Generative Networks Applied to Model Fluid Flows

Mustapha Jolaade, Vinicius L. S. Silva, Claire E. Heaney, and Christopher C. Pain.

Applied Modelling and Computation Group, Imperial College London, UK.
moj20@ic.ac.uk

Abstract. The production of numerous high fidelity simulations has been a key aspect of research for many-query problems in fluid dynamics. The computational resources and time required to generate these simulations can be so large and impractical. With several successes of generative models, we explore the performance and powerful generative capabilities of both generative adversarial network (GAN) and adversarial autoencoder (AAE) to predict the evolution in time of a highly nonlinear fluid flow. These generative models are incorporated within a reduced-order model framework. The test case comprises two-dimensional Gaussian vortices governed by the time-dependent Navier-Stokes equation. We show that both the GAN and AAE are able to predict the evolution of the positions of the vortices forward in time, generating new samples that have never before been seen by the neural networks.

Keywords: Generative adversarial networks · Adversarial autoencoder · two-dimensional turbulence · Spatial-temporal predictions · Deep learning.

1 Introduction

The study of fluid dynamics has involved massive amounts of data generated either from controlled experiments, field measurements or large-scale numerical simulations. The high volume of data, amongst other reasons, means these methods can be relatively slow and require a great deal of computational power to be able to model the underlying physics. While advancements in high performance computing research has boosted speed and accuracy of numerical simulation, obstacles still remain [2]. Thus, the development of computational frameworks that are accurate, robust, cheap and fast enough to model fluid dynamics remains a key aspect of computational science and engineering research.

In this paper, generative models, a branch of machine learning, is applied to a two-dimensional turbulent fluid problem for the purposes of rapidly predicting forward in time while avoiding the high computational cost of traditional numerical methods.

Generative models have garnered a huge amount of interest in recent years [11]. The main idea behind generative models is to build a statistical model around a

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_61

given dataset that is capable of generating new sample instances that appear to be taken from the original dataset. These new samples can further be used for tackling problems related to the case under study. When the building process is based on deep networks (artificial neural networks such as convolutional neural networks (CNN) [12]) that use multiple layers to capture how patterns/features of the dataset are organised or clustered, the resulting model is termed a deep generative model. Once a deep generative model has learned the structure of the training dataset, by being fed a random vector as input, its networks can generate desired samples from complex probability distributions in high-dimensional spaces [8]. In building deep generative networks two main methods have been widely used. The first is a variational autoencoder that uses stochastic variational inference to minimize the lower bound of the data likelihood [11]. The second is a generative adversarial network (GAN) whereby two players (neural network) play a zero-sum game. The game seeks to minimize the distribution divergence between the model output and the real/training dataset by using real samples as a proxy for optimization. A novel third method born out of the amalgamation of these two methods is the use of adversarial autoencoder (AAE) [14]. In this project, attention is given to both GAN and AAE as data-driven methods for prediction and modelling of spatial-temporal turbulent fluid flow.

Although reduced order models have been used for time-dependent turbulent fluid modelling in areas such as subsurface flow [3] and for the solution of the Navier-stokes equation [19]. In this project, for the first time, we use generative models in a reduced-order model framework to carry out efficient predictions in time of a two-dimensional turbulent fluid flow problem.

The rest of this paper is structured as follows: the next section provides a description of the methodology adopted from [16] for spatial-temporal prediction with GANs. Here, we also include the methodology for prediction using the AAE. Section 3, introduces the test fluid system and a relevant discussion about the transformation carried out to make the data suitable for use. The obtained results from predicting single and multiple time levels are also presented in section 3. Finally, conclusion and remarks about possible future work are provided in section 4.

2 Methodology

The use of GANs for time series prediction and data assimilation of real world dynamical systems has been proven to be successful for the spatial-temporal spread of COVID-19 using SEIRS type models [15, 16]. Particularly, the method in [16] has been shown to be independent of the underlying system, thus this project will apply the same method for the two-dimensional turbulent fluid model.

In this project, we start by building a reduced model of the turbulent fluid flow, going from a high-fidelity spatial domain to a lower dimensional representation. Then, a generative model is built and trained to learn a mapping between a input latent vector and the lower dimensional representation. Finally, we apply the processes of simulating forward in time using the capabilities of the gener-

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_61

ative models. The aim is then for the generative networks to serve as surrogate models that can reproduce the high-fidelity numerical model.

2.1 POD-based Non-Intrusive Reduced Order Modelling

The connection between physics-based machine learning and dimensionality reduction has been substantially studied and well-documented [18]. Results of these studies have shown that many methods used to obtain a low-dimensional subspace of a system are related to machine learning methods. In modern computational research, Reduced Order Modelling (ROM) is a well-known technique for dimensionality reduction [3]. By constructing reduced-order models that encapsulate the original features of the fluid systems while maintaining its underlying physics, it is possible to seek solutions to a model in an efficient and much less expensive way [20]. The Non-Intrusive Reduced Order Modelling (NIROM) is a type of ROM so named due to its non-dependent on the system under study. This model reduction approach can use proper orthogonal decomposition (POD) [17] to derive a physics-inspired low-dimensional parameterization that represents the high dimension of the high-fidelity spatial domain of the fluid model (i.e. state of snapshots). POD is closely related to the principal component analysis (PCA) method in statistics and was first used for turbulent flows by [13].

In this project, the dimensionality reduction aspect of our methodology is set within a NIROM framework that involved computing the POD basis vectors (via PCA) using the snapshots of the input data [17].

Consider a three-dimensional field $\boldsymbol{\omega}$, which is dependent on some input parameter and varies in space and time. We can define its function as $\boldsymbol{\omega} : \mathcal{X} \times \mathcal{T} \times \zeta \rightarrow \mathbb{R}$ where $\mathcal{X}$ is the spatial domain, $\mathcal{T}$ is the time domain, and an input domain ζ of initial parameters/condition. The aim of data-driven/non-intrusive dimensionality reduction is to find an approximate model for $\boldsymbol{\omega}$ from the data

$$\mathcal{D} \subset \{\omega(\boldsymbol{x}, t, \boldsymbol{z}) \mid \boldsymbol{x} \in \mathcal{X}, t \in \mathcal{T}, \boldsymbol{z} \in \zeta\} \tag{1}$$

which, in this case, are snapshots in time of the field. The desired approximate model of the field can be expressed as a linear expansion in the POD basis. This POD basis would be computed from many snapshots data developed as solutions of a high-fidelity model that describes the field. To compute the POD basis, we consider a snapshot data to be $\boldsymbol{\omega}(t; \boldsymbol{z}) \in \mathbb{R}^{n_x}$ where n_x is the dimension of the spatial domain (from finite discretization). Thus, the set $\{\boldsymbol{\omega}(t_i; \boldsymbol{z}_j) \mid i = 1, \cdots, n_t;\ j = 1, \cdots, n_z\}$ of snapshots at n_t different time levels/steps of $t_1,\ t_2,\ \cdots,\ t_{n_t} \in \mathcal{T}$ and n_z different initial input conditions of $\boldsymbol{z_1}, \boldsymbol{z_2}, \cdots, \boldsymbol{z_{n_z}} \in \zeta$ comprises of $n_s = n_t n_z$ snapshots. The snapshot matrix can be defined as $\boldsymbol{S} \in \mathbb{R}^{n_x \times n_s}$ with each row corresponding to a spatial location and each column representing a snapshot in the set. At this stage, PCA can then be introduced for dimensionality reduction.PCA seeks a transformation $\boldsymbol{T}$ that maps each vector $\{\boldsymbol{\omega}(t_i; \boldsymbol{z}_j)$ in $\boldsymbol{S}$ (i.e each snapshot) from the original dimensional space of n_x to a new space that only keeps the first r principal components using the first r eigenvectors of the transformed matrix [10].

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_61

The idea is to maximize the variance of the original data while minimising the total least squared errors in the representation of the snapshots. The size of the POD basis/principal component r is chosen by specifying a tolerance in this error calculation. This user-specified tolerance, k also indicates how much information/energy of the data is captured by the resulting snapshot representation. We chose r such that:

$$\frac{\sum_{k=1}^{r} \sigma_k^2}{\sum_{k=1}^{n_s} \sigma_k^2} > k, k = 0.999 \tag{2}$$

This means given a snapshot field we can compute its original state, using the POD coefficients, with 99.9% reconstruction accuracy. Hence, once the dimension reduction is completed, the POD expansion coefficients $\boldsymbol{\theta_{k=1,\cdots,r}(t;\ z)}$ denote the model approximation and parameterization of a snapshot field $\boldsymbol{\omega}(t;\ \boldsymbol{z})$ at time t and input conditions $\boldsymbol{z}$. The coefficients are then employed in the training of generative models for the time series prediction. Results of the POD-based compression are shown and discussed in section 3.2.

2.2 Generative Models

Generative modelling is the process of training a machine learning model with specific data to produce 'fake' data from a distribution that mimics the probability distribution of the original training set. Here, we produce two generative models to perform time series prediction of a turbulent fluid flow. The two models utilized are: a generative adversarial network (GAN)[7] and an adversarial autoencoder (AAE)[14]. The choice of these models was based on their proven successes in the use of nonlinear fluid modelling. In the result section, a comparison between outputs of the two models is presented.

Generative Adversarial Network: A GAN is an artificial learning technology that is composed of two neural networks as shown in Fig. 2.2. GANs have been adopted widely in several research areas, showing huge successes in practical applications including simulating fluid models [5]. The training process is essentially a game between two models competing as adversaries. While the generator module (G) generates fake samples from an input random distribution, a discriminator module (D) tries to distinguish between real samples drawn from the original distribution and the sample output from the generator. D does this by estimating a score which serves as the probability that a particular sample came from the original distribution i.e. $D(G(\boldsymbol{\theta_r})) = 1$. The training process of a GAN is a minimization-maximization problem that is based on a cross-entropy loss function

$$\boldsymbol{J}(D,G) : \min_G \max_D E_{\theta_r \sim p_{data}(\theta_r)}[logD(\theta_r)] + E_{\mathbf{z} \sim p_z(\mathbf{z})}[log(1 - D(G(\theta)))] \tag{3}$$

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_61

where $p_{data}(\theta_r)$ is the probability data distribution of the target output of real samples θ_r and $p_z(z)$ is the prior distribution for the random latent vector $\mathbf{z}$. The training process involves:

- Updating D with gradients that maximize the discriminator function by differentiating with respect to parameters of the discriminator.
- Updating G with gradients that minimize the generator function by differentiating with respect to parameters of the generator.

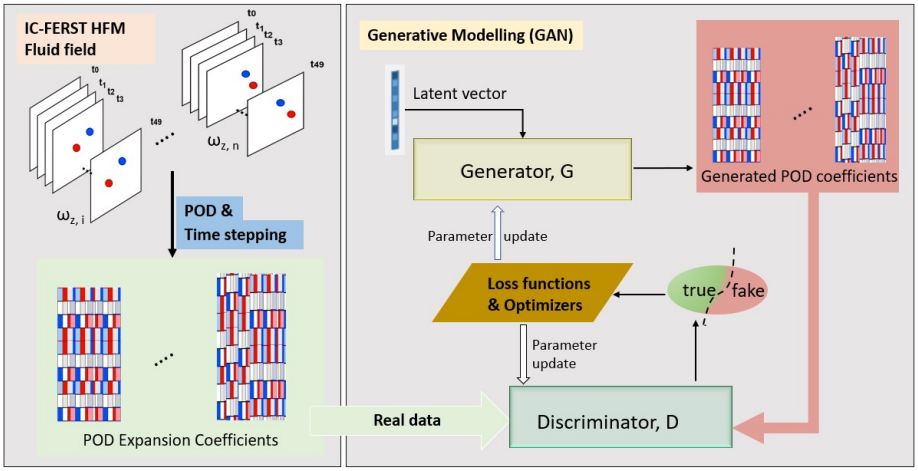

Fig. 1. Generative modelling using GAN. In this workflow, real samples obtained from POD-based NIROM are utilized as training data for the discriminator module of a GAN. Fake data produced by the generator, G from an input latent vector is simultaneously used in the training process, with loss back propagated through both neural network modules.

A common problem in the use of GANs for sample generation is mode collapse. Typically, a GAN is trained to produce a wide variety of outputs that mimic the training data distribution. For example, if a GAN is trained with pictures of different dog breeds, we want a different dog for every random input to the dog generator. However, it is possible that the generator only produces a small set of realistic outputs and learns to generate only that seemingly credible output (or small set of outputs) to the discriminator.

The Wasserstein GAN (WGAN) [1] is a type of GAN that avoids this problem of mode collapse by circumventing the issue of vanishing gradients. This implies that the discriminator is trained to optimality, learning to reject any output/set of outputs the generator tries to stabilize on. The WGAN method introduces a new loss function that alternatively minimizes an approximation of the Earth Mover distance between the distributions completely avoiding mode collapse.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_61

In developing a GAN for the generative modelling of this project, a typical Deep Convolutional GAN (DCGAN) was developed and trained to produce the target output. Following evidence of mode collapse however, an Improved WGAN [9] was also developed by altering the loss functions of the original DCGAN. The WGAN also included a gradient penalty term that led to more diverse output from the generator.

The WGAN loss function uses a Earth mover distance criteria to enforce match of a prior data distribution. The loss function for this type of GAN is given as follows

$$L = \mathbb{E}_{\tilde{\mathbf{x}} \sim \mathbb{P}_g} \left[D\left(\tilde{\mathbf{x}}\right) \right] - \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_r} \left[D\left(\mathbf{x}\right) \right] + \lambda \mathbb{E}_{\hat{\mathbf{x}} \sim \mathbb{P}_{\hat{\mathbf{x}}}} \left[\left(||\nabla_{\tilde{\mathbf{x}}} D\left(\tilde{\mathbf{x}}\right)||_2 - 1 \right)^2 \right] \tag{4}$$

where the second term is a gradient penalty that replaces weight clipping to achieve Lipschitz continuity (gradient with norm at most 1 everywhere). The discriminator in this GAN works as a critic.

The generative model (GAN and/or WGAN) developed and trained using the presented workflow can be used for time-series/forward prediction without any changes to its structure. This is also the case when the model is utilized for the assimilation of given observation/sensor data [16].

Adversarial Autoencoder: A second type of generative model built and implemented in this project is an AAE (Fig. 2.2). Similar to a GAN, the AAE was proposed as a generative model that seeks to match an aggregated posterior distribution of its hidden latent vector with a prior distribution. To be able to function as a deep generative model, the AAE is trained to perform variational inference that enables its decoder to learn a statistical model that maps between the imposed prior and the data distribution. The AAE has a wide range of applications including semi-supervised classification, unsupervised clustering and data visualization [14]. In the field of computational fluid dynamics, Cheng et. al [4] studied the capability of an advanced deep-AAE for parameterizing nonlinear fluid flow and utilized it in the prediction of a water collapse test case. Here, we develop an AAE and test it for prediction of nonlinear turbulent flow.

2.3 Space-time predictions using generative models

The goal of this project is to show that generative models such as GANs and AAEs can be utilized for the time-series prediction of nonlinear turbulent fluid models. This section discusses the time-series prediction and an algorithm for its implementation. The methodology proposed by [16] is further tested on a two-dimensional turbulent fluid model to obtain a surrogate model that is accurate and computationally cheap. The next subsections discuss the this method and its components.

Prediction using GANs: The ability of a GAN to produce realistic samples that seem to belong to a prior distribution is leveraged in this project. To predict

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_61

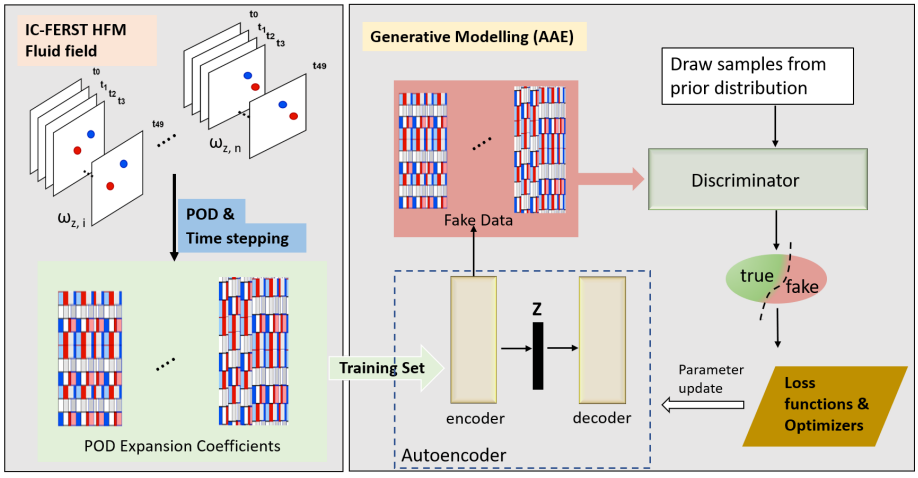

Fig. 2. Generative modelling using AAE. The training process of this workflow attempts to match output of the autoencoder with a prior distribution. While the encoder generates fake samples that matches this distribution, the discriminator attempts to critic against the generated samples.

forward in time, an algorithm, Predictive GAN (PredGAN) algorithm [16] is implemented in this project on two-dimensional turbulent flow data. The PredGAN algorithm begins with training a GAN to generate a data sequence of $p+1$ time levels from an input latent vector. To achieve this, the GAN is trained with $p+1$ consecutive time levels of compressed variables/POD coefficients concatenated to form a trajectory. Once the training is completed, the generator of the GAN is capable of producing fake snapshots at multiple time levels, $n-p$ to n where $n \geq p$. In order to complete prediction with the trained GAN, the first p time levels of a known trajectory/given solution is matched with corresponding time levels of the output of the GAN through loss optimization. Once convergence has been reached, the additional time step $p+1$, in the output of the generator serves as the forward prediction of the trajectory. This process can be repeated by using the predicted $p+1$ solution as a known solution while similar optimization is carried out to predict time level $p+2$. Ultimately, all time steps can be predicted by replicating the process and obtaining a new time step for each iteration of the PredGAN algorithm.

Prediction using Adversarial Autoencoder: To predict with an autoencoder, the following steps were followed:

1. Since the autoencoder does not require a latent variable as input, we use the first $p-1$ time levels of the known solution as input.
2. To predict forward, the $p-1$ time level is used as an initial guess for the desired p time level. and passed into the autoencoder to give a prediction.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_61

Following successful training, the autoencoder then attempts to match the true snapshot at time p from the input initial guess.
3. The output prediction from a single iteration through the autoencoder is further re-used as input guess for the time level p and the time series is passed through the autoencoder till convergence is reached.
4. The final output is the snapshot prediction at time p.
5. For multiple time level predictions, the process is repeated from steps 1- 4 using the last p time levels as initial guesses for subsequent time levels.

3 Implementation and Results

3.1 Case study: Parameter-varying flow in a periodic box

In order to train a GAN capable of time-series prediction of the two-dimensional turbulent fluid problem, a dataset comprising two velocities component (x, y) and the pressure for each discretized node of a two-dimensional incompressible Navier-Stokes simulations has been obtained. Fig. 3.1 shows the spatial properties of each snapshot. This dataset represents a parameter-varying flow in a fixed-wall box. Given that the convolutional layers of a neural network are de-

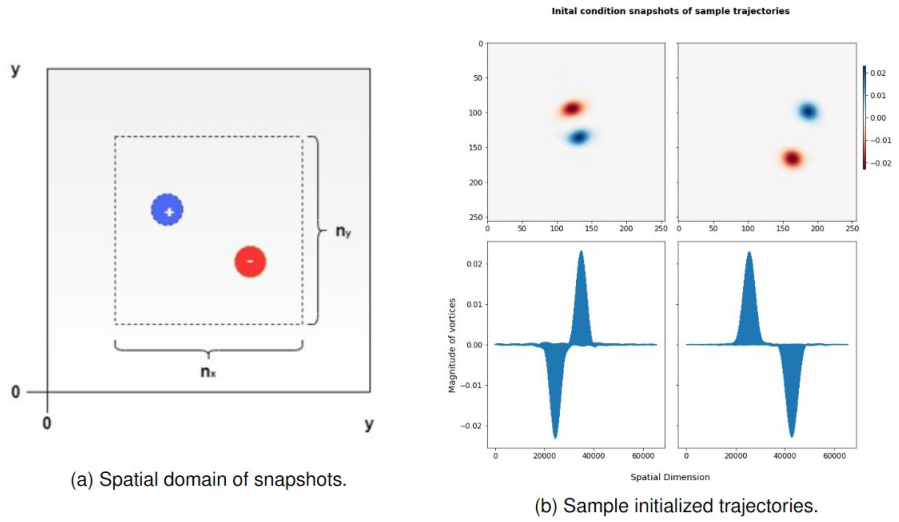

Fig. 3. Two-dimensional Gaussian vortices in a square domain. The positive and negative vortices are of equal strength and each snapshot $S \in \mathbb{R}^{y \times y}$ where $y = 256$, are randomly initialized within a predefined subdomain $n_x \times n_y$. The images on the bottom right show the magnitude of the vortices projected over a 1-D domain.

signed to detect object/features anywhere in an image, it can be used in this

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_61

project since the large-parameter variations implicit in the dataset generation is of a similar nature as object randomly located in an image [6].

The simulations were run on Imperial College-Finite Element Reservoir Simulator (IC-FERST) using the following criteria: turbulent flow with Re=5000, constant viscosity and no slip walls boundary conditions.The first step in the project is to transform the velocity dataset into vorticity data so it represents initial Gaussian (randomly initialized) vortices that decay due to viscosity changes. Following this transformation, the vorticity data are then compressed by carrying out a POD.

The training set for this project included snapshots from 300 separate trajectories. Snapshots from trajectories were obtained such that each trajectory included 50 snapshots - a total of 15K snapshots. Prior to actual training, the data is prepared for time series prediction by concatenating successive snapshots, 5s apart, into a time series of 7 instances (i.e. each time series represents vortices' evolution over a period of 30s). This sums up to 6K distinct time series - one trajectory can be split into a maximum of 20 time series of 7 snapshots. In predicting with generative models post-training, we were able to forecast 1-3 additional instances (evolution over a period of $\leq$15s) for never before seen time series (30 trajectories). See 3.3 for more details.

3.2 POD compression and order reduction

Each snapshot is no longer a 256 by 256 array but now represented using 292 features (POD coefficients). The cumulative information/energy retained measured using explained variance is over 99.99% as shown in Fig.3.2. A visual comparison of the compression is shown in Fig. 3.2.

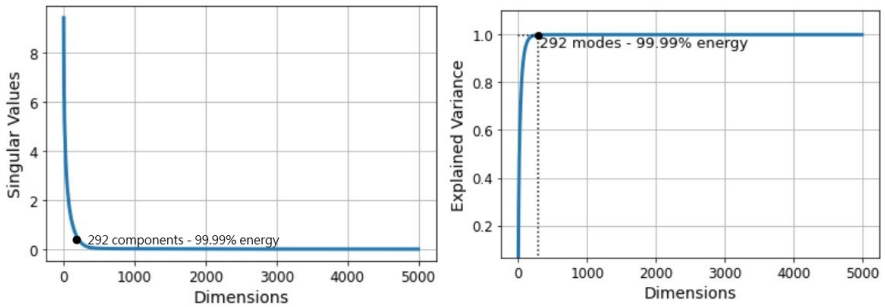

Fig. 4. POD singular values and relative cumulative energy for the two-dimensional Gaussian vorticity filed snapshot set.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_61

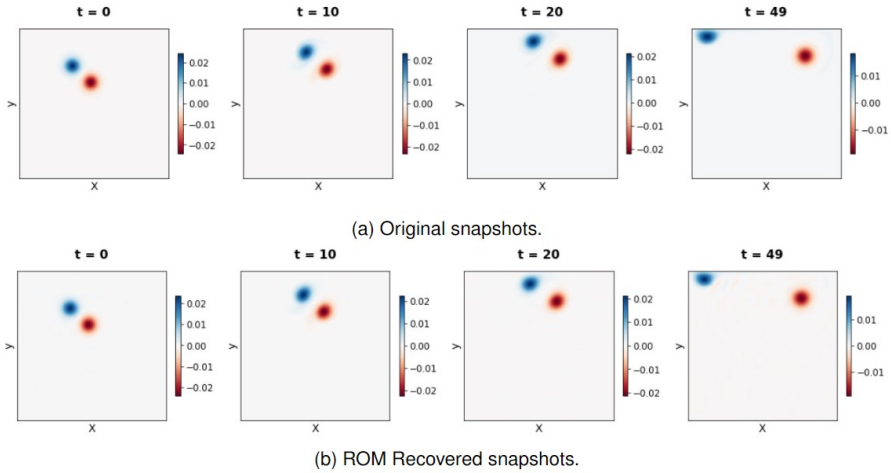

Fig. 5. Original and recovered snapshots following POD-based NIROM. The reduction was specified to retain 99.99% information from the original snapshots. The reduction decreased the dimension from 256-by-256 to 292 POD coefficients.

3.3 Prediction using GAN and AAE

Single time level prediction: In this section, we apply the PredGAN algorithm to a sample trajectory to predict a single time level forward. Following the training of a WGAN-GP and an AAE with 7 time levels from sample trajectories, we proceed to predict a single step forward in the test set using the PredGAN and PredAAE algorithms. In this application, the first 6 time levels (t=0 to t=25) are considered known while the 7th time level (t=30) is the predicted time step. Results of the single time level prediction can be found in Fig.3.3. A sample trajectory (Fig. 3.3a) serves as the input to the generative models (WGAN-GP and AAE). Snapshots of each generative model shows output following convergence of the loss between generated sample and known solution (Fig. 3.3bc). The second row visualizes the mismatch between the magnitude and location of the true data (in blue) and generated prediction (in orange). The vertical axis represents the magnitude of the vortices while the horizontal axis is a one-dimensional projection of the two-dimensional domain. Given these results, AAE is shown to have a better performance both for predicting forward and matching known solutions with samples generated from a random input latent vector.

Multiple time levels prediction: The results from predicting multiple time levels, shown in Fig. 3.3, follows a similar pattern as that of the single time level prediction. Following results for the single time level prediction (t=30), we proceed to predict multiple time levels from t=35 to t=45. It is worth mentioning that this data was not present in the training set. The first row of snapshots (Fig.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_61

Fig. 6. Prediction of one time level (t=30) using WGAN-GP(b) and AAE(c) on a sample trajectory from train dataset.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_61

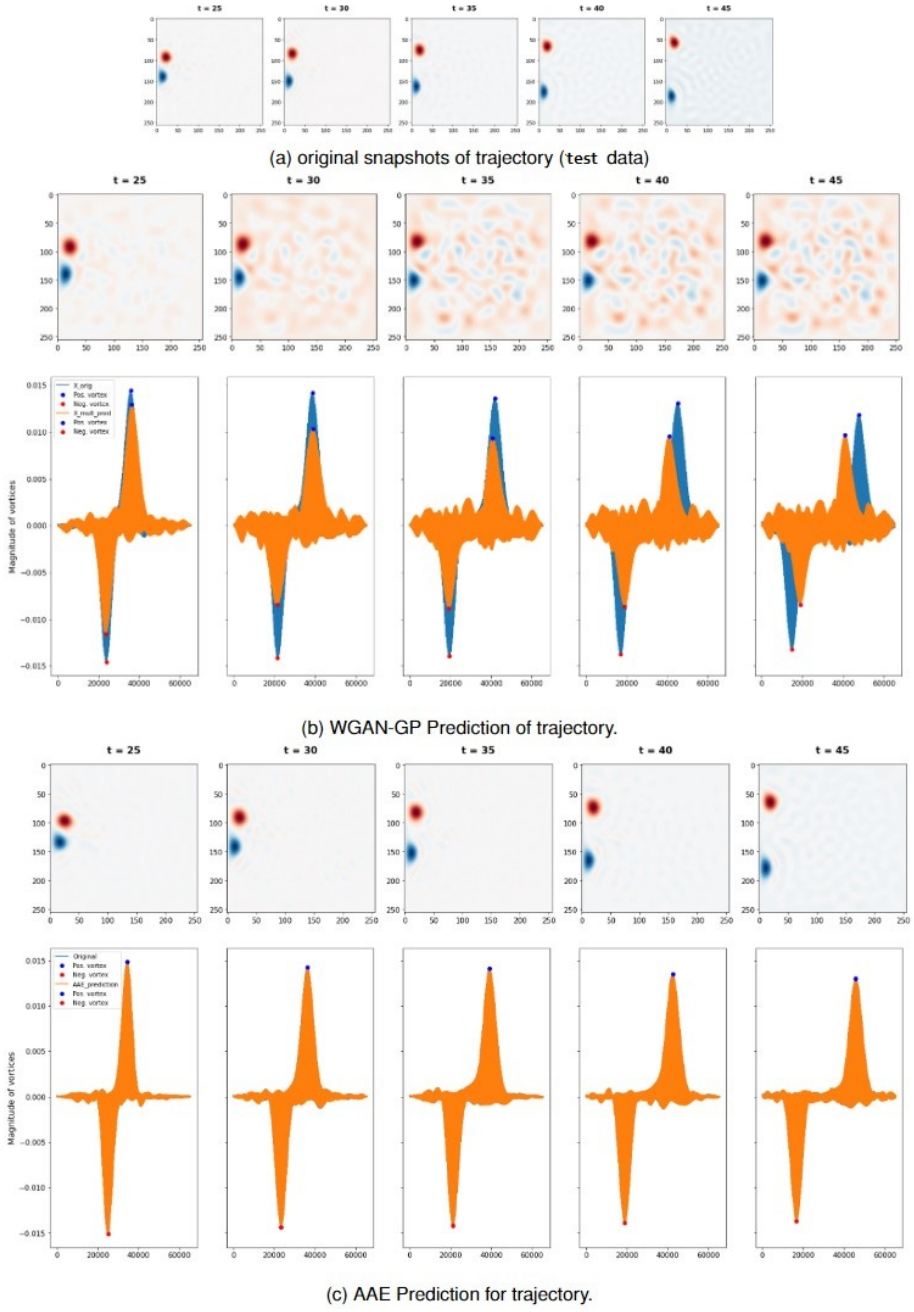

Fig. 7. Multiple time level prediction (t=35 to t=45) using WGAN-GP(b) and AAE(c) on a sample trajectory.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_61

3.3a.) shows the true snapshot of the trajectory at times t=35 to t=45. This is the known/given solution form the high fidelity simulation. Fig. 3.3b. shows predicted output for these time levels using the same WGAN-GP. Here, we see that while the the WGAN-GP is able to predict the spatio-temporal distribution, the prediction ability reduces with forward time. The AAE (Fig. 3.3c), however, shows no such sign.

4 Discussion and Conclusion

The use of machine learning techniques for fluid modelling problems is very promising. The low cost, speedup and relative accuracy provided by machine learning tools, specially generative models, are attractive features in the study of forward modelling. In this project, an exploratory study is done to understand the capabilities of two generative models - generative adversarial network (GAN) and adversarial autoencoder (AAE) - for predicting the evolution in time of a highly nonlinear turbulent fluid flow. We use the capabilities of the generative models within a non-intrusive reduced order model framework. The results demonstrate that both generative models are capable of predicting the evolution of the vortice positions in time, although the AAE has generate more accurate predictions than the WGAN-GP. Furthermore, with the event of mode collapse, we conclude that a 'vanilla' DCGAN may be insufficient for the turbulent flow prediction. We also show that the WGAN-GP and AAE can generalise and generate solutions not present in the training set.

Acknowledgements The authors would like to acknowledge the following EPSRC grants: RELIANT, Risk EvaLuatIon fAst iNtelligent Tool for COVID19 (EP/V036777/1); MAGIC, Managing Air for Green Inner Cities (EP/N010221/1); MUFFINS, MUltiphase Flow-induced Fluid-flexible structure InteractioN in Subsea applications (EP/P033180/1); the PREMIERE programme grant (EP/T000414/1); and INHALE, Health assessment across biological length scales (EP/T003189/1). Most sincere appreciation goes out to the Department of Earth Science and Engineering at Imperial College London for support and resources provided over the period of this project. We would also like to extend gratitude to everyone who was engaged in discussions during the project. Thank you for giving your time and sharing your ideas.

References

1. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein GAN (2017), http://arxiv.org/abs/1701.07875
2. Brunton, S.L., Noack, B.R., Koumoutsakos, P.: Machine Learning for Fluid Mechanics. Annu. Rev. Fluid Mech. **52**, 477–508 (2020). https://doi.org/10.1146/annurev-fluid-010719-060214

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_61

3. Cardos, M., Durlofsky, L., Sarma, P.: Development and application of reduced-order modeling procedures for subsurface flow simulation. International Journal for Numerical Methods in Engineering **77**(9), 1322–1350 (2009). https://doi.org/10.1002/nme, https://doi.org/10.1002/nme.2453
4. Cheng, M., Fang, F., Pain, C.C., Navon, I.M.: An advanced hybrid deep adversarial autoencoder for parameterized nonlinear fluid flow modelling. Comput. Methods Appl. Mech. Eng. **372** (2020). https://doi.org/10.1016/j.cma.2020.113375
5. Cheng, M., Fang, F., Pain, C.C., Navon, I.M.: Data-driven modelling of nonlinear spatio-temporal fluid flows using a deep convolutional generative adversarial network. Comput. Methods Appl. Mech. Eng. **365** (2020)
6. Gonzalez, F.J., Balajewicz, M.: Deep convolutional recurrent autoencoders for learning low-dimensional feature dynamics of fluid systems (2018), http://arxiv.org/abs/1808.01346
7. Goodfellow, I.: NIPS 2016 Tutorial: Generative Adversarial Networks (2016), http://arxiv.org/abs/1701.00160
8. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016), http://www.deeplearningbook.org
9. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.: Improved training of wasserstein GANs. Adv. Neural Inf. Process. Syst. **2017-Decem**, 5768–5778 (2017)
10. Hotelling, H.: Analysis of a complex of statistical variables into principal components. Journal of educational psychology **24**(6), 417 (1933)
11. Kingma, D.P., Welling, M.: Auto-encoding variational bayes (2014)
12. Lecun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015). https://doi.org/10.1038/nature14539
13. Lumley, J.: Atmospheric turbulence and radio wave propagation. Atmospheric Turbulence and Radio Wave Propagation pp. 166–178 (1967), cited By 94
14. Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., Frey, B.: Adversarial Autoencoders (2015), http://arxiv.org/abs/1511.05644
15. 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)
16. Silva, V.L.S., Heaney, C.E., Li, Y., Pain, C.C.: Data Assimilation Predictive GAN (DA-PredGAN): applied to determine the spread of COVID-19 (2021), http://arxiv.org/abs/2105.07729
17. Sirovich, L.: Turbulence and the dynamics of coherent structures. III. Dynamics and scaling. Q. Appl. Math. **45**(3), 583–590 (1987). https://doi.org/10.1090/qam/910464
18. Swischuk, R., Mainini, L., Peherstorfer, B., Willcox, K.: Projection-based model reduction: Formulations for physics-based machine learning. Comput. Fluids **179**, 704–717 (2019). https://doi.org/10.1016/j.compfluid.2018.07.021, https://doi.org/10.1016/j.compfluid.2018.07.021
19. Xiao, D., Fang, F., Buchan, A.G., Pain, C.C., Navon, I.M., Muggeridge, A.: Non-intrusive reduced order modelling of the Navier-Stokes equations. Comput. Methods Appl. Mech. Eng. **293**, 522–541 (2015). https://doi.org/10.1016/j.cma.2015.05.015
20. Xiao, D., Yang, P., Fang, F., Xiang, J., Pain, C.C., Navon, I.M., Chen, M.: A non-intrusive reduced-order model for compressible fluid and fractured solid coupling and its application to blasting. J. Comput. Phys. **330**, 221–244 (2017). https://doi.org/10.1016/j.jcp.2016.10.068

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_61