GAN for time series prediction, data assimilation and uncertainty quantification

Vinicius L. S. Silva
Applied Modelling and Computation Group
Imperial College London, UK
viluiz@gmail.com

Claire E. Heaney
Applied Modelling and Computation Group
Imperial College London, UK
c.heaney@imperial.ac.uk

Christopher C. Pain
Applied Modelling and Computation Group
Imperial College London, UK
c.pain@imperial.ac.uk

Abstract

We propose a new method in which a generative adversarial network (GAN) is used to quantify the uncertainty of forward simulations in the presence of observed data. Previously, a method has been developed which enables GANs to make time series predictions and data assimilation by training a GAN with unconditional simulations of a high-fidelity numerical model. After training, the GAN can be used to predict the evolution of the spatial distribution of the simulation states and observed data is assimilated. In this paper, we describe the process required in order to quantify uncertainty, during which no additional simulations of the high-fidelity numerical model are required. These methods take advantage of the adjoint-like capabilities of generative models and the ability to simulate forwards and backwards in time. Set within a reduced-order model framework for efficiency, we apply these methods to a compartmental model in epidemiology to predict the spread of COVID-19 in an idealised town. The results show that the proposed method can efficiently quantify uncertainty in the presence of measurements using only unconditional simulations of the high-fidelity numerical model.

1 Introduction

Complex physical and engineering systems are usually described in terms of partial differential equations that, for most problems of practical interest, cannot be solved analytically. Then it is necessary to use numerical methods to solve the governing equations [13, 2]. Nonetheless, these methods need a large number of degrees of freedom to solve the partial differential equations accurately. This fact can generate prohibitively expensive simulations in terms of computational time and memory demand. Furthermore, these models are built on limited information, which makes their predictions uncertain. Therefore, it is necessary to assimilate observed data and formally propagate the uncertainties through the numerical simulator. To that end, the purpose of prediction and data assimilation evolved from a purely deterministic perspective (single solution) to a probabilistic one (multiple solutions) [22, 30, 7, 4].

In this context, surrogate models are computationally appealing and have been attracting significant attention in the last decades. Deep neural networks have become one of the most popular surrogate models in science and engineering nowadays [39, 35, 42, 40, 33, 26]. Among them, generative

Source code and data are available at https://github.com/viluiz/gan

Preprint. Under review.

adversarial networks (GAN) have been demonstrating promising results. GANs have been used to predict spatio-temporal solutions for super-resolution fluid flow [38], carbon capture [41], incoming waves from Hokkaido tsunami [11], and the spread of COVID-19 [29]. GANs have also been used in the processes of data assimilation and uncertainty quantification to generate conditional models parameters [21, 16, 28, 10]. Even though in these works they still need to simulate the high-fidelity numerical model to predict forward in time.

In order to quantify the uncertainty of forward numerical simulations, multiple random models conditional to measurements are required. After simulating the conditional models, an empirical distribution of the variables of interest can be obtained [20]. The validity of the uncertainty quantification depends on the quality of the generated conditional simulations. Nonetheless, it is often difficult and computationally expensive to generate a single conditional model, suggesting that the task of quantifying uncertainty must be even more difficult [20, 32]. Methods such as rejection sampling and Markov chain Monte Carlo are unfeasible to propagate uncertainty through most practical computational models due to their computational cost [24, 20, 22, 31]. Therefore, approximate methods need to be used. Among them, Liu et al. [20] showed that the randomized maximum likelihood (RML) [17, 23], also called randomize-then-optimize (RTO) [5], performed better than other approximate methods.

In this work, we propose a new method inspired by the RML in which a GAN is used to quantify the uncertainty of forward simulations in the presence of measurements. The GAN is trained using only unconditional simulations of the high-fidelity numerical model. After training, the GAN can be used as a surrogate model to predict the evolution of the spatial distribution of the simulation states and observed data can be assimilated. We describe the process required in order to quantify uncertainty, during which no additional simulations of the high-fidelity numerical model are required. We apply these methods to quantify the uncertainty of a compartmental model in epidemiology, that represents the spread of COVID-19 in an idealized town. In the authors' opinion, there is no negative societal impact of this research. All the data generated here is synthetic and developed by the authors.

2 Method

In this section, we present a method to quantify uncertainty in the presence of observed data using a GAN. The GAN only needs to be trained with the priors (unconditional simulations) from the high-fidelity numerical simulation. The data assimilation and uncertainty quantification are performed within the GAN using the newly proposed loss functions and taking advantage of automatic differentiation. First, we demonstrate how a GAN within a reduced-order model [14, 37] framework can be used to generate time series predictions. Secondly, the prediction is extended to account for observed data. Finally, we present the proposed method to quantify uncertainty.

2.1 GAN for time series prediction

In order to make predictions in time using a GAN, an algorithm named Predictive GAN (PredGAN) is used here [29]. We train the GAN to produce data at a sequence of $m+1$ time steps, i.e. given a latent vector $\mathbf{z}$, the output of the generator $G(\mathbf{z})$ will be data at time steps $n-m$ to n, no matter which point in time n represents. Then, given known solutions at m consecutive time steps, we can perform an optimization to match the first m time levels in the output of the generator with the known solutions. After convergence, the last time step, $m+1$, in the output of the generator is the prediction. We now can use this last time level $m+1$ as a known solution and perform another optimization to predict the time step $m+2$. The process continues until we predict all time steps. Figure 1 illustrates how the PredGAN works.

In our case, after training, the output of the generator $G(\mathbf{z})$ is made up of $m+1$ consecutive time steps of compressed grid variables $\boldsymbol{\alpha}$ (outputs of the numerical model), and model parameters $\boldsymbol{\mu}$ (inputs of the numerical model). The compressed variables are principal component analysis (PCA) coefficients, but could also be latent variables from an autoencoder. For a GAN that has been trained

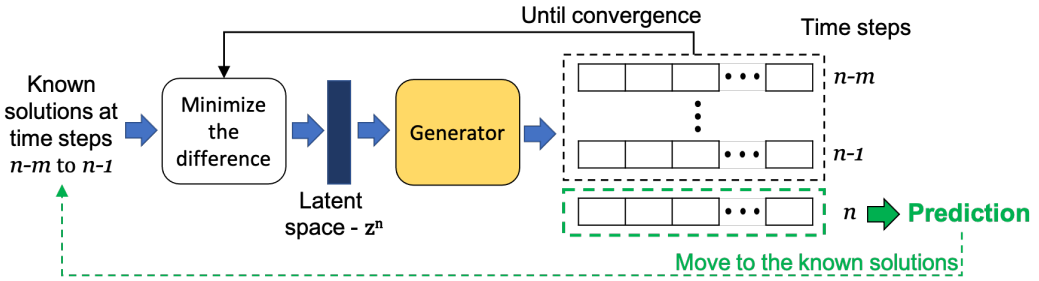

Figure 1: Overview of the PredGAN process.

with $m+1$ time levels, $G(\mathbf{z})$ takes the following form

$$G(\mathbf{z}) = \begin{bmatrix} (\boldsymbol{\alpha}^{n-m})^T, (\boldsymbol{\mu}^{n-m})^T \\ \vdots \\ (\boldsymbol{\alpha}^{n-1})^T, (\boldsymbol{\mu}^{n-1})^T \\ (\boldsymbol{\alpha}^{n})^T, (\boldsymbol{\mu}^{n})^T \end{bmatrix} \tag{1}$$

where the compressed grid variables are defined as $(\boldsymbol{\alpha}^n)^T = [\alpha_1^n, \alpha_2^n, \cdots, \alpha_{N_{\text{PCA}}}^n]$. N_{PCA} is the number of principal components, and α_i^n represents the ith PCA coefficient at time level n. The model parameters are represented as $(\boldsymbol{\mu}^n)^T = [\mu_1^n, \mu_2^n, \cdots, \mu_{N_\mu}^n]$. N_μ is the number of model parameters, and μ_i^n represents the ith parameter at time level n.

In each iteration of the PredGAN, one new time step is predicted. Assume we have solutions at time levels from $n-m$ to $n-1$ for the PCA coefficients, denoted by $\{\tilde{\boldsymbol{\alpha}}^k\}_{k=n-m}^{n-1}$, and also have the model parameters over all time steps $\tilde{\boldsymbol{\mu}}^k$, then to predict the solution at time level n we perform an optimization defined as

$$\begin{gathered} \mathbf{z}^n = \arg\min_{\mathbf{z^n}} \mathcal{L}_p(\mathbf{z}^n), \\ \mathcal{L}_p(\mathbf{z}^n) = \sum_{k=n-m}^{n-1} \left(\tilde{\boldsymbol{\alpha}}^k - \boldsymbol{\alpha}^k\right)^T \boldsymbol{W}_\alpha \left(\tilde{\boldsymbol{\alpha}}^k - \boldsymbol{\alpha}^k\right) + \sum_{k=n-m}^{n-1} \zeta_\mu \left(\tilde{\boldsymbol{\mu}}^k - \boldsymbol{\mu}^k\right)^T \boldsymbol{W}_\mu \left(\tilde{\boldsymbol{\mu}}^k - \boldsymbol{\mu}^k\right), \end{gathered} \tag{2}$$

where $\boldsymbol{W}_\alpha$ is a square matrix of size N_{PCA} whose diagonal values are equal to the weights that govern the relative importance of the PCA coefficients, all other entries being zero. $\boldsymbol{W}_\mu$ is a square matrix of size N_μ whose diagonal values are equal to the model parameter weights, and the scalar ζ_μ controls how much importance is given to the model parameters compared to the compressed variables. It is worth noticing that only the time steps from $n-m$ to $n-1$ are taken into account in the functional which controls the optimization of $\mathbf{z}^n$. After convergence, the newly predicted time level n is added to the known solutions $\tilde{\boldsymbol{\alpha}}^n = \boldsymbol{\alpha}^n$, and the converged latent variables $\mathbf{z}^n$ are used to initialize the latent variables at the next optimization to predict time step $n+1$. The process repeats until all time levels are predicted. The gradient of Eq. (2) is calculated by automatic differentiation [36, 19, 6], which means backpropagating the error generated by the loss function in Equation (2) through the generator.

2.2 GAN for data assimilation

Data assimilation is a type of inverse problem that aims to incorporate observed data into mathematical models. To perform data assimilation with GANs, Silva et al. [29] proposed the Data Assimilation Predictive GAN (DA-PredGAN) that incorporates three main changes in the PredGAN.

1. One additional term is included in the loss function in Eq. (2) to take account of the data mismatch between the observed data and the generated values.

2. The aim of the data assimilation is to match the observed data and to determine the model parameters $\boldsymbol{\mu}^k$ (inputs of the numerical model). Therefore, they are not known a priori, as in the prediction.
3. The forward marching in time is now replaced by forward and backward marching.

The loss function for the optimization at each iteration n of the forward march is given by

$$\mathcal{L}_{da,f}(\mathbf{z}^n) = \sum_{k=n-m}^{n-1} \left(\tilde{\boldsymbol{\alpha}}^k - \boldsymbol{\alpha}^k\right)^T \boldsymbol{W}_\alpha \left(\tilde{\boldsymbol{\alpha}}^k - \boldsymbol{\alpha}^k\right) + \sum_{k=n-m}^{n-1} \zeta_\mu \left(\tilde{\boldsymbol{\mu}}^k - \boldsymbol{\mu}^k\right)^T \boldsymbol{W}_\mu \left(\tilde{\boldsymbol{\mu}}^k - \boldsymbol{\mu}^k\right) + \sum_{k=n-m}^{n-1} \zeta_{obs} \left(\boldsymbol{d}^k - \boldsymbol{d}_{obs}^k\right)^T \boldsymbol{W}_{obs}^k \left(\boldsymbol{d}^k - \boldsymbol{d}_{obs}^k\right), \quad (3)$$

where the observed data at each time step k is stored in the vector $\boldsymbol{d}_{obs}^k$ of size N_{obs}. $\boldsymbol{d}^k$ is the generated data calculated based on the output of the generator at time step k. In or case, it represents data at some points in the grid (high dimensional states) and it is calculated through the PCA coefficients $\boldsymbol{\alpha}^k$ and stored eigenvectors. $\boldsymbol{W}_{obs}^k$ is a square matrix of size N_{obs} whose diagonal values are equal to the observed data weights, and the scalar ζ_{obs} direct controls how much importance is given to the data mismatch. The values in the diagonal of $\boldsymbol{W}_{obs}^k$ are set to zero where we have no observation. After convergence, the new predict time level n is added to the known solutions $\tilde{\boldsymbol{\alpha}}^n = \boldsymbol{\alpha}^n$, and different from the prediction, we also update the model parameters using the newly predicted time step $\tilde{\boldsymbol{\mu}}^n = \boldsymbol{\mu}^n$.

After the forward march, the process continues with a backward march. For the latter instead of working forward in time as in Eq. (3), the process goes backwards in time, from the last time step to the first. The loss function for the optimization at each iteration n of the backward march is defined as

$$\mathcal{L}_{da,b}(\mathbf{z}^n) = \sum_{k=n+1}^{n+m} \left(\tilde{\boldsymbol{\alpha}}^k - \boldsymbol{\alpha}^k\right)^T \boldsymbol{W}_\alpha \left(\tilde{\boldsymbol{\alpha}}^k - \boldsymbol{\alpha}^k\right) + \sum_{k=n+1}^{n+m} \zeta_\mu \left(\tilde{\boldsymbol{\mu}}^k - \boldsymbol{\mu}^k\right)^T \boldsymbol{W}_\mu \left(\tilde{\boldsymbol{\mu}}^k - \boldsymbol{\mu}^k\right) + \sum_{k=n+1}^{n+m} \zeta_{obs} \left(\boldsymbol{d}^k - \boldsymbol{d}_{obs}^k\right)^T \boldsymbol{W}_{obs}^k \left(\boldsymbol{d}^k - \boldsymbol{d}_{obs}^k\right), \quad (4)$$

After performing a forward and backward march using Eqs. (3) and (4), respectively, the average of the data mismatch (last term on the right of Eqs. 3 and 4) at the end of all iterations n is calculated. If the average mismatch has not converged or the maximum number of iterations is not reached, the process continues with a new forward and backward marches. A relaxation factor is also introduced to stabilize the process of marching forward and backward in time as in Silva et al. [29].

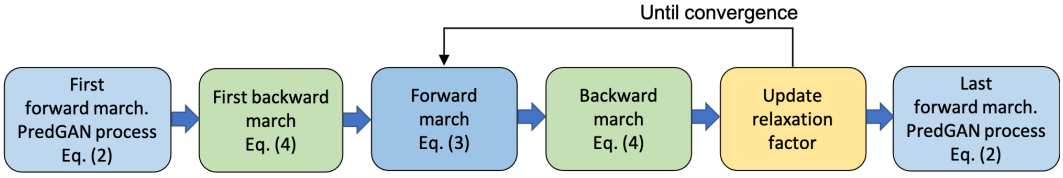

Figure 2: Overview of the DA-PredGAN. Each march represents going through all time steps.

2.3 GAN for uncertainty quantification

The computation of a single model that matches the observed data is usually insufficient to quantify risks and uncertainties. Data assimilation is generally an ill-posed inverse problem [34, 25], hence several models can match the observed data, within some tolerance. In order to quantify uncertainty, we propose in this paper a method named Uncertainty Quantification Predictive GAN (UQ-PredGAN). This method is inspired by the RML as a way of sampling a posterior distribution conditioned to observed data. In the RML, the numerical simulation is used to predict forward, and for each sample,

an optimization (data assimilation) is performed to condition the models to the observed data. The challenge is usually to perform the optimization, since the high-fidelity numerical simulation needs to be run several times and usually adjoints are not present. In this work, the proposed method UQ-PredGAN can compute uncertainties relying just on a set of unconditioned numerical simulations. The predictions, data assimilation and uncertainty quantification are performed using the inherent adjoint capability present on GAN, and no additional high-fidelity numerical simulations, other than those used for training the GAN, are required.

The idea is to generate several models that match the observed data and can quantify the uncertainty in the model states (outputs) and models parameters (inputs). To this end, we perform several data assimilations using the DA-PredGAN algorithm with the modified loss functions

$$\mathcal{L}_{uq,j}(\mathbf{z}^n) = \sum_k \left(\tilde{\boldsymbol{\alpha}}_j^k - \boldsymbol{\alpha}^k\right)^T \boldsymbol{W}_\alpha \left(\tilde{\boldsymbol{\alpha}}_j^k - \boldsymbol{\alpha}^k\right) + \sum_k \zeta_\mu \left(\tilde{\boldsymbol{\mu}}_j^k - \boldsymbol{\mu}^k\right)^T \boldsymbol{W}_\mu \left(\tilde{\boldsymbol{\mu}}_j^k - \boldsymbol{\mu}^k\right) + \sum_k \zeta_{obs} \left(\boldsymbol{d}^k - \boldsymbol{d}_{obs}^k + \boldsymbol{\varepsilon}_j^k\right)^T \boldsymbol{W}_{obs}^k \left(\boldsymbol{d}^k - \boldsymbol{d}_{obs}^k + \boldsymbol{\varepsilon}_j^k\right), \quad (5)$$

where for the forward march $k \in \{n-m, n-m+1, \cdots, n-1\}$ and for the backward march $k \in \{n+m, n+m-1, \cdots, n+1\}$. Considering N_s the number of data assimilations to be performed, then $j = 1, ..., N_s$. In this work, $N_s = 200$. The observed data error is represented by the random vector $\boldsymbol{\varepsilon}$, and we consider that all measurement errors are uncorrelated, thus they are sampled from a normal distribution with zero mean and standard deviation equal to 5% of the corresponding observed data. For each data assimilation j, we use a different prior $\tilde{\boldsymbol{\mu}}_j^k$ with the corresponding initial condition $\{\tilde{\boldsymbol{\alpha}}_j^k\}_{k=0}^m$, and a different perturbation on the observed data $\boldsymbol{\varepsilon}_j^k$.

The UQ-PredGAN is proposed as follows:

1. Sample the model parameters $\tilde{\boldsymbol{\mu}}_j$ from a normal distribution $\mathcal{N}(\overline{\boldsymbol{\mu}}, \mathbf{C}_\mu)$, where $\mathbf{C}_\mu$ is the covariance matrix of the model parameters, and $\overline{\boldsymbol{\mu}}$ is the model parameter mean vector.
2. Sample the measurement error $\boldsymbol{\varepsilon}_j$ from a normal distribution $\mathcal{N}(0, \mathbf{C}_d)$, where $\mathbf{C}_d$ is the covariance matrix of the measurement error.
3. Assimilate data using the DA-PredGAN process with the loss in Eq. (5).

After performing N_s steps of the UQ-PredGAN, accept all realizations that obtained an acceptable level of data mismatch. It is worth mentioning that for the RML, when the case is linear it samples the corrected posterior distribution [23, 31]. In this work, the test case is nonlinear, there is an additional term in the loss function compared to the RML, and the weighting terms are seen as tuning parameters. Thus, the results are an approximate sample of the posterior distribution.

2.4 Calculating the weighting terms

The weighting terms in the loss function of Eq. (5) are calculated as

$$\zeta_{obs} = \hat{\zeta}_{obs} \left(\frac{\Delta\alpha}{\Delta d}\right)^2 \left(\frac{m \sum_{i=1}^{N_{\text{POD}}} (w_\alpha)_{ii}}{\sum_k \sum_{i=1}^{N_c} (w_{obs})_{ii}^k}\right), \quad (6)$$

where $\hat{\zeta}_{obs}$ is a tuning parameter and in this work it is set to 10. $\Delta\alpha$ and Δd are the ranges of the compressed variables and the observed data, respectively. $(w_\alpha)_{ii}$ are the the terms on the diagonal of $\boldsymbol{W}_\alpha$, and $(w_{obs})_{ii}^k$ are the terms on the diagonal of $\boldsymbol{W}_{obs}^k$.

$$\zeta_\mu = \hat{\zeta}_\mu \left(\frac{\Delta\alpha}{\Delta\mu}\right)^2 \left(\frac{\sum_{i=1}^{N_{\text{POD}}} (w_\alpha)_{ii}}{\sum_{i=1}^{N_\mu} (w_\mu)_{ii}}\right), \quad (7)$$

where $\Delta\mu$ represents the range of the scalar parameters, $(w_\mu)_{ii}$ are the the terms on the diagonal of $\boldsymbol{W}_\mu$, and $\hat{\zeta}_\mu$ is a tuning parameter. ζ_μ controls how quickly one lets the parameters $\boldsymbol{\mu}_j^k$ change within the data assimilation method. In order to let $\boldsymbol{\mu}^k$ change more rapidly at the beginning and more slowly when the process is near convergence, during the data assimilation, we choose to dynamically update ζ_μ. Therefore, we start with $\hat{\zeta}_\mu = 10^{-4}$ and increase it by a factor of 1.2 after each forward-backward iteration. For the prediction, we use $\hat{\zeta}_\mu = 10^{-2}$.

3 Test case description

The test case used here is the spatio-temporal variation of a virus infection in an idealized town. The extended SEIRS model [29, 26] is a nonlinear model with compartments of susceptible (S), exposed (E), infectious (I), and recovered (R). It extends the traditional theory of the dynamics of infectious diseases [3, 9, 8] to account for variations not only in time but also in space.

3.1 Extended SEIRS model

The extended SEIRS model used in this work consists of four compartments (Susceptible - Exposed - Infections - Recovered) and two people groups (Home - Mobile). Figure 3 shows the diagram of how individuals move between compartments and groups. The model starts with some individuals in the infectious compartments (Home-I/Mobile-I). The members of these compartment will spread the pathogen to the susceptible compartments (Home-S/Mobile-S). Upon being infected, the members of the susceptible compartments are moved to the exposed compartments (Home-E/Mobile-E) and remain there until they become infectious. Infectious individuals remain in the infectious compartment until they become recovered (Home-R/Mobile-R). Recovered people can also become susceptible again due to the loss of immunity.

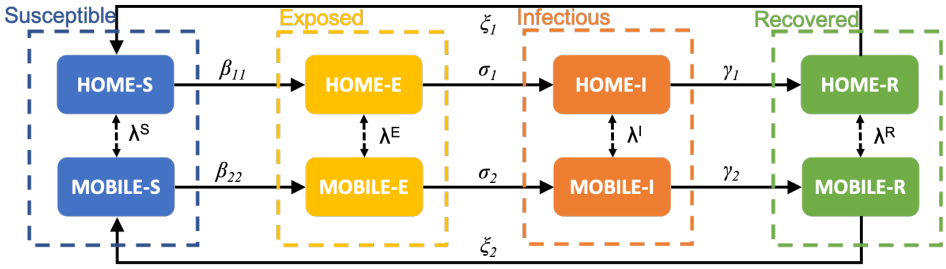

Figure 3: Diagram of the extended SEIRS model. The diagram shows how people move between groups and compartments within the same point in space (or the same cell in the grid). The vital dynamics and the transport via diffusion is not displayed here.

Modeling the movement of people is of the utmost importance in the spread of infectious diseases, such as COVID-19. Therefore, the goal of the extended SEIRS model is to reproduce the daily cycle of night and day, in which there is a pressure for mobile people to go to their homes at night, and there will be many people leaving their homes during the day and thus joining the mobile group. To this end, the extended SEIRS model uses a diffusion term (last term on the right of Eqs. (8)) and an interaction term (penultimate term on the right of Eqs, (8)) to model this process:

$$\frac{\partial S_h}{\partial t} = \eta_h N_h - \frac{S_h \sum_{h'}(\beta_{h\,h'} I_{h'})}{N_h} + \xi_h R_h - \nu_h^S S_h - \sum_{h'=1}^{\mathcal{H}} \lambda_{h\,h'}^S S_{h'} + \nabla \cdot (k_h^S \nabla S_h), \tag{8a}$$

$$\frac{\partial E_h}{\partial t} = \frac{S_h \sum_{h'}(\beta_{h\,h'} I_{h'})}{N_h} - \sigma_h E_h - \nu_h^E E_h - \sum_{h'=1}^{\mathcal{H}} \lambda_{h\,h'}^E E_{h'} + \nabla \cdot (k_h^E \nabla E_h), \tag{8b}$$

$$\frac{\partial I_h}{\partial t} = \sigma_h E_h - \gamma_h I_h - \nu_h^I I_h - \sum_{h'=1}^{\mathcal{H}} \lambda_{h\,h'}^I I_{h'} + \nabla \cdot (k_h^I \nabla I_h), \tag{8c}$$

$$\frac{\partial R_h}{\partial t} = \gamma_h I_h - \xi_h R_h - \nu_h^R R_h - \sum_{h'=1}^{\mathcal{H}} \lambda_{h\,h'}^R R_{h'} + \nabla \cdot (k_h^R \nabla R_h), \tag{8d}$$

where $\mathcal{H}$ represents the number of groups. Here, we have two groups of people, hence $\mathcal{H} = 2$, one representing people at home $h = 1$, and the second representing people that are mobile $h = 2$ and outside their homes therefore. N_h represents the total number of individuals in each group, $\beta_{h\,h'}$ is

the transmission rate between groups, σ_h is the rate of exposed individuals becoming infectious, γ_h is the recovered rate, and ξ_h is the rate recovered individuals return to the susceptible group due to loss of immunity. The vital dynamics are represented by η_h and ν_h, where η_h is the birth rate and ν_h is the death rate. The diffusion coefficient is represented by k_h and describes the movement of people around the domain. The interaction terms, $\lambda_{h\,h'}$, control how people move between groups, for example, how people that are in the mobile group move to the home group.

One important factor in dynamics of infectious diseases is the basic reproduction number ($\mathcal{R}_0$), it represents the expected number of new cases caused by a single infectious member in a completely susceptible population [12, 15]. The $\mathcal{R}_0$ controls how rapidly the disease could spread and for each group it is define as

$$\mathcal{R}_{0\,h} = \frac{\sigma_h}{(\sigma_h + \nu_h)} \frac{\beta_{h\,h}}{(\gamma_h + \nu_h)}, \tag{9}$$

where we assume $\beta_{h\,h'} = 0$ when $h \neq h'$ because people in their homes never directly meet mobile people (who are outside their homes). For this case, we can also calculate an effective $\mathcal{R}_0$ representing the $\mathcal{R}_0$ seen by the whole population at an specific time. It can be calculated as $\sum_h S_h \mathcal{R}_{0\,h}$.

3.2 Problem set up

The idealized town occupies an area of 100km by 100km as shown in Figure 4. This area is divided in 25 regions, where those labelled as 1 are regions where people do not travel, the region labelled as 2 is where homes are located, and regions from 2 to 10 are where people in the mobile group can travel. Thus people in the home group stay in the region 2. The aim is that most people move from home to mobile group in the morning, travel to locations in regions 2 to 10, and return to the home group later on in the day.

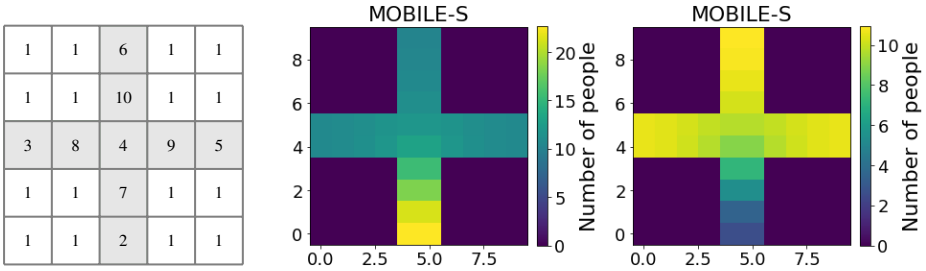

Figure 4: Idealized town (100km $\times$ 100km) showing the different regions. The two plots on the right show the number of people in each cell of the grid, for one simulation, and at two times in the day.

To generate the high-fidelity numerical simulations, the domain in Figure 4 is discretized on a regular grid of 10×10 control volume cells. Therefore, each region in Figure 4 comprises four control volumes. We start the simulation with 2000 people in each control volume of region 2 and belonging to the home group. All other fields are set to zero. The initial condition is that 0.1% of people at home have been exposed to the virus and will thus develop an infection. The epidemiological parameters used in this work are chosen to be consistent with those of COVID-19, and are described in [29, 26]. Further information about the discretization and solution methods of the high-fidelity numerical simulation can be also found in [29, 26].

4 Dataset and training process

For the training process 40 high-fidelity numerical simulations were performed in order to generate the training dataset. Each simulation consists of two different $\mathcal{R}_{0\,h}$, one for people at home and another for mobile people. We divide the whole region in Figure 4 in a regular grid of 10×10 (100 cells) for the numerical simulation. Considering that each type of people (people at home and mobile) has the four quantities of the extended SEIRS model (Susceptible, Exposed, Infectious and Recovered), there will be eight variables for each cell in the grid per time step, which gives a total number of $100 \times 8 = 800$ variables per time step. Principal component analysis is performed in the 800 variables, in order to work with a low dimensional space in the GAN. The first 15 principal components were chosen and they capture $> 99.9999\%$ of the variance held in the time snapshots.

Hence the GAN is trained to generate the 15 PCA coefficients ($\boldsymbol{\alpha}^n$) and the two $\mathcal{R}_{0\,h}$ ($\boldsymbol{\mu}^n$) over a sequence of 10 time steps. We choose this time length because it represents a cycle (one day) in the results.

The GAN architecture is based on DCGAN [27] and is implemented using TensorFlow [1] (Apache 2.0 license). We train the generator and discriminator over $5,000$ epochs, and we choose the size of the latent vector $\mathbf{z}$ to be 100. The networks receive/generate the 10 time levels as a two-dimensional array ("an image") with 10 rows and 17 columns. Each row represents a time level and each column comprises the 15 PCA coefficients and the two $\mathcal{R}_{0\,h}$. We choose this configuration, instead of a linear representation, to exploit the time dependence in the two-dimensional array. We also carried out initial tests using a linear representation of the time level outputs and a multi-layer perceptron as a generator and discriminator. However, it generated worse results than the two-dimensional representation. The success of the convolutional neural network (CNN) suggests correlations between the successive principal components that the CNN can exploit.

5 Results and discussion

In this section, we apply the UQ-PredGAN to quantify uncertainty in the extended SEIRS model considering the presence of measurements. The model represents the spread of COVID-19 in an idealized town. We generate the observed data from a high-fidelity numerical simulation ($\mathcal{R}_{0\,1} = 7.7$, $\mathcal{R}_{0\,2} = 17.4$) that was not included in the training set. Observed data was collected at five points of the domain in Figure 4, bottom-left corner of regions 2, 3, 4, 5 and 6. In order to make the case more realistic, the measurements are available every two days and we measure only infectious and recovered people. The $\mathcal{R}_{0\,h}$ are not used as observed data. For generating the priors (unconditional simulations) 200 model parameters $\mathcal{R}_{0\,h}$ were sampled from a normal distribution with a mean of 10 and standard deviation of 4. The mean was chosen in accordance with Kochańczyk et al. [18]. The 200 model parameters and their corresponding initial conditions were used to start the UQ-PredGAN process. For each of the model parameters, one data assimilation was performed as described in Section 2.3. After the data assimilation, 104 realizations were accepted based on their data mismatch error. It is worth noting that for the whole uncertainty quantification process using the UQ-PredGAN, we required only 40 high-fidelity numerical simulations (for training the GAN).

Figure 5 shows the UQ-PredGAN results for each group and compartment at one point in space (bottom-left corner of region 2 in Figure 4). The priors (gray lines) are the first forward march of each data assimilation, and the posteriors (blue lines) are the last forward march of the accepted realizations. The posterior mean (black line) is also shown in the plots. We can see from these figures that the conditional simulations (posteriors) generated by the UQ-PredGAN match the observed data, within some tolerance (we considered a measurement error of 5%), and the uncertainty is propagated through the simulation time. The high frequency oscillation presented in the results corresponds to a daily cycle, when mobile people leave their homes during the day and return to them at night. Comparable results were observed at other points in domain, hence they are not presented here.

In order to quantify the uncertainty at specific times, we generated a probability density function (PDF) of the number of people in each group and compartment. Figure 6 shows these plots at day 12 (last day with observed data). The results demonstrate that the proposed method can predict the evolution of the number of people in each group and compartment and generate the corresponding uncertainties conditioned to measurements. Figure 7 shows the PDF of the $\mathcal{R}_{0\,h}$ and effective $\mathcal{R}_0$ for the priors and posteriors. The result shows that the UQ-PredGAN was able to reduce the uncertainty in the model parameters approaching the true values used to generate the observed data. Note that the data assimilation is an inverse and usually ill-posed problem, hence different values of $\mathcal{R}_{0\,h}$ could match the measurements within some tolerance. We can also notice that for the mobile group the posterior PDFs do not match the observed data and the ground truth as well as for the home group. This can be because the number of mobile people is one order of magnitude smaller than the number of people at home, which gives the latter more importance during the data assimilation process, and the relative rate of change of the number of people in the mobile group is much greater than in the home group, thus small perturbations in the former can cause huge relative deviations.

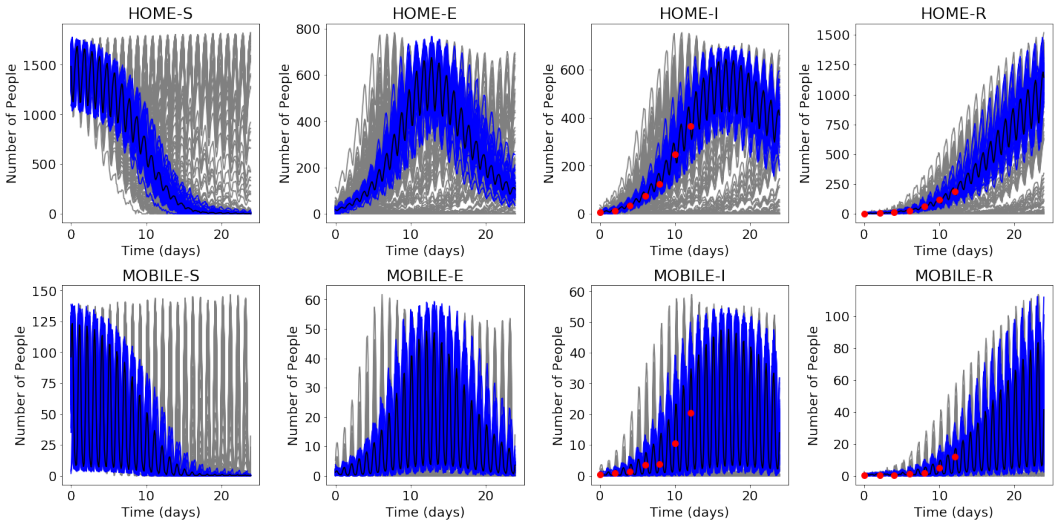

Figure 5: Results of the UQ-PredGAN applied to the spread of COVID-19 in an idealized town. In each plot, the red dots represent the observed data (measurements), the gray lines the unconditional simulations (before data assimilation), the blue lines the conditional simulations (after the data assimilation), and the black line the posterior mean.

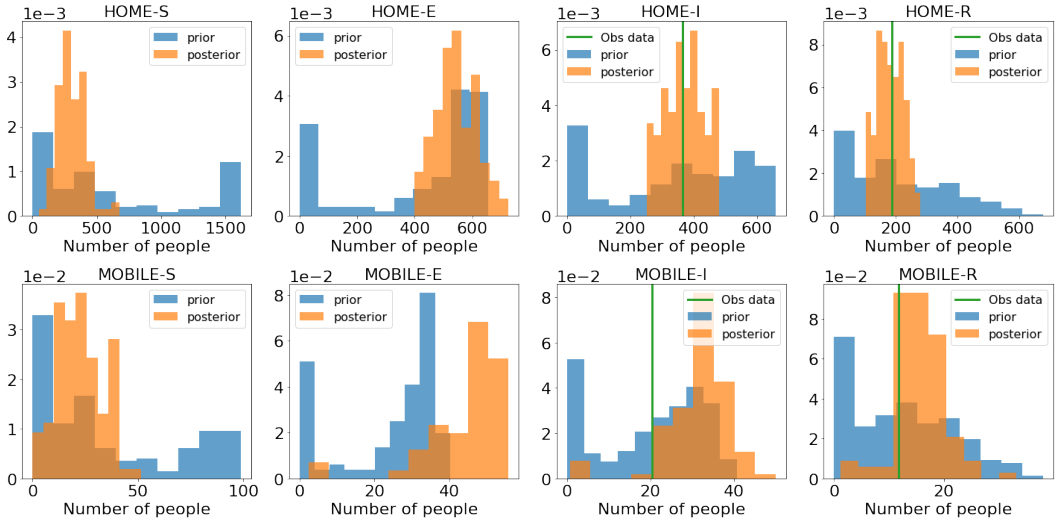

Figure 6: Probability density function of each group and compartment at day 12.

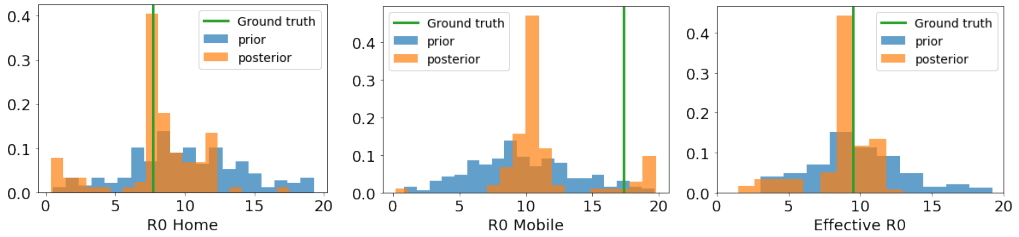

Figure 7: Probability density functions of the $\mathcal{R}_{0\,h}$ and effective $\mathcal{R}_0$ at day 12.

6 Conclusion

In this work, we proposed a novel use of a generative adversarial network (UQ-PredGAN) that is able to quantify uncertainty in time series predictions, considering the presence of measurements. The aim is to generate a surrogate model of the high-fidelity numerical simulation, that can assimilate observed data and generate the corresponding uncertainties. We applied the proposed method to an extended SEIRS model that represents the spread of COVID-19 in an idealized town. The results show that the UQ-PredGAN accurately matches the observed data and efficiently quantifies uncertainty in the model states (groups and compartments) and model parameters (basic reproduction numbers). The method used only a few unconditional simulations of the high-fidelity numerical model to train the network. The UQ-PredGAN is not limited to the underlying physics of this application: it is a general framework for time series prediction, data assimilation and uncertainty quantification.

Acknowledgements

This work is supported by the following EPSRC grants: RELIANT, Risk EvaLuatIon fAst iNtelligent Tool for COVID19 (EP/V036777/1); MUFFINS, MUltiphase Flow-induced Fluid-flexible structure InteractioN in Subsea applications (EP/P033180/1); the PREMIERE programme grant (EP/T000414/1); INHALE, Health assessment across biological length scales (EP/T003189/1); and MAGIC, Managing Air for Green Inner Cities (EP/N010221/1). This work has been undertaken, in part, as a contribution to 'Rapid Assistance in Modelling the Pandemic' (RAMP), initiated by the Royal Society. In particular, we would like to acknowledge the useful discussion had within the Environmental and Aerosol Transmission group of RAMP, coordinated by Profs Paul Linden and Christopher Pain. The first author also acknowledges the financial support from Petrobras.

Data and code availability

The source code and data used in this work are available at `https://github.com/viluiz/gan`.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

[2] W. F. Ames. *Numerical methods for partial differential equations*. Academic press, 2014.

[3] R. M. Anderson, B. Anderson, and R. M. May. *Infectious diseases of humans: dynamics and control*. Oxford University Press, 1992.

[4] R. Arcucci, J. Zhu, S. Hu, and Y.-K. Guo. Deep data assimilation: integrating deep learning with data assimilation. *Applied Sciences*, 11(3):1114, 2021.

[5] J. M. Bardsley, A. Solonen, H. Haario, and M. Laine. Randomize-then-optimize: A method for sampling from posterior distributions in nonlinear inverse problems. *SIAM Journal on Scientific Computing*, 36(4):A1895–A1910, 2014.

[6] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind. Automatic differentiation in machine learning: a survey. *The Journal of Machine Learning Research*, 18(1):5595–5637, 2017.

[7] M. Biloš, B. Charpentier, and S. Günnemann. Uncertainty on asynchronous time event prediction. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[8] O. Bjørnstad, K. Shea, M. Krzywinski, and N. Altman. The SEIRS model for infectious disease dynamics. *Nature Methods*, 17(6):557–558, 2020.

[9] O. N. Bjørnstad. *Epidemics: Models and data using R*. Springer, 2018.

[10] S. W. Canchumuni, J. D. Castro, J. Potratz, A. A. Emerick, and M. A. C. Pacheco. Recent developments combining ensemble smoother and deep generative networks for facies history matching. *Computational Geosciences*, 25(1):433–466, 2021.

[11] M. Cheng, F. Fang, C. C. Pain, and I. Navon. Data-driven modelling of nonlinear spatio-temporal fluid flows using a deep convolutional generative adversarial network. *Computer Methods in Applied Mechanics and Engineering*, 365:113000, 2020.

[12] K. Dietz. The estimation of the basic reproduction number for infectious diseases. *Statistical Methods in Medical Research*, 2(1):23–41, 1993.

[13] G. H. Golub, J. M. Ortega, et al. *Scientific computing and differential equations: an introduction to numerical methods*. Academic press, 1992.

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

[15] H. W. Hethcote. The mathematics of infectious diseases. *SIAM review*, 42(4):599–653, 2000.

[16] B. Kang and J. Choe. Uncertainty quantification of channel reservoirs assisted by cluster analysis and deep convolutional generative adversarial networks. *Journal of Petroleum Science and Engineering*, 187:106742, 2020.

[17] P. K. Kitanidis. Quasi-linear geostatistical theory for inversing. *Water resources research*, 31 (10):2411–2419, 1995.

[18] M. Kochańczyk, F. Grabowski, and T. Lipniacki. Super-spreading events initiated the exponential growth phase of COVID-19 with $\mathcal{R}_0$ higher than initially estimated. *Royal Society Open Science*, 7(9):200786, 2020.

[19] S. Linnainmaa. Taylor expansion of the accumulated rounding error. *BIT Numerical Mathematics*, 16(2):146–160, 1976.

[20] N. Liu, D. S. Oliver, et al. Evaluation of monte carlo methods for assessing uncertainty. *SPE Journal*, 8(02):188–195, 2003.

[21] L. Mosser, O. Dubrule, and M. J. Blunt. Deepflow: history matching in the space of deep generative models. *arXiv preprint arXiv:1905.05749*, 2019.

[22] D. S. Oliver and Y. Chen. Recent progress on reservoir history matching: a review. *Computational Geosciences*, 15(1):185–221, 2011.

[23] D. S. Oliver, N. He, and A. C. Reynolds. Conditioning permeability fields to pressure data. In *ECMOR V-5th European conference on the mathematics of oil recovery*, pages cp–101. European Association of Geoscientists & Engineers, 1996.

[24] D. S. Oliver, L. B. Cunha, and A. C. Reynolds. Markov chain monte carlo methods for conditioning a permeability field to pressure data. *Mathematical geology*, 29(1):61–91, 1997.

[25] D. S. Oliver, A. C. Reynolds, and N. Liu. *Inverse theory for petroleum reservoir characterization and history matching*. Cambridge University Press, 2008.

[26] C. Quilodrán-Casas, V. S. Silva, R. Arcucci, C. E. Heaney, Y. Guo, and C. C. Pain. Digital twins based on bidirectional lstm and gan for modelling the covid-19 pandemic. *arXiv preprint arXiv:2102.02664*, 2021.

[27] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[28] S. M. Razak and B. Jafarpour. History matching with generative adversarial networks. In *ECMOR XVII*, volume 2020, pages 1–17. European Association of Geoscientists & Engineers, 2020.

[29] V. L. Silva, C. E. Heaney, Y. Li, and C. C. Pain. Data Assimilation Predictive GAN (DA-PredGAN): applied to determine the spread of COVID-19. *arXiv preprint arXiv:2105.07729*, 2021.

[30] V. L. S. Silva, A. A. Emerick, P. Couto, and J. L. D. Alves. History matching and production optimization under uncertainties–application of closed-loop reservoir management. *Journal of Petroleum Science and Engineering*, 157:860–874, 2017.

[31] A. S. Stordal and G. Nævdal. A modified randomized maximum likelihood for improved bayesian history matching. *Computational Geosciences*, 22(1):29–41, 2018.

[32] B. Sudret, S. Marelli, and J. Wiart. Surrogate models for uncertainty quantification: An overview. In *2017 11th European conference on antennas and propagation (EUCAP)*, pages 793–797. IEEE, 2017.

[33] M. Tang, Y. Liu, and L. J. Durlofsky. Deep-learning-based surrogate flow modeling and geological parameterization for data assimilation in 3d subsurface flow. *Computer Methods in Applied Mechanics and Engineering*, 376:113636, 2021.

[34] A. Tarantola. *Inverse problem theory and methods for model parameter estimation*. SIAM, 2005.

[35] R. K. Tripathy and I. Bilionis. Deep uq: Learning deep neural network surrogate models for high dimensional uncertainty quantification. *Journal of computational physics*, 375:565–588, 2018.

[36] R. E. Wengert. A simple automatic derivative evaluation program. *Communications of the ACM*, 7(8):463–464, 1964.

[37] D. Xiao, C. E. Heaney, L. Mottet, F. Fang, W. Lin, I. M. Navon, Y.-K. Guo, O. K. Matar, A. G. Robins, and C. C. Pain. A reduced order model for turbulent flows in the urban environment using machine learning. *Building and Environment*, 148:323–337, 2019.

[38] Y. Xie, E. Franz, M. Chu, and N. Thuerey. tempoGAN: A temporally coherent, volumetric GAN for super-resolution fluid flow. *ACM Transactions on Graphics (TOG)*, 37(4):1–15, 2018.

[39] C. Yang, X. Yang, and X. Xiao. Data-driven projection method in fluid simulation. *Computer Animation and Virtual Worlds*, 27(3-4):415–424, 2016.

[40] V. Zantedeschi, D. De Martini, C. Tong, C. S. de Witt, A. Kalaitzis, M. Chantry, and D. Watson-Parris. Towards data-driven physics-informed global precipitation forecasting from satellite imagery. In *Proceedings of the AI for Earth Sciences Workshop at NeurIPS*, 2020.

[41] Z. Zhong, A. Y. Sun, and H. Jeong. Predicting CO_2 Plume Migration in Heterogeneous Formations using Conditional Deep Convolutional Generative Adversarial Network. *Water Resources Research*, 55(7):5830–5851, 2019.

[42] Y. Zhu, N. Zabaras, P.-S. Koutsourelakis, and P. Perdikaris. Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data. *Journal of Computational Physics*, 394:56–81, 2019.