Imperial College London

Department of Earth Science and Engineering

MSc in Applied Computational Science and Engineering

Independent Research Project
Final Report

Embedding sensor information into Generative Adversarial Networks to determine COVID-19 infection risks in enclosed spaces

by

Jamesson Ipock

jamesson.ipock19@imperial.ac.uk
GitHub login: acse-jsi18

Supervisors:

Dr. Claire E. Heaney

Dr. Laetitia Mottet

Prof. Christopher Pain

August 2021

GitHub repository: https://github.com/acse-2020/acse2020-acse9-finalreport-acse-jsi18

Abstract

Modelling air quality in enclosed spaces, specifically the spread of viral infection, has grown in urgency as infections such as COVID-19 transmit across global populations. Predictive modelling with use of a Generative Adversarial Network (GAN) in a novel approach of Enhanced Training, sees the direct implementation of pre-defined sensor locations in making spatio-temporal predictions of viral laden air. As inference of viral infection distribution we make use of experimental and a computational fluid flow model of CO_2, acting as surrogate for viral air. In this research, models of predicting in time (PredGAN) coupled with data assimilation (DA-PredGAN), are developed effectively mapping the evolution of the high-fidelity flow. This is carried out within a non-intrusive reduced order model (NIROM) framework; coupling the high-accuracy predictions with increased efficiency over computationally expensive numerical simulations.

Keywords:

generative adversarial networks; data assimilation; non-intrusive reduced order model; spatio-temporal predictions; deep learning; COVID-19.

Acknowledgements

I would like to thank and express my upmost appreciation to several individuals who have shown unwavering support throughout this research project. Firstly, I would like to gratefully thank my project supervisors, Dr. Claire E. Heaney, Dr. Laetitia Mottet, and Professor Christopher Pain for their invaluable guidance and reassurance throughout. I must also give my greatest thanks to my friends and fellow peers within the AMCG MSc team – it has been a blast.
Finally, I would also like to show appreciation to my family for their boundless enthusiasm, encouragement, and interminable support.

Table of Contents

1 Introduction

The COVID-19 outbreak which spread across the world has infected greater than 153 million individuals [5], giving rise to an ever-growing urgency for models tackling the importance of indoor air quality. With predictions of an estimated two-thirds of the total population living in urban areas by 2050 [3], the urbanisation habits of the global population, are resulting in increased building densities with low air quality and circulations. This issue of over-crowdedness in dense regions and within building complexes, coupled with both the devastating short term and possible long-lasting effects of COVID-19 [30], expresses a crucial need for models to estimate the spread of viral diseases, further tackling the ever-growing Indoor Air Quality issue.

Mitigation of COVID-19 viral spread has become a dynamic and engrossed area of research with models aiming to study the epidemiological issues and complex scenario that is viral fluid-flow. Modelling the viral spread and further accurately predicting the probability of airborne infection transmissions has therefore become paramount in garnering greater understanding of how to minimise the detrimental effect of viral disease throughout populations.

1.1 Description of Problem

Numerical simulations and Computational Fluid Dynamics (CFD) solvers have been widely used throughout differing specialisms from medical, structural through to environmental engineering, providing powerful and high-fidelity simulations of complex systems. However, these simulations often produce high-fidelity results at the expense of large computational time and resource cost [17].

Machine learning is a branch of artificial intelligence (AI), which enables a software system to systematically improve accuracy of predictions through outcome experience and use of large data automatically. Through the combination of large experimental and simulated datasets, the advancement of algorithms, has seen a rise in the use of Machine Learning techniques. Solving practical problems through statistical frameworks for a given dataset using these techniques can yield impressive results efficiently.

Estimations of the risk of airborne COVID-19 infections within an enclosed space sees the implementation of a computational model through inference of fluid flows.
This project investigates the applicability and possible novel use of Artificial Intelligence (AI) models, specifically Generative Adversarial Networks (GAN), to obtain estimates of the CO_2 distributions within the enclosed space; which acts as a surrogate for virus-laden air. The TensorFlow [1] open-source software library will be utilised to build AI models based on simulation data from Fluidity [2], an open source computational fluid dynamics library. Experimental observations from the MAGIC field campaign [25] will further be assimilated into the AI model.

1.2 Related research

As mentioned previously, modelling complex dynamics can impose heavy computational expense with models consisting of millions of parameters (degrees of freedom) for a single time step solution. This demand for computational speed-up has led to the use of reduced order models (ROMs) [20]. Non-intrusive reduced order models (NIROM) have seen previous success providing a significant speed-up without losing too much accuracy from the original high-fidelity model [12]. The work carried out by Hesthaven et al. [10] employed the use of proper orthogonal decomposition (POD) with multi-layer perceptrons. The method showed suitability for parametrised steady-state partial differential equations, with the benefit of no alterations to the high-fidelity model required.
NIROM has seen its integration with neural networks [10, 28, 12] within their offline-online framework. Bidirectional Long Short-term memory networks (BDLSTM) has been one example which has been made effective in numerous applications from phenome classification and recognition [8], text classifi-

cation [14], through to urban air pollution modelling [19]. Research fulfilled by Quilodrán-Casas et al. [20] made use of two methodologies in BDLSTM and Predictive Generative Adversarial Networks (GAN), for modelling the spatial and temporal spread of COVID-19. Success of predictions made by the predictive GAN, was evaluated to outperform the data-corrected BDLSTM, deeming the use of GANs to show promise for time-series prediction.

Generative adversarial networks (GAN) first proposed by Ian Goodfellow and colleagues [7], describes a machine learning framework where two models, generator(G) and discriminator (D), are respectively trained to capture data distribution and estimate probability of the input being a real sample. The corresponding neural networks 'compete' in a zero-sum game with the objective to generate new, synthetic instances of real data [4]. GANs have since shown effective application across diverse fields including synthetic medical image augmentation [6], sketch to image generation [15], domain adaptation [22], and more.

Research carried out by Silva et al. [23] proposed novel use of GANs in the development of algorithms that (1) make predictions in time (PredGAN), and (2) assimilate observations (DA-PredGAN). The techniques developed enabled the accurate prediction of the evolution of high-fidelity numerical simulations, efficiently assimilating observed data and determining the corresponding model parameters. As a result the proposed algorithms proved successful in its application to predict the spread of COVID-19 in an idealised town.
These algorithms will thereby form the foundation of what will be expanded upon in this project through the novel application of an enhanced GAN training depicting CO_2 spread in an enclosed space.

1.3 Project Objectives

The framework of the proposed model integrates the use Generative Adversarial Networks (GANs) to make spatio-temporal predictions. Upholding the objective of reducing computational expense, the algorithm is set within a NIROM framework. The framework reduces the degrees of freedom the GAN is trained and operates over, approximating a high-dimensional system with a lower-dimensional space; whilst retaining reasonable accuracy of the high-fidelity simulation [10]. Through successful reduction of the snapshots in the lower decomposed space, the GAN is trained to learn the evolution of the numerical simulation and resulting in a surrogate model of the high-fidelity simulation.

Using GANs within this NIROM framework to capture the low-dimensional high-fidelity data distribution, spatio-temporal fluid flow predictions can be obtained by a method of Predictive GANs outlined by Silva et al. [23]. The developed models will see to predict CO_2 and flow field POD coefficients, whilst formulating the CO_2 levels at pre-defined sensor locations.
Novelty of this research project remains in the formation of enhanced training in using GANs to predict in time the quasi non-cyclical fluid flow modelled by numerical simulations within the enclosed space. Enhanced training embeds the information from sensor locations within the GAN avoiding the need to reconstruct the primitive variables at said locations, and thereby compounding efficiency gains. In addition, predictions in time can further be coupled with data assimilation (DA-PredGAN), exploting the inherent adjoint-like properties of generative models further simulating both forward and back in time [23]. With this method of DA, inclusions of observations can become a corrective measure for the model, interpolating and reducing loss discrepancy at known solutions without requiring further computation of the high fidelity model.

2 System Description

2.1 Non-Intrusive Order Model, NIROM

Efficiency of a NIROM is gained through an offline & online scheme. The offline stage refers to the attainment of basis functions. At this stage the forward model is run, the results of which represent the behaviour of the system. In this project using the linear method of proper orthogonal decomposition (POD) will be carried out for compression.
Further, the online stage enables the training in time and thereby predictions in time to occur.

Overview the process is depicted in Figure 1 and described as follows:

Offline computation stage:

(a) Solve the full model generating snapshots producing solutions at different time levels

(b) Obtain the reduced basis functions through the use of proper orthogonal decomposition (POD)

(c) Produce reduced data that will be used in the training process

(d) Train a neural network with the data produced above

Online computation stage:

(a) Choosing arbitrary time range for prediction N_t

(b) Use a Neural Network for predictions of the POD coefficients

(c) Map back from reduced space to physical space

(d) Update the domain according to the values obtained

The process of NIROM ensures several advantages including holding generality, non-intrusiveness [29] and providing a low-dimensional representation of a highly dimensional system for accurate and orders of magnitude faster computations. This, with the added advantage of legacy or complex source code requiring no need for modifications; expresses a method appropriate for general computational speed-up and further for this case of spatio-temporal variation modelling.
In this project, minimal loss of accuracy from the original high-fidelity CFD model, whilst enabling more real-time computation adheres to this system for the desired research output.

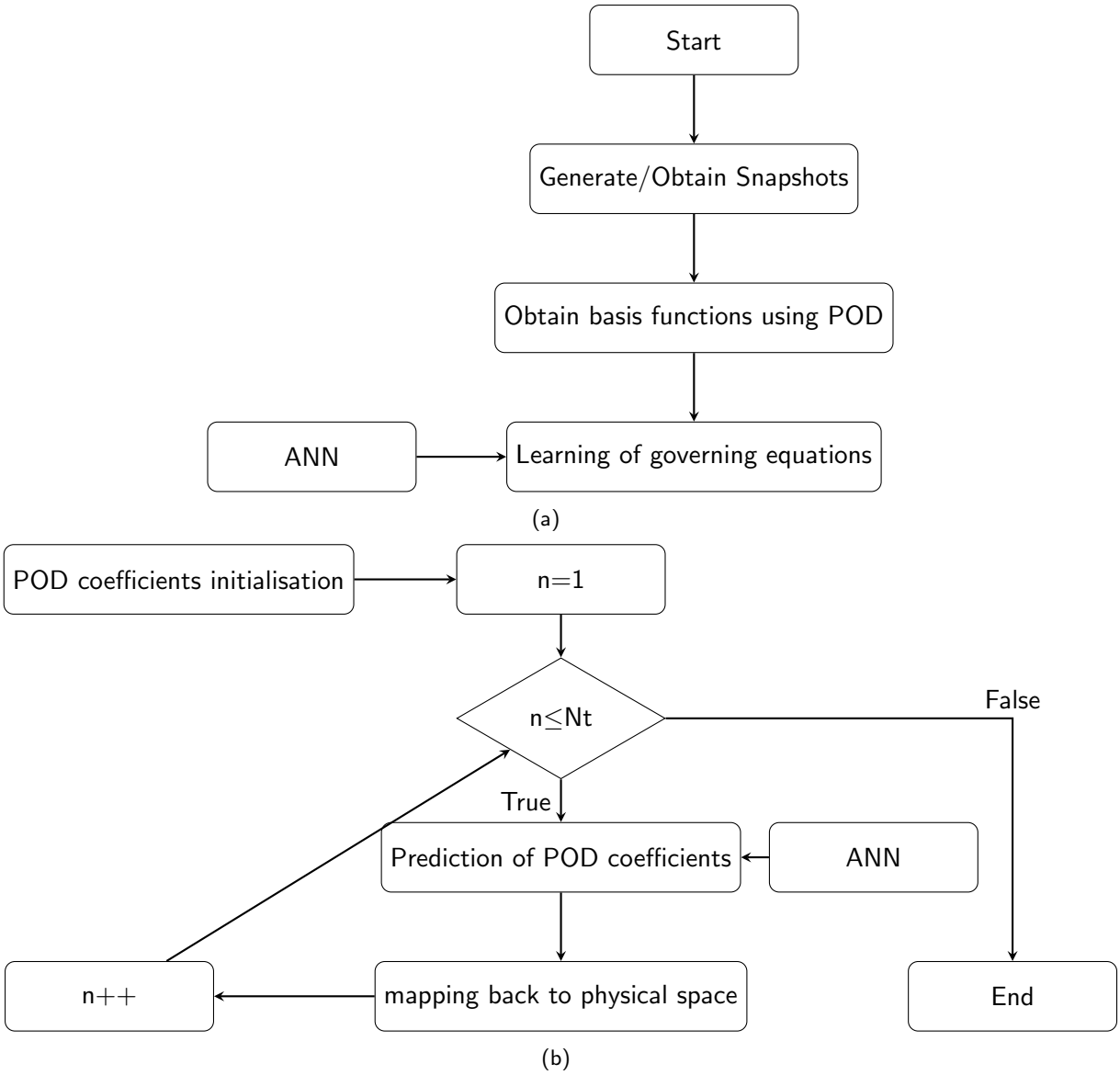


Figure 1: Flow chart of the NIROM (a) Offline, (b) Online stages.

2.2 Enhanced Training

As mentioned within the project objectives section, whilst obtaining the POD coefficients which encompass the CO_2 and flow fields throughout the domain, the GAN is further used to predict observations directly at specified sensor locations.
With this in mind, the neural network takes as input embedded sensor location information coupled with the POD coefficients; learning the input distribution during training. The Adversarial Neural Network, is therefore able to generate outputs in time containing a sample from the concatenated distribution map. Although this embedding of sensor observations process modifies the reduced order NIROM framework, it holds the principality of increasing efficiency through two main aspects:

(1) Does not require the reconstruction of the primitive variables from the POD coefficients, to attain the values at specified sensor locations.

(2) Enables the direct assimilation of data, without the need of additional simulations from the high-fidelity model.

2.3 Clarence Centre enclosed space

The devised and implemented predictive models within this project are developed to encapsulate the progression of CO_2 in time within a well described enclosed space. The results of these algorithms presented in section 4 are applied to a well-ventilated room located at the top floor of the Clarence Centre building, London, UK [3]. The geometry of the room contains three windows: one window facing onto a courtyard, with the other two on the opposing end directly overlooking onto a busy London Road (Figure 2).

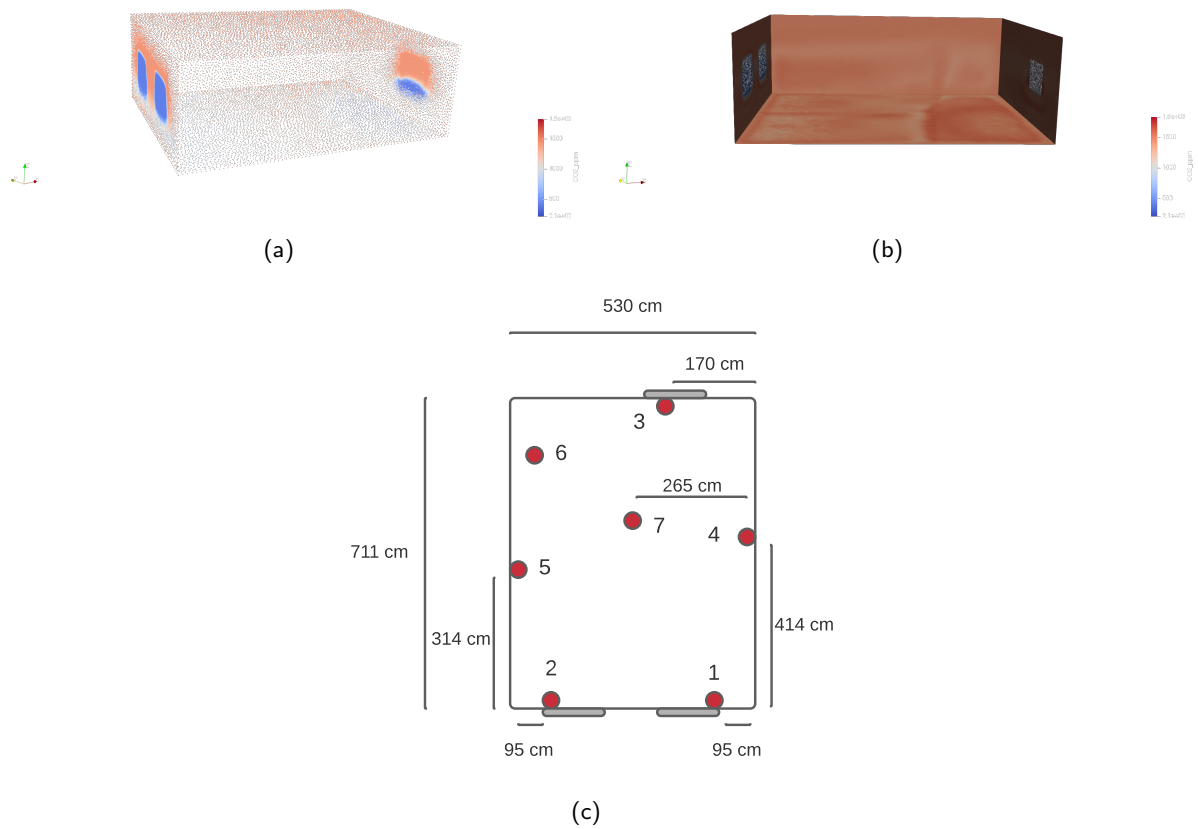


Figure 2: Depiction of the enclosed space geomtery containing three windows (two on one side and one of the opposing side). (a) Points plot of the encapsulated space. (b) Visualisation of three walls and the floor. (c) Schematics of the room with highlighted sensor locations as the red dots.

The Computational Fluid Dynamics (CFD) software, Fluidity, was used by Dr. Laetitia Mottet to generate numerical solutions that are used within this project to train the Generative Adversarial Neural Network. The same simulation has previously been used in a paper by Amendola et al. [3] which reflects the computational flow representation of the MAGIC field campaign [25].
Setup conditions such as the cross-wind, initial and boundary conditions of the CFD simulation reflected the experimental conditions at the time of the field study. This project therefore sees applicability in the training of the GAN with use of this simulated data.
The computational domain is an unstructured mesh composed of 148,906 nodes, with initial CO_2 conditions of 1420ppm and 400ppm indoor and outdoor concentrations respectively. The experimental study recorded CO_2 concentration, temperature and other parameters with seven sensors spread throughout the room. In this research project, the parameter of interest remains the CO_2 and its respective flow in velocities.

2.4 Code metadata

Implementation of the predictive models and associated scripts followed an Agile methodology framework due to the highly iterative process of GAN training.
The programming language this project utilises is Python 3.5 (or higher) [26]. This decision was due to several factors, however primarily to the applicability of relevant Machine Learning libraries available; making this the most widely used language in this domain. The system was developed in a Windows 10 environment, with development carried out on the *Google Colab* platform. *Jupyter Notebook* [13], and specifically use of *Google Colab*, was chosen as the platform for development as it enabled ease of use and access to relevant package installations, linking of scripts, but principally due to the significant speed-up obtained with use of the GPU's. Additionally, the notebooks lends itself better than console outputs for graphical pre and post-processing.
The computational solution is self-contained within several *jupyter notebooks*, each pertaining to carry out a different functionality or variation of predictive models.

Dependencies/Libraries:

- Numpy (1.19.2) [9]
- Scipy (1.5.2) [27]
- TensorFlow ($\geq$2.0) [1]
- Matplotlib (3.3.2) [11]
- Scikit-learn ($\geq$0.24.2) [18]

Amongst several python packages detailed above, the notebooks made use of functionality from the open source fluid dynamics library Fluidity [2]. Additionally, the methodology presented throughout the thesis is an extension of that first devised by Silva et al. [23]. With this, the code foundations which was extended upon can be found at https://github.com/viluiz/gan.git.

Version: 1.1
Github repository & Documentation:
https://github.com/acse-2020/acse2020-acse9-finalreport-acse-jsi18

3 System Design

3.1 Dimensionality Reduction using POD

Proper Orthogonal Decomposition (POD), or Principal Component Analysis (PCA), is a linear method of dimensionality reduction based on singular value decomposition [12]. It expands a solution in terms of coefficients and orthogonal basis functions whereby the representation error is minimised.
For spatio-temporal inferences and how the evolution of these parameters in relation to viral laden air evolve in time, both the CO_2 and velocity fields are abstracted and used.

POD expansion of the CO_2 field u:

$$\boldsymbol{u} = \overline{\boldsymbol{u}} + \boldsymbol{R}\boldsymbol{\alpha} \tag{1}$$

with $\overline{\boldsymbol{u}}$ referring to the mean of the field, $\boldsymbol{R}$ the POD basis functions and $\boldsymbol{\alpha}$ the POD coefficient.

The snapshot matrix, $\boldsymbol{X}$ thereby takes the form:

$$\boldsymbol{X} = [\boldsymbol{u}^{t_1} - \overline{\boldsymbol{u}}, \boldsymbol{u}^{t_2} - \overline{\boldsymbol{u}}, \ldots, \boldsymbol{u}^{t_{N^s}} - \overline{\boldsymbol{u}}] \tag{2}$$

where $\boldsymbol{u}^{t_i}$ is the snapshot solution at a particular time, i. N^s refers to the total number of known time levels.

We then apply SVD to the snapshot matrix, $\boldsymbol{X}$, to obtain the POD basis functions:

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\boldsymbol{T}} \tag{3}$$

with the snapshot matrix, $\boldsymbol{X} \in \mathbb{R}^{N \times N^s}$, N representing the total number of nodes within the domain. Through this decomposition, $\boldsymbol{U}$ contains the POD basis functions, $\boldsymbol{\Sigma}$ the ordered singular values which thereby gives importance to the columns of $\boldsymbol{U}$. In reducing dimensionality, we define the POD basis matrix $\boldsymbol{R}$ as the first $M \leq N^s \ll N$, i.e. $\boldsymbol{R} \in \mathbb{R}^{N \mathsf{X} M}$.

This enables the neural network to learn the parameter dependence and evolution of the POD coefficients, α, of the snapshots in time.

3.2 Time series predictions using GANs

This project utilises a GAN for predictions in time with an algorithm first developed by Silva et al. [23] referred to as Predictive GAN (Pred-GAN).

The method sees the GAN trained to produce data in a consecutive sequence of m+1 time steps, giving the output of the Generator to be:

$$\boldsymbol{G}(\boldsymbol{z}) = \Phi = \begin{pmatrix} \boldsymbol{\alpha}^{t_{n+2}}, \boldsymbol{\mu}^{t_{n+2}} \\ \boldsymbol{\alpha}^{t_{n+1}}, \boldsymbol{\mu}^{t_{n+1}} \\ \boldsymbol{\alpha}^{t_n}, \boldsymbol{\mu}^{t_n} \end{pmatrix} \tag{4}$$

where z is a given vector of random latent variables, producing POD coefficient solutions, $\boldsymbol{\alpha}$, at three (in the given example above) successive time levels, $\boldsymbol{\alpha}^{t_n}$, $\boldsymbol{\alpha}^{t_{n+1}}$, $\boldsymbol{\alpha}^{t_{n+2}}$ and through enhanced training the sensor location observations, $\boldsymbol{\mu}^{t_n}$, $\boldsymbol{\mu}^{t_{n+1}}$, $\boldsymbol{\mu}^{t_{n+2}}$.
Through knowledge of solutions at m consecutive time steps (from simulation data) which act as initial conditions, $\boldsymbol{\alpha}^{t_k}$, $\boldsymbol{\alpha}^{t_{k+1}}$ that we intend to predict forward in time, an optimisation step is required. The optimisation loop minimises the difference between the known values and the generator predicted values with the loss function:

$$\boldsymbol{L_p}(\boldsymbol{z^n}) = \sum_{k=n-m}^{n-1} (\tilde{\alpha}^k - \alpha^k)^T W_\alpha (\tilde{\alpha}^k - \alpha^k) + \sum_{k=n-m}^{n-1} \zeta_\mu (\tilde{\mu}^k - \mu^k)^T W_\mu (\tilde{\mu}^k - \mu^k) \tag{5}$$

where W_α is a square matrix of size N_{PCA}, with diagonal entries equal to the weights representing the PCA coefficient importance. W_μ, a square matrix of size N_μ, ζ_μ a scalar representing embedded sensor value importance, $\tilde{\alpha}^k$ the PCA coefficients and the embedded sensor values over all time steps, $\tilde{\mu}^k$ [23]. In implementation, the importance weights, W_α & W_μ, may be altered for appropriate training.

The loss function L_p's gradient is determined with regards to the input latent values, z^n, which incurs a process of back-propagation. It's implementation however can be carried out directly through the use of Tensorflow [1] functionality.
With this, the L_p is minimised leading to an optimisation step, producing an updated (improved) set of latent variables. This in turn refines the latent values until the convergence criteria, number of epochs is met. Once converged, the next time step prediction is determined, with it, being included as a known solution: $\tilde{\alpha}^k = \alpha^k$.

Stepping through in time creating full pass predictions (over the complete input time levels), continues to follow this process of predictive optimisation steps, resulting in realistic and an interpolated model.

3.3 Data Assimilation in time using GANs

Data assimilation is a technique which incorporates information from observed data combined with mathematical models. It is defined as an inverse problem whereby this project will use the adjoint nature of GANs to evaluate and predict the observations at sensor locations with time. The algorithm formulating the data assimilation time predictions with GANs (DA-PredGAN) was proposed by Silva et al. [23], which identifies three modifications to the previous Predictive GAN method [24]:

1. Inclusion of an additional term within the loss function, to represent the difference between the predicted and observed values.
2. The inputs are no longer defined as a priori, due to the DA process attempting to determine the sensor location values, μ^k.
3. Both forward and backward marching in time takes place.

The functional for the optimisation that occurs with each iteration of both the forward and backward march for the DA is depicted as:

$$\begin{aligned} \boldsymbol{L_{\mathrm{DA}}}(\boldsymbol{z^n}) = \sum_k (\tilde{\alpha}^k - \alpha^k)^T W_\alpha (\tilde{\alpha}^k - \alpha^k) + \sum_k \zeta_\mu (\tilde{\mu}^k - \mu^k)^T W_\mu (\tilde{\mu}^k - \mu^k) \\ + \sum_k \zeta_{\mathrm{obs}} (\mu^k - {u^{\mathrm{obs}}}^k)^T {W_u}^k (\mu^k - {u^{\mathrm{obs}}}^k) \end{aligned} \tag{6}$$

where ${W_u}^k$ represents the weights of the observed data, whilst ζ_{obs} associates the importance of the given data mismatch. Further, between the forward and backward marches, the loss functional remains the same however acts over differing indices of summation, k. In order to predict the solution at time level $n-1$, we require solutions at time levels $n+m-2, n+m-3, ..., n$. Further, to then predict for time level $n-2$, we use obtained solutions at $n+m-3, n+m-4, ..., n, n-1$, continuing this process until prediction at the first time level is met.
Through this backward march we can then obtain the average data mismatch, between the predicted and observed data at sensor locations, before continuing again for a forward pass. Repetition of stepping forward and then backward in time occurs until either:

(a) Convergence has occured whereby the relaxation factor drops below the set value of 0.05.

(b) Total number of consecutive forward & backward marches reach the maximum.

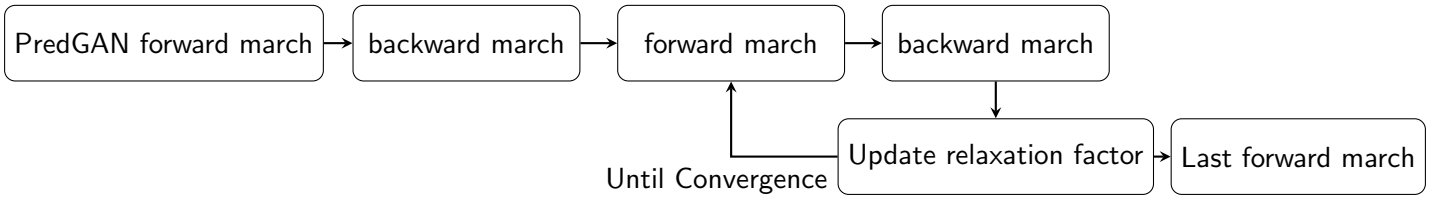


Figure 3: DA-PredGAN process overview.

4 Results & Implementation

In this section the implementation of the NIROM framework will be discussed, their associated results and validation testing shown.

4.1 Preprocessing

As mentioned throughout this report, data from a numerical solution (CFD) model is used as vital input for the training of the GAN. To this end, open source modules from Fluidity [2] were used to extract and obtain relevant velocity and CO_2 fields from the given dataset.

The computational mesh is unstructured with 148,906 nodes per field, resulting in an approximate $\sim$0.6 million with velocity (directions x, y & z), coupled with CO_2 concentration. Traversing through this large number of nodes would be significantly impractical and extremely computationally expensive for GAN training. Further, due to the unstructured domain at which the sensor locations did not exist on any particular nodes, and through alterations of code provided by supervisors - the CO_2 values at sensor locations could be obtained. This was a significant and necessary pre-processing step as the project follows a method of embedding representative (scaled) sensor values for direct prediction from both the PredGAN and DA-PredGAN methods.

4.1.1 Proper Orthogonal Decomposition, POD

In order to uphold the efficiency objective in this project which is set out within the NIROM's framework, the linear dimensionality reduction method of POD is used. To retain appropriate accuracy of the high-fidelity simulation, 99.5% of the cumulative variance was chosen effective and resulted in dimensionality reduction to 43 principal components.

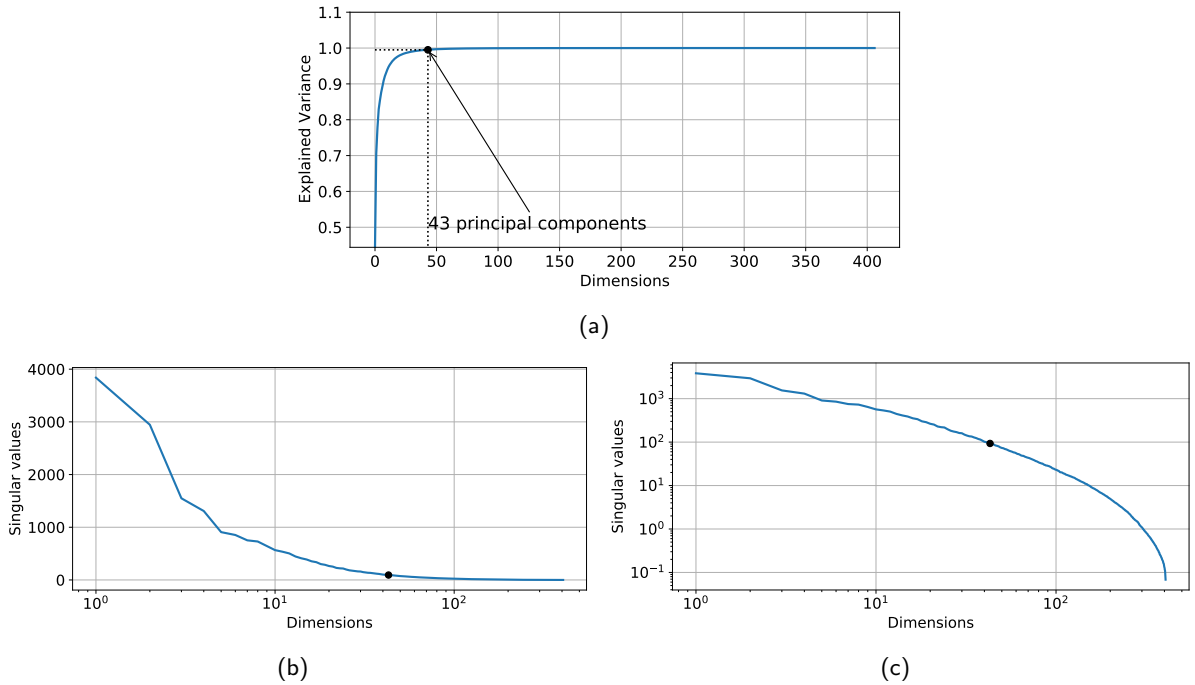


Figure 4: Applied Proper Orthogonal Decomposition to dataset, reducing dimensionality from $\sim$0.6 million (148,906$\times$4) to 43 principal components. (a) Plot of explained variance against dimensions with 43 principal components chosen representing 99.5% of the cumulative variance. Plot of singular value decay (b) in linear scale (c) in logarithmic scale.

4.2 GAN Training

GANs are characteristically extremely hyper-parameter sensitive, requiring both validations and hyper-parameter sweeps to be implemented with a feedback approach. This lent itself to a very modular approach to tuning.
The Adversarial Network architecture chosen is based on a Deep Convolutional GAN (DCGAN) first proposed by Radford et al [21]. DCGANs has previously shown great success in producing low sampling cost yet high performance image generations through learning the target data distribution if trained appropriately. In a similar way, the main objective of this work is to reproduce realistic sample outputs of the high-fidelity CFD model, producing high acccuracy spatio-temporal predictions.

Through the enhanced training, the inputs to the GAN remain as a two-dimensional array of 410 rows by 50 columns. The rows represent the number of time levels used to train the GAN and 50 columns with each time step containing a concatenated 43 POD coefficients $\tilde{\alpha}^k$, with the 7 sensor location values $\tilde{\mu}^k$. The number of 410 time levels used to train the GAN represents an approximate 90-10 percent split from the complete dataset of 455 time steps. The last 45 are therefore further used as a test set, giving forth a possibility to not only validate the model, but evaluate the error that may possibly accumulate through predicting over consecutive time steps beyond seen time.
The Adversarial Network is trained over 30,000 epochs, with an input of noise of dimension 50. The DCGAN generator and discriminator configuration is shown in figure 5.

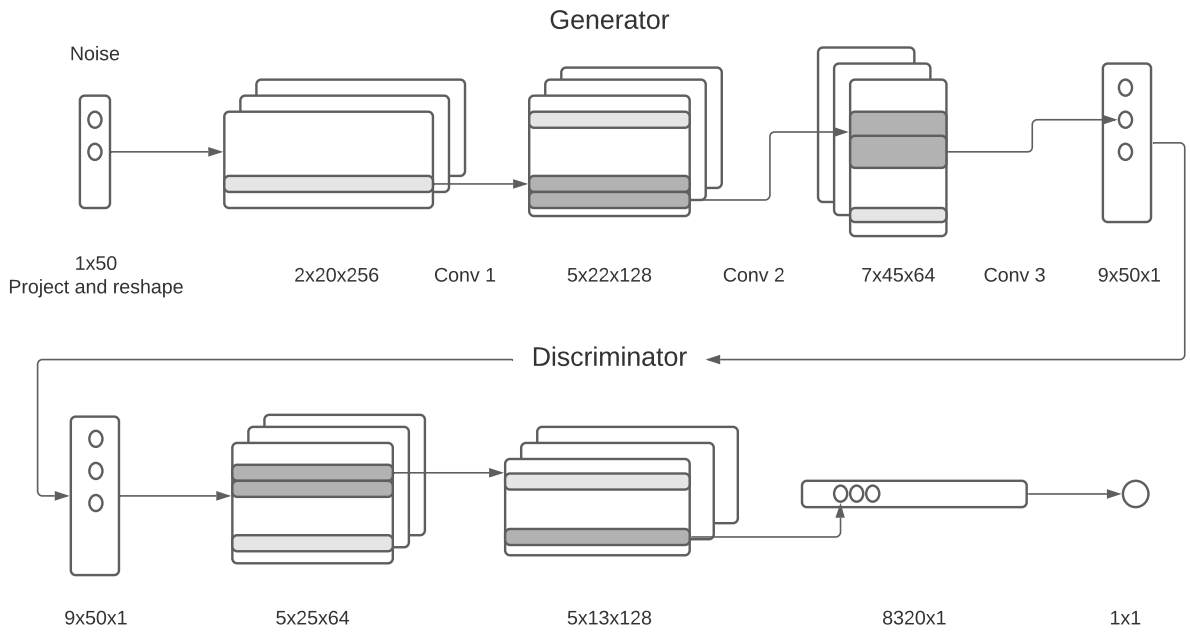


Figure 5: DCGAN Architecture

The DCGAN is a sensitive network to hyper-parameter choices. With such, they are further extremely data specific and require a significant tuning sweep when training to ensure all values within the latent space produce high-fidelity realistic samples. With the above network architecture the hyperparameters as a result of a grid search are as follows:

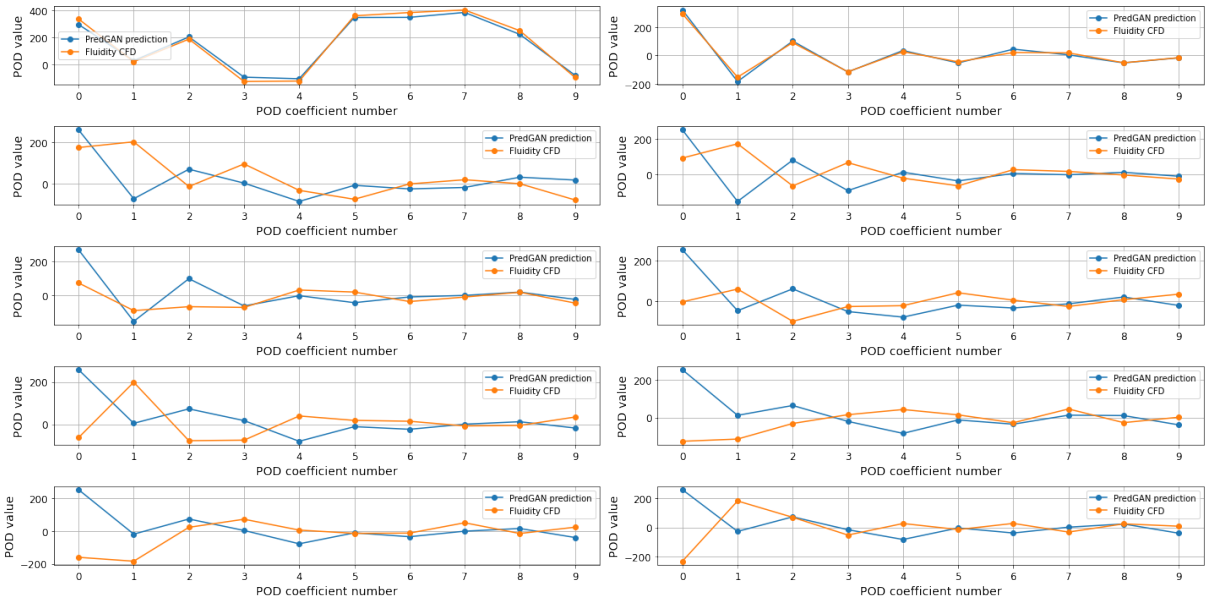
- Activation function: Leaky ReLU ($\alpha = 0.2$), Tanh (final layer).
- Dropout: discriminator (0.3)
- Optimiser: Adam
- Learning rate: generator (2×10^{-3}), discriminator (5×10^{-3})
- Momentum decay: $\beta_1 = 0.7$, $\beta_2 = 0.999$
- Number of epochs: $30,000$
- Batch size: 256
- Latent space: 50

4.3 Predictive-GAN

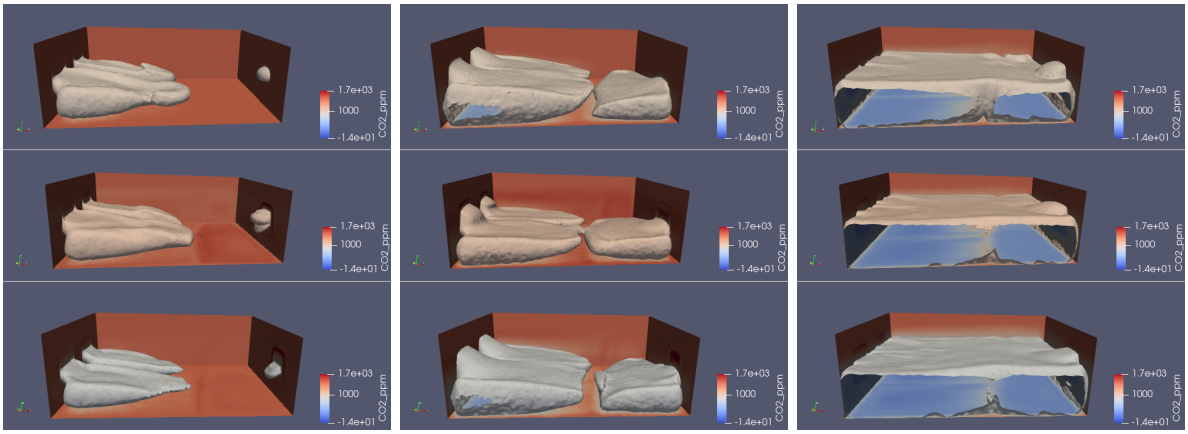
The development and implementation of the PredGAN sees to predict the spatial and temporal spread of COVID-19 within an enclosed environment. As justified previously, this is carried out through evaluating the flow of CO_2; representative of viral laden air.

The model is initiated optimizing the latent space with knowledge of an initial condition taken from the CFD simulation. Through this, we use known solutions from eight time levels to predict the next time step. Next, the tenth prediction is produced traversing the previous eight time levels, including the ninth level prediction just made. This process continues for all time levels, enabling to make predictions up to and beyond the given numerical simulation time steps. In this manner, information from the high-fidelity CFD is not used past the 17^{th} time step.

The figure 6 below, a spatial variation comparison between the PredGAN and the Fluidity CFD model, with the first ten POD coefficients shown. The results are expressed as snapshots in time with every sub-graph representing the 40th time step from the previous (0, 40, 80, etc.). The first ten POD coefficients were chosen for analysis as cumulatively they amount to over 93% of the explained variance in the compressed dataset. The results show that throughout the domain and over the entirety of time steps, the model is able to effectively predict and represent the POD coefficients accurately. With this, contour subplots (b), (c) and (d) further express the successful predictive spatio-temporal evolution across the domain.
For the PredGAN, domain convergence of POD coefficients however does show greater accuracy within the early time-steps and with a slightly increased mean squared difference from time level 120. These implications are as a result of the direct use of initial conditions from the Numerical (CFD) simulation, enabling a direct convergence to the true values early on. As we time-step along, the method aims to find convergence of latent values that minimises the mean square error (MSE) of the previous 8 predicted in forming the next and compounding predictive differences. Although slight perturbations occur at latter time levels, the results significantly highlights the probability distribution map created by the training of the GAN, and the overall success of the PredGAN model in predicting spatial variation throughout.



(a)

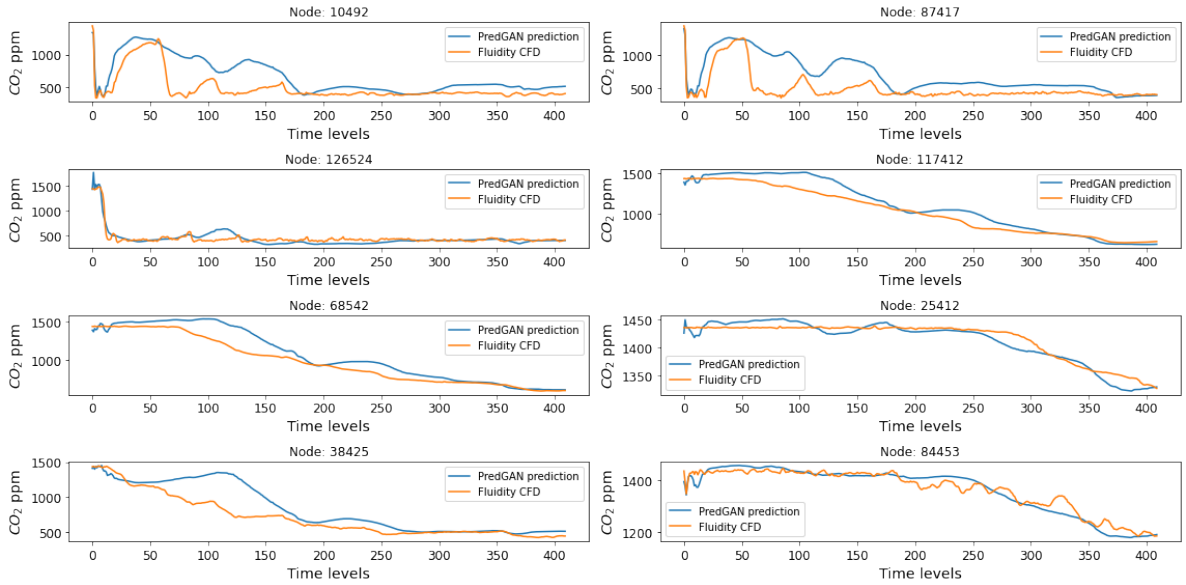


(b) (c) (d)

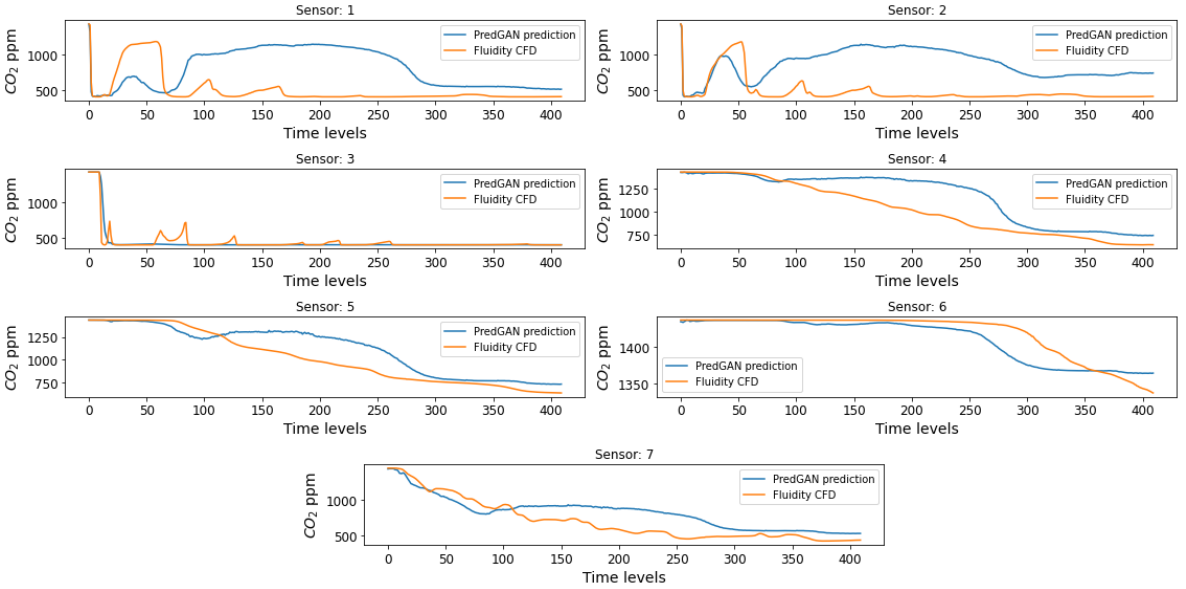
Figure 6: (a) A PredGAN comparison of the first ten pod coefficients against the CFD Fluidity low-dimensional representation over 40 incremental time level intervals.
Isosurface plot comparisons of CO_2 evolution across the enclosed domain showing CFD Fluidity, DA-PredGAN and PredGAN at time levels (b) 10 (∼20s), (c) 80 (∼160s), and (d) 190 (∼380s) respectively. Refer to figure 11 in appendix for more spatio-temporal isosurface plots.

With the project objective of also predicting CO_2 ppm levels at specified sensor locations, figure 7 highlights the PredGAN method's effectiveness across (a) peripheral nodes and (b) sensor locations. Peripheral nodes are described as locations specifically chosen surrounding the pre-defined sensors. As depicted in the graphs the PredGAN is able to effectively and accurately predict in time at surrounding nodes.

Additionally, the temporal plot of figure 7, sub-figure (b), shows progressive and an effective evolution of CO_2 at sensors in time. Despite the overall successful temporal predictions, sensor locations 1 and 2 characterises a comparatively higher MSE from ~ 80 to 275 time levels. Throughout development, hyperparameter tuning and effective testing for overall convergence properties was carried out iteratively. The sensor 1 and 2 difference is as a result of a convergence step occurring to enter and remain at a local minima. This effect in turn has been sought to be minimised through tuning of several parameters: including increasing the momentum parameter at which the optimizer acts, number of epochs and altering optimizer choice whilst also re-evaluating architecture. With this, the overall characteristic of the PredGAN at not only sensor locations but across the domain was evaluated and hyperparameters chosen reducing larger detrimental failure modes.



(a)

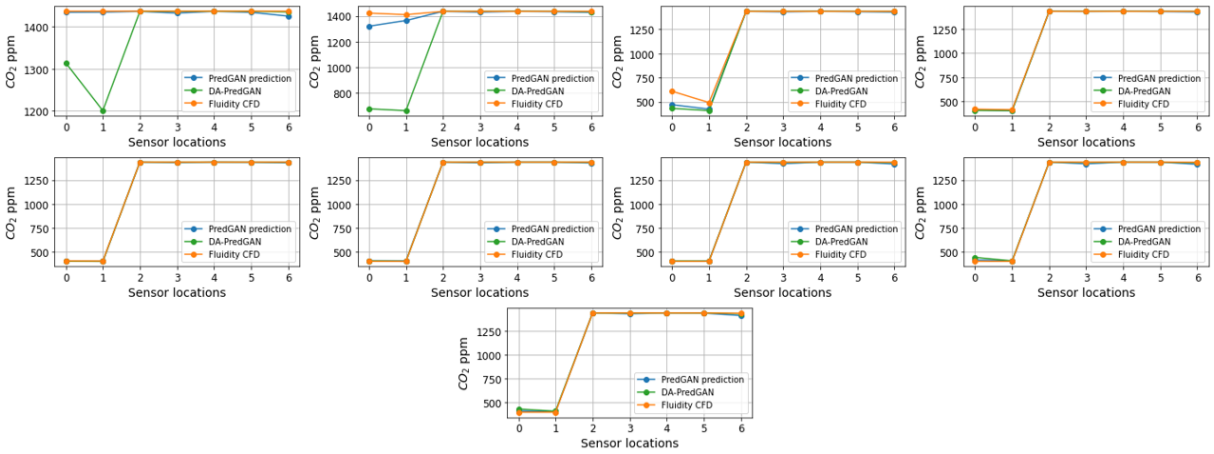


(b)

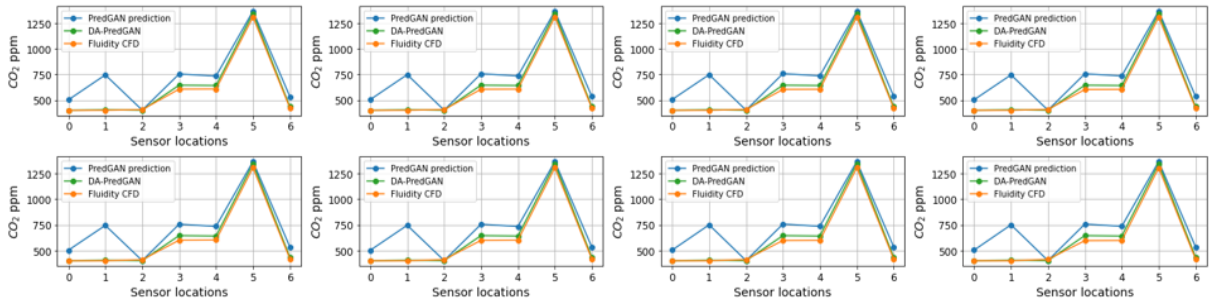
Figure 7: PredGAN prediction plots of CO_2 ppm levels at (a) Peripheral Nodes (explained in main-body text) and (b) at the seven sensor locations, across 410 time-levels.

Figure 8 depicts the sensor location CO_2 ppm value predictions at consecutive snapshots in time. Sub-figure (a) expresses explicitly the first 8 time levels, with the ninth, one step prediction. Focussing on the convergence of this ninth prediction is vital for developing the full-pass predictive model, as it remains the first true prediction given the set of initial known numerical solutions. It therefore sets up further optimisation steps and predictive time levels. As shown, the PredGAN is able to effectively optimise the latent values converging almost exactly. Focusing on predicting beyond time-steps the GAN has been trained on, remains one of the major testing criteria set for effective and impartial analysis of the developed system. The GANs training makes use of 410 time-levels, adhering to hold back a test set corresponding to a 90-10 split. Figure 8 (b), depicts this graphical representation of this model validation/test-case, showing the sensors at the last 8 subsequent snapshots in time (488-455). At each sensor location, and at each snapshot there remains significant convergence to the true numerical solution.

However, with the PredGAN model, there exists several nodes including sensor locations 1, which although is predicted within a very acceptable margin, expresses errors higher than other nodal locations.
As a result, this brings light to the possibility of implementing an element of data-correction; which further lends itself to the advantages of a Data Assimilation predictive model in the DA-PredGAN.



(a)



(b)

Figure 8: Snapshot comparisons at sensor locations of PredGAN and DA-PredGAN predictive models against Fluidity at (a) the first nine time levels, (b) last 8 time levels (448-455)

4.4 DA-PredGAN

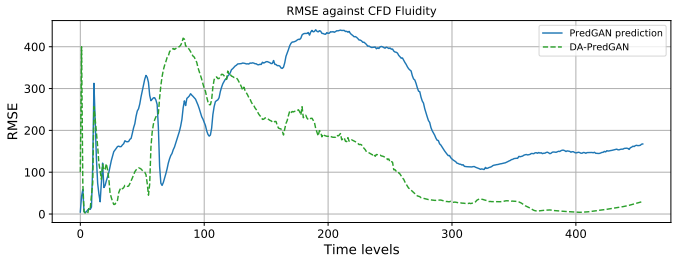
Assimilating data in time, enabling an augmented model may result in various advantages when attempting to interpolate spatial-temporal variation. As mentioned in the System Design section, its implementation carries out both forward as well as backward marches.
The importance of the backward march is to enable the model to effectively form initial or prior conditions to the points in time that observations are available. This can have an important implication and extremely advantageous properties, as forward marching would make little use of sensor observations until the model meets that respective time step.
Below details the pseudo code algorithm for the DA-PredGAN proposed.

Algorithm 1: DA-PredGAN Algorithm

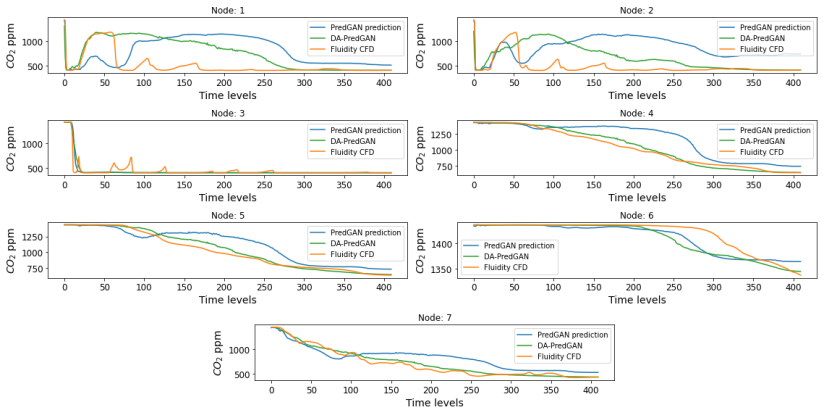
```
relax ← 1
j ← 0
while j < 10 do
    if j == 0 then
        Initial first step prediction optimising Eq. (5)
        March forward in time optimising each time step using Eq. (6)
    else
        March forward in time optimising Eq. (6)
    end
    Time march backwards optimising Eq. (6)
    Calculate observational loss
    if j == 0 then
        continue
    else
        if observational loss < 0.98 (observational loss at previous iteration (j − 1)) then
            relax *= 1.5
            if relax > 1 then
                relax ← 1
            else
                continue
            end
        else
            relax *= 0.5
            if relax < 0.05 then
                break
            else
                continue
            end
        end
    end
    j ← j + 1
end
Final forward march optimising Eq. (5)
```

Throughout the Data assimilation process, we aim to minimise the data mismatch between the observations and the GAN predictions generated. In light of this, priority of the minimisation of this functional is given, further considering the overall convergence weighting of this additional term within the loss equation, Eq. 6. With regards to the implementation, this is carried out configuring the tuning parameters ζ_{obs} and ζ_{μ} appropriately. In the model developed, values for ζ_{obs} and ζ_{μ} were

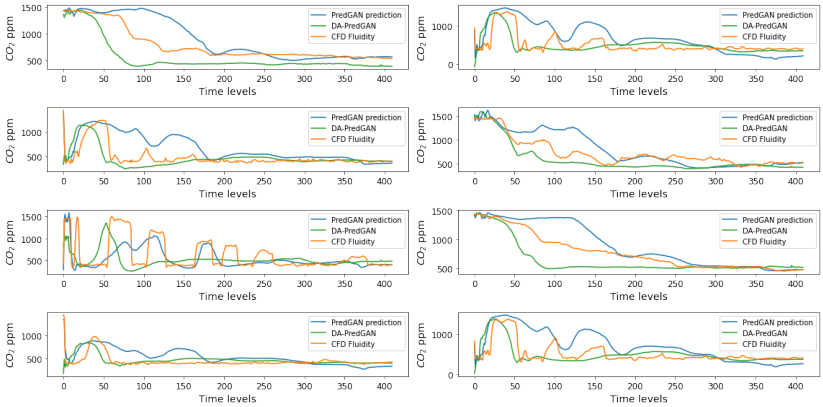
tuned to be 0.005 and 0.5 respectively; forming overall convergence losses of magnitudes between $\sim 10^{-4}$ and 10^{-5}.



(a)



(b)



(c)

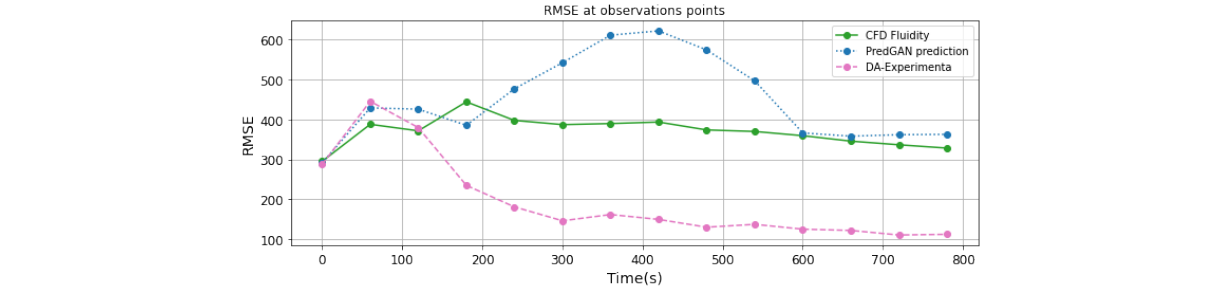
Figure 9: Comparative plots of convergence against Fluidity CFD for both predictive models, PredGAN and DA-PredGAN. (a) Root mean square error of the respective convergences to CFD over the 410 time levels. (b) CO_2 variation at the seven sensor locations and (b) at 8 randomly chosen nodes across the whole 148,906 node domain. Random node locations: 18071, 46784, 86376, 34646, 144594, 102752, 133064, 136683.

The implementation of the DA method saw an increased efficiency due to the direct use of the GAN which was trained to output sensor values explicitly in a novel technique of 'enhanced training'. This incurred no computational cost to inverse transform the POD coefficients, isolate the surrounding nodal positions and then requiring interpolation of the primitive values to calculate the observed data mismatch. Through this technique of enhanced training, direct observational loss is attained at each time level output from the GAN.
With respect to the focal aspect of the data assimilated observations, figure 9 conveys a temporal plot across the seven sensor locations. As shown, although both models follow the numerical simulations trend evolving in time, across all nodes the DA-PredGAN saw greater accuracy in convergence.
However, 9 (c), expresses the nature of the predictive models with respect to randomly chosen nodes. As can be seen, the graph depicts the smooth nature of the DA-PredGAN as opposed to the PredGAN attempting to converge more accurately at even sharp features. Analysis of sub-figure (a) however, details that through DA there is a significantly reduced data mismatch, which is more expressive at time levels beyond ~ 120. The issue of high RMSE at earlier time-levels can be determined due to the nature of Data Assimilation being an ill-posed inverse problem, granting more than one possible solution.

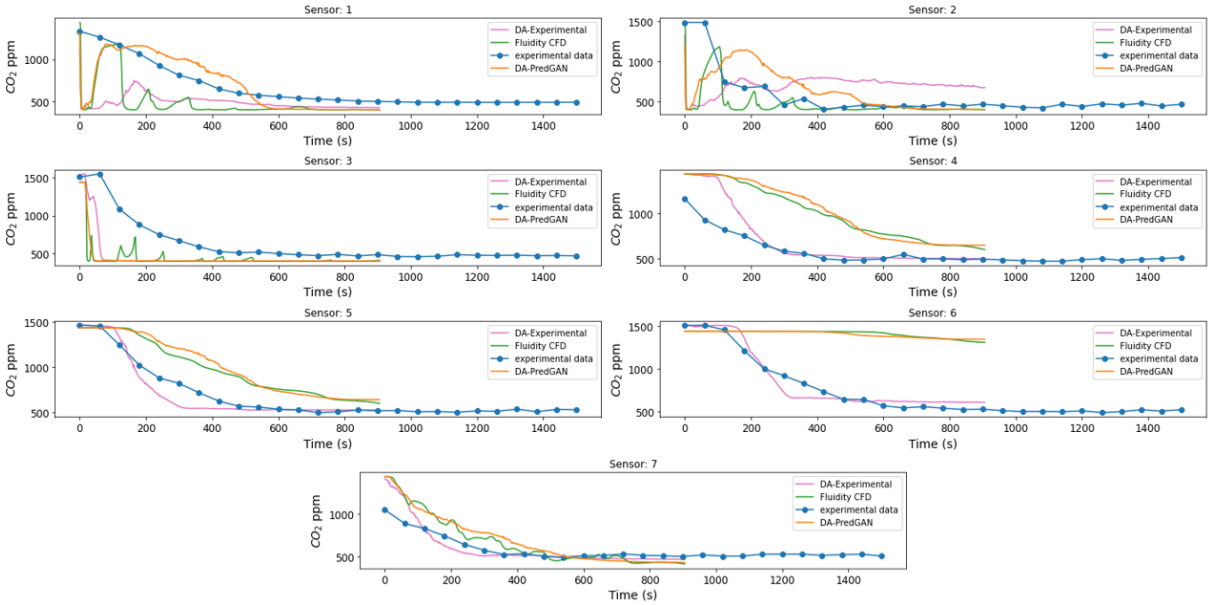
Snapshot comparisons between the PredGAN and DA-PredGAN models at sensor locations are shown in Figure 8 previously displayed. The convergence of the next step prediction from the initial known solutions in (a) shows successful for both models.
The subsequent success of the DA-PredGAN model can really be expressed through analysing sensor locations snapshots in time. Specifically, beyond the first 8 time levels and with more emphasis with how the prediction fares outside the range of trained time-levels. Sub-figure (b) shows the predictions of time levels 448 up to 455. The data mismatch scheme with the inclusion of observations removed the consistent deviation of localisation sensors (e.g.sensor 1).
Moreover, the model shows an extreme improvement across all the sensor locations, with even more successful convergence in comparison with the effective PredGAN model.

4.4.1 Experimental Data

Extending further to real experimental data collected by the MAGIC field campaign [25] pertains to a test case pertinent to the true fluid flow observed at the 7 physical sensors. The results for Data assimilating at only 14 time-levels, with the observations present in 60 second intervals between 0 and 780s are shown in Figure 10 (b). This sub-figure expresses two Data Assimilated methods: DA-PredGAN being assimilated with CFD observations whilst DA-Experimental with limited experimental (true) data.



(a)



(b)

Figure 10: Assimilating experimental data with the DA-PredGAN, and further showing convergence comparisons between the DA-PredGAN, PredGAN and Fluidity CFD with respect to the experimental (true) data. (a) A convergence comparison showing the Root Mean Square Error of the different models in time against the experimental observations. (b) A temporal plot comparing the convergence of the CFD (green), DA-PredGAN (Assimilated with CFD observations) & DA-Experimental (Assimilated with experimental dataset) against the experimental sensor data.

With matching the observations and determining the sensor location values using the DA predictive model presents it to convergence more optimally overall to the true field campaign than both the CFD assimilated prediction as well as the Fluidity CFD simulation.
Sub-figure (a) reiterates the effectiveness of the DA-PredGAN model - showing that across simulated time, the DA-PredGAN is able to minimise root mean square error (RMSE) with experimental observations exceptionally well. This data mismatch minimisation predictive method indicates promising results with albeit only a limited set of corrective observational points.

5 Conclusion & Future work

In the work presented within this research thesis, the developed generative adversarial networks within the non-intrusive reduced order model framework showed effectiveness in spatio-temporal predictions of CO_2 in an enclosed space. Through the challenge of using GANs to accurately model and provide a more efficient alternative to high-fidelity, computationally costly numerical simulations; the PredGAN algorithm was able to effectively model a quasi non-cyclical dataset. Further to this, the research was extended to show the applicability and impressive nature of predictive GANs in the application of data assimilation. Its capability of observational sensor location loss reduction with no requirement for additional numerical simulations might prove significant for future viral infection fluid flow modelling. Specifically, with analysis of the RMSE using experimental observations (figure 10 (a)), showed noteworthy results in its significant convergence to the real (experimental) dataset truer than both the PredGAN and numerical simulation.
Through the development of the technique of enhanced training, the research project showed an amplification of the performance efficiency. Thereby expressing that with appropriately trained predictive models, spatio-temporal and embedded sensor convergence predictions can effectively be achieved.

Future developments might see the employment of a Space Filling Curve Convolutional Auto-Encoder (SFC-CAE) as a possible method of dimensionality reduction. As Proper Orthogonal Decomposition remains a linear mapping between features, SFC produces a topological mapping which is spatially non-distruptive thereby preserving local data correlations [16].
Additionally, due to the hyper-parameter sensitiveness of DCGANs, with their respectively model failures including mode collapse and vanishing gradient problems, possible exploration of differing and perhaps more stable GAN architectures (e.g. Wasserstein GAN) may prove to be beneficial.
Extending the framework to identify regions which pertain to a reduced CO_2 flow, coupled with further being able to generalise to different room geometries; might see highly impactful in mitigation of viral spread in enclosed spaces such as classrooms, trains and planes.

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] I.C.L. AMCG. Fluidity manual v4.1.12, 2015. Accessed: 2021-06-20.

[3] Maddalena Amendola, Rossella Arcucci, Laetitia Mottet, Cesar Quilodran Casas, Shiwei Fan, Christopher Pain, Paul Linden, and Yi-Ke Guo. Data assimilation in the latent space of a neural network. *arXiv preprint arXiv:2012.12056*, 2020.

[4] Steven L Brunton, Bernd R Noack, and Petros Koumoutsakos. Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics*, 52:477–508, 2020.

[5] Ensheng Dong, Hongru Du, and Lauren Gardner. An interactive web-based dashboard to track COVID-19 in real time. *The Lancet infectious diseases*, 20(5):533–534, 2020.

[6] Maayan Frid-Adar, Idit Diamant, Eyal Klang, Michal Amitai, Jacob Goldberger, and Hayit Greenspan. Gan-based synthetic medical image augmentation for increased CNN performance in liver lesion classification. *Neurocomputing*, 321:321–331, 2018.

[7] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.

[8] Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. Bidirectional lstm networks for improved phoneme classification and recognition. In *International conference on artificial neural networks*, pages 799–804. Springer, 2005.

[9] Charles R. Harris, K. Jarrod Millman, Stéfan J van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585:357–362, 2020.

[10] Jan S Hesthaven and Stefano Ubbiali. Non-intrusive reduced order modeling of nonlinear problems using neural networks. *Journal of Computational Physics*, 363:55–78, 2018.

[11] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[12] Mahdi Kherad, Mohammad Kazem Moayyedi, and Faranak Fotouhi. Reduced order framework for convection dominant and pure diffusive problems based on combination of deep long short-term memory and proper orthogonal decomposition/dynamic mode decomposition methods. *International Journal for Numerical Methods in Fluids*, 93(3):853–873, 2021.

[13] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Jason Grout, Sylvain Corlay, Paul Ivanov, Damián Avila, Safia Abdalla, and Carol Willing. Jupyter notebooks – a publishing format for

reproducible computational workflows. In F. Loizides and B. Schmidt, editors, *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pages 87 – 90. IOS Press, 2016.

[14] Gang Liu and Jiabao Guo. Bidirectional LSTM with attention mechanism and convolutional layer for text classification. *Neurocomputing*, 337:325–338, 2019.

[15] Yifan Liu, Zengchang Qin, Tao Wan, and Zhenbo Luo. Auto-painter: Cartoon image generation from sketch by using conditional wasserstein generative adversarial networks. *Neurocomputing*, 311:78–87, 2018.

[16] Baback Moghaddam, Kenneth J Hintz, and Clayton V Stewart. Space-filling curves for image compression. In *Automatic object recognition*, volume 1471, pages 414–421. International Society for Optics and Photonics, 1991.

[17] Nina Morozova, FX Trias, R Capdevila, Carlos David Pérez-Segarra, and A Oliva. On the feasibility of affordable high-fidelity cfd simulations for indoor environment design and control. *Building and Environment*, 184:107144, 2020.

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[19] César Quilodrán-Casas, Rossella Arcucci, Christopher Pain, and Yike Guo. Adversarially trained lstms on reduced order models of urban air pollution simulations. *arXiv preprint arXiv:2101.01568*, 2021.

[20] César Quilodrán-Casas, Vinicius Santos Silva, Rossella Arcucci, Claire E Heaney, Yike Guo, and Christopher C Pain. Digital twins based on bidirectional lstm and gan for modelling the covid-19 pandemic. *arXiv preprint arXiv:2102.02664*, 2021.

[21] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[22] Swami Sankaranarayanan, Yogesh Balaji, Carlos D Castillo, and Rama Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8503–8512, 2018.

[23] Vinicius LS Silva, Claire E Heaney, Yaqi Li, and Christopher C Pain. Data assimilation predictive gan (da-predgan): applied to determine the spread of covid-19. *arXiv preprint arXiv:2105.07729*, 2021.

[24] Vinicius LS Silva, Claire E Heaney, and Christopher C Pain. Gan for time series prediction, data assimilation and uncertainty quantification. *arXiv preprint arXiv:2105.13859*, 2021.

[25] Jiyun Song, S Fan, William Lin, L Mottet, H Woodward, M Davies Wykes, R Arcucci, D Xiao, J-E Debay, H ApSimon, et al. Natural ventilation in cities: the implications of fluid mechanics. *Building Research & Information*, 46(8):809–828, 2018.

[26] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009.

[27] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore,

Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[28] Zheng Wang, Dunhui Xiao, Fangxin Fang, Rajesh Govindan, Christopher C Pain, and Yike Guo. Model identification of reduced order fluid dynamics systems using deep learning. *International Journal for Numerical Methods in Fluids*, 86(4):255–268, 2018.

[29] D Xiao, F Fang, CC Pain, IM Navon, P Salinas, and Z Wang. Non-intrusive model reduction for a 3d unstructured mesh control volume finite element reservoir model and its application to fluvial channels. *International Journal of Oil, Gas and Coal Technology*, 19(3):316–339, 2018.

[30] Dana Yelin, Eytan Wirtheim, Pauline Vetter, Andre C Kalil, Judith Bruchfeld, Michael Runold, Giovanni Guaraldi, Cristina Mussini, Carlota Gudiol, Miquel Pujol, et al. Long-term consequences of COVID-19: research needs. *The Lancet Infectious Diseases*, 20(10):1115–1117, 2020.

Appendices

Supplementary Graphs:

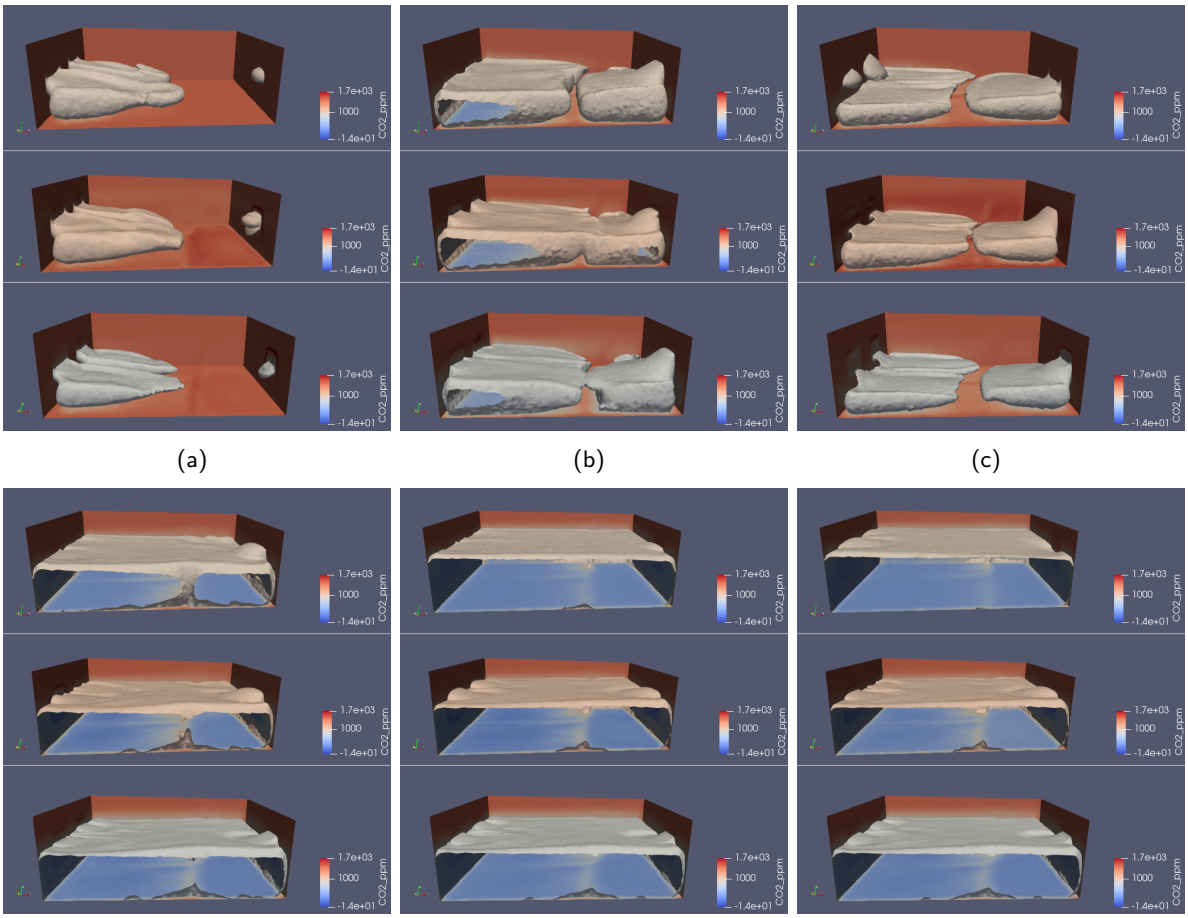


Figure 11: Isosurface CO_2 plot comparison between CFD model(top), PredGAN(middle) and DA-PredGAN(bottom) at the (a) 10th (∼2s), (b) 100th (∼200s), (c) 60th (∼120s), (d) 200th (∼400s), (e) 390th (∼780s), and (f) 455th (∼910s) time steps.

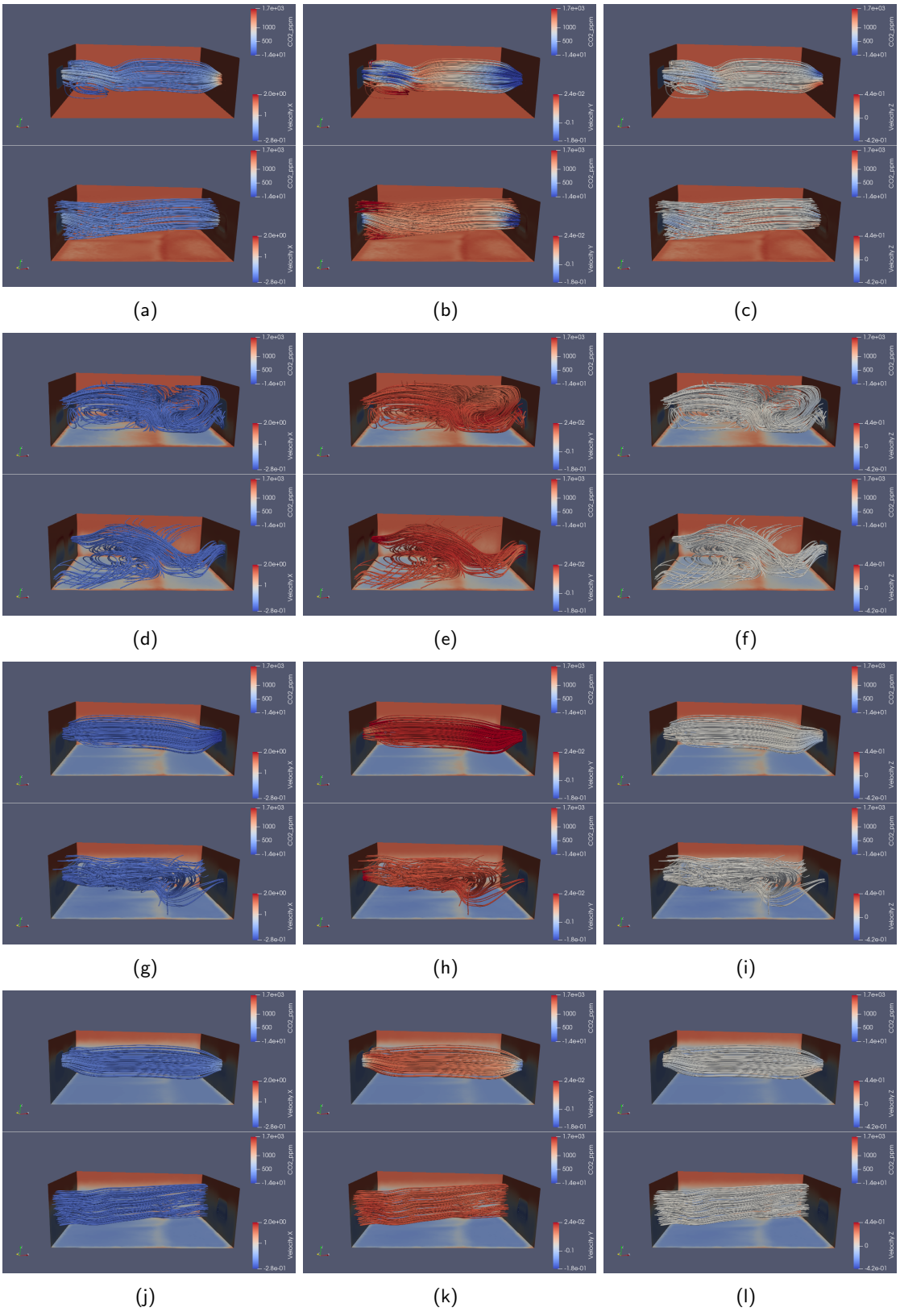
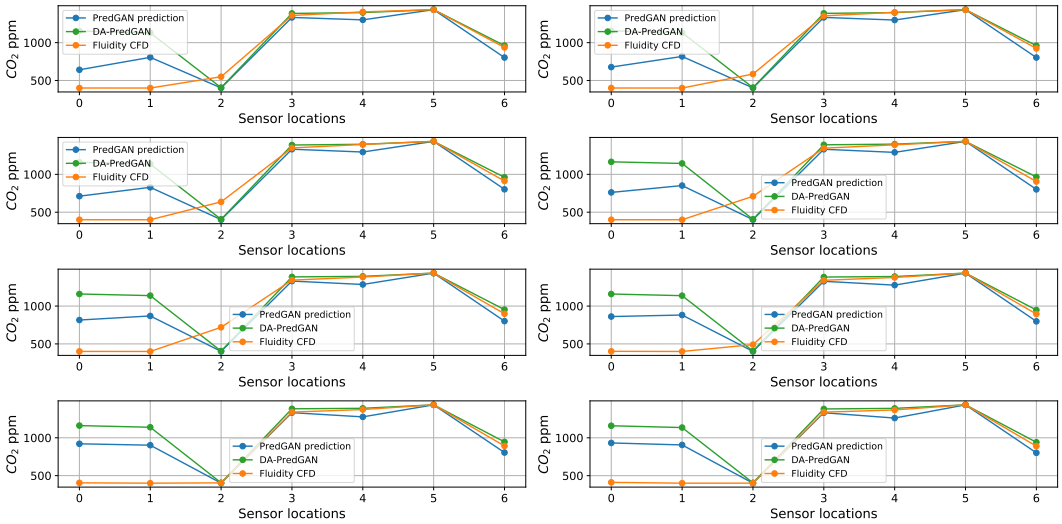
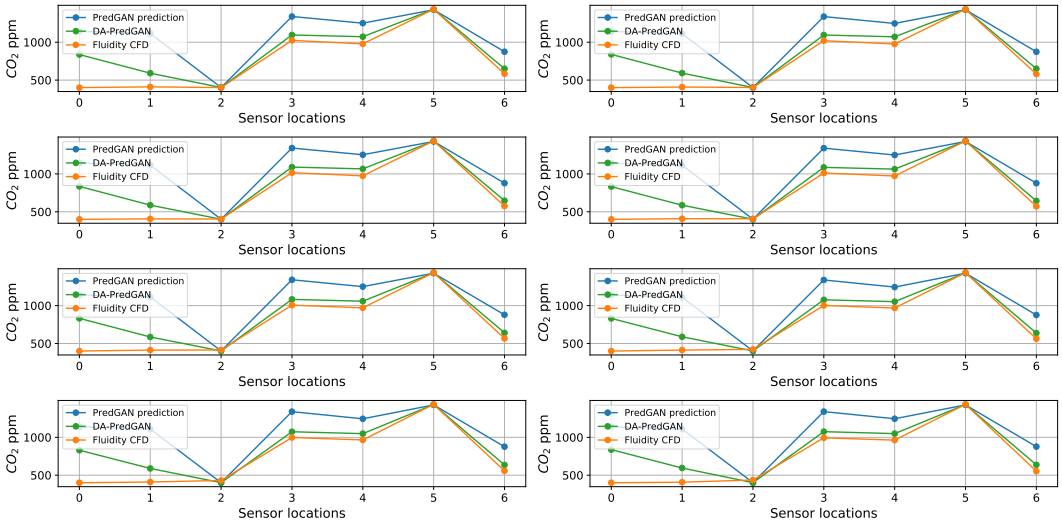


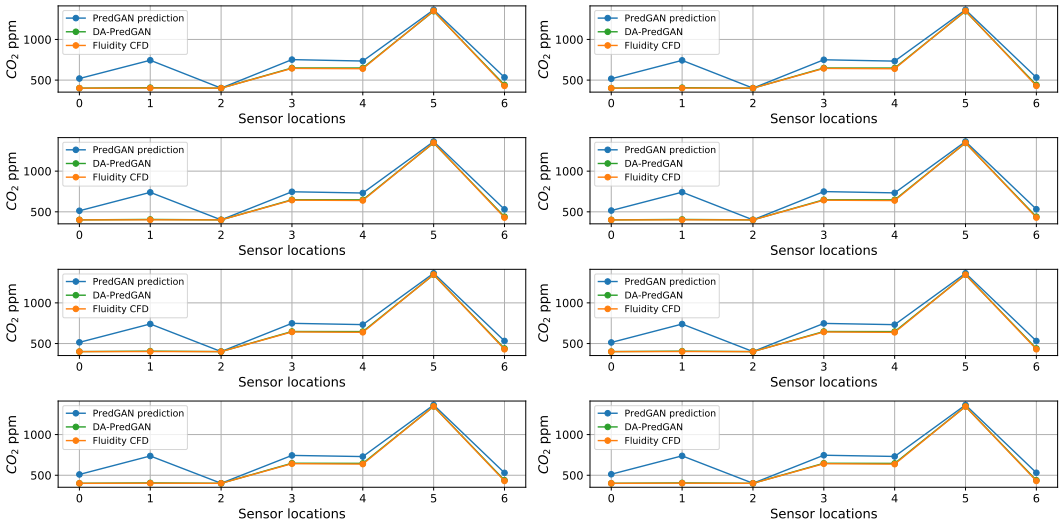
Figure 12: Streamline velocities from the CFD model (top) and predicted DA-PredGAN (bottom) at each subplot. Time level 1 (∼2s) corresponding to sub-figures a, b & c. Time level 100 (∼200s) corresponding to sub-figures d, e & f. Time level 220 (∼440s) corresponding to sub-figures g, h & i. Time level 455 (∼910s) corresponding subplots j, k & l. The first column of sub-figures a, d, g & j represent the velocity x component. Second column with sub-figures b, e, h & k the y component. Finally, the last column with sub-figures c, f, i & l corresponding to the z velocity component.



(a)

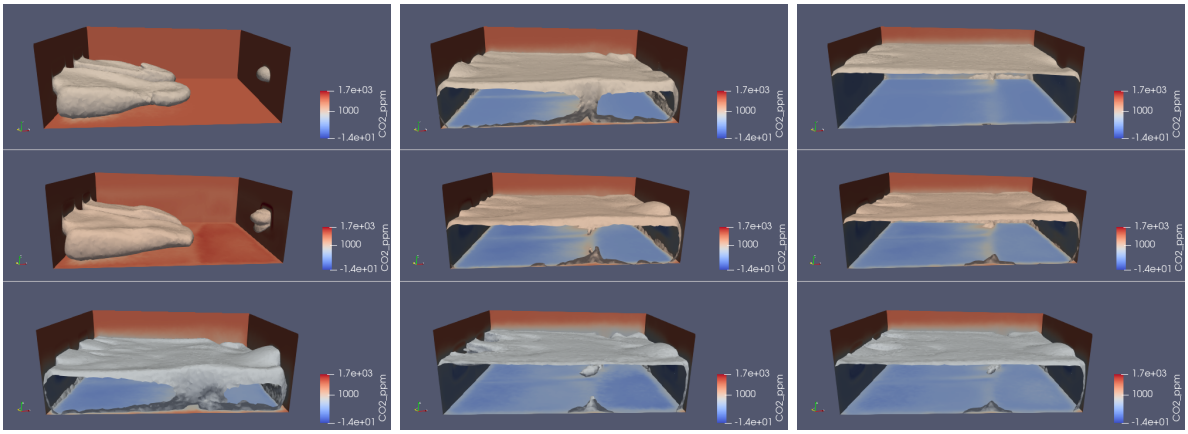


(b)



(c)

Figure 13: Snapshots in time of sensor locations predictions compared to Fluidity CFD at time levels (a) 80-88 (∼160-176s), (b) 200-208(∼400-416s) and (c) 390-398 (∼780-796s).



(a) (b) (c)

Figure 14: Isosurface plots comparing the CFD(top), PredGAN(bottom) and DA-PredGAN with experimental observations (bottom) at the (a) 10th (∼20s), (b) 200th (∼400s), and (c) 455th (∼910s) time steps.

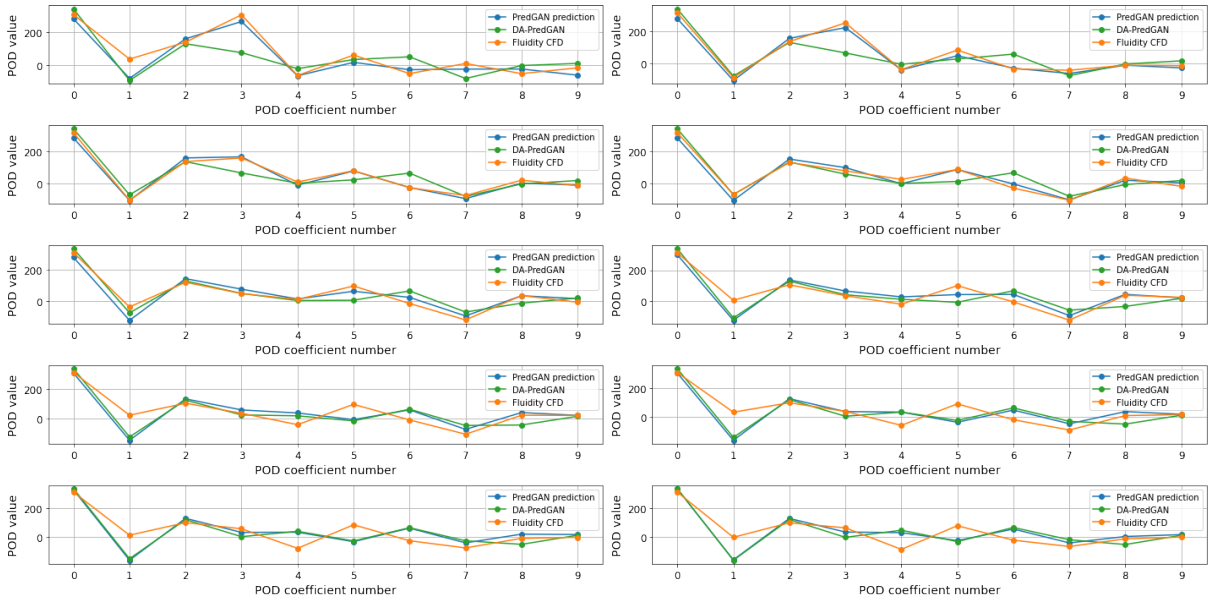


Figure 15: PredGAN and DA-PredGAN comparisons of the first 10 POD coefficients against Fluidity.

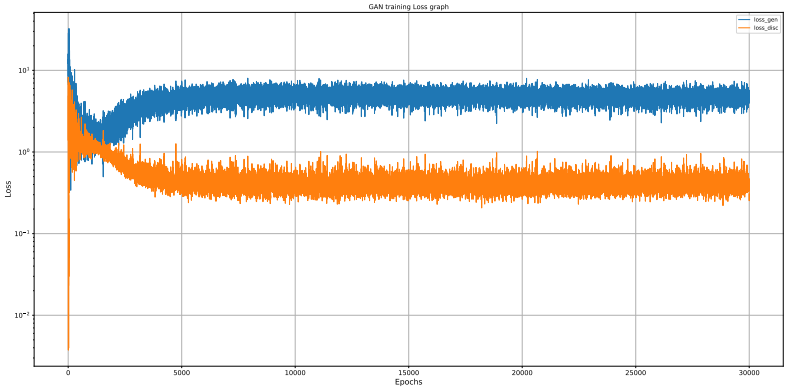


Figure 16: Losses of the generator and discriminator during enhanced GAN training over 30,000 epochs.