
Adversarially trained LSTMs on reduced order models of urban air pollution simulations.

César Quilodrán-Casas*

Data Science Institute
Imperial College London
cesar.quilodran-casas13@imperial.ac.uk

Rossella Arcucci

Data Science Institute
Imperial College London
r.arcucci@imperial.ac.uk

Christopher Pain

Department of Earth Science & Engineering
Imperial College London
c.pain@imperial.ac.uk

Yike Guo

Data Science Institute
Imperial College London
y.guo@imperial.ac.uk

Abstract

This paper presents an approach to improve computational fluid dynamics simulations forecasts of air pollution using deep learning. Our method, which integrates Principal Components Analysis (PCA) and adversarial training, is a way to improve the forecast skill of reduced order models obtained from the original model solution. Once the reduced order model (ROM) is obtained via PCA, a Long Short-Term Memory network (LSTM) is adversarially trained on the ROM to make forecasts. Once trained, the adversarially trained LSTM outperforms a LSTM trained in a classical way. The study area is in London, including velocities and a concentration tracer that replicates a busy traffic junction. This adversarially trained LSTM-based approach is used on the ROM in order to produce faster forecasts of the air pollution tracer.

1 Introduction

Given the amount of data in Computational Fluid Dynamics (CFD) simulations, data-driven approaches can be seen as attractive solutions to produce reduced order models (ROMs). Furthermore, forecasts produced by these surrogates can be obtained at a fraction of the cost of the original CFD model solution when used together with a ROM. Recurrent neural networks (RNN) have been used to model and predict temporal dependencies between inputs and outputs of ROMs.

Non-intrusive ROMs and RNNs have been used together in previous studies, e.g. [Quilodrán Casas et al., 2020, Reddy et al., 2019]. The surrogate forecast systems can easily reproduce a time-step in the future accurately, when data from the original training is available [Wu et al., 2020]. However, the problem of these surrogates relies on the error propagation either produced by the ROM or the forecast system, especially when the forecasts wander off the training data. When the predicted output is used as an input for the prediction of the subsequent time sequence, the results can diverge quickly when encountering out-of-distribution data. In real-time applications, like urban air pollution, fast forecasts are urgently needed and an accurate system that can provide reliable forecasts is extremely useful.

*Corresponding author

A way to obtain reliable forecasts comes from adversarial losses via adversarial training. Generative adversarial networks (GANs) [Goodfellow et al., 2014] are a class of unsupervised machine learning algorithms. In essence, a GAN learns a generative model with the guidance of a discriminator model which is trained jointly, and the loss function provided by the discriminator is referred to as an adversarial loss. However, the idea of adversarial losses can also be applied to supervised scenarios and have advanced the state of the art in many fields over the past years [Dong and Yang, 2019, Wang et al., 2019]. Additionally, robustness may be achieved by detecting and rejecting adversarial examples by using adversarial training [Shafahi et al., 2019, Meng and Chen, 2017]. Data-driven modelling of nonlinear fluid flows incorporating adversarial networks have been successfully being studied previously [Cheng et al., 2020, Xie et al., 2018].

This extended abstract applies adversarial training to a LSTM network, based on a ROM of an urban air pollution simulation in an unstructured mesh. The robustness added by the adversarial training allows us to reduce the divergence of the forecast prediction over time, with similar execution times than a LSTM non-adversarially trained.

2 Methods

The methodology applies an adversarial training to a supervised ROM-based LSTM, in order to obtain reliable fast forecasts. The non-intrusive ROM is obtained via PCA [Arcucci et al., 2019]. Given that the CFD simulation used in this study contains $O(10^5)$ dimensions and it is in an unstructured mesh, there are two issues that arise if fast forecasts are needed.

Firstly, a simulation of this size requires the use of a supercomputer, due to processing power and memory constraint, in order to produce the next time-step. One way to reduce the dimensionality is using PCA, which decomposes the data into linear basis functions that describe the original problem data. The caveat of using PCA is that the dimension reduction comes from truncating the Principal Components (PC), and by doing so the retained variance decreases. However, for the purposes of the forecast only, only a truncated PCA dimension reduction is used here.

Secondly, a traditional 3D Convolutional Neural Network (CNN) is not trivial since the CFD simulation lies on a unstructured mesh rather than on a regular grid [Kim et al., 2019, Quilodr  n Casas et al., 2020].

The workflow is presented in Figure 1.

2.1 Principal Components Analysis

As described by Lever et al. [2017] PCA is an unsupervised learning method that simplifies high-dimensional data by transforming it into fewer dimensions. Let

$$\mathbf{x} = \{\mathbf{x}_t\}_{t=1,\dots,n} \quad (1)$$

$\mathbf{x} \in \mathbb{R}^{n \times m}$, with $n < m$, denotes the matrix of the model vectors at each time step. The PCA consists in decomposing this dataset as:

$$\mathbf{x} = \mathbf{P}\mathbf{\Pi} + \bar{\mathbf{x}} \quad (2)$$

$$\mathbf{x}_\tau = \mathbf{P}_\tau \mathbf{\Pi}_\tau + \bar{\mathbf{x}} \quad (3)$$

where $\mathbf{P} \in \mathbb{R}^{n \times n}$ are the principal components of \mathbf{x} ; $\mathbf{\Pi} \in \mathbb{R}^{n \times m}$ are the Empirical Orthogonal Functions; and $\bar{\mathbf{x}}$ is the mean vector of the model. The dimension reduction of the system comes from truncating \mathbf{P} at the first τ PCs, with $\mathbf{P}_\tau \in \mathbb{R}^{n \times \tau}$.

2.2 Adversarially trained long short-term memory

Once the truncated PCs are obtained, they can be used to train a LSTM [Hochreiter and Schmidhuber, 1997] to make predictions. In this paper, a vanilla LSTM network takes N

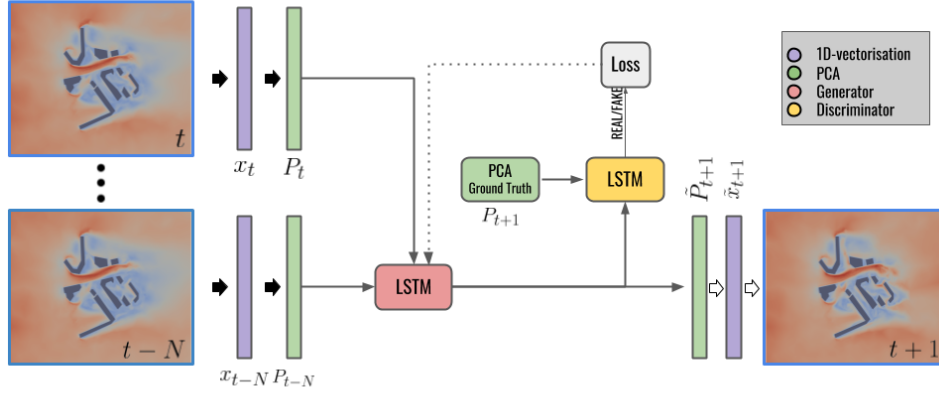


Figure 1: Proposed workflow for adversarial training of LSTM. The dimension-reduced data is used as an input in the LSTM, and its output is judged by the LSTM discriminator. The LSTM takes N previous time-steps to produce a forecast of $t + 1$. The vectorisation translates each point of the unstructured mesh onto a 1-dimensional vector \mathbf{x} . Then, \mathbf{P} are the principal components of \mathbf{x}

previous time-steps of $\mathbf{P}_{t-N}, \dots, \mathbf{P}_t$ as input and predicts an approximation of \mathbf{P}_{t+1} , named $\tilde{\mathbf{P}}_{t+1}$:

$$f^{LSTM} : \mathbf{P}_{t-N}, \dots, \mathbf{P}_t \rightarrow \tilde{\mathbf{P}}_{t+1}. \quad (4)$$

Our methodology proposes to add adversarial training to the LSTM network in order to further improve the forecasts.

The supervised adversarial training includes a discriminator to distinguish between the real samples of the Principal Components \mathbf{P} and predictions $\tilde{\mathbf{P}}$ produced by f^{LSTM} . The \mathbf{P} are fed to the discriminator as real sequences (ground truth). Let, $D(x, y)$ represent the discriminator function with an input x and a target label y such that, for $x = \mathbf{P}_{t+1}$, $y = 1$ and for $x = \tilde{\mathbf{P}}_{t+1}$, $y = 0$. The training of D is based on the minimisation of the binary cross-entropy loss (\mathcal{L}^{bce}), using the Nesterov Adam optimizer (Nadam) [Dozat, 2016]. The loss of f^{LSTM} also includes the mean squared error (mse) between $\tilde{\mathbf{P}}_{t+1}$ and \mathbf{P}_{t+1} .

The adversarial losses \mathcal{L}^{adv} for D and f^{LSTM} are then defined as:

$$\mathcal{L}_D^{adv}(\mathbf{P}_{t+1}) = \mathcal{L}^{bce}(D(\mathbf{P}_{t+1}, 1)) + \mathcal{L}^{bce}(D(f^{LSTM}(\mathbf{P}_{t-N}, \dots, \mathbf{P}_t), 0)) \quad (5)$$

$$\mathcal{L}_{f^{LSTM}}^{adv}(\mathbf{P}) = \mathcal{L}^{bce}(D(f^{LSTM}(\mathbf{P}_{t-N}, \dots, \mathbf{P}_t), 1)) + \mathcal{L}^{mse}(f^{LSTM}(\mathbf{P}_{t-N}, \dots, \mathbf{P}_t)) \quad (6)$$

3 Study area, model data and results

The computational fluid dynamics (CFD) simulations were carried out using Fluidity [Davies et al., 2011] (<http://fluidityproject.github.io/>). The study area is a 3D realistic representation of a part of South London. The dispersion of the pollution is described by the classic advection-diffusion equation (eq. (7)).

$$\frac{\partial c}{\partial t} + \nabla \cdot (\mathbf{u}c) = \nabla \cdot (\bar{\kappa} \nabla c) + F \quad (7)$$

where $\bar{\kappa}$ is the diffusivity tensor (m^2/s), c is the concentration and F represents the source terms ($\text{kg}/\text{m}^3/\text{s}$). The passive tracer is the transport of a scalar field representing a point source mimicking pollution in a traffic congested junction whose advection and diffusion depend on velocity, pressure and density.

The 3D case is composed of an unstructured mesh including $m = 100,040$ nodes per dimension and $n = 1500$ time-steps. The wind profile of the atmospheric boundary layer is represented by a log-profile velocity. The top and the sides of the model domain have a perfect slip boundary condition, while the facades of the buildings and the bottom of the model domain have a no-slip boundary. The pollution background is modelled as a sinusoidal function. This background pollution mimics waves of pollution in an urban environment.

4 Experiments and results

Four experiments were set up to assess the improved forecast of the adversarially trained LSTM. A PCA was applied to two output fields from the CFD simulation: Tracer (1-dimensional, unitless), and Velocity (3-dimensional, ms^{-1}). To each of these set of PCs, a truncation of 64 and 128 PCs was applied which explains over 90% of variance in each case. The truncation to 64 PCs and 128 PCs reduces the size of the dataset by 4 and 3 orders of magnitude, respectively. Only 90% of the data is used for training, and 10% is used for validation.

Firstly, a grid search for optimal hyperparameters was set up for each of these experiments, including dropouts, hidden nodes in the LSTM, batch sizes, activation function of the output layer, and lags. Once the optimal set of hyperparameters for each experiment is found, the trained LSTM (named $LSTM^{classic}$) is saved.

Secondly, the adversarially trained LSTM (named $LSTM^{adv}$) is setup with the same optimal hyperparameters found for each combination. The adversarial training includes a discriminator D . The discriminator D is a LSTM that takes $\tilde{\mathbf{P}}$ and outputs between 0 and 1 (sigmoid function) as described in Section 2.2. The hyperparameters used to train the LSTM networks for each experiment are summarised in table 1. The adversarial training adds the same mirrored architecture as a discriminator D with an output dense layer of size 1 with a sigmoid activation function, and binary cross-entropy as loss. Both $LSTM^{classic}$ and $LSTM^{adv}$ were trained for 5000 epochs with a piece-wise mean squared error loss.

Table 1: Hyperparameters for each experiment. AF is the activation function in the output layer, and Time-lag is the number of time-steps used to predict the following one.

| Experiments | Batch size | Hidden nodes LSTM | Dropout | AF | Time-lag |
|-----------------------|------------|-------------------|---------|---------|----------|
| Tracer $\tau = 64$ | 32 | 256 | 0.5 | ReLU | 2 |
| Tracer $\tau = 128$ | 32 | 256 | 0.3 | Sigmoid | 2 |
| Velocity $\tau = 64$ | 32 | 128 | 0.5 | ReLU | 2 |
| Velocity $\tau = 128$ | 32 | 256 | 0.5 | Sigmoid | 2 |

Figure 2 presents the error from an ensemble of forecasts, of velocities in X, Y, Z and Tracer, starting from different time-steps. The solid line represents the mean of the error ensemble and the shaded area is the standard deviation from the mean. The red shaded area is $LSTM^{adv}$, while the blue shaded area is $LSTM^{classic}$. The forecasts are created by using previous time-steps from data and producing a forecast. This forecast is subsequently used as an input for the prediction of the next time-step. After 50 iterations, it is very clear that $LSTM^{adv}$ outperforms $LSTM^{classic}$. The forecasts are 4 orders of magnitude faster than the CFD simulation. Table 2 shows further results for all the experiments.

Table 2: Forecast improvement, error reduction in % of $LSTM^{adv}$ over $LSTM^{classic}$.

| PCs | Within training data | | | | Within validation data | | | |
|--------------|----------------------|-------|-------|------------|------------------------|-------|-------|------------|
| | Velocity | | | Tracer (c) | Velocity | | | Tracer (c) |
| | X | Y | Z | | X | Y | Z | |
| $\tau = 64$ | 21.07 | 22.21 | 83.32 | 18.95 | 11.90 | 12.14 | 13.52 | 9.01 |
| $\tau = 128$ | 51.09 | 51.39 | 49.13 | 56.29 | 8.13 | 8.18 | 10.02 | 6.46 |

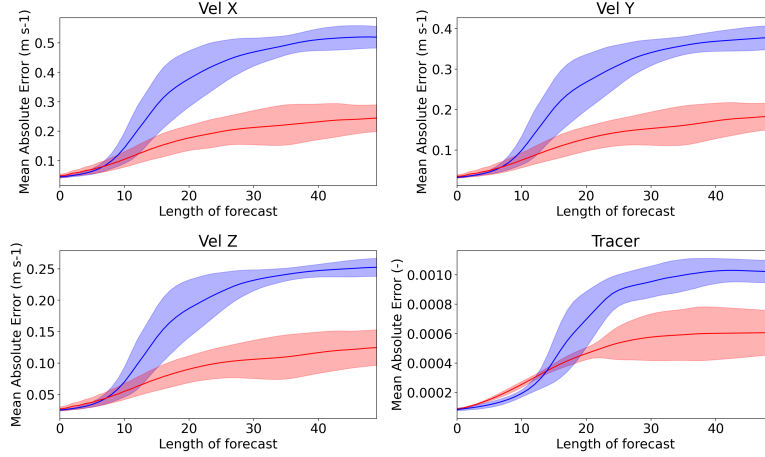


Figure 2: Comparison of $LSTM^{classic}$ (blue) and $LSTM^{adv}$ (red). The shaded areas show an ensemble of errors of 50 time-step long forecast from different starting points from $t = 350$ to $t = 400$ (within training data) with 128 PCs. The solid line is the mean and the shaded area is one standard deviation from the mean.

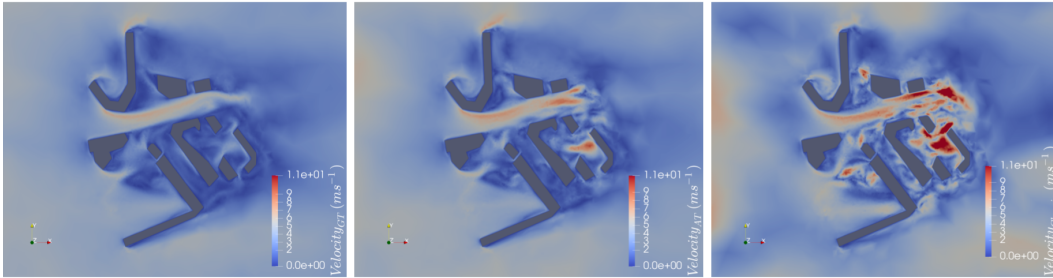


Figure 3: Comparison of forecasted velocity in ms^{-1} (magnitude) fields by $LSTM^{adv}$ and $LSTM^{classic}$ with 64 PCs. This is a 25 time-step forecast starting from $t = 350$. Left: Ground truth at $t = 375$, Middle: $LSTM^{adv}$ forecasted 25 time-steps from $t = 350$, Right: $LSTM^{classic}$ forecasted 25 time-steps from $t = 350$.

Figure 3 shows the comparison of forecasted magnitude velocity (in ms^{-1}) fields of $LSTM^{adv}$ and $LSTM^{classic}$ using 64 PCs from $t = 350$. The snapshots show clearly that after 25 time-steps of forecasting, $LSTM^{classic}$ diverges quickly from the underlying model state, while $LSTM^{adv}$ preserve more underlying physics.

5 Summary and future work

This paper presented an adversarially trained LSTM that improves the forecast of a LSTM trained in a classical way. This is important when accurate near real-time predictions are needed and not enough data is available. It can be observed that adversarially trained LSTM does not diverge greatly from the data it has learned, given the constraint of the discriminator network.

The replacement of the CFD solution by these models will speed up the forecast process towards a real-time solution. And, with the application of adversarial training could potentially produce more physically realistic flows. Future work will apply the same methodology to different dimension reduction schemes. Furthermore, this framework is data-agnostic and could be applied to different CFD models where enough data is available.

Acknowledgements

This work is supported by the EPSRC Grand Challenge grant ‘Managing Air for Green Inner Cities (MAGIC) EP/N010221/1, the EPSRC Centre for Mathematics of Precision Healthcare EP/N0145291/1 and the EP/T003189/1 Health assessment across biological length scales for personal pollution exposure and its mitigation (INHALE). Thanks to Dr. Laetitia Mottet for the set up of the full model in Fluidity.

Broader Impact

In this paper, researchers introduced an adversarially trained LSTM to improve the forecast of reduced order models of urban air pollution. The impact of this research is likely to drive the development of improved urban air pollution forecasts, at a fraction of the computational costs but equally as accurate. Our framework comes from the urgency to create data-driven real-time forecasts from expensive computational fluid dynamics simulations. Furthermore, our framework is data-agnostic and can be applied to different variables or simulations where there is enough available data.

References

- R. Arcucci, C. Quilodr  n Casas, D. Xiao, L. Mottet, F. Fang, P. Wu, C. C. Pain, and Y.-K. Guo. A Domain Decomposition Reduced Order Model with Data Assimilation (DD-RODA). In *PARCO*, pages 189–198, 2019.
- 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.
- D. R. Davies, C. R. Wilson, and S. C. Kramer. Fluidity: A fully unstructured anisotropic adaptive mesh computational modeling framework for geodynamics. *Geochemistry, Geophysics, Geosystems*, 12(6), 2011.
- H.-W. Dong and Y.-H. Yang. Towards a deeper understanding of adversarial losses. *arXiv preprint arXiv:1901.08753*, 2019.
- T. Dozat. Incorporating Nesterov momentum into Adam. 2016.
- I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.
- S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.
- B. Kim, V. C. Azevedo, N. Thuerey, T. Kim, M. Gross, and B. Solenthaler. Deep fluids: A generative network for parameterized fluid simulations. In *Computer Graphics Forum*, volume 38, pages 59–70. Wiley Online Library, 2019.
- J. Lever, M. Krzywinski, and N. Altman. Points of significance: Principal component analysis, 2017.
- D. Meng and H. Chen. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pages 135–147, 2017.
- C. Quilodr  n Casas, R. Arcucci, and Y. Guo. Urban air pollution forecasts generated from latent space representation. In *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*, 2020.
- C. Quilodr  n Casas, R. Arcucci, P. Wu, C. Pain, and Y.-K. Guo. A Reduced Order Deep Data Assimilation model. *Physica D: Nonlinear Phenomena*, 412:132615, 2020.

- S. B. Reddy, A. R. Magee, R. K. Jaiman, J. Liu, W. Xu, A. Choudhary, and A. Hussain. Reduced order model for unsteady fluid flows via recurrent neural networks. In *International Conference on Offshore Mechanics and Arctic Engineering*, volume 58776, page V002T08A007. American Society of Mechanical Engineers, 2019.
- A. Shafahi, M. Najibi, M. A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein. Adversarial training for free! In *Advances in Neural Information Processing Systems*, pages 3358–3369, 2019.
- D. Wang, C. Gong, and Q. Liu. Improving neural language modeling via adversarial training. *arXiv preprint arXiv:1906.03805*, 2019.
- P. Wu, J. Sun, X. Chang, W. Zhang, R. Arcucci, Y. Guo, and C. C. Pain. Data-driven reduced order model with temporal convolutional neural network. *Computer Methods in Applied Mechanics and Engineering*, 360:112766, 2020.
- 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.