

Reduced order surrogate modelling and Latent Assimilation for dynamical systems

Sibo Cheng^{1,†}, César Quilodrán-Casas^{1,2,†}, and Rossella Arcucci^{1,2}

¹ Data Science Institute, Imperial College London

² Department of Earth Science and Engineering, Imperial College London
sibo.cheng, c.quilodran, r.arcucci@imperial.ac.uk

[†] Equal contribution to this work

Keywords: Reduced order modelling · Deep learning · Data assimilation · Adversarial neural networks

Abstract. For high-dimensional dynamical systems, running high-fidelity physical simulations can be computationally expensive. Much research effort has been devoted to develop efficient algorithms which can predict the dynamics in a low-dimensional reduced space. In this paper, we developed a modular approach which makes use of different reduced-order modelling for data compression. Machine learning methods are then carried out in the reduced space to learn the dynamics of the physical systems. Furthermore, with the help of data assimilation, the proposed modular approach can also incorporate observations to perform real-time corrections with a low computational cost. In the present work, we applied this modular approach to the forecasting of wildfire, air pollution and fluid dynamics. Using the machine learning surrogate model instead of physics-based simulations will speed up the forecast process towards a real-time solution while keeping the prediction accuracy. The data-driven algorithm schemes introduced in this work can be easily applied/extended to other dynamical systems.

1 Introduction

Reduced-order modelling is a reduced dimensionality model surrogate of an existing system in a reduced-order space. reduced-order modelling (ROM), in combination with machine learning (ML) algorithms is of increasing interest of research in engineering and environmental science. This approach improves the computational efficiency for high-dimensional systems. Since forecasting the full physical space is computationally costly, much effort has been given to develop ML-based surrogate models in the pre-trained reduced-order space.

In recent years, the algorithm schemes which combine ROM and ML surrogate models have been applied in a variety of engineering problems, including computational fluid dynamics [33, 32], numerical weather prediction [25] and nuclear science [30], among others, to speed up computational models without

losing the resolution and accuracy of the original model. Typically, the first stage consists of reducing the dimension of the problem by compression methods such as Principal Component Analysis (PCA), autoencoder (AE), or a combination of both [31, 29]. Solutions from the original computational models (known as snapshots) are then projected onto the lower-dimensional space, and the resulting snapshot coefficients are interpolated in some way, to approximate the evolution of the model.

In order to incorporate real-times observations, data assimilation (DA), originally developed in meteorological science, is a reference method for system updating and monitoring. Some recent researches [10, 2, 5, 26] have focused on combining DA algorithms and ROMs so that the system correction/adjusting can be performed with a low computational cost. Adversarial training and Generative Adversarial Networks (GAN), introduced by [17], has also been used with ROM. Data-driven modelling of nonlinear fluid flows incorporating adversarial networks has been successfully studied previously [6]. GANs are also being used to capture physics of molecular dynamics [41] and have potential to aid in the modeling and simulation of turbulence [24].

The aim of this work is to create general workflows in order to tackle different applications combining DA and ML approaches. In the present work, we propose a modular approach, which combines ROM, ML surrogate models, and DA for complex dynamic systems with applications in computational fluid dynamic (CFD), wildfire spread and air pollution forecasting. The algorithms described in this work can be easily applied/extended to other dynamical systems. Numerical results in both applications show that the proposed approach is capable of real-time predictions, yielding an accurate result and considerable speed-up when compared to the computational time of simulations.

The paper is structured as follows: Section 2 presents the modular approach and its components. The applications are shown in Section 3 and 4. And finally, Discussions and Conclusions are presented in Section 5.

2 Components of the modular approach

The modular approach shown in this paper can be summarised by Figure 1. The state model (\mathbf{u}_t) is compressed using ROMs approaches such as PCA, AE or a combination of both, followed by a ML-based forecast in the reduced space. This forecast is then corrected using DA, incorporating real-time observations (\mathbf{v}_t). This is an iterative process that can be used to improve the starting point of the next time-level forecast, thus improving its accuracy [3].

2.1 Reduced order modelling

In this section, we introduce two types of ROMs, namely the PCA and the convolutional autoencoder (CAE).

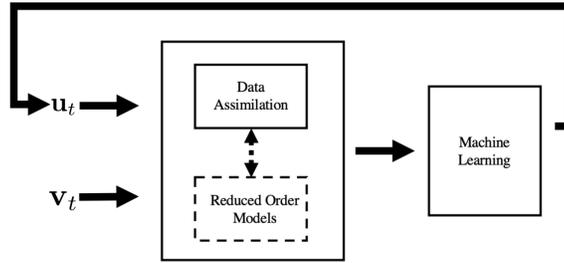


Fig. 1: Flowchart of the modular approach, modified using figure 1 of [4]

2.1.1 Principle component analysis

principal component analysis, also known as Karhunen–Loève transform or Hotelling transform, is a reference method of ROM via an orthogonal linear projection. This approach has been widely applied in dynamical systems [38] with snapshots at different time steps. Applications can be found in a large range of engineering problems, including numerical weather prediction [22], hydrology [10] or nuclear engineering [16]. More precisely, a set of n_u simulated or observed fields $\{\mathbf{u}_{t_0, t_1, \dots, t_{n_u-1}}\}$ at different time are flattened and combined vertically to a matrix,

$$\mathbf{U} = \left[\mathbf{u}_{t_0} \mid \mathbf{u}_{t_1} \mid \dots \mid \mathbf{u}_{t_{n_u-1}} \right]. \quad (1)$$

The principle components are then extracted by computing the empirical covariance matrix, that is,

$$\mathbf{C}_{\mathbf{u}} = \frac{1}{n_u - 1} \mathbf{U} \mathbf{U}^T = \mathbf{L}_{\mathbf{U}} \mathbf{D}_{\mathbf{U}} \mathbf{L}_{\mathbf{U}}^T, \quad (2)$$

where each column of $\mathbf{L}_{\mathbf{U}}$ which represents an eigenvector of $\mathbf{C}_{\mathbf{u}}$ and $\mathbf{D}_{\mathbf{U}}$ is the associated eigenvalue diagonal matrix. The dynamical field \mathbf{u}_t can then be compressed to

$$\tilde{\mathbf{u}}_t = \mathbf{L}_{\mathbf{U}, q}^T \mathbf{u}_t, \quad (3)$$

where $\tilde{\mathbf{u}}_t$ denotes the compressed state vectors and q is the truncation parameter and the reconstruction to the full physical space reads

$$\mathbf{u}_t^{\text{PCA}} = \mathbf{L}_{\mathbf{U}, q} \tilde{\mathbf{u}}_t = \mathbf{L}_{\mathbf{U}, q} \mathbf{L}_{\mathbf{U}, q}^T \mathbf{u}_t. \quad (4)$$

2.1.2 Convolutional autoencoder

PCA, by design, is a linear method for ROM. In the last several years, much effort has been given to dimension reduction for chaotic dynamical systems via deep learning (DL) AEs [15]. Typically, an AE is an unsupervised neural network (NN) which consists two parts: an encoder E which maps the input variables to latent (i.e., compressed) vectors and a decoder D which performs reconstructions from

the low-dimensional latent space to the full physical space. These processes can be summarised as:

$$\tilde{\mathbf{u}}_t = E(\mathbf{u}_t) \quad \text{and} \quad \mathbf{u}_t^{\text{AE}} = D(\tilde{\mathbf{u}}_t). \quad (5)$$

It is found that employing convolutional layers in AEs is helpful to i) reduce the number of parameters for high-dimensional systems, and ii) take into account local spatial patterns for structured data (e.g., images and times series) [19]. Following this idea, CAE was developed [19, 35] where both the encoder E and the decoder D consist of a series of convolutional layers.

In general, the encoder and the decoder of an AE should be trained jointly, for instance, with the Mean square error (MSE) or the Mean absolute error (MAE) loss function of reconstruction accuracy, i.e.,

$$J_{\text{MSE}}(E, D) = \sum_{j=0}^{n_u-1} \frac{\|\mathbf{u}_{t_j} - D \circ E(\mathbf{u}_{t_j})\|^2}{n_u}, \quad J_{\text{MAE}}(E, D) = \sum_{j=0}^{n_u-1} \frac{\|\mathbf{u}_{t_j} - D \circ E(\mathbf{u}_{t_j})\|}{n_u}. \quad (6)$$

2.2 Machine Learning for surrogate models

2.2.1 Recurrent neural network

Long short-term memory networks neural networks, introduced in [20], is a variant of recurrent neural network (RNN), capable of dealing long term dependency, and vanishing gradient problems that traditional RNN could not handle. One of the components of our modular approach in the present work, is the sequence-to-sequence long short-term memory networks (LSTM). The LSTM learns the dynamics in the latent space from compressed training data.

LSTM can be unidirectional or bidirectional. Recently developed Bidirectional LSTM (BDLSTM) [36] differs from the unidirectional ones, as the latter can capture the forward and backward temporal dependencies in spatiotemporal data [12, 21, 27, 34]. LSTMs are widely recognised as one of the most effective sequential models for times series predictions in engineering problems [40, 28].

The LSTM network comprises three gates: input (\mathbf{i}_{t_k}), forget (\mathbf{f}_{t_k}), and output (\mathbf{o}_{t_k}); a block input, a single cell \mathbf{c}_{t_k} , and an output activation function. This network is recurrently connected back to the input and the three gates. Due to the gated structured and the forget state, the LSTM is an effective and scalable model that can deal with long-term dependencies [20]. The vector equations for a LSTM layer are:

$$\begin{aligned} \mathbf{i}_{t_k} &= \phi(\mathbf{W}_{xi}\mathbf{u}_{t_k} + \mathbf{W}_{Hi}\mathbf{H}_{t_{k-1}} + \mathbf{b}_i) \\ \mathbf{f}_{t_k} &= \phi(\mathbf{W}_{xf}\mathbf{u}_{t_k} + \mathbf{W}_{Hf}\mathbf{H}_{t_{k-1}} + \mathbf{b}_f) \\ \mathbf{o}_{t_k} &= \phi(\mathbf{W}_{xo}\mathbf{u}_{t_k} + \mathbf{W}_{Ho}\mathbf{H}_{t_{k-1}} + \mathbf{b}_o) \\ \mathbf{c}_{t_k} &= \mathbf{f}_{t_k} \circ \mathbf{c}_{t_{k-1}} + \mathbf{i}_{t_k} \circ \tanh(\mathbf{W}_{xc}\mathbf{u}_{t_k} + \mathbf{W}_{Hc}\mathbf{H}_{t_{k-1}} + \mathbf{b}_c) \\ \mathbf{H}_{t_k} &= \mathbf{o}_{t_k} \circ \tanh(\mathbf{c}_{t_k}) \end{aligned} \quad (7)$$

where ϕ is the sigmoid function, \mathbf{W} are the weights, $\mathbf{b}_{i,f,o,c}$ are the biases for the input, forget, output gate and the cell, respectively, \mathbf{x}_{t_k} is the layer input, \mathbf{H}_{t_k} is the layer output and \circ denotes the entry-wise multiplication of two vectors. This is the output of a unidirectional LSTM.

For a BDLSTM, the output layer generates an output vector \mathbf{u}_{t_k} :

$$\mathbf{u}_{t_k} = \psi(\overrightarrow{\mathbf{H}_{t_k}}, \overleftarrow{\mathbf{H}_{t_k}}) \quad (8)$$

where ψ is a concatenating function that combines the two output sequences, forwards and backwards, denoted by a right and left arrow, respectively.

2.2.2 Adversarial network

The work of [18] introduced the idea of adversarial training and adversarial losses which can also be applied to supervised scenarios and have advanced the state-of-the-art in many fields over the past years. Additionally, robustness may be achieved by detecting and rejecting adversarial examples by using adversarial training [37]. GAN are a network trained adversarially. The basic idea of GAN is to simultaneously train a discriminator and a generator, where the discriminator aims to distinguish between real samples and generated samples. By learning and matching the distribution that fits the training data \mathbf{x} , the aim is that new samples, sampled from the matched distribution formed by the generator, will produce, or generate, ‘realistic’ features from the latent vector \mathbf{z}

The GAN is composed by a discriminator network (\mathcal{D}) and a generator network (\mathcal{G}) The GAN losses, binary cross-entropy, therefore, can be written as:

$$\begin{aligned} \mathcal{L}_{\mathcal{D}}^{adv} &= - \sum \log \mathcal{D}(\mathbf{x}) + \log(1 - \mathcal{D}(\mathcal{G}(\mathbf{z}))) \\ \mathcal{L}_{\mathcal{G}}^{adv} &= - \sum \log(\mathcal{D}(\mathcal{G}(z))) \end{aligned} \quad (9)$$

This idea can be developed further if we consider similar elements of the adversarial training of GAN and applied to other domains, e.g. time-series, extreme events detection, adversarial attacks, among others.

2.3 Data assimilation

Data assimilation algorithms aim to estimate the state variable \mathbf{u} relying on a prior approximation \mathbf{u}_b (also known as the background state) and a vector of observed states \mathbf{v} . The theoretical value of the state vector is denoted by \mathbf{u}_{true} , so called the true state, which is out of reach in real engineering problems. Both the background and the observation vectors are supposed to be noisy in DA, characterised by the associated error covariance matrices \mathbf{B} and \mathbf{R} , respectively, i.e.,

$$\mathbf{B} = \text{Cov}(\epsilon_b, \epsilon_b), \quad \mathbf{R} = \text{Cov}(\epsilon_o, \epsilon_o), \quad (10)$$

with the prior errors ϵ_b and ϵ_o defined as:

$$\epsilon_b = \mathbf{u}_b - \mathbf{u}_{\text{true}} \quad \epsilon_o = \mathcal{H}(\mathbf{u}_{\text{true}}) - \mathbf{v}. \quad (11)$$

Since the true states are out of reach in real applications, the covariance matrices \mathbf{B} and \mathbf{R} are often approximated through statistical estimations [7, 14]. The \mathcal{H} function in equation 11 is called the transformation operator, which maps the state variables to the observable quantities. $\mathcal{H}(\mathbf{u}_{\text{true}})$ is also known as the model equivalent of observations.

By minimizing a cost function J defined as

$$\begin{aligned} J(\mathbf{u}) &= \frac{1}{2}(\mathbf{u} - \mathbf{u}_b)^T \mathbf{B}^{-1}(\mathbf{u} - \mathbf{u}_b) + \frac{1}{2}(\mathbf{v} - \mathcal{H}(\mathbf{u}))^T \mathbf{R}^{-1}(\mathbf{v} - \mathcal{H}(\mathbf{u})) \\ &= J_b(\mathbf{u}) + J_o(\mathbf{u}) \end{aligned} \quad (12)$$

DA approaches attempt to find an optimally weighted *a priori* analysis state,

$$\mathbf{u}_a = \underset{\mathbf{u}}{\operatorname{argmin}}(J(\mathbf{u})). \quad (13)$$

The \mathbf{B} and the \mathbf{R} matrices, determining the weights of background and observation information (as shown in equation 12), is crucial in DA algorithms [39, 11]. When \mathcal{H} can be approximated by a linear function H and the error covariances B and R are well specified, equation 12 can be solved via Best Linear Unbiased Estimator (BLUE) [7]:

$$\mathbf{u}_a = \mathbf{u}_b + K(\mathbf{v} - \mathbf{H}\mathbf{u}_b) \quad (14)$$

where \mathbf{K} denotes the Kalman gain matrix,

$$\mathbf{K} = \mathbf{B}\mathbf{H}^T(\mathbf{H}\mathbf{B}\mathbf{H}^T + \mathbf{R})^{-1}. \quad (15)$$

The optimisation of equation 12 often involves gradient descent algorithms (such as "L-BFGS-B") and adjoint-based numerical techniques. In the proposed modular approach of the present paper, we aim to perform DA in the low-dimensional latent space to reduce the computational cost, enabling a real-time model updating. The latent assimilation (LA) approach was first introduced in the work of [2] for CO_2 spread modeling. A generalised Latent Assimilation algorithm was proposed in the recent work of [9]. The observation quantities \mathbf{v}_t are first preprocessed to fit the space of the state variables \mathbf{u}_t , i.e.,

$$\tilde{\mathbf{v}}_t = E(\mathbf{v}_t). \quad (16)$$

As a consequence, the transformation operator becomes the identity function in the latent space, leading to the loss function of LA:

$$\tilde{J}(\tilde{\mathbf{u}}_t) = \frac{1}{2}(\tilde{\mathbf{u}}_t - \tilde{\mathbf{u}}_{t,b})^T \mathbf{B}^{-1}(\tilde{\mathbf{u}}_t - \tilde{\mathbf{u}}_{t,b}) + \frac{1}{2}(\tilde{\mathbf{u}}_t - \tilde{\mathbf{v}}_t)^T \mathbf{R}^{-1}(\tilde{\mathbf{u}}_t - \tilde{\mathbf{v}}_t), \quad (17)$$

where the latent background state $\tilde{\mathbf{u}}_{t,b}$ is issued from the RNN predictions as mentioned in section 2.2.1. The analysis state,

$$\mathbf{u}_{t,a} = \underset{\mathbf{u}_t}{\operatorname{argmin}}(\tilde{J}(\tilde{\mathbf{u}}_t)), \quad (18)$$

can then replace the background prediction $\tilde{\mathbf{u}}_{t,b}$, which can be used as the starting-point for the next-level prediction in ML algorithms.

3 Application to Wildfires

The first application is real-time forecasting of wildfire dynamics. Wildfires have increasing attention recently in fire safety science world-widely, and it is an extremely challenging task due to the complexities of the physical models and the geographical features. Real-time forecasting of wildfire dynamics which raises increasing attention recently in fire safety science world-widely, is extremely challenging due to the complexities of the physical models and the number of geographical features. Running physics-based simulations for large-scale wildfires can be computationally difficult, if not infeasible. We applied the proposed modular approach for fire forecasting in near real-time, which combines reduced-order modelling, recurrent neural networks RNN, data assimilation DA and error covariance tuning. More precisely, based on snapshots of dynamical fire simulations, we first construct a low-dimensional latent space via proper orthogonal decomposition or convolution AE. A LSTM is then used to build sequence-to-sequence predictions following the simulation results projected/encoded in the reduced space. In order to adjust the prediction of burned areas, latent DA coupled with an error covariance tuning algorithm is performed with the help of daily observed satellite wildfire images as observation data. The proposed method was tested on two recent large fire events in California, namely the Buck fire and the Pier fire, both taking place in 2017 as illustrated in figure 2.

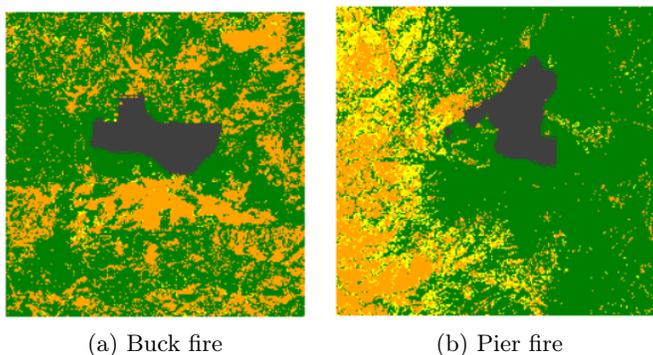


Fig. 2: Observed burned area of the Buck fire and the Pier fire at the first day

We first employed an operational cellular automata (CA) fire spread model [1] to generate the training dataset for ROM and RNN surrogate modelling. This CA model is a probabilistic simulator which takes into account a number of local geographical features, such as vegetation density (see figure 2) and ground elevation. Once the latent space is acquired, the ML-based surrogate model is then trained using the results of stochastic CA simulations in the corresponding area of fire events. With a much shorter online execution time, the so-obtained data-driven model provides similar results as physics-based CA simulations (stochastic) in

the sense that the mean and the standard deviation of CA-CA and CA-LSTM differences are similar as shown in figure 3 for the Pier fire. In fact, the ROM- and ML-based approach run roughly 1000 times faster than the original CA model as shown in figure 3(b).

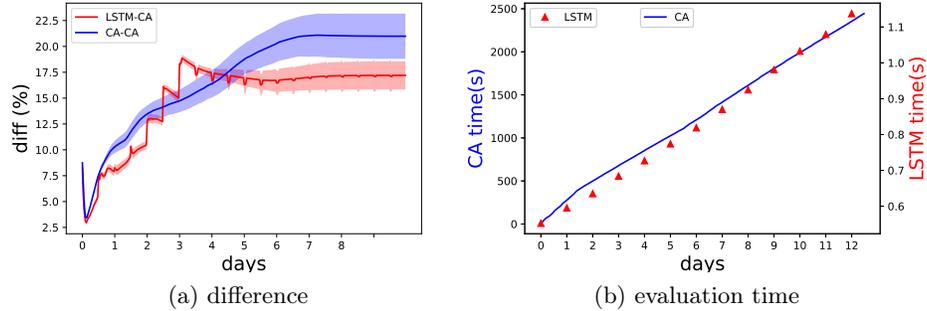


Fig. 3: Model difference and computational time

Each step in CA-LSTM predictions is roughly equivalent to 30 minutes in real time while the satellite observations are of daily basis. The latter is used to adjust the fire prediction consistently since the actual fire spread also depends heavily on other factors such as real-time climate or human interactions which are not included in the CA modelling. The evolution of the averaged relative root mean square error (R-RMSE) is shown in figure 4. The numerical results show that, with the help of error covariance tuning [14, 8], DA manages to improve the model prediction accuracy in both fire events.

4 Application to Computational Fluid Dynamics and air pollution

Similar to the wildfire problem, we also present an general workflow to generate and improve the forecast of model surrogates of CFD simulations using deep learning, and most specifically adversarial training. This adversarial approach aims to reduce the divergence of the forecasts from the underlying physical model. Our two-step method, similar to the wildfire application, integrates a PCA AE with adversarial LSTM networks. Once the reduced-order model (ROM) of the CFD solution is obtained via PCA, an adversarial autoencoder (AAE) is used on the principal components time series. Subsequently, a LSTM model is adversarially trained, named adversarial LSTM (ALSTM), on the latent space produced by the principal component adversarial autoencoder (PC-AAE) to make forecasts. Here we show, that the application of adversarial training improves the rollout of the latent space predictions.

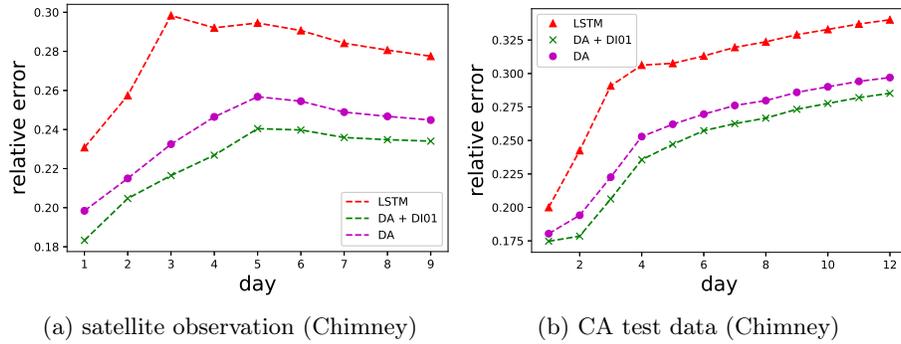


Fig. 4: Relative error of pure LSTM and assimilated prediction compared to satellite observations

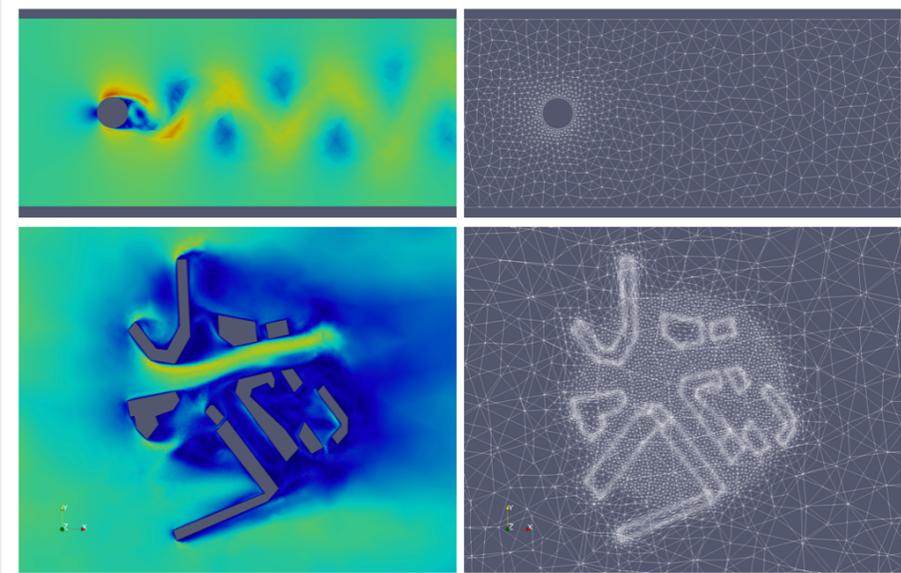
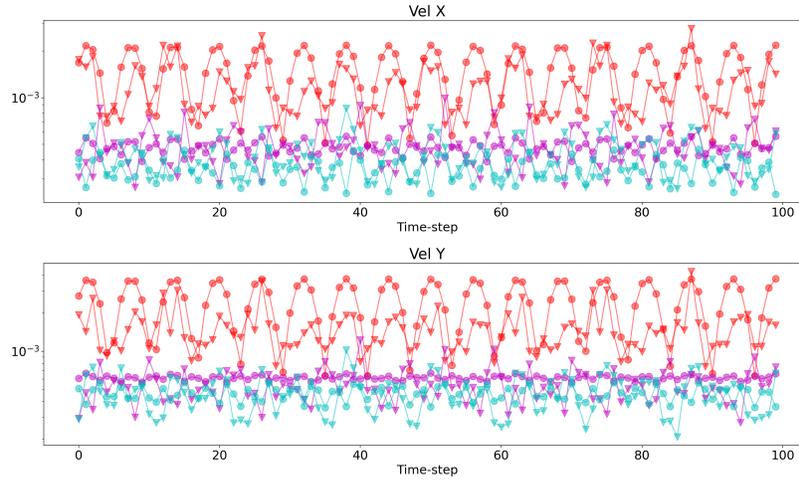


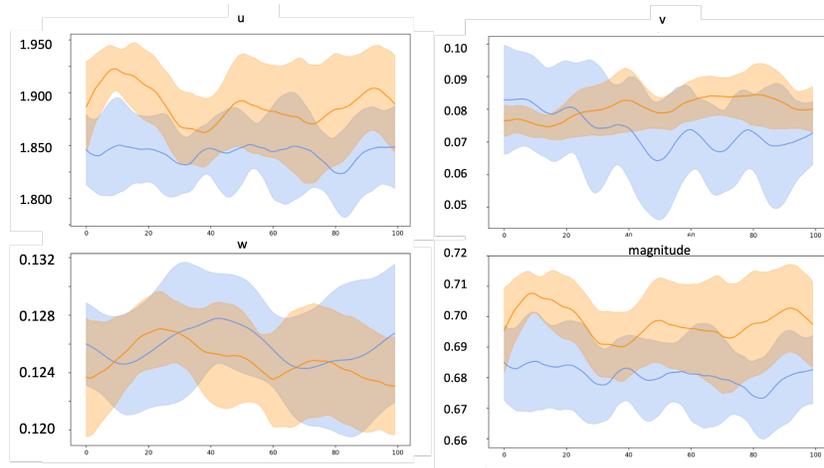
Fig. 5: Top: Flow past the cylinder - 2D CFD simulation. Bottom: 3D Urban Air pollution CFD simulation. Right: Unstructured meshes of both domains

Different case studies are shown in Figure 5:

- FPC: the 2D case describes a typical flow past the cylinder CFD, in which a cylinder placed in a channel at right angle to the oncoming fluid making the steady-state symmetrical flow unstable. This simulation has a Reynolds number (Re) of 2,300 with $m = 5,166$ nodes and $n = 1,000$ time-steps.
- 3DAirPollution: The 3D case is a realistic case including 14 buildings representing a real urban area located near Elephant and Castle, South London,



(a) Compression



(b) Forecast

Fig. 6: Top: Mean absolute error (MAE) of the different compression methods on the velocities (ms^{-1}) of the flow past the cylinder domain. Red, magenta, and cyan represents a compression to 8, 4, and 2 dimensions, respectively. Circle and triangle markers are a PC and a PC-AAE compression, respectively. Bottom: Ensemble, of 50 different starting points, of the MAE of the forecast, for 100 time-levels, of the velocities (ms^{-1}) of the 3D Urban Air Pollution domain. Blue is the original test data, and orange is the prediction obtained with the PC-AAE and the ALSTM.

UK. The 3D case ($720m \times 676m \times 250m$) is composed of an unstructured mesh including $m = 100,040$ nodes per dimension and $n = 1,000$ time-steps.

The two-dimensional (2D) CFD case study was performed using Thetis [23] and the three-dimensional (3D) CFD simulations were carried out using Fluidity [13]. For these domains, the framework was trained and validated on the first 1000 time-steps of the simulation, and tested on the following 500 time-steps.

A PCA as applied to a 2-dimensional velocity field (ms^{-1}) in Flow past the cylinder and likewise to the velocities of the 3D model. The full-rank PCs were used as input for the AAE and divided in 3 different experiments named LS_τ and compared to the corresponding reconstruction \mathbf{x}_τ with $\tau = \{2, 4, 8\}$ PCs. The results of the mean absolute error using the different dimension reduction approaches are shown in Figure 6a for the flow past the cylinder case. The AAE outperforms a simple truncation of the PCs in both domains.

In terms of forecasting, our framework generalises well on unseen data 6b. This is because of the Gaussian latent space obtained with the adversarial AE constraints the further predictions and forces the predictions back into the distribution. Furthermore, the adversarial training of the LSTM learns how to stay within the distribution data. Followed by the training of an adversarial LSTM, we can assess the forecasts using our workflow. An ensemble of 50 different starting points from the test dataset were used to be forecasted in time for 100 time-levels. The ensemble of Mean Absolute Errors results are based on a dimension reduction 8 dimensions in the latent space of the AAE, which is a compression of 5 orders of magnitude. The error percentage of the means of these forecasts is 5% in the test dataset.

5 Conclusions

In the present paper, we introduced a ROM- and ML-based modular approach for efficient predictions of high-dimensional dynamical systems. In addition, this method can also incorporate real-time observations for model correction/adjusting with a low computational cost. A variety of ROM and RNN approaches can be included in the algorithm scheme regarding different applications. The replacement of the physics-based simulation/resolution by these models will speed up the forecast process towards a real-time solution. And, the application of adversarial training could potentially produce more physically realistic scenarios. We showed the strength of the proposed method in predicting wildfire spread and air pollution diffusion in this work. Furthermore, this framework is data-agnostic and could be applied to different physical models when enough data is available.

Acknowledgements

This research is funded by the Leverhulme Centre for Wildfires, Environment and Society through the Leverhulme Trust, grant number RC-2018-023. This work is partially supported by the EP/T000414/1 PREdictive Modelling with

Quantification of Uncertainty for Multiphase Systems (PREMIERE) and by the EPSRC grant EP/T003189/1 Health assessment across biological length scales for personal pollution exposure and its mitigation (INHALE).

References

1. Alexandridis, A., Vakalis, D., Siettos, C., Bafas, G.: A cellular automata model for forest fire spread prediction: The case of the wildfire that swept through spetses island in 1990. *Applied Mathematics and Computation* **204**(1), 191–201 (2008)
2. Amendola, M., Arcucci, R., Mottet, L., Casas, C.Q., Fan, S., Pain, C., Linden, P., Guo, Y.K.: Data assimilation in the latent space of a neural network (2020)
3. Asch, M., Bocquet, M., Nodet, M.: *Data assimilation: methods, algorithms, and applications*, vol. 11. SIAM (2016)
4. Buizza, C., Quilodrán-Casas, C., Nadler, P., Mack, J., Marrone, S., Titus, Z., Le Cornec, C., Heylen, E., Dur, T., Ruiz, L.B., et al.: Data learning: Integrating data assimilation and machine learning. *Journal of Computational Science* **58**, 101525 (2022)
5. Casas, C.Q., Arcucci, R., Wu, P., Pain, C., Guo, Y.K.: A reduced order deep data assimilation model. *Physica D: Nonlinear Phenomena* **412**, 132615 (2020)
6. Cheng, M., Fang, F., Pain, C.C., Navon, I.: Data-driven modelling of nonlinear spatio-temporal fluid flows using a deep convolutional generative adversarial network. *Computer Methods in Applied Mechanics and Engineering* **365**, 113000 (2020)
7. Cheng, S., Argaud, J.P., Iooss, B., Lucor, D., Ponçot, A.: Background error covariance iterative updating with invariant observation measures for data assimilation. *Stochastic Environmental Research and Risk Assessment* **33**(11), 2033–2051 (2019)
8. Cheng, S., Argaud, J.P., Iooss, B., Lucor, D., Ponçot, A.: Error covariance tuning in variational data assimilation: application to an operating hydrological model. *Stochastic Environmental Research and Risk Assessment* **35**(5), 1019–1038 (2021)
9. Cheng, S., Chen, J., Anastasiou, C., Angeli, P., Matar, O.K., Guo, Y.K., Pain, C.C., Arcucci, R.: Generalised latent assimilation in heterogeneous reduced spaces with machine learning surrogate models. *arXiv preprint arXiv:2204.03497* (2022)
10. Cheng, S., Lucor, D., Argaud, J.P.: Observation data compression for variational assimilation of dynamical systems. *Journal of Computational Science* p. 101405 (2021)
11. Cheng, S., Qiu, M.: Observation error covariance specification in dynamical systems for data assimilation using recurrent neural networks. *Neural Computing and Applications* pp. 1–19 (2021)
12. Cui, Z., Ke, R., Pu, Z., Wang, Y.: Stacked bidirectional and unidirectional LSTM recurrent neural network for forecasting network-wide traffic state with missing values. *Transportation Research Part C: Emerging Technologies* **118**, 102674 (2020)
13. Davies, D.R., Wilson, C.R., Kramer, S.C.: Fluidity: A fully unstructured anisotropic adaptive mesh computational modeling framework for geodynamics. *Geochemistry, Geophysics, Geosystems* **12**(6) (2011)
14. Desroziers, G., Ivanov, S.: Diagnosis and adaptive tuning of observation-error parameters in a variational assimilation. *Quarterly Journal of the Royal Meteorological Society* **127**(574), 1433 – 1452 (2001)
15. Dong, G., Liao, G., Liu, H., Kuang, G.: A review of the autoencoder and its variants: A comparative perspective from target recognition in synthetic-aperture radar images. *IEEE Geoscience and Remote Sensing Magazine* **6**(3), 44–68 (2018)

16. Gong, H., Cheng, S., Chen, Z., Li, Q.: Data-enabled physics-informed machine learning for reduced-order modeling digital twin: Application to nuclear reactor physics. *Nuclear Science and Engineering* pp. 1–26 (2022)
17. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: *Advances in neural information processing systems*, pp. 2672–2680 (2014)
18. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: *Advances in neural information processing systems*. pp. 2672–2680 (2014)
19. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. *Science* **313**(5786), 504–507 (2006)
20. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural computation* **9**(8), 1735–1780 (1997)
21. Huang, Z., Xu, W., Yu, K.: Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991* (2015)
22. Jaruszewicz, M., Mandziuk, J.: Application of pca method to weather prediction task. In: *Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP'02. vol. 5*, pp. 2359–2363. IEEE (2002)
23. Kärnä, T., Kramer, S.C., Mitchell, L., Ham, D.A., Piggott, M.D., Baptista, A.M.: Thetis coastal ocean model: discontinuous galerkin discretization for the three-dimensional hydrostatic equations. *Geoscientific Model Development* **11**(11), 4359–4382 (2018)
24. Kim, B., Azevedo, V.C., Thuerey, N., Kim, T., Gross, M., Solenthaler, B.: Deep fluids: A generative network for parameterized fluid simulations. In: *Computer Graphics Forum. vol. 38*, pp. 59–70. Wiley Online Library (2019)
25. Knol, D., Leeuw, F.d., Meirink, J.F., Krzhizhanovskaya, V.V.: Deep learning for solar irradiance nowcasting: A comparison of a recurrent neural network and two traditional methods. In: *International Conference on Computational Science*. pp. 309–322. Springer (2021)
26. Liu, C., Fu, R., Xiao, D., Stefanescu, R., Sharma, P., Zhu, C., Sun, S., Wang, C.: Enkf data-driven reduced order assimilation system. *Engineering Analysis with Boundary Elements* **139**, 46–55 (2022)
27. Liu, G., Guo, J.: Bidirectional lstm with attention mechanism and convolutional layer for text classification. *Neurocomputing* **337**, 325–338 (2019)
28. Nakamura, T., Fukami, K., Hasegawa, K., Nabae, Y., Fukagata, K.: Convolutional neural network and long short-term memory based reduced order surrogate for minimal turbulent channel flow. *Physics of Fluids* **33**(2), 025116 (2021)
29. Phillips, T.R.F., Heaney, C.E., Smith, P.N., Pain, C.C.: An autoencoder-based reduced-order model for eigenvalue problems with application to neutron diffusion. *International Journal for Numerical Methods in Engineering* (2021)
30. Phillips, T.R., Heaney, C.E., Smith, P.N., Pain, C.C.: An autoencoder-based reduced-order model for eigenvalue problems with application to neutron diffusion. *International Journal for Numerical Methods in Engineering* **122**(15), 3780–3811 (2021)
31. Quilodrán Casas, C., Arcucci, R., Guo, Y.: Urban air pollution forecasts generated from latent space representations. In: *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations* (2020)
32. Quilodrán-Casas, C., Arcucci, R., Mottet, L., Guo, Y., Pain, C.: Adversarial autoencoders and adversarial LSTM for improved forecasts of urban air pollution simulations. Published as a workshop paper at ICLR 2021 SimDL Workshop (2021)

33. Quilodrán Casas, C., Arcucci, R., Wu, P., Pain, C., Guo, Y.K.: A Reduced Order Deep Data Assimilation model. *Physica D: Nonlinear Phenomena* **412**, 132615 (2020)
34. Quilodrán-Casas, C., Silva, V.L., Arcucci, R., Heaney, C.E., Guo, Y., Pain, C.C.: Digital twins based on bidirectional lstm and gan for modelling the covid-19 pandemic. *Neurocomputing* **470**, 11–28 (2022)
35. Rawat, W., Wang, Z.: Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation* **29**(9), 2352–2449 (2017)
36. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing* **45**(11), 2673–2681 (1997)
37. Shafahi, A., Najibi, M., Ghiasi, M.A., Xu, Z., Dickerson, J., Studer, C., Davis, L.S., Taylor, G., Goldstein, T.: Adversarial training for free! In: *Advances in Neural Information Processing Systems*. pp. 3358–3369 (2019)
38. Sirovich, L.: Turbulence and the dynamics of coherent structures. II. Symmetries and transformations. *Quarterly of Applied mathematics* **45**(3), 573–582 (1987)
39. Tandeo, P., Ailliot, P., Bocquet, M., Carrassi, A., Miyoshi, T., Pulido, M., Zhen, Y.: A review of innovation-based methods to jointly estimate model and observation error covariance matrices in ensemble data assimilation. *Monthly Weather Review* **148**(10), 3973–3994 (2020)
40. Tekin, S.F., Karaahmetoglu, O., Ilhan, F., Balaban, I., Kozat, S.S.: Spatio-temporal weather forecasting and attention mechanism on convolutional lstms. arXiv preprint arXiv:2102.00696 (2021)
41. Wu, H., Mardt, A., Pasquali, L., Noe, F.: Deep generative markov state models. arXiv preprint arXiv:1805.07601 (2018)