulting in the following update equations:

$$
\begin{aligned}
\mathbf{x}^f_{i,k} &= \mathcal{M}_{k-1:k}(\mathbf{x}^a_{i,k-1}) & P^f_k &= Cov(\{\mathbf{x}^f_{i,k}\}^m_{k=1}) \\
\mathbf{z}_{x_i}, \mathbf{z}_{c_i} &= f_L(\mathbf{x}^f_{i,k}, P^f_k, \mathbf{y}_k, g(\mathbf{x}^f_{i,k}), \mathbf{c}_{i,k-1}) & \boldsymbol{\lambda}_{x_i}, \boldsymbol{\lambda}_{c_i} &= f_N(\mathbf{x}^f_{i,k}, P^f_k, \mathbf{y}_k, g(\mathbf{x}^f_{i,k}), \mathbf{c}_{i,k-1}) \quad (5) \\
\mathbf{x}^a_{i,k} &= \boldsymbol{\lambda}_{x_i} \odot \mathbf{x}^f_{i,k} + \mathbf{z}_{x_i} & \mathbf{c}_{i,k} &= \boldsymbol{\lambda}_{c_i} \odot \mathbf{c}_{i,k-1} + \mathbf{z}_{c_i}
\end{aligned}
$$

where f_L and f_N are neural networks with linear and sigmoid final activations respectively. We implement these as a single network which is split only at the final layer. We observed significant gains to training stability through the inclusion of this sigmoid-gated final layer. For scalability, we do not use the full covariance matrix as an input to the model. We include local covariance entries as additional feature channels.

The inclusion of recurrent memory may seem out of place when the dynamics are assumed to be Markovian. However, while the dynamics themselves are Markovian, the data assimilation process is not. The inclusion of memory allows the model to learn from the past without explicitly including previous states and observations. This is a significant benefit in practice. The other non-standard inclusion, the local dynamics, led to a small performance boost. As our method is not reliant on generative models of the inputs, additional features could be added trivially.

3.2 Self-Supervised Assimilation

Recent work in image denoising has demonstrated a promising alternative to explicitly minimizing a noise model in the form of self-supervised denoising. Lehtinen et al. [25] initially demonstrated the ability to denoise images without clean ground truth images using different samples of noise. Batson and Royer [16] expanded this idea to the more general $\mathcal{J}$-invariance framework which utilizes denoising functions where the function inputs and outputs have statistically independent noise to ensure that minimizing the squared error over noisy targets is equivalent in expectation to minimizing the squared error between the denoised outputs and the ground truth. Techniques derived from this

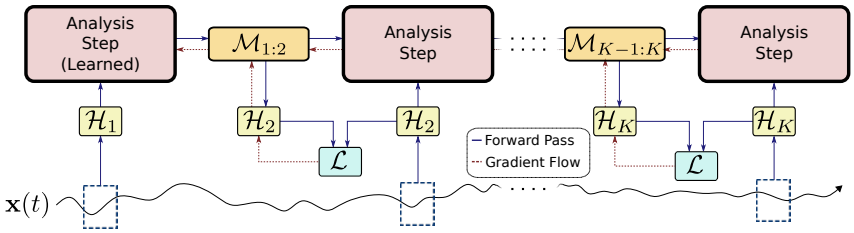


Figure 4: The self-supervised training process is unrolled across multiple filtering steps. Loss is evaluated between "pseudo-observations" and noisy observations obtained from the system.

approach have exhibited comparable performance to explicitly supervised denosiing across multiple applications [26–28]. We extend this approach to the dynamical systems setting through the use of differentiable simulation.

The core idea of amortized assimilation (Figure 4) is to exploit differentiable simulation and use the composition $\mathcal{M}_{k:k+1} \circ f_\theta$ as our denoising function during training rather than just f_θ. Under the generative model described in Section 2, the prediction inputs and targets are now assumed to have statistically independent noise, letting us use the framework from [16] to analyze the problem of learning to assimilate. While we only require simulating one step ahead to obtain this property, similar to prior work [29], we observe significantly improved performance by unrolling the procedure over multiple steps.

Apart from the previously specified generative model assumptions, our theoretical results rely on two additional assumptions:

Assumption 1 (Uniqueness of Initial Value Problem (IVP) Solution)**.** *The autonomous system evolution equation $x'(t) = g(x(t))$ is deterministic and L-Lipschitz continuous in x thus admitting a unique solution to the initial value problem under the Picard-Lindelöf theorem [30].*

Assumption 2 (Unbiased Observation Operators)**.** *For all observation operators $\mathcal{H}_k$, the observation y_k is an unbiased estimator of some subset of x_k, ie. if x_k^S is the restriction of x_k to some subset of features S then $\mathbb{E}[y_k \mid x_k^S] = x_k^S$.*

For the purposes of our analysis, we'll assume that S is the full state such that $\mathbb{E}[y_k \mid x_k] = x_k$ though this will not be the case in our numerical experiments. In data assimilation problems, the objective that we'd actually like to minimize is:

$$\mathcal{L}^a(\theta) = \frac{1}{K-1} \sum_{k=1}^{K-1} \left\| \left(\frac{1}{m} \sum_{i=1}^{m} f_\theta(\mathbf{x}_{i,k}^f, P_k^f, \mathbf{y}_k, g(\mathbf{x}_{i,k}^f), \mathbf{c}_{i,k-1}) \right) - \mathbf{x}_k \right\|^2 \tag{6}$$

as our goal is to produce the best initial condition estimates at the start of the forecast window. We call this objective the supervised analysis loss. Unfortunately, $\mathbf{x}_k$ is unknown. Instead, we aim to show that minimizing the self-supervised forecast loss:

$$\mathcal{L}^f(\theta) = \frac{1}{K-1} \sum_{k=1}^{K-1} \left\| \left(\frac{1}{m} \sum_{i=1}^{m} \mathcal{H}_{k+1}(\mathcal{M}_{k:k+1} \circ f_\theta(\mathbf{x}_{i,k}^f, P_k^f, \mathbf{y}_k, g(\mathbf{x}_{i,k}^f), \mathbf{c}_{i,k-1}) \right) - \mathbf{y}_{k+1} \right\|^2 \tag{7}$$

which uses readily available noisy observations as the training target is an effective and well-motivated way to learn to assimilate. It will be conceptually useful to further define the supervised forecast loss $\mathcal{L}^{sf}$ to be the mean squared error between the forecast state and the true state.

Proposition 1 (Interpolation Regime)**.** *Under the Hidden Markov assumptions of our generative model and given that assumptions 1 and 2 are satisfied, let f_θ denote a family of functions parameterized by θ for which $\min_\theta \ \mathcal{L}^f(\theta) = 0$. Any θ satisfying this criteria also satisfies $\mathcal{L}^a(\theta) = 0$.*

Proof: In this regime, the two losses exactly coincide on the training set. As observation noise is uncorrelated in time, Proposition 1 of [16] gives us that under Assumption 2, minimizing the self-supervised forecast loss $\mathcal{L}^f$ minimizes the supervised forecast loss $\mathcal{L}^{sf}$. Assumption 1 then implies that the resolvent map $\mathcal{M}_{k:k+1}$ is one-to-one such that the preimage $\mathcal{M}_{k:k+1}^{-1}[x_{k+1}]$ contains a single element for all values x_{k+1} in the range of $\mathcal{M}_{k:k+1}$. Thus if $\mathbb{E}_{y_k}[\mathcal{M}_{k:k+1} \circ f_\theta(\cdot)))] = \mathbf{x}_{k+1}$, then $\mathbb{E}_{y_k}[f_\theta(\cdot)] = \mathbf{x}_k$.

The interpolation regime is a common target of study due to the fact that many real-world neural networks are highly overparameterized [31]. While for the more general case, for sufficiently smooth dynamics, one might expect that over short time horizons, a function which minimizes the forecast loss should perform well for the analysis loss, this is not guaranteed and depends on the properties of the system under study. For our bound, we only consider the supervised components as, per [16], minimizing the self-supervised forecast loss minimizes the supervised forecast loss.

Proposition 2 (Non-zero Loss Setting)**.** *Given data generated from a HMM under assumptions 1 and 2, suppose the supervised analysis loss can be bounded by the supervised forecast loss as:*

$$\frac{1}{K-1}\sum_{k=1}^{K-1}\left\|\left(\frac{1}{m}\sum_{i=1}^{m} f_\theta(\cdot)\right)-\boldsymbol{x}_k\right\| \leq \frac{e^{Lt}}{K-1}\sum_{k=1}^{K-1}\max_{i\in[m]}\left\|f_\theta(\boldsymbol{x}^f_{i,k},\cdot)-\boldsymbol{x}_{k+1}\right\| \tag{8}$$

We include the full proof in the supplementary materials. Proposition 2 illustrates that even when we are not operating in the interpolation regime, minimizing the self-supervised forecast loss is equivalent to minimizing an upper bound on our supervised loss of interest motivating our self-supervised approach for learning to assimilate.

3.3 Implementation Details

Avoiding Ensemble Collapse Ensemble-based uncertainty estimates are an elegant solution to the problem of computing the evolution of uncertainty under nonlinear dynamics, but without assumptions on the noise model, evaluating the analysis uncertainty is non-trivial. A well-known problem in deep learning-based uncertainty quantification is feature collapse [32, 33], a phenomenon in which the hidden representations resulting from regions of the input space are pushed arbitrarily closely together by the neural network. In our ensemble-based uncertainty quantification scheme, this poses an issue as ensemble members can quickly converge over a small number of assimilation steps resulting in ensemble collapse.

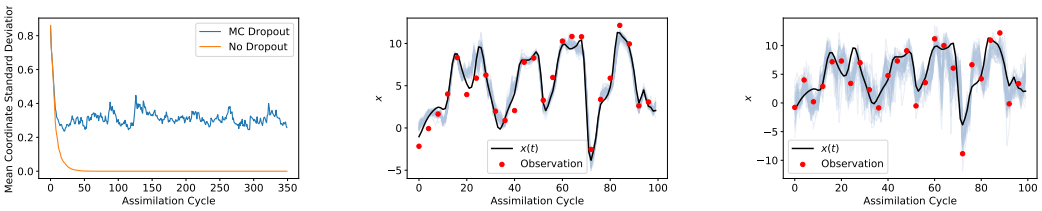
We address this by combining ensemble estimation with MC Dropout [34] which interprets a pass through a dropout-regularized neural network as a sample from a variational distribution over network weights. Thus, for standard usage of MC Dropout one can estimate uncertainty in the predictive distribution by using multiple dropout passes to integrate out weight uncertainty. In our case, we further have to account for uncertainty over the input distribution which is represented by our forecast ensemble. We employ a relatively simple Monte Carlo scheme in which the forecast ensemble members are assimilated by a single pass through a dropout-regularized network with independent dropout masks giving us m samples from the predictive distribution as our analysis ensemble. While this is a crude approximation, we found it resulted in more consistent performance with less tuning compared to the sensitivity constraints found in recent UQ methods [32, 35, 36].

Incomplete Observations One important concern in data assimilation is handling a variety of observation operators as the full state will rarely be observed at one time point. We discuss two processes for addressing this in amortized filters. The first, which is far more flexible but less effective at exploiting prior knowledge, is to create a single feedforward network corresponding to each observation type. This can be used with any observation type, even ones with unknown relationships to the system state. These unknown observation operators can be by evaluating the loss at points with future, known observation operators and passing the gradient signal back in time through the differentiable simulation connecting the time points.

The second, which we implement in the AmEnF, is to use our knowledge of the spatial distribution of the operation to assign the observed values corresponding to certain locations as channels in the input representation. Coordinates which are not observed can be masked and an additional channel is appending indicating whether a particular coordinate was observed during the given assimilation cycle. This is depicted by the shading in the observation channels of Figure 3. More details on the masking process can be found in the supplementary material.

4 Related Work

Work on the intersection of deep learning and data assimilation has exploded in recent years. Some early efforts explicitly used reanalysis data produced by traditional assimilation methods as a target



(a) Mean ensemble spread over assimilation cycles

(b) Ensemble trajectories with observation noise $\sigma = 1.0$

(c) Ensemble trajectories with observation noise $\sigma = 2.5$

Figure 5: Empirical behavior of the ensemble under MC dropout.

for supervised training [37, 38]. These approaches are limited as they cannot reasonably expect to outperform the method used to produce the reanalysis data in terms of accuracy. More recently, significant effort has gone into using traditional assimilation alongside learned surrogate models to stabilize training, learn augmented dynamics, or accelerate simulation with larger ensembles [39–42].

The work most comparable to our own is that which uses machine learning methods to learn a component of the assimilation mechanism directly [43, 44] in continuous nonlinear systems. Our method differs from Grönquist et al. [43] in that we use differentiable simulation to enable self-supervised learning. While Frerix et al. [44] incorporates differentiable simulation, their Learned Inverse Operator framework acts as a component of the 4D-Var algorithm and still requires solving an expensive optimization problem at each assimilation step rather than using using a closed form update. Outside of the field of data assimilation specifically, there has also been significant work into neural parameterization of Bayesian filters [23, 45–53].

5 Experiments

We examine the empirical performance of amortized assimilation from two perspectives. In the first, we derive empirical justification for our design choices, comparing the performance of the various objective functions and qualitatively examining the evolution of uncertainty in the ensemble.

The second objective is to evaluate the competitiveness of our amortized assimilation architecture, the AmEnF, against standard data assimilation methods across multiple dynamical systems commonly used to benchmark data assimilation methods. All methods tested have explicit knowledge of the system dynamics. Due to the challenge of learning chaotic dynamics, we found that methods which rely on learning the dynamics were uncompetitive. In this section, we demonstrate the improved performance of the AmEnF especially at low ensemble sizes and in high noise regimes.

For all experiments, we generate a training set consisting of 6000 sequences consisting of 40 assimilation steps each. The validation set consists of a single sequence of an additional 1000 steps and the test set is a further 10,000 steps. AmEnF models are developed in PyTorch [54] using the torchdiffeq [55] library for ODE integration. Models are trained on a single GTX 1070 GPU for 500 epochs using the Adam [56] optimizer with initial learning rate $8e-4$ with a warm-up over 50 iterations followed by halving the learning rate every 200 iterations.

5.1 AmEnF Evaluation

All experiments in this section are performed using the Lorenz 96 system [57]. Lorenz 96 is a system of coupled differential equations intended to emulate the evolution of a single atmospheric quantity across a latitude circle.

$$\dot{x}_i = (x_{i+1} - x_{i-2})x_{i-1} - x_i + F \tag{9}$$

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 $F = 8$. This is perhaps the most widely used test system in data assimilation and is what we will use the explore the behavior of the AmEnF model.

Ensemble Behavior Figure 5 provides justification for the use of MC Dropout. In the left figure, we see that the ensemble variance quickly collapses to zero using determinisitic networks while it

varies along a stable range using MC Dropout. More interesting are the central and right figures which compare the ensemble member trajectories (light blue) with the true trajectory (black) for a single spatial coordinate across several noise settings. We can see that the uncertainty patterns in our ensemble behave largely as one would hope. Less certainty in the system results in a larger ensemble spread.

The assimilation behavior is similarly interesting. We can see that while the ensemble trajectories occasionally approach outlier observations, when the ensemble spread is tight, the learned assimilator trusts the current forecast trajectory and the ensemble members do not move onto a worse path to meet the outlier.

Training Objective Figure 6 shows a comparison between the test-time supervised analysis loss across AmEnF models trained using several objective functions. The problem used for comparison is the fully observed Lorenz 96 system with observation noise $\sigma = 1$. As one might anticipate, the self-supervised analysis objective, the only objective available without differentiable simulation, results in the AmEnF learning an identity mapping from the observation so that the error is exactly the observation variance. The supervised analysis (which we are able to use since this is simulated data) and self-supervised forecast training approaches result in learning a legitimate denoising function. What is intriguing is that the self-supervised forecast loss actually outperforms the supervised analysis loss for this system. The optimization problem is nonconvex, so neither approach is guaranteed to reach a global minimizer, so this doesn't violate our theoretical results. It does, however, suggest that there may be some value to including differentiable simulation in the training pipeline beyond simply enabling the denoising property.

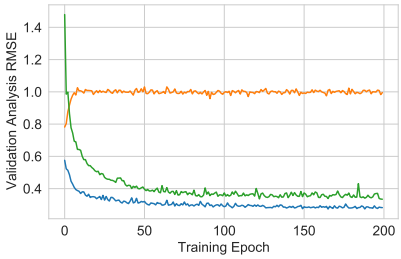


Figure 6: Test analysis RMSE across models trained using different approach.

5.2 Method Comparison

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 on the test 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). All results are reported in Figure . Error bars correspond to deviation across 10 repeated experiments with different noise samples.

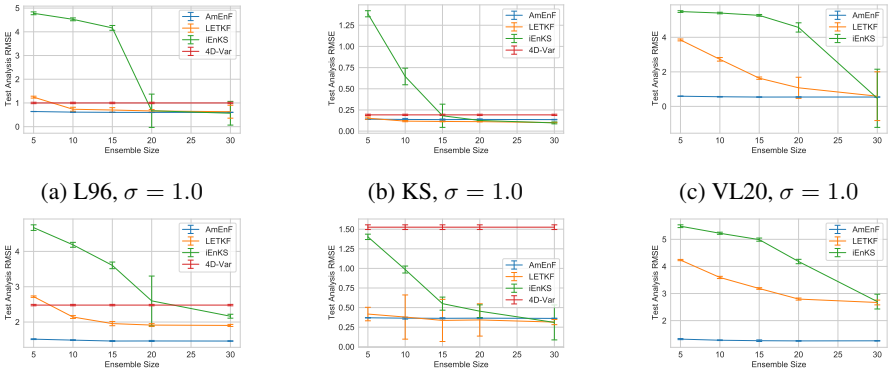
We compare performance against a set of widely used filtering methods for data assimilation implemented in the Python DAPPER library [58]. These methods include 4D-Var [22], the Local Ensemble Transform Kalman Filter (LETKF) [59], and the Iterative Ensemble Kalman Smoother (iEnKS) [60]. 4D-Var and the iEnKF are used in a filtering capacity, using only the latest observation rather than a long trajectory. Full experiment settings including hyperparameters searched are including in the supplementary materials.

Lorenz 96 For method comparisons, we examine the more challenging case of a partially observed system. At every assimilation step, we observe only a rotating subset consisting of every fourth spatial dimension so that only 25% of the system is observed at any time. These experiments use an RK-4 [61] integrator with step sizes of .05 and partial observations every two integration steps.

Kuramoto-Shivashinsky While our theoretical results focus on ordinary differential equations, the method can also be used to assimilation partial differential equations like the KS equation. The KS equation [62] is a fourth-order partial differential equation known to exhibit chaotic behavior. In one spatial dimension, the governing equation can be written as:

$$u_t + u_x + u_{xxxx} + uu_x = 0 \tag{10}$$

with periodic boundary conditions. These experiments were fully observed. The system was integrated with an ETD-RK4 method with step sizes of .5 with observations every two steps.



(a) L96, $\sigma = 1.0$ (b) KS, $\sigma = 1.0$ (c) VL20, $\sigma = 1.0$

(d) L96, $\sigma = 2.5$ (e) KS, $\sigma = 2.5$ (f) VL20, $\sigma = 2.5$

Figure 7: Results from numerical comparisons across multiple ensemble sizes and observation noises.

Vissio-Lucarini 20 The system described by Vissio and Lucarini [63] is a coupled system intended to represent a minimal model of the earth's atmosphere. It augments the Lorenz system by augmenting it with "temperature" variables allowing for more complicated behavior:

$$\begin{aligned} \frac{dX_j}{dt} &= X_{j-1}(X_{j+1} - X_{j-2}) - \alpha\theta_j - \gamma X_j + F \\ \frac{d\theta_j}{dt} &= X_{j+1}\theta_{j+2} - X_{j-1}\theta_{j-2} + \alpha X_j - \gamma\theta_j + G \end{aligned} \tag{11}$$

We set $F = 10$, $G = 10$, $\gamma = 1$, $\alpha = 1$, and use 36 spatial points. Experiments using this system are also partially observed with every fourth spatial coordinate being observed at one time. The system is integrated using RK-4 with steps of length .05 and partial observations every two steps. 4D-Var was excluded in this comparison as performance on prior experiments did not justify further examination.

Results As we can see in Figure 7, the AmEnF models match or outperform the competitive methods with more significant improvements coming at smaller ensemble sizes and higher noise levels which are associated with stronger nonlinear effects [60]. The only system in which the AmEnF does not notably outperform the comparison models is the KS equation. This is interesting as our theoretical results rely on assumptions about ODEs while the KS equation is a PDE. Furthermore, the bound we derived has a time component and the KS experiments occur at the largest time scales. Nonetheless, despite these potential theoretical snags, the AmEnF model near matches the the top performing models in accuracy while exhibiting stronger stability under repeated experiments.

6 Discussion and Conclusion

Limitations Despite the successful numerical results, the method is not without challenges. The largest consideration is related to scaling. First, while the use of differentiable simulation allowed our method to outperform competitive methods, the availability of adjoints in numerical forecasting is not certain. Forecasts in domains like NWP often use legacy code where adjoints need to be maintained by hand. We expect this to change over time, but this will limit the adoption of similar approaches in the near future. Assuming the presence of adjoints, training cost is also a concern as it requires the repeated simulation of the dynamical systems. However, numerical models tend to change slowly and applications like NWP are among the largest users of compute in the world [1] with assimilation cycles occurring multiple times per day, thus we expect the amortization to pay off over time.

Conclusion We theoretically and experimentally motivate the use of a self-supervised approach for learning to assimilate in chaotic dynamical systems. This approach uses a hybrid deep learning-numerical simulation architecture which outperforms or matches several standard data assimilation methods across several numerical experiments. The performance gains are especially profound at low ensemble sizes, which could enable higher resolution, more accurate simulation for assimilation problems leading to stronger predictions for high value problems like numerical weather prediction in the future.

References

[1] Peter Bauer, Alan Thorpe, and Gilbert Brunet. The quiet revolution of numerical weather prediction. *Nature*, 525(7567):47–55, 2015.

[2] David A Randall, Richard A Wood, Sandrine Bony, Robert Colman, Thierry Fichefet, John Fyfe, Vladimir Kattsov, Andrew Pitman, Jagadish Shukla, Jayaraman Srinivasan, et al. Climate models and their evaluation. In *Climate change 2007: The physical science basis. Contribution of Working Group I to the Fourth Assessment Report of the IPCC (FAR)*, pages 589–662. Cambridge University Press, 2007.

[3] M. Bocquet, H. Elbern, H. Eskes, M. Hirtl, R. Žabkar, G. R. Carmichael, J. Flemming, A. Inness, M. Pagowski, J. L. Pérez Camaño, P. E. Saide, R. San Jose, M. Sofiev, J. Vira, A. Baklanov, C. Carnevale, G. Grell, and C. Seigneur. Data assimilation in atmospheric chemistry models: current status and future prospects for coupled chemistry meteorology models. *Atmospheric Chemistry and Physics*, 15(10):5325–5358, 2015. doi: 10.5194/acp-15-5325-2015. URL `https://acp.copernicus.org/articles/15/5325/2015/`.

[4] Andreas Fichtner. *Full seismic waveform modelling and inversion*. Springer Science & Business Media, 2010.

[5] Didier Bresch, Thierry Colin, Emmanuel Grenier, Benjamin Ribba, and Olivier Saut. Computational modeling of solid tumor growth: the avascular stage. *SIAM Journal on Scientific Computing*, 32(4):2321–2344, 2010.

[6] Cristóbal Bertoglio, Philippe Moireau, and Jean-Frederic Gerbeau. Sequential parameter estimation for fluid–structure problems: application to hemodynamics. *International Journal for Numerical Methods in Biomedical Engineering*, 28(4):434–455, 2012.

[7] Heinz W Engl, Christoph Flamm, Philipp Kügler, James Lu, Stefan Müller, and Peter Schuster. Inverse problems in systems biology. *Inverse Problems*, 25(12):123014, 2009.

[8] Stephen Wiggins, Stephen Wiggins, and Martin Golubitsky. *Introduction to applied nonlinear dynamical systems and chaos*, volume 2. Springer, 1990.

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

[10] 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`.

[11] Eugenia Kalnay. *Atmospheric modeling, data assimilation and predictability*. Cambridge university press, 2003.

[12] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2014.

[13] Tianju Xue, Alex Beatson, Sigrid Adriaenssens, and Ryan Adams. Amortized finite element analysis for fast PDE-constrained optimization. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 10638–10647. PMLR, 13–18 Jul 2020. URL `http://proceedings.mlr.press/v119/xue20a.html`.

[14] Sachin Ravi and Alex Beatson. Amortized bayesian meta-learning. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=rkgpy3C5tX`.

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

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

[17] 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.

[18] A H Jazwinski. Stochastic processes and filtering theory. *IEEE Transactions on Automatic Control*, 1972.

[19] 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.

[20] 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.

[21] Chris Snyder, Thomas Bengtsson, Peter Bickel, and Jeff Anderson. Obstacles to high-dimensional particle filtering. *Monthly Weather Review*, 2008. ISSN 00270644. doi: 10.1175/2008MWR2529.1.

[22] 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`.

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

[24] P.J. Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990. doi: 10.1109/5.58337.

[25] 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.

[26] Samuli Laine, Tero Karras, Jaakko Lehtinen, and Timo Aila. High-quality self-supervised deep image denoising. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper/2019/file/2119b8d43eafcf353e07d7cb5554170b-Paper.pdf`.

[27] Shreyas Fadnavis, Joshua Batson, and Eleftherios Garyfallidis. Patch2self: Denoising diffusion mri with self-supervised learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 16293–16303. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/file/bc047286b224b7bfa73d4cb02de1238d-Paper.pdf`.

[28] Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. Noise2void - learning denoising from single noisy images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[29] Kiwon Um, Robert Brand, Yun (Raymond) Fei, Philipp Holl, and Nils Thuerey. Solver-in-the-loop: Learning from differentiable physics to interact with iterative pde-solvers. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6111–6122. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/file/43e4e6a6f341e00671e123714de019a8-Paper.pdf`.

[30] Earl A Coddington and Norman Levinson. *Theory of ordinary differential equations*. Tata McGraw-Hill Education, 1955.

[31] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019. ISSN 0027-8424. doi: 10.1073/pnas.1903070116. URL `https://www.pnas.org/content/116/32/15849`.

[32] Joost van Amersfoort, Lewis Smith, Andrew Jesson, Oscar Key, and Yarin Gal. Improving deterministic uncertainty estimation in deep learning for classification and regression, 2021.

[33] Vardan Papyan, X. Y. Han, and David L. Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663, 2020. ISSN 0027-8424. doi: 10.1073/pnas.2015509117. URL https://www.pnas.org/content/117/40/24652.

[34] 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.

[35] Joost van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. Uncertainty estimation using a single deep deterministic neural network, 2020.

[36] Mihaela Rosca, Theophane Weber, Arthur Gretton, and Shakir Mohamed. A case for new neural network smoothness constraints, 2020.

[37] 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.

[38] 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.

[39] 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.

[40] 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.

[41] 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.

[42] Ashesh Chattopadhyay, Mustafa Mustafa, Pedram Hassanzadeh, Eviatar Bach, and Karthik Kashinath. Towards physically consistent data-driven weather forecasting: Integrating data assimilation with equivariance-preserving deep spatial transformers, 2021.

[43] Peter Grönquist, Chengyuan Yao, Tal Ben-Nun, Nikoli Dryden, Peter Dueben, Shigang Li, and Torsten Hoefler. Deep learning for post-processing ensemble weather forecasts. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 379 (2194):20200092, Feb 2021. ISSN 1471-2962. doi: 10.1098/rsta.2020.0092. URL http://dx.doi.org/10.1098/rsta.2020.0092.

[44] Thomas Frerix, Dmitrii Kochkov, Jamie A. Smith, Daniel Cremers, Michael P. Brenner, and Stephan Hoyer. Variational data assimilation with a learned inverse observation operator, 2021.

[45] 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.

[46] 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.

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

[48] 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`.

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

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

[51] 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.

[52] 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.

[53] Philipp Becker, Harit Pandya, Gregor Gebhardt, Cheng Zhao, James Taylor, and Gerhard Neumann. Recurrent kalman networks: Factorized inference in high-dimensional deep feature spaces, 2019.

[54] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary Devito Facebook, A I Research, Zeming Lin, Alban Desmaison, Luca Antiga, Orobix Srl, and Adam Lerer. Automatic differentiation in PyTorch. In *Advances in Neural Information Processing Systems 32*, 2019.

[55] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. *Advances in Neural Information Processing Systems*, 2018.

[56] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, 2014.

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

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

[59] 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.

[60] 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.

[61] C. Runge. Ueber die numerische Auflösung von Differentialgleichungen. *Mathematische Annalen*, 1895. ISSN 00255831. doi: 10.1007/BF01446807.

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

[63] Gabriele Vissio and Valerio Lucarini. Mechanics and thermodynamics of a new minimal model of the atmosphere. *The European Physical Journal Plus*, 135(10), Oct 2020. ISSN 2190-5444. doi: 10.1140/epjp/s13360-020-00814-w. URL `http://dx.doi.org/10.1140/epjp/s13360-020-00814-w`.

References

References follow the acknowledgments. Use unnumbered first-level heading for the references. Any choice of citation style is acceptable as long as you are consistent. It is permissible to reduce the font size to `small` (9 point) when listing the references. Note that the Reference section does not count towards the page limit.

[1] Alexander, J.A. & Mozer, M.C. (1995) Template-based algorithms for connectionist rule extraction. In G. Tesauro, D.S. Touretzky and T.K. Leen (eds.), *Advances in Neural Information Processing Systems 7*, pp. 609–616. Cambridge, MA: MIT Press.

[2] Bower, J.M. & Beeman, D. (1995) *The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural SImulation System.* New York: TELOS/Springer–Verlag.

[3] Hasselmo, M.E., Schnell, E. & Barkai, E. (1995) Dynamics of learning and recall at excitatory recurrent synapses and cholinergic modulation in rat hippocampal region CA3. *Journal of Neuroscience* **15**(7):5249-5262.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] **See Discussion**
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] **The method is designed with numerical weather forecasting in mind.**
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] **They're listed immediately before the theoretical results are discussed.**
   (b) Did you include complete proofs of all theoretical results? [Yes] **Though in the supplementary materials.**
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] **Yes, all code is included in the supplementary materials. If accepted, the link to the GitHub will be added to the paper. Or if not accepted, the arxiv version will have the link**
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] **though moved to the supplement for space reasons**
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] **Yes, all comparisons to other methods include error bars**
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] **The hardware used was listed.**
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] Y**es. All existing libraries are cited under experiments**
   (b) Did you mention the license of the assets? [No]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No] **unless the execution code counts**
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] **All data is simulated by the authors**
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] **All data is simulated from governing equations**
5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]