LEARNING FLOW-DEPENDENT CHARACTERISTICS WITH NEURAL OPERATORS FOR DATA ASSIMILATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Data assimilation is a key component of numerical weather forecasting systems, which is aimed at reconstructing earth systems' states from observation data using consistency constraints on physical properties. One of the major challenges in data assimilation is to properly exploit the flow dependencies to reconstruct physical states with higher accuracies. The traditional variational methods introduce flow-dependent characteristics by optimizing a target function in which flow dependencies are quantified, but at a relatively high computational cost. In this paper, we develop a deep learning framework with three different operators for data assimilation. The key novelty of our proposed approach is that we train a so-called flow operator through unsupervised learning to explicitly learn flow-dependent characteristics for reconstructed states. Our numerical experiments on Lorenz-63 and Lorenz-96 systems show that the proposed method may achieve higher evaluation efficiency than state-of-the-art neural-networks while maintaining, if not improving, accuracies. It's also demonstrated that our method is robust to biases in the prior dynamic model. We believe that our findings can inspire new possible solutions for data assimilation systems that incorporate model dynamics into reconstructed states.

1 INTRODUCTION

In earth science, data assimilation can be defined as a technique to combine the prior dynamics with a sequence of noisy and irregularly-sampled observations. Data assimilation is an important aspect in numerical weather forecasting systems. For instance, European Centre for Medium-Range Weather Forecasts (ECMWF) employs the 4D-Var assimilation algorithm in its operational systems to take full advantage of both in situ and satellite-derived data, as well as state-of-the-art numerical models (Rabier et al., 2000).

Classical data assimilation methods, from early empirical analysis (Bergthorsson et al., 1955) and optimal interpolation (Gandin, 1963) to later variational-based assimilation algorithms (Sasaki, 1970; Lewis & Derber, 1985) and the filtering methods based on statistical estimation theory, such as Kalman filter (Welch et al., 1995), ensemble Kalman filter (Evensen, 2003), and particle filter (Carpenter et al., 1999), have been instrumental in improving the capability of major numerical weather prediction centers worldwide (Kalnay, 2003). However, as the number of observations increases and the resolution of the models continues to enhance, it poses a great challenge for traditional assimilation algorithms to realize both high-performance and low-cost assimilation, as well as effectively utilizing meteorological data.

Data assimilation issue can be essentially classified as a regression problem, thus neural network techniques may be a beneficial weapon to deal with the challenges facing it. Hsieh & Tang (1998) are the first to suggest that neural networks could be used for data assimilation. After that, research work focuses on building different neural networks, such as multilayer perceptron (Vijaykumar et al., 2002; de Campos Velho et al., 2002; Härter & de Campos Velho, 2008; Cintra & de Campos Velho, 2018) and recurrent neural network (Härter & de Campos Velho, 2008), to do three-dimensional variational assimilation on different models, such as Lorenz systems (Vijaykumar et al., 2002; de Campos Velho et al., 2002), DYNAMO Härter & de Campos Velho (2008) and SPEEDY (Cintra & de Campos Velho, 2018). Thanks to the development of AI hardware and software, since 2018, much more novel work with bold attempts has emerged in the research area of AI for data assimila-

tion. Among them, advanced neural network techniques, such as auto-differentiation (Fablet et al., 2021b;a; Hatfield et al., 2021) and more complicated network structures (Mack et al., 2020) are applied to substitute part of the traditional three-dimensional or four-dimensional variational assimilation schemes, such as initial value determination (Frerix et al., 2021), target function optimizer (Fablet et al., 2021b; Hatfield et al., 2021) and dynamical model surrogate (Brajard et al., 2020; Arcucci et al., 2021; Farchi et al., 2021; Bonavita & Laloyaux, 2020). Despite these efforts, the development of AI for assimilation is still in the early stage, and traditional methods remain unaltered for the integration of dynamic features and observations.

Here, we develop a new approach of combing data-driven strategies with model-driven methods to deal with the data assimilation issue. Unlike previous work in which a target function involving a dynamical prior and an observation term is explicitly optimized, we don't directly rely on a variational data assimilation formulation. Instead, we put forth a deep learning framework with three operators for assimilation issue, among which the so-called observation operator extracts features from observations to generate coarse reconstructions, the so-called flow operator explicitly corrects the dynamics of given states to a certain degree via unsupervised learning, while the discriminator introduces high-frequency disturbances to the badly-reconstructed regions. The alternating operations of discriminator and flow operator form an essential part of the data assimilation architecture. This resulting end-to-end architecture, blending dynamical prior, maps a sequence of observations to its corresponding reconstructed state sequence. Experiment results prove that our architecture is not only efficient and accurate, but also robust to deviations of the prior dynamic model.

2 Problem Statement

We refer to the previous state-of-the-art work to introduce the strict definition and assumptions under considered problem settings (Fablet et al., 2021b). Denote x for the time-dependent state in some space $\mathcal{X}$ satisfying $x(t) \in \mathcal{X}$, $\mathcal{M}_{gt}$ for the ground truth dynamic model, then the dynamic system can be formulated as

$$\frac{\partial x(t)}{\partial t} = \mathcal{M}_{gt}\left(x(t)\right) \tag{1}$$

In real scenarios, the dynamic model of earth systems is unknown and meteorologists develop physical models to estimate it, based on laws of kinetics, thermodynamics and etc. Denote $\mathcal{M}_{pr}$ for the prior model. In the ideal case, the bias of prior model should be bounded, that is to say, there exits a constant C, such that,

$$\|\mathcal{M}_{pr}(x) - \mathcal{M}_{gt}(x)\| \leq C \tag{2}$$

Observations $\{y(t_i)\}$ are assumed to be discrete and partial, with white Gaussian noise. Suppose all the observations are obtained at time $t_i = t_0 + i\Delta t$, $i = 0, 1, ..., N$ and at time t_i, only components in domain Ω_{t_i} are observed. The formulation can be written as

$$y(t, p) = x(t, p) + \epsilon(t), \forall t \in \{t_0, t_0 + \Delta t, ..., t_0 + N\Delta t\}, \forall p \in \Omega_t \tag{3}$$

where $\epsilon(t)$ represents the white Gaussian noise.

The problem setting can be further explained within a deep learning paradigm. Our main goal is to construct an inverse map, implemented in the form of a neural network, from a time sequence of noisy and partial observations y to its corresponding reconstructed states x. The a priori knowledge is that the reconstructed states approximately satisfy certain dynamic equations with a bounded error. The major challenge is that it is difficult to incorporate the prior knowledge into neural network structures. Most of the previous work relies on the minimization of 4D-Var target function to tackle it. In our work, a pure neural network solution is given.

3 Learning Framework with Three Operators

To deal with the problem, we propose an end-to-end learning framework with three operators, including observation operator, flow operator and discriminator, as shown in Fig 1.

3.1 Three Operators

Observation Operator The observation operator is aimed at learning lower-dimensional observations and converting them into physical states (Frerix et al., 2021). We design the structure of

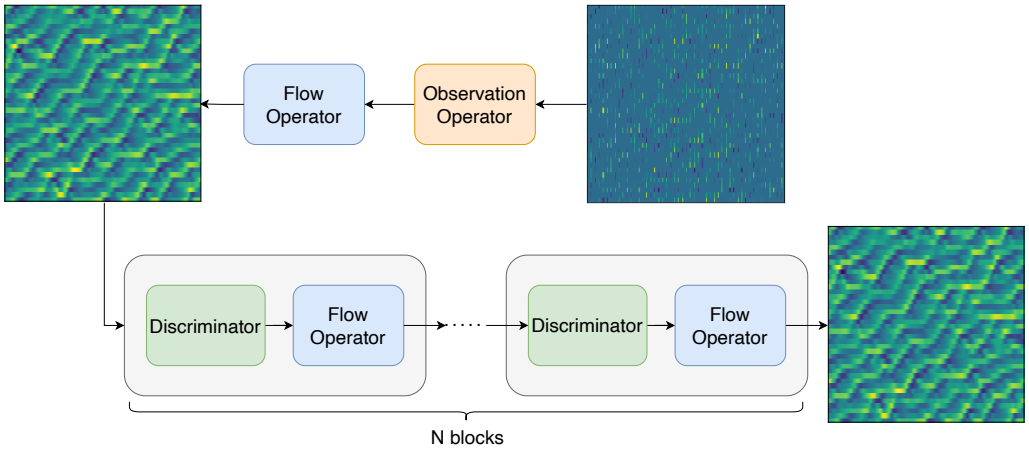

Figure 1: Proposed end-to-end learning framework with three operators. The observation operator (orange box) is aimed at converting sparse observations to complete physical states, as illustrated in the first row, while the second row demonstrates the process of refining reconstructed physical states with the help of discriminators and flow operators.

observation operator on the basis of Residual U-Net structure (Ronneberger et al., 2015; Siddique et al., 2021). The hierarchical structure of U-Net allows it to capture both the large-scale and small-scale information of observation states and is ubiquitously used in the reconstruction domain (Guan et al., 2019; Siddique et al., 2021; Punn & Agarwal, 2022; Feng et al., 2020). Moreover, the residual structure helps alleviate the vanishing gradient problem (He et al., 2016), thus allowing for U-net models with deeper layers to be designed.

In addition, this paper also modifies the network according to the specificity of the assimilation issue, as illustrated below. Unlike images, of which the three dimensions are length, height and channels, the inputs of assimilation issue are sequences of observations (states), whose dimensions are time, space and channels. Length and height in images are symmetric, while time and space are generally not. To tackle with this difference, the dimensionality reductions in U-Net for assimilation are carried out only in the temporal dimension. What's more, our proposed observation operator adds a skip layer from the input to the bottom hidden variables. It is implemented by extracting valid observations in the observed moments from the input and concatenating them directly with the bottom hidden variables. The advantage of this operation is that the information of sparse observations can be more effectively transferred through the network. The number of layers for U-Net can be set to three (Figure 2), four or even larger, depending on the sparsity of observations.

Discriminator and Flow Operator The discriminator and flow operator always work in pairs. The discriminator uses as input the roughly reconstructed states and outputs the diagnosed states. Badly reconstructed regions of the input are recognized and high frequency interference is added by the discriminator. Usually, the disturbances are biased towards the true value. After that, the flow operator inputs the diagnosed states and smooths the high frequency disturbances to generate the refined reconstructed states with higher accuracies.

Both the discriminator and flow operator adopt the U-Net architecture. The only difference between the structure of the discriminator and that of the flow operator is that for the latter we add an residual structure from the input to the output. This subtle modification is to make the unsupervised learning process of flow operator converge faster.

3.2 Network Training

The training process of the proposed framework can be divided into two parts: pretraining and finetuning. During pretraining, network parameters of three operators in the framework are roughly determined, while the purpose of finetuning is to further adjust and modify the parameters to fit the recurrent structure.

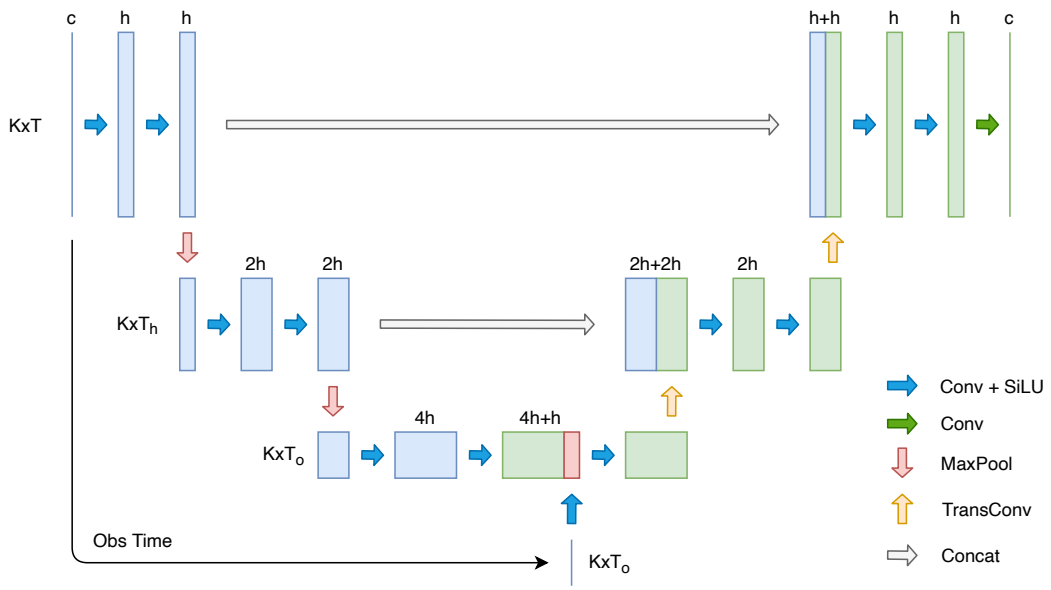

Figure 2: A example of the 3-layer U-Net architecture of the observation operator. We denote K for the spatial variables of physical states, T for the total sequence length, T_o for the sequence length with valid observations, T_h for the intermediate hidden sequence length. We require $T > T_h > T_o$.

Pretraining Observation Operator The procedure of pretraining is demonstrated in Figure 3. The observation operator is firstly trained to learn a map from observational space to physical states, in which a reconstruction loss $\mathcal{L}_{rec}$ is applied. If we denote the network by Ψ, the output by $\hat{x} = \Psi(y)$ then the loss function is

$$\mathcal{L}_{rec}(\hat{x}) = \|\hat{x} - x\|^2 \tag{4}$$

The number of epochs should be determined in advance to avoid overfitting.

Pretraining Flow Operator After that, a flow operator is added. We fix the parameters of the observation operator and train only the flow operator. Here, we define a so-called dynamic loss $\mathcal{L}_{dyn}$ for the loss function of the flow operator.

$$\mathcal{L}_{dyn}(\hat{x}) = \|\Phi(\hat{x}) - \hat{x}\|^2 \tag{5}$$

where Φ, defined by

$$\Phi(x)(t) = x(t - \Delta t) + \int_{t-\Delta t}^{t} \mathcal{M}(x(u))du \tag{6}$$

is a flow-dependent operator, which measures how well the given sequence of states fits the dynamic system. For example, if the sequence of states x perfectly matches the dynamic system, then $\Phi(x) = x$ holds and the dynamic loss is zero. We emphasize that the dynamic loss is unsupervised, hence generally it does not lead to overfitting as long as the difference between validation loss and training loss remains bounded during training. Whereas if the prior dynamic model is inaccurate, we stipulate a stopping criterion for the training procedure:

$$\mathcal{L}_{dyn-pr}(\hat{x}) \leq E_{x_{gt}}\left[\mathcal{L}_{dyn-pr}(x_{gt})\right] \tag{7}$$

where $E_{x_{gt}}\left[\mathcal{L}_{dyn-pr}(x_{gt})\right]$ measures the difference between the prior dynamics and ground truth. Since the error of prior dynamics is bounded, $E_{x_{gt}}\left[\mathcal{L}_{dyn-pr}(x_{gt})\right]$ is also a bounded value smaller than infinity. As long as inequality 7 is not satisfied, we end the current training phase and move on to the next one. Inequality 7 guarantees that the dynamic loss of the reconstructed states should be no smaller than that of the true states. It is a *necessary but not sufficient* condition to avoid overfitting in terms of dynamics and any reconstructed states with smaller dynamic loss than true states indicate that more information than expected has been learned from the prior dynamics.

Pretraining Discriminator In the last step, both a new discriminator and a flow operator whose parameters share those of the previous one are added. The parameters of the flow operator are fixed and we train the observation operator and discriminator together with the reconstruction loss.

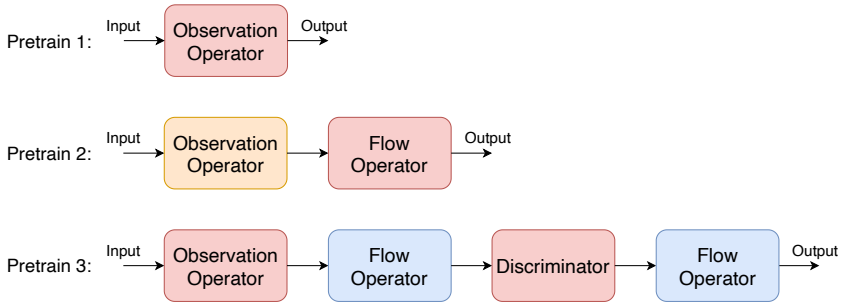

Figure 3: Pretraining procedure for the framework. There are totally three steps in this procedure. In each step, only the parameters of the network modules that are represented by red boxes are trained, while the parameters of those represented by other colors are fixed.

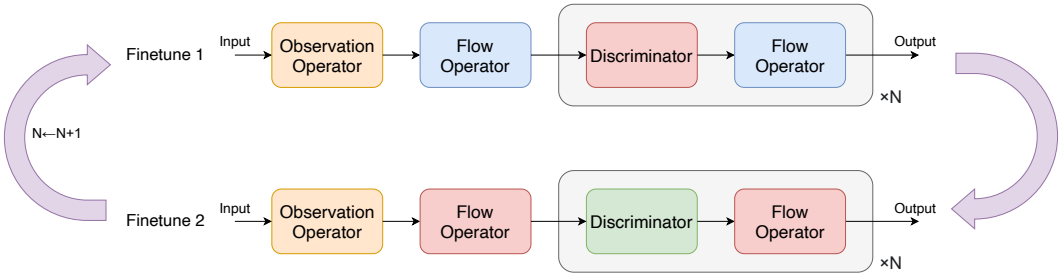

Figure 4: Finetuning procedure for the framework. We let the reader refer to Figure 3 and the main text for details on the procedure and different boxes in the figure.

Finetuning So far, the pretraining procedure is finished. If we'd like to further improve the accuracy of recurrent structure, we can finetune the parameters of discriminators and flow operators. While finetuning, the parameters of discriminators and flow operators are trained alternatively. When a new finetuning round starts, the number of recurrent sections of the recurrent neural network is increased by one, as is shown in Figure 4.

4 Numerical Experiments

In this section, numerical experiments with the proposed framework are reported for Lorenz-63 and Lorenz-96 systems. Owing to their chaotic properties and similarities with meteorological systems, Lorenz systems are widely used for data assimilation issues, especially in data-driven and machine learning case-studies (Lguensat et al., 2017; Raissi, 2018; Fablet et al., 2021b; Frerix et al., 2021).

4.1 Lorenz-63

The first experiment involves Lorenz-63 dynamic system which is governed by the following ODE (Lorenz, 1963):

$$\begin{cases} \frac{dX_1}{dt} &= \sigma\left(X_2 - X_1\right) \\ \frac{dX_2}{dt} &= \rho X_1 - X_2 - X_1 X_3 \\ \frac{dX_3}{dt} &= X_1 X_2 - \beta X_3 \end{cases} \tag{8}$$

We refer to the previous work for the selection of parameterization (Fablet et al., 2021b), in which $\sigma = 10$, $\rho = 28$, and $\beta = 8/3$. In this experiments, we simulate Lorenz-63 time series with a 0.01 time step using a RK45 integration scheme (Dormand & Prince, 1980). The assimilation window length is 200 time steps. The observation data is sampled every 8 steps and only the first component is observed. The observation involves a Gaussian noise with a variance of 2. The training data comprises 10000 sequences sampled from time series with 12000 steps. The validation data

Model	# of Blocks	R-score	LossDyn	Test Time (seconds)
NN with Three-Op	2	1.28	1.10e-3	3.1
NN with Three-Op	4	0.923	3.82e-4	4.5
NN with Three-Op	6	0.799	2.08e-4	5.8
NN with Three-Op	8	**0.751**	**1.52e-4**	6.4
GENN aG-Conv	20	1.34	>1.00e-2	27.1

Table 1: Lorenz-63 Experiments. We report the reconstruction and dynamical performance of the proposed framework for Lorenz-63 system. The results of state-of-the-art neural network draws directly from the original paper (Fablet et al., 2021b). Bold values refer to the best score. All of the models are tested on an independent test set with 2000 samples. The batch size is set to 8 and all the evaluations are performed on one GPU. Test time may vary from experiment to experiment and we average over 10 independent tests.

comprises 2000 sequences from time series with 5000 steps and test data with 3000 sequences from 10000 time steps.

We adopt similar structures in Figure 2 to construct the observation operator, discriminator and flow operator respectively and the only difference is that a 4-layer architecture is adopted. We choose $c = 3$, $K = 1$, $T = 200$, $h = 32$ for the U-Net parameters.

Firstly, we consider the case in which the prior dynamics are identical to the ground truth dynamic, that is $\mathcal{M}_{gt} = \mathcal{M}_{pr}$. In this case, we don't need to take into consideration the stopping criterion (inequality 7) for training the flow operator.

We use a total of eight blocks to train the deep learning framework, including an observation operator and a flow operator in the front, followed by three pairs of discriminators and flow operators, as shown in Figure 1 with $N = 3$. The observation operator and flow operator are pretrained for 20 epochs and 200 epochs, respectively. Pretraining for the discriminator and all the finetuning procedures takes 10 epochs each.

In Table 1, we compare our finetuned neural network structure with the previous state-of-the-art network in terms of R-score metric and evaluation time on test set. Our proposed method achieves better reconstruction performance than the state-of-the-art on the metric of R-score. The reconstruction error is 44% smaller and the dynamic loss is two orders of magnitude smaller than it, which is a significant achievement. As long as the network ends with the flow operator, the dynamic loss is at least one order of magnitude smaller than the state-of-the-art, regardless of the number of blocks. We also find that adding more blocks to the structure will generally leads to better performance both on R-score and dynamic loss. As for the evaluation time, our proposed framework with eight blocks is over four times faster than the state-of-the-art on one GPU.

We select an example from test set in Figure 5 to further illustrate how our architecture works. Focus on the blue oval box area in the upper right corner, where we find how the cooperation of discriminator and flow operator succeeds in correcting the initial states. In the beginning, the states generated by two blocks are clearly inconsistent with ground truth in the upper right region. After a discriminator is added (block 3), disturbances with high frequencies are involved in the misaligned area to distinguish them from well-constructed areas. If a flow operator is further added behind the discriminator (block 4), the high-frequency noise disappears and the inconsistency is corrected to some extent in the new states. Such identification-refinement step is repeated over and over again (block 5 & block 6) and the misalignment becomes smaller and smaller. The lower left panel also shows a continuous decrease in reconstruction error from two, four, to six blocks (orange line), although the addition of supervisors alone brings a temporary increase in error (blue line).

In the following experiment, additional error is introduced to the prior dynamics by perturbing the ODE parameters. In the case of Lorenz-63 systems, σ in Equation 8 is selected to be the perturbed parameter and the ODE for the prior dynamics can be formulated as

$$\begin{cases} \frac{dX_1}{dt} &= (\sigma + \Delta\sigma)(X_2 - X_1) \\ \frac{dX_2}{dt} &= \rho X_1 - X_2 - X_1 X_3 \\ \frac{dX_3}{dt} &= X_1 X_2 - \beta X_3 \end{cases} \tag{9}$$

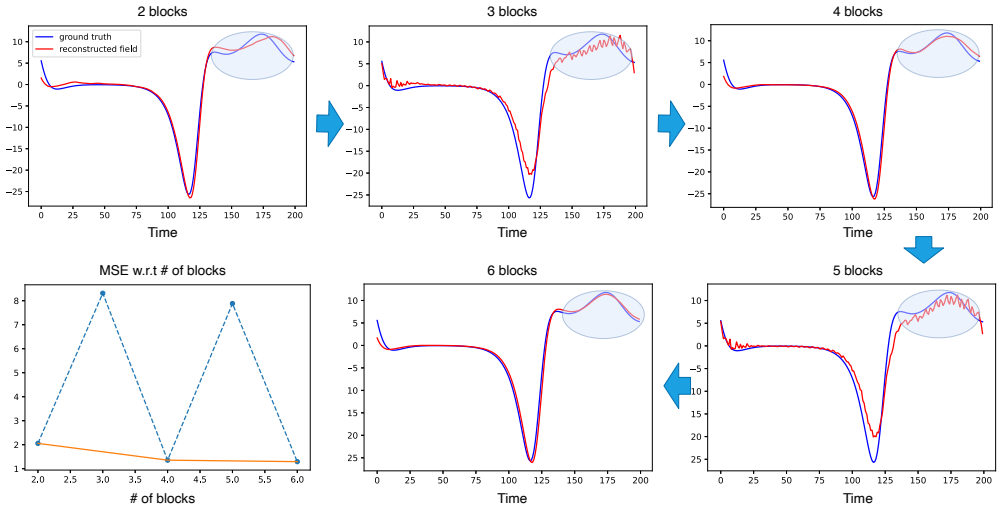

Figure 5: An intuitive presentation of how the proposed structure works. One sample from the test set is randomly selected to evaluate the pretrained and finetuned neural network with eight blocks. We truncate the model into shorter lengths, from two blocks with only one observation operator and one flow operator, to six blocks, and evaluate their performances. The five upper right panels demonstrate the change of reconstructed states when more blocks are added, while the bottom left panel shows the relationship between R-score on test set and the number of blocks.

Specifically, we select $\Delta\sigma$ from the set $\{0.25, 0.50, 0.75, 1.00, 1.50, 2.00, 2.50, 3.00, 3.50, 4.00, 4.50, 5.00\}$ and conduct numerical experiments accordingly. The neural network structures and training losses are almost identical to those in the experiment with no bias, except that we use the prior dynamics rather than ground truth dynamics to construct the dynamic loss.

We report our results in Figure 6. As illustrated in the right figure, if $\Delta\sigma$ is smaller than 2.00, R-score increases as $\Delta\sigma$ increases, whereas if $\Delta\sigma$ is larger than 2.00, R-score oscillates up and down without monotonic changes. Such phenomenon could be explained by looking into how our network structure works: When $\Delta\sigma$ is small, the difference between the prior dynamics and ground truth is small and the information of the prior dynamics can be successfully exploited by the flow operator. Therefore, as $\Delta\sigma$ gets larger, useful information provided by the prior dynamics decreases and the performance of our framework gets worse. However, if $\Delta\sigma$ is too large, it becomes no longer necessary for the flow operator to learn from the bad prior dynamics and it is cheaper to learn directly from ground truth labels instead. Hence, in this case, the value of $\Delta\sigma$ does not affect the performance of the neural network. Results shown in the left figure can further confirm our explanation. When $\Delta\sigma < 2$, the difference between $\mathcal{L}_{dyn-pr}(\hat{x})$ and $\mathcal{L}_{dyn-pr}(x_{gt})$ is close to zero, which is consistent with our stopping criterion (Inequality 7), thus the network works as expected. On the other hand, if $\Delta\sigma > 2$, the bias of prior dynamics is too large for the flow operator to learn and solely learning from the ground truth is sufficient for the neural network to generate states whose dynamic loss is smaller than $\mathcal{L}_{dyn-pr}(x_{gt})$.

4.2 Lorenz-96

In the second experiment, we consider Lorenz-96 dynamic system, which can be formulated by the following ODE

$$\frac{dX_i}{dt} = (x_{i+1} - x_{i-2})\, x_{i-1} - x_i + F \tag{10}$$

with i ranging from 1 to 40, F set to 8 and a periodic boundary constraint $x_i = x_{i+40}$.

Again, we adopt the structures in Figure 2 to construct the observation operator, discriminator and flow operator respectively. We choose $c = 1$, $K = 40$, $T = 200$, $h = 32$ for the U-Net parameters.

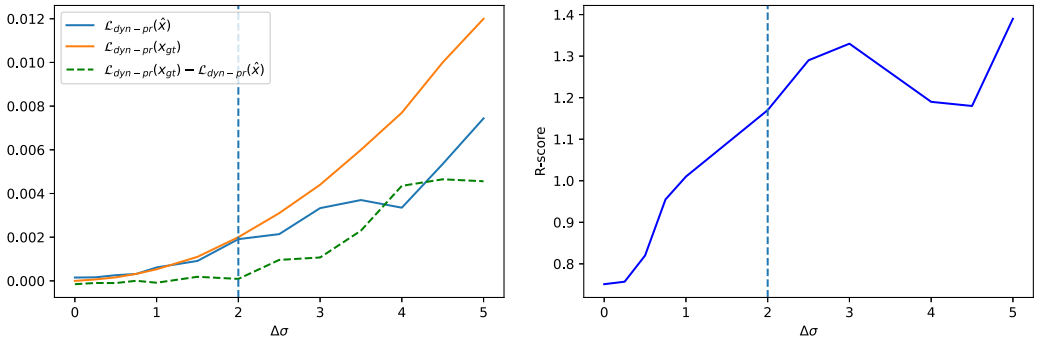

Figure 6: The left panel shows the dynamic loss of reconstructed states (blue line), the biases of prior dynamics (orange line) and their differences (green dotted line) against $\Delta\sigma$. The right panel shows the reconstruction loss (R-score) in test set against $\Delta\sigma$.

Model	# of Blocks	R-score	LossDyn	Test Time (seconds)
NN with Three-Op	2	0.571	1.40e-3	7.8
NN with Three-Op	4	0.450	5.09e-4	9.8
NN with Three-Op	6	0.397	3.14e-4	11.5
NN with Three-Op	8	0.373	2.66e-4	13.5
NN with Three-Op	10	**0.367**	**2.36e-4**	15.8
GENN aG-Conv	20	0.38	>1.00e-2	47.3

Table 2: Lorenz-96 Experiments. We report the reconstruction and dynamical performance of the proposed framework for Lorenz-96 system. We let reader refer to main text for details. Bold values refer to the best score. All of the models are tested on an independent test set with 3000 samples. The batch size is set to 8 and all the evaluation is performed on one GPU. The model is trained with 8 blocks at most. We add two more blocks to test the generalization ability of the architecture.

Similarly to Lorenz-63 experiments, we first consider the case in which the ground truth satisfies the prior dynamics perfectly, that is, $\mathcal{M}_{gt} = \mathcal{M}_{pr}$. The deep learning framework is also trained with totally 8 blocks, as shown in Figure 1 with $N = 3$. The observation operator and flow operator are pretrained for 20 epochs and 200 epochs. Pretraining for the discriminator and all the finetuning procedures takes 10 epochs respectively.

A quantitative comparison between the proposed networks with different blocks and the state-of-the-art structure is reported in Table 2. If the number of blocks is larger than six, the reconstruction performance of our neural network is as good as the state-of-the-art neural network. Again, the more blocks the framework is composed of, the better performance it achieves, in terms of not only R-score but also dynamic loss. Whichever the case, the dynamic loss is also at least one magnitude smaller than the state-of-the-art, thanks to the flow operator. Specifically, the network shows better performance when two more blocks are added to the 8-block structure without further finetuning, which may indicate that our model has some generalization ability. For every additional two blocks, the test time increases by approximately two seconds. Our framework with 10 blocks takes 15.8 seconds for evaluation, approximately three times faster than state-of-the-art framework, which takes 47.3 seconds. The network with 10 blocks reduces reconstruction error by 3%, compared with the state-of-the-art. In contrast to the Lorenz-63 system, the accuracy gain for Lorenz-96 is relatively small. It could be boiled down to the fact that the observations in Lorenz-96 system are irregularly sampled, thereby bringing greater difficulty for flow operators to learn from dynamics, than the regularly sampled Lorenz-63 case.

Further results are illustrated in Figure 7. The variation of reconstruction and dynamic loss with respect to the training epochs is similar to those in the Lorenz-63 system. Visually, the light regions in the error map becomes smaller and darker as the number of block grows, which confirms the effects of the collaboration of discriminator and flow operator in this example.

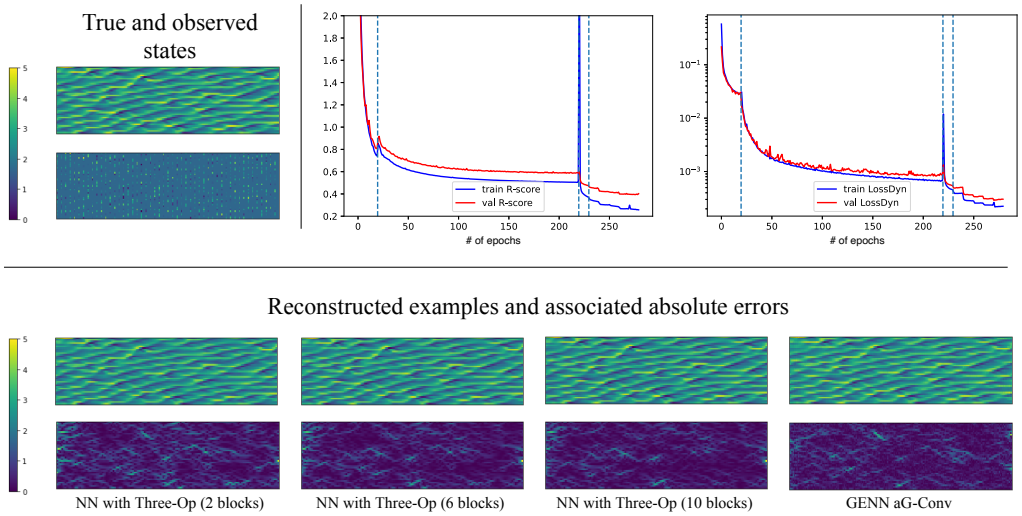

Figure 7: Reconstruction of Lorenz-96 dynamics. The upper left panel is an example of true state sequences and its corresponding observations, with the horizontal axis being the time axis and the vertical axis being the spatial axis. The upper right panel demonstrates the variation of reconstruction and dynamic loss with respect to the training epochs. The reconstructed states from observations in the upper left panel by different neural networks are displayed in the first row of the lower panel. The second row demonstrates absolute error maps of their associated reconstructions. All the displayed figures of states and errors share the same colormap.

To continue, we add additional error to the prior dynamics by perturbing the ODE parameters and for the Lorenz-96 systems, F in Equation 10 is the only parameter. ODE for the prior dynamics can be formulated as

$$\frac{dX_i}{dt} = (x_{i+1} - x_{i-2})\, x_{i-1} - x_i + F + \Delta F \tag{11}$$

Similar to what we have done for the Lorenz-63 system, we select ΔF from the set $\{0.50, 1.00, 1.50, 2.00, 2.50, 3.00, 3.50, 4.00, 4.50, 5.00\}$ and repeat the numerical experiments. The patterns of both dynamic loss and reconstruction loss resemble those in the Lorenz-63 system. The threshold changes from 2.0 to around 2.5. If ΔF is smaller than 2.5, the reconstruction error rises as the bias of prior dynamics increases; otherwise, the error remains high and does not change with the prior dynamics. Again, this finding provides further evidence that our network learns features from prior dynamics rather than simply remembering ground truth labels, and that our framework works properly despite biases in prior dynamics.

5 Conclusions

We have introduced a deep learning framework with three operators for data assimilation. The key novelty of our research is the cooperation of discriminator and flow operator to refine reconstructed states. Numerical experiments on Lorenz-63 and Lorenz-96 support the relevance of our framework. In terms of assimilation accuracy, we have shown that our framework reduces reconstruction error by 44% on Lorenz-63 system and 3% on Lorenz-96 system, compared with the state-of-the-art. In terms of efficiency, our framework is four times and three times faster than state-of-the-art networks on Lorenz-63 and Lorenz-96 systems respectively. We have also shown that our framework is robust to biases in the prior dynamics.

Our findings inspire a new method of integrating ODE information into neural networks. They suggest that ODE information may be applied to modify coarse reconstruction states instead of being explicitly embedded into neural network structures. Future work will explore the generalization of this framework to PDE scenarios and attempt to construct pure unsupervised learning schemes with high accuracies.

REFERENCES

Rossella Arcucci, Jiangcheng Zhu, Shuang Hu, and Yi-Ke Guo. Deep data assimilation: integrating deep learning with data assimilation. *Applied Sciences*, 11(3):1114, 2021.

P Bergthorsson, BR Döös, S Fryklund, O Haug, and R Lindquist. Routine forecasting with the barotropic model. *Tellus*, 7(2):272–274, 1955.

Massimo Bonavita and Patrick Laloyaux. Machine learning for model error inference and correction. *Journal of Advances in Modeling Earth Systems*, 12(12):e2020MS002232, 2020.

Julien Brajard, Alberto Carrassi, 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. *Journal of Computational Science*, 44:101171, 2020.

James Carpenter, Peter Clifford, and Paul Fearnhead. Improved particle filter for nonlinear problems. *IEE Proceedings-Radar, Sonar and Navigation*, 146(1):2–7, 1999.

Rosangela Saher Cintra and Haroldo F de Campos Velho. Data assimilation by artificial neural networks for an atmospheric general circulation model. *Advanced applications for artificial neural networks*, 265, 2018.

HF de Campos Velho, S Stephany, AJ Preto, NL Vijaykumar, and AG Nowosad. A neural network implementation for data assimilation using mpi. *WIT Transactions on Information and Communication Technologies*, 27, 2002.

John R Dormand and Peter J Prince. A family of embedded runge-kutta formulae. *Journal of computational and applied mathematics*, 6(1):19–26, 1980.

Geir Evensen. The ensemble kalman filter: Theoretical formulation and practical implementation. *Ocean dynamics*, 53(4):343–367, 2003.

Ronan Fablet, Mohamed Mahmoud Amar, Quentin Febvre, Maxime Beauchamp, and Bertrand Chapron. End-to-end physics-informed representation learning for satellite ocean remote sensing data: Applications to satellite altimetry and sea surface currents. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, (3), 2021a.

Ronan Fablet, Bertrand Chapron, Lucas Drumetz, Etienne Mémin, Olivier Pannekoucke, and François Rousseau. Learning variational data assimilation models and solvers. *Journal of Advances in Modeling Earth Systems*, 13(10):e2021MS002572, 2021b.

Alban Farchi, Patrick Laloyaux, Massimo Bonavita, and Marc Bocquet. Using machine learning to correct model error in data assimilation and forecast applications. *Quarterly Journal of the Royal Meteorological Society*, 147(739):3067–3084, 2021.

Jinchao Feng, Jianguang Deng, Zhe Li, Zhonghua Sun, Huijing Dou, and Kebin Jia. End-to-end res-unet based reconstruction algorithm for photoacoustic imaging. *Biomedical optics express*, 11(9):5321–5340, 2020.

Thomas Frerix, Dmitrii Kochkov, Jamie Smith, Daniel Cremers, Michael Brenner, and Stephan Hoyer. Variational data assimilation with a learned inverse observation operator. In *International Conference on Machine Learning*, pp. 3449–3458. PMLR, 2021.

Lev Semenovich Gandin. Objective analysis of meteorological field. *Gidrometeorologicheskoe Izdate'stvo*, 286, 1963.

Steven Guan, Amir A Khan, Siddhartha Sikdar, and Parag V Chitnis. Fully dense unet for 2-d sparse photoacoustic tomography artifact removal. *IEEE journal of biomedical and health informatics*, 24(2):568–576, 2019.

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, 2008.

Sam Hatfield, Matthew Chantry, Peter Dueben, Philippe Lopez, Alan Geer, and Tim Palmer. Building tangent-linear and adjoint models for data assimilation with neural networks. *Journal of Advances in Modeling Earth Systems*, 13(9):e2021MS002521, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

William W Hsieh and Benyang Tang. Applying neural network models to prediction and data analysis in meteorology and oceanography. *Bulletin of the American Meteorological Society*, 79(9): 1855–1870, 1998.

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

John M Lewis and John C Derber. The use of adjoint equations to solve a variational adjustment problem with advective constraints. *Tellus A*, 37(4):309–322, 1985.

Redouane Lguensat, Pierre Tandeo, Pierre Ailliot, Manuel Pulido, and Ronan Fablet. The analog data assimilation. *Monthly Weather Review*, 145(10):4093–4107, 2017.

Edward N Lorenz. Deterministic nonperiodic flow. *Journal of atmospheric sciences*, 20(2):130–141, 1963.

Julian Mack, Rossella Arcucci, Miguel Molina-Solana, and Yi-Ke Guo. Attention-based convolutional autoencoders for 3d-variational data assimilation. *Computer Methods in Applied Mechanics and Engineering*, 372:113291, 2020.

Narinder Singh Punn and Sonali Agarwal. Modality specific u-net variants for biomedical image segmentation: a survey. *Artificial Intelligence Review*, pp. 1–45, 2022.

Florence Rabier, Heikki 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.

Maziar Raissi. Deep hidden physics models: Deep learning of nonlinear partial differential equations. *The Journal of Machine Learning Research*, 19(1):932–955, 2018.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Yoshikazu Sasaki. Some basic formalisms in numerical variational analysis. *Monthly Weather Review*, 98(12):875–883, 1970.

Nahian Siddique, Sidike Paheding, Colin P Elkin, and Vijay Devabhaktuni. U-net and its variants for medical image segmentation: A review of theory and applications. *Ieee Access*, 9:82031–82057, 2021.

Nandamudi Lankalapalli Vijaykumar, Stephan Stephany, HF Campos Velho, AJ Preto, and AG Nowosad. Optimized neural network code for data assimilation. In *Brazilian Congress on Meteorology*, volume 12, pp. 38413849, 2002.

Greg Welch, Gary Bishop, et al. An introduction to the kalman filter. 1995.

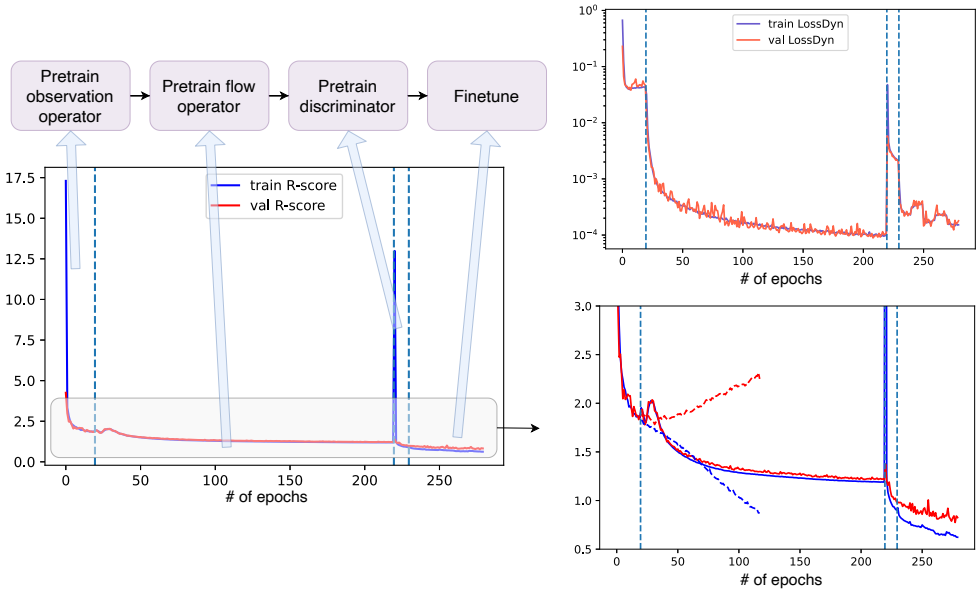

Figure 8: The variation of R-score and dynamic loss with the number of epochs. R-score is a synonym for reconstruction loss (mean square error). Three vertical dashed lines divide the training procedure into four periods: pretraining observation operator, pretraining flow operator, pretraining discriminator and finetuning. The left panel demonstrates the changes in R-score during training, the details of which are enlarged and shown in the bottom right panel. The red and blue dashed lines in the bottom right panel show how continuing to train the observation operator without adding new operators would result in a change in the reconstructed loss function. The top right panel demonstrates the variation of dynamic loss during training.

A Appendix

A.1 Supplementary Experimental Results

The variation of R-score and dynamic loss against the number of epochs is reported in Figure 8. During pretraining, the observation operator learns a map from observational space to physical states at first. Since a shift between two different spaces takes place in this period, we see a steep decrease of loss within the first 20 epochs. After the pretraining of observation operator is finished, we start the pretraining of flow operator to correct the flow dynamics of reconstructed states. The pretraining of flow operator is relatively slow because an unsupervised loss is employed in this procedure. We claim that the flow operator is indispensable in further decreasing the reconstruction loss and dynamic loss, for the reason that without the flow operator, the observation operator would quickly lead to overfitting, in other words, the difference between the training loss and validation loss will diverge, as shown in the red and blue dashed lines in the figure. During the pretraining of discriminator and finetuning procedure, the reconstruction loss drops steadily while the dynamic loss jumps low and high. This is not surprising considering that the finetuning procedure alternates between discriminator and flow operator. If the training object is discriminator, the dynamic loss is relatively high because the optimized target is the reconstruction loss; if the training object is flow operator, the dynamic loss is relatively low because the optimized target is just the dynamic loss. Such alternating finetuning process contributes to the continuous drop of R-score.

A.2 Related Work

In Fablet et al. (2021b), an end-to-end learning framework for 4D-Var assimilation issue is proposed, which combines a neural-network representation of the dynamics and a neural-network solver for the considered variational cost. What our work has in common with theirs is that we both con-

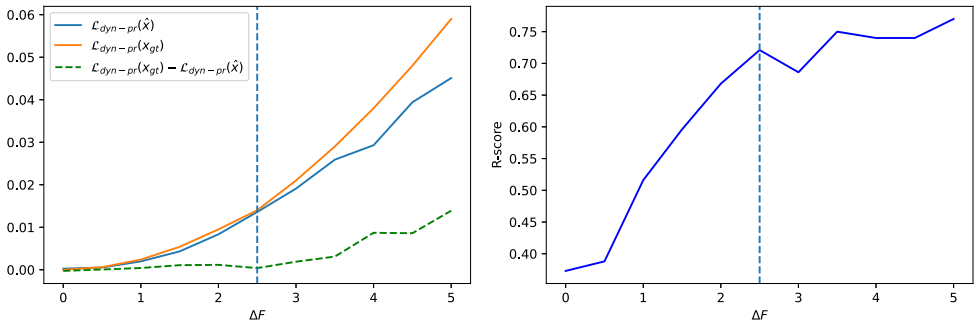

Figure 9: The left panel demonstrates the dynamic loss of reconstructed states (blue line), the biases of prior dynamics (orange line) and their differences (green dotted line) against $\Delta\sigma$. The right panel shows the variation of reconstruction loss (R-score) in test set with respect to $\Delta\sigma$.

sider assimilation issue over a time series of states; the observations are noisy and incomplete; the dynamics are known as prior information. On this basis, we made some improvements and more in-depth exploration of their work. Firstly, their work only considers a simplified ideal case, in which the prior dynamics are precisely consistent with ground truth. We extend their previous work to the case where the prior model is in error and explore the relationship between network performance and prior dynamic bias. Moreover, Fablet et al. (2021b) claimed in their work that numerical ODE-based representations which are optimal for forecasting performance may not be optimal for reconstruction purposes. Whereas in our work, we directly use the numerical ODE-based loss function to train the flow operator without explicitly learning a neural representation for dynamics and obtain even better results, both in terms of accuracy and efficiency. This finding may suggest that it is possible to properly construct a deep learning scheme so that the network can directly learn from ODE, instead of its neural representations. By comparing the experimental results, we claim that our work makes better use of numerical ODE information.

Frerix et al. (2021) is another highly relevant work, in which an observation operator is also proposed to realize 4D-Var assimilation. Our work differs from theirs in at least two major aspects. The first is that in their work, the strong constraint four dimensional variation is considered, with the presumption that the prior model is not unbiased. Whereas our work sticks to a weak constraint assumption, in other words, the prior model is assumed to be biased. Such difference allows our works to be applied to a wider range of scenarios. The second is that the observation operator is only used to find the initial states for gradient descent optimization in their work. Traditional procedure of solving the reconstructed states by iteratively optimizing the 4D-Var target function is not avoided. By contrast, our proposed framework completely replaces the iterative process of traditional algorithms with a neural network framework composed of three operators, thereby significantly reducing the number of iteration rounds, from hundreds to less than ten.

A.3 Application in Real Scenarios

In our research, both supervised learning and unsupervised learning are employed to train the deep learning framework. Therefore, ground truth is required to serve as labels. In real scenarios, although it is not possible to obtain ground truth states, we can regard historical reanalysis data as ground truth training data because they are the highest precision state information available to humans. By training the proposed learning framework, we can realize a neural assimilation method. Despite that its accuracy may not exceed that of historical reanalysis data, the efficiency is greatly improved (Fablet et al., 2021b).

A.4 Further Improvement

The first aspect that needs to be improved for our work is unsupervised learning. As discussed in section A.3, our work does not realize a completely unsupervised learning scheme yet. It poses

a great challenge to build a neural network which combines data assimilation and unsupervised machine learning, though previous work has partially addressed this problem. For example, in Fablet et al. (2021b), an unsupervised learning scheme is also proposed, but the accuracy can hardly exceed that of no learning schemes.

Another aspect is about higher dimensions. Up to now, our experiments are conducted on one dimensional Lorenz-63 and Lorenz-96 systems. Higher dimensional systems are more difficult in some areas. For instance, the dynamics are usually formulated by PDE instead of ODE; the non-linearity is stronger. Whether these findings generalize to higher dimensional systems requires further study.