Development of generative adversarial networks for spatiotemporal fluid flow, atmospheric and flood predictions

A PhD Dissertation

Presented to

Faculty of Engineering

By

Meiling Cheng

Supervised by:

Dr. Fangxin Fang

Prof. Christopher Pain

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy in the

Department of Earth Science and Engineering

Imperial College London

April 2022

Abstract

Accurate prediction of fluid flow is of vital importance to many applications in engineering and physics. Long-standing challenges in flow simulations are the simultaneous high accuracy and efficiency demanded in numerical computations. Recent advances in machine learning technologies are increasingly of interest for the efficient simulation of nonlinear and complex systems. Deep learning techniques are capable of capturing the physical dynamics without prior knowledge of underlying physical relationships.

However, there remain several challenges in nonlinear fluid flow modelling, i.e., spatiotemporal nonlinear fluid flow modelling, large data-driven fluid flow modelling, real-time forecasting of nonlinear fluid flows beyond the training period, improvement of the forecasting accuracy in long lead-time. To overcome these challenges, in this work, Generative Adversarial Networks (GANs) have been first introduced to spatiotemporal nonlinear fluid flow prediction while data assimilation techniques are used for improving the accuracy of long term forecasting.

The key contributions of this thesis are: firstly, an Artificial Intelligence (AI) fluid model based on a deep convolutional generative adversarial network (DCGAN) has been developed for modelling spatiotemporal flow distributions. Secondly, a hybrid deep adversarial autoencoder (VAE-GAN) model to integrate generative adversarial network (GAN) and variational autoencoder (VAE) has been proposed for large data-driven nonlinear fluid flow prediction. Thirdly, a real-time predictive machine learning model, i.e., artificial neural network (ANN), long short term memory (LSTM), DCGAN, and VAE-GAN has been developed for basin streamflow, urban flooding, and national air pollution forecasting problems respectively. Fourthly, an ensemble Kalman filter (EnKF) for GAN and convolutional LSTM (GAN-ConvLSTM) based forecasting system has been proposed for accurate long lead-time forecasting.

The presented models have been validated by various test cases and the results are in good agreement with high-fidelity models and observations. Promising results have shown that the DCGAN and VAE-GAN models are capable of accurately predicting the spatiotemporal flow features as the flow evolves, with CPU speed-up of several orders of magnitude.

In addition, the ANN, LSTM, DCGAN and VAE-GAN models have demonstrated to provide efficient and accurate forecasts for a long lead-time in applications of streamflow, flooding and ozone forecasting in spatial and temporal spaces. Finally, the two-hybrid forecast models (DCGAN-EnKF and ConvLSTM-EnKF) are able to yield long lead-time forecasts of dynamic states and the use of EnKF in ConvLSTM and DCGAN models successfully corrects online model errors and significantly improves the real-time forecasting of dynamic systems.

Copyright Declaration

The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work.

Declaration

I hereby declare that this thesis entitled "**Development of generative adversarial networks for spatiotemporal fluid flow, atmospheric and flood predictions**" is a genuine research work carried out by me under the guidance of Dr. Fangxin Fang and Prof. Christopher Pain, Imperial College London, Department of Earth Science and Engineering, AMCG - Applied Modelling & Computation Group. Text and results obtained from other sources are referenced and properly acknowledged.

Meiling Cheng

Acknowledgements

I would like to express my deepest gratitude to my principal supervisor, Dr. Fangxin Fang. She is a very responsible and patient teacher, and her provident vision, rigorous attitude, and strong enthusiasm for scientific research have deeply inspired and guided me for a lifetime. She creates a relaxed atmosphere for research and discussions, and continuously encourages me to take every chance to improve and move forward. I am extremely grateful for her friendship, care, and great support for me in all struggle times. In addition, my sincere thanks to my second supervisor Prof. Christopher Pain for all the help, support and encouragement he's given me. I would like to thank Prof. Ionel Michael Navon for his valuable suggestions after promptly reading and revising my manuscripts.

I am very grateful for the financial support for my PhD from China Scholarship Council. It provided me the opportunity to study at Imperial College London. Without the funding support, I would not have had the chance to chase my dream and do my PhD research.

I would also like to extend my sincere appreciation to friends and colleagues throughout my PhD. Special thanks go to Rong Hu and Ying Gao for their continuous help and patient suggestions on my study and life. I cherish our friendship and shall never forget the time we spent together. In addition, I would like to thank Wenzhuo Cao, Dunhui Xiao, Pan Yang, and Linfeng Li, for giving me help, support and company in London.

Finally, I would like to thank my father Mr. Yongbin Cheng, my mother Mrs. Sanrong Zuo and families for their encouragement and love at all times. A very special thank to Mr. Wenli Shi, who has always been supportive and encouraged me in these years. Without his love, I cannot imagine how my life would be on this adventurous journey. Thanks for being in my life and making it special.

Contents

Abstract i

Copyright Declaration iii

Declaration iv

Acknowledgements v

1 Introduction 1

1.1 Research motivation and objectives 1

1.2 Literature review of machine learning for spatiotemporal prediction . . . 4

1.2.1 Machine Learning methods 4

1.2.1.1 Deep neural networks 4

1.2.1.2 Generative adversarial networks (GANs) 8

1.2.2 Challenges of machine learning methods in fluid dynamics modelling 9

1.2.2.1 Spatiotemporal fluid flow modelling 9

1.2.2.2 Large data-driven prediction 11

1.2.2.3 Real-time forecasting 12

1.2.2.4 Accuracy improvement in real-time forecasting 14

1.3 Original contributions 17

1.4 Thesis outline 19

2 Development of a deep convolutional generative adversarial network for nonlinear spatiotemporal fluid flows 22

2.1 Introduction 23

2.2 Governing equations for nonlinear fluid flow prediction problems 24

2.3 Adversarial network architecture 25

2.4 Deep convolutional GAN for spatiotemporal data-driven modelling 28

2.4.1 Spatiotemporal datasets for deep convolutional networks 28

2.4.1.1 Input and output datasets 28

2.4.1.2 Scaled datasets for deep convolutional networks 29

2.4.2 Forward propagation 30

2.4.2.1 Convolutional neural network in simulations of spatiotemporal flows 30

2.4.2.2 CNNs in the generator and discriminator 31

2.4.3 Backward propagation 32

2.4.4 Training and predictive processes 33

2.5 Numerical examples 34
2.5.1 Descriptions of study cases 34
2.5.2 Data collections and predictive analysis 37
2.5.3 Performance assessment 38
2.6 Results 39
2.6.1 Case 1: Wave phase variation 39
2.6.2 Case 2: Wave peak variation 43
2.6.3 Computational efficiency 47
2.7 Conclusion 48

3 Development of an advanced hybrid deep adversarial autoencoder for large data-driven nonlinear fluid flow modelling 50

3.1 Introduction 51
3.2 Governing equations 52
3.3 Variational Autoencoder for model reduction 53
3.4 Hybrid deep adversarial autoencoder for nonlinear fluid flow modelling . 55
3.4.1 Hybrid deep adversarial autoencoder 56
3.4.1.1 Model architecture 57
3.4.1.2 Loss function 60
3.4.1.3 Offline training and online predictive processes 62
3.5 Numerical examples 63

3.5.1 Case 1: flow past a cylinder 65
3.5.1.1 Sensitivity analysis 65
3.5.1.2 Model prediction 68
3.5.2 Case 2: water column collapse 69
3.5.2.1 Model application 71
3.5.2.2 Model prediction 72
3.5.3 Model accuracy and efficiency of hybrid VAE-GAN 78
3.6 Conclusion 79

4 Long lead-time streamflow forecasting using machine learning methods 81

4.1 Introduction 82
4.2 Methodology 83
4.2.1 Multi-step-ahead daily and monthly streamflow forecast strategy . 83
4.2.1.1 One-step-ahead daily and monthly streamflow forecast . 83
4.2.1.2 Multi-step-ahead daily and monthly streamflow forecast 84
4.2.2 Artificial neural network for daily and monthly streamflow forecast 87
4.2.3 Long short-term memory model (LSTM) for daily and monthly streamflow forecast 89
4.3 Application 90
4.3.1 Study area 90
4.3.2 Daily and monthly model development 91

4.3.3 Model performance evaluation 92

4.4 Results and Discussions 93

4.4.1 Daily streamflow forecast 94

4.4.1.1 Modelling parameter optimization in daily streamflow forecast 94

4.4.1.2 One-day-ahead streamflow forecast 95

4.4.1.3 Multi-day-ahead streamflow forecast 97

4.4.2 Monthly streamflow forecast 100

4.4.2.1 Modelling parameter optimization in monthly streamflow forecast 100

4.4.2.2 One-month-ahead streamflow forecast 102

4.4.2.3 Multi-month-ahead streamflow forecasting 103

4.5 Conclusion 106

5 A real-time flooding forecasting using deep convolutional GAN and application to Denmark 108

5.1 Introduction 109

5.2 Spatiotemporal flood forecasting problem 110

5.3 Deep convolutional GAN for real-time flood forecasting 111

5.3.1 Dynamic flow learning of DCGAN 113

5.3.2 Real-time forecast of DCGAN 115

5.3.3 Model evaluation 117

5.4 Application and results 118
5.4.1 Study area 118
5.4.2 Uncertainty quantification 121
5.4.3 Real-time flooding forecast 124
5.5 Conclusion 129

6 Spatiotemporal hourly and daily ozone forecasting in China using GAN and ConvLSTM 131

6.1 Introduction 132
6.2 Data 133
6.2.1 Reanalysis data 133
6.2.2 Monitoring and physical modelling data 135
6.3 Spatiotemporal ozone simulation 136
6.3.1 Method 136
6.3.1.1 ConvLSTM 136
6.3.1.2 DCGAN and LDCGAN 137
6.3.2 Results 139
6.3.2.1 Data processing 139
6.3.2.2 Model sensitivity analysis 142
6.3.2.3 Spatiotemporal ozone prediction analysis 146
6.3.2.4 Machine learning-based ozone prediction in comparison to physical air pollution modelling and observations 153

6.4 Real-time forecasting . . . 157
6.4.1 Methods . . . 157
6.4.1.1 Dynamic learning . . . 160
6.4.1.2 Real-time forecasting . . . 163
6.4.2 Results . . . 165
6.4.2.1 Data processing . . . 165
6.4.2.2 Hourly ozone forecasting analysis over China in summer and winter . . . 166
6.4.2.3 Comparison of hourly ozone forecasting between VAE-GAN and the physical model-NAQPMS . . . 171
6.4.2.4 Daily ozone forecasting analysis over China . . . 177
6.4.2.5 Impact of meteorological factors . . . 182
6.5 Conclusions . . . 186

7 Ensemble Kalman filter for GAN-ConvLSTM based long lead-time forecasting 187

7.1 Introduction . . . 188
7.2 Ensemble Kalman filter . . . 189
7.3 Long lead-time forecasting using hybrid machine learning models and EnKF . . . 191
7.3.1 Multi-Input Multi-Output (MIMO) forecast using ConvLSTM and DCGAN . . . 193

7.3.2 Long lead-time forecast using ConvLSTM and DCGAN 194

7.3.3 Extension of forecasting horizon integrating ConvLSTM and DC-GAN with EnKF 195

7.4 Numerical examples 197

7.4.1 Model setup 198

7.4.2 One MIMO step forecast 199

7.4.3 Long lead-time forecasts with ML-EnKF 203

7.5 Conclusions 211

8 Conclusion and Future work **212**

8.1 Conclusions 212

8.2 Future Work 215

8.2.1 Combining machine learning with physical models 215

8.2.2 Machine learning for problems with multi-inputs 216

8.2.3 Realistic applications 217

8.2.4 Machine learning for unresolved subgrid parametrization 218

Bibliography **218**

Appendix A Long lead-time daily and monthly streamflow forecasting using machine learning methods **250**

A.1 Supplementary information related to structure of ANN and LSTM models 250

A.2 Multi-day-ahead streamflow forecast at PRB 252

A.2.1 Study area 252

A.2.2 Multi-day-ahead streamflow forecast 253

Appendix B Spatio-temporal hourly and daily ozone forecasting in China using a hybrid machine learning model: Autoencoder and Generative Adversarial Networks 257

B.1 Model comparison 257

B.2 Ozone forecasting using NO_2 259

B.2.1 Hourly ozone forecasting using NO_2 from reanalysis datasets . . 260

B.2.2 Impact of NO_2 emissions on ozone forecasting 262

Appendix C Software and error analysis 265

C.1 Fluidity 265

C.2 MIKE 3 FM 266

C.3 Error analysis 266

List of Figures

1.1 Thesis structure and inter-relationships between individual chapters. . . . 19

2.1 Deep convolutional generative adversarial network architecture. The ϕ and ϕ_d can be represented by fluid flow solutions (e.g. water depth). . . . 27

2.2 Overview of the DCGAN in offline training process and online predictive process of constructing fluid fields. 34

2.3 Input varied waves for two cases: (a) representing the wave phase variation and (b) representing the wave height variation. 36

2.4 Computational domain area and unstructured mesh (the incoming wave is the boundary condition along the costal line, and the simulation starts with a static state). 37

2.5 Comparison of the spatial distribution of water depth fields obtained from the DCGAN (left) and the original high fidelity model (right) at timesteps $60, 90, 216, 441$. 41

2.6 Differences of water depth fields between the DCGAN and the original high fidelity model at timesteps $60, 90, 216, 441$. 42

2.7 Comparison of the temporal variation of water depth at points ID = 2597, 2637, 3082, 5604. 42

2.8 The RMSE and correlation coefficients of water depth between the generated fields and original high fidelity fields during whole simulational period. 43

2.9 Comparison of spatial distribution of water depth fields obtained from the DCGAN (left) and the original high fidelity model (right) at timesteps $50, 95, 205, 493$. 45

2.10 The correlation coefficients of water depth solutions between the DCGAN and the original high fidelity model at timesteps $50, 95, 205, 493$. 46

2.11 Comparison of water depth solutions between the DCGAN and the original high fidelity model at four points during whole simulational period. . 46

2.12 The RMSE and correlation coefficients of water depth solutions between the DCGAN and the original high fidelity model during whole simulational period. 47

3.1 Illustration of a VAE model. Yellow lines denotes the generative process in a probabilistic decoder $p_\theta(\hbar|\zeta)$, blue lines represent the variational approximation $q_\phi(\zeta|\hbar)$ in a probabilistic encoder to the intractable posterior $p_\theta(\hbar|\zeta)$. ϕ and θ are parameters which are learned jointly (see Kingma and Welling (2013)). 55

3.2 (a) Illustration of a hybrid deep adversarial autoencoder (VAE-GAN), which consists of an encoder (acted as a generator) (b), a discriminator (c), and a decoder (d). The encoder and decoder compose of convolutional layers and full-connected layers. The discriminator is a stack of full-connected layers. 57

3.3 The offline training and online predictive processes. 63

3.4 The sketch of flow past a cylinder (mesh size ranges from 0.005 to 0.05) and the settings of boundary velocities. 66

3.5 The RMSE of velocity between the generated fields and original high fidelity fields during whole simulational period. 67

3.6 The relative error of velocity solutions between the generated fields and original high fidelity fields during whole simulational period. 68

3.7 Comparison of the spatial distribution of velocity fields obtained from the hybrid VAE-GAN (left) and the original high fidelity model (right) at time levels $t = 1.8, 4, 5.6, 9.2s$. 69

3.8 The sketch of water collapse experiment, where bottom to up and right to left have denser meshes. The mesh size ranges from 0.003 to $0.2m$. 70

3.9 The settings of length and height in water collapse experiment. 71

3.10 Comparison of the spatial distribution of pressure fields obtained from the hybrid VAE-GAN (left) and the original high fidelity model (right) at time levels $t = 0.2, 0.65, 1.275, 1.675s$. 73

3.11 Differences and correlation coefficients of pressure fields between the hybrid VAE-GAN and the original high fidelity model at time levels $t = 0.2, 0.65, 1.275, 1.675s$. 74

3.12 Comparison of the temporal variation of pressure at points ID = 7452, 7596, 9315, 9865, 12798, 13267, 16428, 16524. 76

3.13 The correlation coefficients and RMSE of pressure between the generated fields and original high fidelity fields during whole simulational period. . 77

3.14 The correlation coefficients and RMSE of pressure in time series in the computational domain area Ω. 78

4.1 Model forecasting strategy using a recursive way at daily and monthly scales. 86

4.2 Location of the Nan River Basin (NRB) and Ping River Basin (PRB). . . 91

4.3 Impact of the length of the time lag on streamflow daily forecasting, where the RMSE of streamflow results using (a) ANN, and (b) LSTM. . . 95

4.4 Impact of the maximum number of epochs on streamflow daily forecasting, where the RMSE of streamflow results using (a) ANN, and (b) LSTM. 95

4.5 Comparison of forecasting and observed daily streamflow using the ANN and LSTM models with the optimal parameters in Table 4.3. 96

4.6 Forecast performance at the lead-times 1, 2, 3, 4 and 5 days: (a) with the ANN model and (b) with the LSTM model. 99

4.7 Forecast performance with the ANN and LSTM models over the validation period of year 2007 to 2011 at the lead-times of 1-20 days: (a) RMSE, (b) NSE, (c) CC and (d) MAE. 100

4.8 Impact of the time lag on streamflow monthly forecasting, where the RMSE of streamflow results using (a) ANN, and (b) LSTM. 102

4.9 Impact of the maximum number of epochs on streamflow monthly forecasting, where the RMSE of streamflow results using (a) ANN, and (b) LSTM. 102

4.10 Comparison of forecasting and observed monthly streamflow using the ANN and LSTM models with the optimal parameters in Table C.1. 103

4.11 Forecast performance at lead-times 1, 2 and 3 months: (a) with the ANN model and (b) with the LSTM model. 104

4.12 Forecast performance with the ANN and LSTM models at lead-times of 1-12 months: (a) RMSE, (b) NSE, (c) CC and (d) MAE. 106

5.1 The schematic diagram of the dynamic flow learning and forecast processes (taking the wave as an example) 112

5.2 Illustration of the dynamic flow learning system: a deep convolutional GAN (DCGAN), which consists of a generator and a discriminator. The generator and discriminator are composed of convolutional layers and full-connected layers. The combination of the loss functions in two modules forms the adversarial training. 113

5.3 Illustration of the real-time forecast of the DCGAN. The forecast of flood fields employs a cascade manner, i.e. the last one-timestep-ahead forecast from the DCGAN can act as a new input for the next step-ahead forecast, which forms a long lead-time forecast in a recursive way. 117

5.4 Study area in Greve, Municipality of Denmark (left side see Hu et al. (2019b)) 119

5.5 Unstructured mesh of study area and extreme sea water level along the coastal line, where the mesh is composed by 25203 elements, 38437 faces, 13235 nodes and 2 sections (triangular and rectangular). 120

5.6 Different types of incoming waves obtained from the Gaussian distribution of the extreme sea water level (the solid line representing the 2100 upper sea flood level in Fig. 5.5) from 0 to $24h$. (a) represents 15 types of incoming waves, and (b) and (c) are the Gaussian distribution of water depths at $t = 6h$ and $12.5h$, respectively. 121

5.7 Comparison of the spatial distribution of predictive water depth obtained from the DCGAN and the original high fidelity model at $t = 5, 8, 14, 18h$. Here the rectangles regions are the urban cities with dense buildings. . . . 123

5.8 The correlation coefficients of water depth solutions between the DCGAN and the original high fidelity model at different time levels: (a) $t = 5h$, (b) $t = 8h$, (c) $t = 14h$, and (d) $t = 18h$. 124

5.9 Comparison of the temporal distribution of predictive water depth obtained from the DCGAN and the original high fidelity model at eight observed points, where the water depth from the high fidelity model is taken as the real (original) depth while the predictive depth is referred as the results from the DCGAN. 126

5.10 Comparison of the spatial distribution of predictive water depth obtained from the DCGAN and the original high fidelity model at time steps $t = 15.5, 17.9, 22.8h$. 127

5.11 Forecast performance with the DCGAN during the long lead-time of the computational domain area Ω: (a) RMSE, (b) NSE, (c) MAE and (d) R. . 128

6.1 Spatial distribution of the average hourly temperature and ozone concentration in China. (a) and (b) June; (c) and (d) December in 2018. 135

6.2 Variable importance plot using the random forest model in predicting the ozone concentrations over the BTH region at hourly and daily scales, respectively. 141

6.3 Schematic of real-time ozone concentration predictive strategy by using ConvLSTM, DCGAN and LDCGAN. (a) Selection of the spatiotemporal meteorological variables (including temperature, humidity and wind velocity) over the BTH region as model inputs and ozone as model output. (b)(c)(d) Architectures of the ConvLSTM, DCGAN and LDCGAN models for ozone prediction. The LDCGAN model is an improved version of the DCGAN model with an additional loss function, details shown in section 6.3.1.2. (e) Uncertainty and heterogeneity analysis of spatiotemporal ozone prediction in the model validation period. 142

6.4 Performance evaluation of the three machine learning models in: (a) and (b) temporal (hourly) variation of spatial-averaged predicted ozone concentrations, and (c) spatial distribution of hourly-averaged predicted ozone concentrations at a high spatial resolution of $15km \times 15km$, which are assessed by RMSE, the standard deviation, correlation in Taylor diagrams and cumulative density functions (CDF) for the RMSE, where multiple meteorological factors are considered in case 1 (temperature), case 2 (temperature and humidity) and case 3 (temperature, humidity and wind velocity). 145

6.5 Temporal variation of hourly predictive ozone concentration and uncertainty analysis in 2018 by the ConvLSTM, DCGAN and LDCGAN models. (a) Temporal variation of spatial-averaged variation hourly ozone concentration over the BTH region using three machine learning-based models and reanalysis data (as reference). (b) Correlation coefficients of predictive results from the three machine learning-based models compared to the hourly reanalysis data in 2018. (c) Biases of predictive results from the three machine learning-based models compared to the hourly reanalysis data in 2018. 148

6.6 Spatial variation of hourly predictive ozone concentrations and error analysis by the ConvLSTM, DCGAN and LDCGAN models. (a) Spatial distribution of temporal-averaged variation hourly ozone concentration over the BTH region in four seasons and the whole year of 2018 by three models. (b) Relative errors of temporal-averaged predictive results from the three machine learning-based models compared to the reanalysis data in four seasons and the whole year. Note that the spatial resolution is $15km \times 15km$. 149

6.7 Same as Fig. 6.5, except for daily ozone concentration prediction. 151

6.8 Same as Fig. 6.6, except for daily ozone concentration prediction. 152

6.9 The spatial distribution and RMSE of predictive results from the ConvLSTM, DCGAN and LDCGAN and physical model compared to the reanalysis data in the predictive period of June in 2018. (a) RMSE of predictive results from the three machine learning-based models and physical model compared to the hourly reanalysis data in June of 2018; (b) The predictive ozone distributions using ConvLSTM, DCGAN, LDCGAN and physical model compared to the ozone field from reanalysis data on 12:00 13 June of 2018. Note that the spatial resolution is $15km \times 15km$. 154

6.10 Ozone time series comparison at the eleven main cities (Beijing, Tianjin, Zhangjiakou, Chengde, Langfang, Tangshan, Baoding, Cangzhou, Shijiazhuang, Xingtai and Hengshui) over the BTH region in the predictive period of June 2018, where the station observations at the eleven main cities are used as the 'true' references. 156

6.11 Error analysis of results from the reanalysis data, physical model, ConvLSTM, DCGAN and LDCGAN models, where the observations in June of 2018 are taken as references (true values). (a) Correlation coefficient and (b) RMSE of predictive results from reanalysis data, physical model, ConvLSTM, DCGAN and LDCGAN models compared to the monitoring observations in eleven cities over the BTH region. 157

6.12 Schematic of real-time ozone forecasting. The architecture consists of two processes: dynamic learning and real-time forecasting. In real-time forecasting process, one-step ozone forecast can be obtained by a direct prediction for one-step (hourly/daily) lead-time. To get forecasts for a long lead-time, a recursive strategy is adopted, which uses the last one-step ozone forecast as a new input to predict the next-step ozone concentration. 159

6.13 Comparison of the spatial distribution of ozone fields obtained from the VAE-GAN model (left) and the reanalysis data (middle) at lead-times of $30h$ and $60h$ in June and December in 2018. 168

6.14 The spatial-averaged RMSEs and correlation coefficients of ozone concentration between the VAE-GAN and the reanalysis data in June and December in 2018. 170

6.15 The spatial distribution of temporal-averaged RMSEs and the comparison of the cumulative density functions (CDF) for the temporal-averaged RMSEs in June and December in 2018. 170

6.16 Comparison of the spatial distribution of ozone fields obtained from the VAE-GAN model (right), physical model-NAQPMS (middle) and the reanalysis data (left) at lead-times $t = 40, 60h$ in June of 2018 respectively. 172

6.17 The spatial-averaged RMSE of predictive results from the VAE-GAN and physical model-NAQPMS during predictive period of June in 2018. There are six regions: the North China Plain (NCP), Northeast China (NE), Southwest China (SW), Southeast China (SE), Northwest China (NW), and Central China (refer to Kong et al. (2021)). The reanalysis data is used as 'true' reference. 173

6.18 Ozone time series at the cities in three densely populated regions (Jing-Jin-Ji, Yangtze River Delta, and Pearl River Delta) using the observations, reanalysis data, physical model-NAQPMS, VAE-GAN and VAE-GAN-LSTM models. 176

6.19 A comparison of the temporal variation of spatial-average daily ozone concentrations from the VAE-GAN model and the reanalysis data in 2018. 179

6.20 A comparison of the spatial distribution of temporal-average daily ozone concentrations from the VAE-GAN model and the reanalysis data in the JJJ, YRD and PRD regions. 181

6.21 A relative error histograms of the temporal-average daily ozone concentrations between the VAE-GAN model and the reanalysis data over the whole year of 2018. The blue bins aggregate the respective grid points over three regions. Numbers on top of the bins are percentage values with respect to the total number of grid points. Yellow numbers indicate the area proportion of the globe, where values of relative error are within the $\pm 10\%$ interval. 182

6.22 Predictive daily results from the cases of VAE-GAN and DCGAN model with/without the introduction of the meteorological factors into the model inputs, where the densely populated JJJ region in China is selected as the study area and the predictive period is from the year 2018 to 2019. Note that in the case of no meteorological factors the models only take the last-timestep ozone as the model input. 185

7.1 Illustration of a hybrid model combining ensemble data assimilation and ConvLSTM and DCGAN models for long lead-time forecasting. (a) Spatiotemporal dynamic features are used for Multi-Input Multi-Output (MIMO) forecast. (b) Architectures of the ConvLSTM and DCGAN models for MIMO forecast. (c) Extension of forecasting horizon integrating ConvLSTM and DCGAN with EnKF in a recursively MIMO strategy. 192

7.2 One step MIMO forecasts of X_2, X_8, X_{16} and X_{32} state variables conducted by the ConvLSTM and DCGAN models, where the length of the known time series (for inputs) $M = 60$ (0.06 MTU) and the predicted time series $H = 30$ (0.03 MTU), i.e., 30-time-step-ahead forecasts. 200

7.3 Comparison of the trajectories (top panel) and residuals (bottom panel) of all state variables between the physical modeling-L96, the ConvLSTM and DCGAN models for the lead-times $180.6 - 180.9$ MTU, where the results from the physical modelling-L96 is considered as 'references'. . . 201

7.4 Residuals of all states from the ConvLSTM and DCGAN models in comparison to those from the physical modeling-L96. Left column: full state trajectory from the physical modelling-L96; the difference of results between the machine learning model and physical modelling-L96: middle column-ConvLSTM and right column-DCGAN, where the data at the last-60-step time instances (shown in the left) as inputs while the corresponding forecast periods are $181.2-181.5$ MTU, $181.8-182.1$ MTU, and $182.4-182.7$ MTU. 202

7.5 The RMSE of predicted all states from the ConvLSTM and DCGAN models for the lead-time periods, $180.6-180.9$ MTU, $181.2-181.5$ MTU, $181.8-182.1$ MTU, and $182.4-182.7$ MTU. 203

7.6 The trajectories of X_2, X_8, X_{16} and X_{32} predicted using the ConvLSTM-EnKF1 and DCGAN-EnKF1, where the length of the known series is $M = 60$ (0.06 MTU) and the predicted series is $\Phi \times H = 300$ (3 MTU), i.e., 300-steps-ahead forecasts. 206

7.7 The RMSEs and correlation coefficients of the ConvLSTM, DCGAN, ConvLSTM-EnKF1, and DCGAN-EnKF1 models during the forecasting period of $180.6-183.6$ MTU, where (a) the spatial-averaged RMSEs and (b) the temporal-averaged RMSEs. Note that EnKF are not introduced in the ConvLSTM and DCGAN models while EnKF is used for updating the inputs in ConvLSTM-EnKF1 and DCGAN-EnKF1. 207

7.8 Same as Fig. 7.6 except for using the ConvLSTM-EnKF2 and GAN-EnKF2 models, where both inputs and outputs obtained from the trained machine learning models are updated using EnKF. 208

7.9 Tylor diagrams representing correlation coefficient together with the standard deviation difference for proposed ConvLSTM, ConvLSTM+EnKF1, ConvLSTM+EnKF2, DCGAN, DCGAN+EnKF1, DCGAN+EnKF2 for (a) X_2 (b) X_8 (c) X_{16}, and (d) X_{32} state variables during the forecasting period of $180.6 - 183.6$ MTU. 209

7.10 Comparison of 300-step-ahead forecast trajectories of full states between the ConvLSTM-EnKF1, ConvLSTM-EnKF2, DCGAN-EnKF1 and DCGAN-EnKF2 models with physical modelling-L96 during the forecasting period of $180.6 - 183.6$ MTU. 210

A.1 A typical three-layer feedforward artificial neural networks. 250

A.2 The structure of the cell connection in the Long Short-Term Memory (LSTM) neural network. 251

A.3 Loss variation in the ANN model training process. 251

A.4 The map of the RMSE of daily forecasting streamflow varying with the number of time lag and epochs in (a) ANN modelling, and (b) LSTM modelling. 252

A.5 The map of the RMSE of monthly forecasting streamflow varying with the number of time lag and epochs in (a) ANN modelling, and (b) LSTM modelling. 252

A.6 Forecast performance at the lead-times 1, 2, 3, 4 and 5 days: (a) with the ANN model and (b) with the LSTM model. 255

A.7 Forecast performance with the ANN and LSTM models over the validation period of year 2012 to 2014 at the lead-times of 1-20 days: (a) RMSE, (b) NSE, (c) CC and (d) MAE. 256

B.1 Spatial-averaged ozone concentration variation during predictive period of June in 2018. The VAE-GAN and VAE-GAN-LSTM models take both the temperature and last-step predictive ozone as the model inputs. The VAE-GAN-LSTM model adopts the convolutional long short-term memory (ConvLSTM) layer into the encoder and decoder of the VAE-GAN. . 258

B.2 Comparison of the spatial distribution of ozone fields obtained from the VAE-GAN model and the VAE-GAN-LSTM at the lead-time of $60h$ in June of 2018. 259

B.3 Comparison of the spatial distribution of ozone fields obtained from the VAE-GAN model (left) and the reanalysis data (middle) at the lead-times of $30h$ and $60h$ in June and December in 2018 respectively. 261

B.4 The RMSE and correlation coefficients of ozone concentration obtained from the VAE-GAN and the reanalysis data in two months in 2018. 262

B.5 The predictive ozone (left) and emission of NO_2 (right) distribution of China in June of 2018. 263

List of Tables

3.1 Architectures of $\mathcal{Q}$, $\mathcal{P}$, and $\mathcal{D}$ 64

3.2 The variation of length and height of the tank in cases 72

3.3 The relative error of variables between the generated fields and original high fidelity fields during whole simulational period for all the cases. . . . 79

3.4 Online CPU time required for running the hybrid VAE-GAN and high fidelity model during the whole simulation period for two cases. 79

4.1 The training and validation datasets at daily and monthly scales. 92

4.2 The parameter settings of ANN and LSTM models at daily and monthly forecast. 92

4.3 The optimal parameter settings in daily forecasting. 95

4.4 The optimal parameter settings in monthly forecasting. 102

5.1 The definition of variables used in this work. 111

5.2 Architecture and model parameter setting of the DCGAN 115

5.3 Statistical metrics for water depth solutions between the DCGAN and the original high fidelity model at $t = 5, 8, 14, 18h$. 124

6.1 Architecture of encoder, decoder and discriminator in the VAE-GAN. . . 162

6.2 Summary of training and validation datasets used in hourly and daily ozone forecasting (where the model inputs are the meteorological conditions (temperature, humidity and wind speed) and the ozone concentrations at the last time level while the targeted outputs are the ozone concentrations at the predictive time level.) 166

6.3 Residuals of ozone fields between the VAE-GAN model and the reanalysis data at the lead-times $t = 30, 60h$ in June and December in 2018. . . . 169

6.4 Error analysis of results from the reanalysis data, the physical model-NAQPMS, VAE-GAN and VAE-GAN-LSTM models where the observations in June of 2018 are taken as references (true values). 177

C.1 List of the statistical metrics used in model evaluation. 267

CHAPTER 1

Introduction

1.1 Research motivation and objectives

Numerical simulations of nonlinear fluid flows are used to describe the complex evolution of many physical processes. Accurate simulations of nonlinear fluid flows are of great importance to many fields such as flooding, ocean modelling, and atmosphere, in which systems often exhibit rich flow dynamics in both space and time. Traditional numerical models have become increasingly complicated by using high spatiotemporal resolutions, and coupling with subgrid parameterization models in combination with a large volume of data from experiments and field measurements. This unavoidably leads to uncertainties in modelling and high computational cost. Recent advances in machine learning technologies are increasing interest in efficiently simulating nonlinear and complex systems. This has motivated the development of machine learning applied in fluid dynamic systems.

Recently, deep learning (LeCun et al., 2015), as one of machine learning techniques, has gained popularity in rapidly predicting nonlinear fluid flows (Kalteh, 2013; Kasiviswanathan et al., 2016; Kisi et al., 2012; Lohani et al., 2014; Yang et al., 2017; Wang et al., 2018). The deep learning techniques are capable to handle large datasets and provide nonlinear fluid flow prediction without prior knowledge of underlying physical relationships. The

striking advantage of deep learning is the deep neural networks with flexible nodes, layers and activation functions for determining the most appropriate model architecture and weights in applications (Zadeh et al., 2010; LeCun et al., 2015). This flexible architecture of deep learning makes it effective to extract high orders of features from raw data, which improves computation efficiency. Different architectures of neural networks have been introduced in section 1.2.1.

Tremendous advances in deep learning have been made in recent years for fluid flows, for example, in turbulence modelling (Mohan and Gaitonde, 2018; Raissi et al., 2019), water level prediction (Assem et al., 2017), rainfall-runoff simulation (Hu et al., 2018a), and air pollution modelling (Xiao et al., 2019a). In these various applications, deep learning has demonstrated a powerful capability in solving complex environmental applications that include large amounts of independent parameters and nonlinear relationships (Yetilmezsoy et al., 2011). This opens up huge opportunities for researchers to exploit the potential of deep learning in numerous research fields and industrial applications. Despite the need and growing attention to deep learning, there remain several challenges in nonlinear fluid flow modelling, i.e., spatiotemporal nonlinear fluid flow modelling, large data-driven fluid flow modelling, real-time forecasting of nonlinear fluid flows beyond the training period, improvement of the forecasting accuracy in long lead-time (see section 1.2.2).

To overcome these challenges, more recently, Generative Adversarial Networks (GANs) representing the most recent progress in deep learning techniques (Goodfellow, 2016; Laloy et al., 2018; Ströfer et al., 2018), have been introduced to fluid flow modelling (Xie et al., 2018). The main specificity of GANs is the competition game between the generator and discriminator modules. The generator consists of deep neural networks, which can produce high-dimensional distributions of data to "fool" the discriminator. The discriminator is responsible for distinguishing the real data from the generated data. Nowadays applications of GANs focus mostly on images (Reed et al., 2016; Li and Wand, 2016; Isola et al., 2017), rarely on fluid flow problems (Farimani et al., 2017; Xie et al., 2018).

Thus, the data-driven GAN for dynamic system modelling is the primary focus of this thesis.

Deep learning for spatiotemporal engineering applications is promising and challenging. This thesis focuses on two important engineering areas: flooding and air pollution problems. Floods as one of the most dangerous natural disasters, cause loss of life and property in many countries in the world. Rapid real-time flood forecasting is of great significance for flood hazard mitigation and advanced management of policy responses to flooding disasters (Ham et al., 2019; Wang et al., 2015; Bontempi and Taieb, 2011). In general, floods are characterized by nonlinearity and non-stationarity fluid flows (Yaseen et al., 2015; Farajpour et al., 2018; Dai et al., 2014), which exhibit complex spatiotemporal dynamics. In addition, the long lead-time flood forecasting faces increasing uncertainties from various sources, for example, model inputs and parameters. Thus, it is particularly difficult to make accurate long lead-time flood forecasts in traditional computational dynamic models (Bontempi et al., 2012). A deep convolutional GAN in this thesis has been proposed to tackle challenges presented by the long lead-time flooding forecasting. Another example requiring machine learning techniques is ozone forecasting. Efficient and accurate real-time forecasting of national spatial ozone distribution is critical to the provision of effective early warning. Traditional numerical air quality models require a high computational cost associated with running large-scale numerical simulations.

In this thesis, my aim is to develop efficient and accurate machine learning-based models for spatial and temporal nonlinear fluid flow, atmospheric and flood predictions. The specific objectives of this work include spatiotemporal modelling, large data-driven modelling and real-time forecasting in flooding and ozone prediction, as well as improvement of forecasting capability in long lead-time (as shown in Fig. 1.1). Therefore, I have developed a deep convolutional GAN (DCGAN) for predicting spatiotemporal nonlinear fluid flow distributions for the first time. For large data-driven prediction, the proposed DCGAN further couples with a variational autoenocder (VAE-GAN) to predict large data-

driven flows in spatial and temporal dimensions. Additionally, the proposed DCGAN and VAE-GAN, as well as artificial neural network (ANN) and a long short term memory (LSTM) have been developed for real-time streamflow, flooding and ozone forecasting in spatial and temporal spaces. Both forecasting methodologies consist of dynamic flow learning and real-time forecasting processes. To further reduce the accumulated error in real-time forecasting, an ensemble Kalman filter (EnKF) data assimilation technique has been introduced into a GAN and a convolutional long short term memory (ConvLSTM) models for nonlinear fluid flow forecasting. The details of the models developed here are given in Chapters 2 and 3 while the applications to stream flows, flooding and ozone forecasts are provided in Chapters 4 to 6.

1.2 Literature review of machine learning for spatiotemporal prediction

1.2.1 Machine Learning methods

This section provides a summary of deep learning methods, mostly focusing on deep neural networks, i.e., artificial neural networks, convolutional neural networks, and recurrent neural networks, as well as introduces Generative adversarial networks.

1.2.1.1 Deep neural networks

Machine learning algorithms have proven to be an effective tool for extracting multi-scale features and capturing arbitrarily complex input-output relationships from large datasets (Merghadi et al., 2020; Humphrey et al., 2016; Tahmasebi et al., 2020). According to the nature of learning signals or feedback, machine learning algorithms are classified broadly into supervised learning and unsupervised learning. The main difference between the two algorithms is the use of labelled datasets. Supervised learning uses labelled input

and output datasets, while unsupervised learning deals with unlabelled datasets. In this thesis, I will focus on supervised learning algorithms. In recent decades, many supervised learning algorithms, such as neural networks (NNs) (Nourani and Komasi, 2013; Noori and Kalin, 2016), support vector machines (SVMs) (Sudheer et al., 2014; Rasouli et al., 2012; Yaseen et al., 2016; Adnan et al., 2019), fuzzy logic (Alvisi and Franchini, 2011), and wavelet transform (WT) (Kisi and Cimen, 2011; Fang et al., 2019), have been widely used in earth science applications. Various studies have shown that these methods are capable of capturing non-linear processes numerically from large scale datasets (Yaseen et al., 2015; Rathinasamy et al., 2013; Prasad et al., 2017; Alvisi and Franchini, 2011).

Deep learning (LeCun et al., 2015), as a kind of machine learning, uses deep neural networks to extract high-level features from large datasets. Deep neural networks are technologically advanced as a generalization of mathematical models of human cognition or neural biology. The development of deep neural networks is based on the following rules, (i) information processing between neurons, (ii) signals passing between neurons through connection links, (iii) weights to represent its connection strength, (iv) a nonlinear transformation on the net input to determine the output. In this section, the popular deep neural network architectures are introduced in plain and accessible language, and the following models are considered in detail: artificial neural network (ANN), convolutional neural network (CNN), and recurrent neural network (RNN).

Artificial neural networks: ANNs, mathematical models of a biologically inspired system, provide a novel and appealing solution to the problem of relating input and output variables in complex systems (Basheer and Hajmeer, 2000; Nourani and Komasi, 2013). It requires no information about the underlying complex physical process, while constructing black-box models of complex and nonlinear relationships between the input and output variables. In general, an ANN model consists of three typical layers. Each layer processes a series of neurons, which are fully connected with those of the following layer. The first layer is the input layer that receives the input data. The hidden layer is the information

processing section, which uses weights to achieve nonlinear transformation between the connection links to determine the output. The output layer receives the processed information from the last hidden layer and then outputs the result. The weights between different neurons adjust as learning proceeds with the aim of minimizing the difference between predicted output and desired output by a back-propagation procedure (Yang et al., 1998). Since the early 1990s, ANNs started to receive much attention in earth and environment sciences and were used in many applications, such as remote sensing (Gopal and Woodcock, 1996; Ingram et al., 2005; Meresa, 2019), hydrology modelling (Kisi et al., 2012; Humphrey et al., 2016), turbulence modelling (Gamahara and Hattori, 2017; Chenyue et al., 2021), and atmosphere forecasting (Perez-Roa et al., 2006; Pazikadin et al., 2020). Although widely used, there are some issues associated with ANNs (Shortridge et al., 2016; Sun et al., 2014; Guo et al., 2011; Kalra et al., 2013). For example, the number of trainable parameters increases drastically in dealing with high-dimensional datasets because it is a feed-forward and no parameter-sharing neural network.

Convolutional neural network (CNNs): CNNs, developed by LeCun et al. (1989), were initially designed for computer vision tasks. In general, a basic structure of CNN includes input, convolution, pooling, fully connected, and output layers. The convolutional layer is the main operation in the CNNs, which involves the convolutional operation. Specifically, the convolutional operation moves a filter across the input layer in a certain order and produces a feature map. Each movement produces one unit in the feature map by calculating a dot product between the filter and a small local region of the input, thus enabling a CNN model to extract features more effectively with spatial contextual information and reduce the dimension of input (Tahmasebi et al., 2020). Further detailed explanations on the convolutional function and its variants and pooling layers, can be found in detail in (Dumoulin and Visin, 2016; Rawat and Wang, 2017). Due to the advantages of convolutional layers, such as local connection, weight sharing, and downsampling dimensionality reduction (Li et al., 2021), CNNs have the powerful capability of learning hierarchical representations of complex high-dimensional datasets (e.g., images and videos in 2D or

3D) and have been extended to various applications, such as image classification and recognition (Howard, 2013; Hussain et al., 2018), signal processing (Kwon et al., 2020) and natural language processing (Young et al., 2018). One of limitations of CNNs is the computation cost associated with large data training (Hosseini et al., 2017). To receive good performance of prediction and accomplish more complex goals, the CNN architecture becomes deeper and more complicated, thus leading to intensive computations (Zhang et al., 2019).

Recurrent neural networks (RNNs): RNNs, are motivated to take advantage of sequential data (Williams and Zipser, 1989). It is a straightforward adaptation of the standard feed-forward neural network by processing real data sequences at time steps and predicting the next time step. The key component of RNNs is the recurring mechanism in hidden layers, which can integrate information over many timesteps and use it to make accurate predictions. This design is significant in many applications where the output relies on previous computations, such as streamflow forecasting (Chang et al., 2004) and natural language processing (Tarwani and Edem, 2017). Long short term memory (LSTM), as an excellent variation of RNNs, was introduced by Hochreiter and Schmidhuber (1997). An LSTM consists of special units (called memory blocks) in the recurrent hidden layers. Each memory block contains the self-connected memory cells and multiplicative units. The memory cells are used to store the temporal state of the networks. The multiplicative units, including the input, output and forget gates, control the flow of information between the cells. The input gate is responsible for controlling the flow of inputs into the memory cell, while the output gate conducts the output flow of cell activations. The forget gate scales the internal state of the cell. LSTM has proved to be a powerful tool for addressing time-series prediction problems (Hu et al., 2019a). Compared to classical NNs, LSTMs are able to capture both the periodic and chaotic behaviors of time series data, and learn their long-range dependencies with greater accuracy (Mouatadid et al., 2019).

1.2.1.2 Generative adversarial networks (GANs)

GANs introduced by Goodfellow et al. (2014), have emerged as having a leading role in producing high-resolution datasets (e.g. images) (Xie et al., 2018; Cheng et al., 2020c). GAN defines a learning generative network (generator) by transforming a latent variable into a state variable using nonlinear functions. Then GAN drives the learning process by discriminating the observed data from the generated data in a discriminator network (discriminator). Due to the special adversarial architecture, GAN has demonstrated great capability in producing high-resolution samples in many areas (Liu et al., 2020b; Creswell et al., 2018), e.g, image synthesis, semantic image editing, super-resolution, style transfer and video stylization. For example, Chen et al. (2018) proposed a CartoonGAN to generate high-quality cartoon images from real-world photos, and follow specific artists' styles with clear edges and smooth shading. Thawakar et al. (2019) proposed a SR2GAN for image and video super-resolution to overcome the difficulty of learning transformation function for synthesizing realistic high-resolution images. Reed et al. (2016) used GANs to generate photorealistic object images such as flowers and birds given text captions. Pascual et al. (2017) developed a SEGAN for speech enhancement. They incorporated 28 speakers and 40 different noise conditions at the waveform level into the SEGAN model, and trained the model end-to-end. The proposed model was validated using an independent, unseen test set with two speakers and 20 alternative noise conditions.

Except for the above applications in the image and computer vision areas, GAN-based techniques have shown promising potential for spatiotemporal-based prediction applications such as trajectory prediction and link prediction (Gao et al., 2022). Compared to traditional deep learning methods (e.g., RNNs) which fail to capture the stochastic behaviour of spatiotemporal datasets, GANs have superior performance in producing spatiotemporal datasets with high resolution and fidelity. In addition, as the loss functions in the GAN focus on probability distribution comparison instead of true values, the inputs in the prediction process which only need to fit the trained data distribution could get a good

prediction result. Therefore, GAN also has the key advantage that, in principle, any point in the latent space of inputs could be associated with a realistic-looking solution (Cheng et al., 2020c).

1.2.2 Challenges of machine learning methods in fluid dynamics modelling

Recent advances in machine learning technologies are increasing of interest for efficiently simulating flows in dynamical systems (Lusch et al., 2018; Lee and Carlberg, 2018; Murata et al., 2020). Machine learning has demonstrated its potential capability in fluid flow applications (Brunton et al., 2020), such as feature extraction (Geneva and Zabaras, 2019; Ling et al., 2016; Kutz, 2017), and super-resolution (Xie et al., 2018). In this section, I will focus on the challenges of machine learning in fluid dynamics modelling: spatiotemporal prediction, large data-driven prediction, real-time forecasting, and improvement of forecast accuracy.

1.2.2.1 Spatiotemporal fluid flow modelling

Numerical simulations of fluid dynamics have been indispensable in many applications relevant to physics and engineering. For improving predictive capability, numerical algorithms have become increasingly sophisticated and complicated by using more spatial and temporal resolution of datasets (Erichson et al., 2019; Hu et al., 2019a). This often leads to very large-scale models and unavoidably raises computational challenges.

Recently, deep learning has gained popularity in rapidly predicting nonlinear fluid flows (Kalteh, 2013; Kasiviswanathan et al., 2016; Kisi et al., 2012; Lohani et al., 2014; Yang et al., 2017; Wang et al., 2018; Brunton et al., 2019). Various deep learning techniques have been applied to nonlinear fluid flow prediction (Ling et al., 2016; Moghaddam and Sadaghiyani, 2018; Raissi et al., 2018). For example, Miyanawala and Jaiman (2017)

successfully demonstrated to combine fluid solvers with deep neural networks. By employing deep convolutional neural networks for the model reduction of Navier-Stokes equations, the Reynolds stress anisotropy was predicted with respect to high-fidelity simulation data. Furthermore, Moghaddam and Sadaghiyani (2018) used deep learning methods to extract features of numeric and non-numeric characteristics for multimodal turbulence, containing the information of velocity, pressure, locations of extreme points and fluid characteristics. It was demonstrated that the deep learning method could provide accurate predictions. Geneva and Zabaras (2019) proposed the Bayesian deep neural networks to improve predictions of Reynolds-Averaged Navier Stokes and provide probabilistic bounds for fluid quantities such as velocity and pressure. Despite the fact that these research efforts demonstrated how to successfully reconstruct flow dynamics, it is noted that they commonly do not take into account the temporal and spatial evolution of inputs or parameters, which is crucial for realistic dynamical systems.

More recently, GANs have been introduced to reconstruction of fluid dynamic features (Farimani et al., 2017; Xie et al., 2018). GANs represent the most recent progress in deep learning techniques (Goodfellow, 2016; Laloy et al., 2018; Ströfer et al., 2018) and have been attractive for producing high-resolution samples (e.g. images) (Ledig et al., 2017; Xie et al., 2018; Creswell et al., 2018). Nowadays applications of GANs focus mostly on images (see Section 1.2.1.2), rarely on fluid flow problems (Farimani et al., 2017; Xie et al., 2018). Farimani et al. (2017) first introduced the conditional generative adversarial networks (cGAN) to simulate steady state heat conduction and incompressible fluid flow. This study proves that GANs can be effective in exhibiting complex nonlinear fluid structures. Xie et al. (2018) introduced a temporal discriminator to GANs for solving the super-resolution problem for smoke flows. In this thesis, I have developed a data-driven deep convolutional GAN for spatiotemporal fluid flow modelling problems. This is the first work of GAN applied in nonlinear spatiotemporal fluid flow modelling.

1.2.2.2 Large data-driven prediction

The numerical simulations have benefited from the availability of high-resolution spatiotemporal data due to recent advances in measurement techniques (Taira et al., 2019), which make the studies of more complex flows possible. However, the computational cost involved in solving complex problems is intensive which still precludes the development in these areas.

Various machine learning-based methods have been adopted for large data-driven fluid flow modelling (Kani and Elsheikh, 2017; Kutz, 2017; Fresca et al., 2021). For example, Kim et al. (2019) adopted a CNN to extract flow features and reconstruct fluid fields. Gonzalez and Balajewicz (2018a) developed a deep convolutional recurrent autoencoder for nonlinear model reduction, which identified the low-dimensional feature dynamics by the high-dimensional datasets and modelled flow dynamics on the underlying manifold space. Wang et al. (2020b) designed a novel deep neural network model reduction approach for multiscale simulations of porous media flows problems. These models developed for model reduction are generally based on deep neural networks. The way of dimensionality reduction in the deep neural networks is similar to that of Galerkin reduced order model (ROM) (Fang et al., 2013, 2014; Xiao et al., 2019a), Variational Autoencoder (VAE) (Gonzalez and Balajewicz, 2018b) and cluster-based reduced order model (CROM) (Kaiser et al., 2014; Nair et al., 2019; Östh et al., 2015). These ROMs are able to compress spatiotemporal flow snapshots into a small number of low-dimensional states for dynamic modelling, which is also a strategy to reduce computational requirements of ROMs (Kaiser et al., 2014). Nevertheless, the prominent difference between the neural networks and existing ROMs lies in the fact that the CROMs or POD based ROMs result in linear combinations of the original snapshots (Kaiser et al., 2014; Xiao et al., 2016), while the deep neural networks are able to capture the nonlinear flow transients since the effective basis functions are non-linear (Miyanawala and Jaiman, 2017).

To meet the requirements of more complex and huge amount of data in the big data era, the hidden layers of neural networks have scaled from 1-2 layers to hundreds of layers (Kutz, 2017). Model architecture, associated with model parameters, determines the relationship between model inputs and outputs. If the network structure is too simple, it may be unable to capture the relationship. If a complex network is chosen, more computation will be needed for a large number of parameters (Kutz, 2017). Therefore, a balance between the model performance and model complexity needs to be considered (Maier et al., 2010). In this thesis, I have introduced a variational autoencoder into GAN, which can compress high-dimensional datasets into low-dimensional representations. The low-dimensional representations, several orders of magnitude smaller than the dimensional size of the original datasets, are then applied to the adversarial network for representation learning and parameter optimization, thus accelerating the computation process and maintaining the accuracy.

1.2.2.3 Real-time forecasting

Natural disasters, such as flooding and air pollution, cause loss of life and property in many countries in the world. For example, according to a report by the Parliamentary Office of Science and Technology (POST, 2007), urban flooding causes losses of 270 million pounds per year in England and Wales with 80,000 homes at risk. Therefore, rapid real-time forecasting is of great significance for natural hazard mitigation and advanced management of policy responses to natural disasters (Ham et al., 2019; Wang et al., 2015; Bontempi and Taieb, 2011). In general, fluid flows are characterized by nonlinearity and non-stationarity (Yaseen et al., 2015; Farajpour et al., 2018; Dai et al., 2014), which exhibit complex spatiotemporal dynamics. In addition, the long lead-time forecasting faces increasing uncertainties from various sources, for example, model inputs and parameters. Thus, it is particularly difficult to make accurate long lead-time fluid flow forecasts in traditional computational dynamic models (Bontempi et al., 2012).

Recent progress in machine learning has shown how to learn the dynamics of a model from its history datasets and forecast beyond the training period, to some extent (Ferreira and da Cunha, 2020; Papacharalampous et al., 2019). For example, Shi et al. (2015) first introduced CNN to LSTM (ConvLSTM) for precipitation nowcasting. In the work of Lin and Wu (2011), radial basis function neural network (RBFNN) were respectively utilized to provide the 3 hours and 6 hours ahead forecasting. In addition, Random Forest (RF), ANN, and Support Vector Regression (SVR) are employed and compared with respect to their capabilities were used to predict 1 month-ahead reservoir inflows in the study of Yang et al. (2017). Adamowski and Sun (2010) developed a coupled wavelet transform and neural networks method for 1 day and 3 days flow forecasting. Furthermore, the study of Badrzadeh et al. (2013) indicated that for 1, 2, 3, 4 and 5 days ahead forecasting could be achieved by a combination of different input datasets with ANN and ANFIS. Chen et al. (2020a) proposed an Auto-Reservoir Neural Network to make multi-step-ahead predictions based on a short-term time series. Similarly, da Silva et al. (2021) combined a Complete Ensemble Empirical Mode Decomposition (CEEMD) with Stacking-ensemble learning (STACK) based on machine learning algorithms to forecast up to 30-minute-ahead of wind turbine power generation in a wind farm at Parazinho city, Brazil.

Considering that the site-specific data-driven machine learning models may disregard the potential nonlinear spatial correlation among stations (Zhou et al., 2019), some researchers have attempted to explore multi-step-ahead forecasts on a regional scale in recent years. Wiewel et al. (2019) also proposed the hybrid CNN-LSTM for predicting the one-step-ahead fluid field of the physic system. Zhang et al. (2020) used a spatiotemporal deep learning algorithm based on bidirectional gated recurrent unit integrated with attention mechanism (BiAGRU) for 24h ahead air pollution prediction using Huaihai Economic Zone dataset. Kow et al. (2020) made 10-step-ahead PM2.5 forecasts for multiple stations at multiple horizons by combining a CNN and a Back Propagation Neural Network (BPNN). Zhou et al. (2019) proposed a Deep Multi-output LSTM (DM-LSTM) neural network model for 4-step-ahead air quality forecasting in Taipei City, Taiwan.

Notwithstanding the success of machine learning in the aforementioned applications, most of these models are applicable to sequence nowcasting using the history datasets, and only one-timestep-ahead or very short lead-time forecast could be achieved (Yaseen et al., 2018; Prakash et al., 2014; Huang et al., 2015; Bento et al., 2020; Fan et al., 2020). There are two challenges in real-time forecasting for a long lead-time, i.e., uncertainty and accumulated forecast error. The uncertainty in future flow dynamics may derive from meteorological, hydrological and boundary condition uncertainty. It is necessary to explore whether the trained machine learning model using the datasets in the past is able to predict the future flow dynamics (beyond the training period). This is dependent on whether the available training datasets are good/big enough for capturing various complex fluid flow dynamics in the future. In real-time forecasting, the predicted variable solution from the previous time levels is used as inputs for prediction at the next time levels. With increasing the length of the forecast, the cumulative error in time series prediction may become larger, thus leading to a predictive failure in long lead-time forecasting. In the work of Hu et al. (2019a), it was found that the predictive accuracy in LSTM modelling decayed rapidly when the predictive lead-time was extended beyond the training period. This indicates that capturing dependencies of features between successive time periods in the model becomes more intractable as the prediction horizon extends in time. In this thesis, I have developed a recursively forecasting strategy in the GANs for long lead-time spatiotemporal forecasting.

1.2.2.4 Accuracy improvement in real-time forecasting

As discussed in the last section, due to the instabilities and biases in forecasting the overall effect of subgrid-scale processes, the performance of the model generally decays as the forecasting horizon extends beyond the given period (Xu et al., 2021; Cheng et al., 2020a), while data assimilation may be a suitable approach to address this issue (Fang et al., 2017; Buizza et al., 2021).

Data assimilation techniques are used to integrate the state of the dynamic system at a given time from its time-distributed observations and numerical solutions in an optimal way, thus reducing its uncertainty (Eyre, 2016; Fang et al., 2006; Tang et al., 2018). In particular, the Kalman filter originally proposed by Kalman (1960), is a widely used optimal sequential data assimilation method for linear state-space systems and measurement processes with Gaussian error statistics. Ensemble Kalman filter (EnKF) as a special Kalman filter (Evensen, 1994), has been widely are being extensively utilized in many fields, such as geosciences (Keller et al., 2021), hydrology (Clark et al., 2008; Bao et al., 2020) and physics (Yang et al., 2019; Fujita et al., 2007; Sampson et al., 2021; Pawar et al., 2020), to correct background predictions based on a combination of heterogeneous measurement data coming from ground observations, experiments and remote sensing (Pawar and San, 2021).

Recent works have also drawn attention to combining EnKF with machine learning models. Recent works have also drawn attention to combining data assimilation with machine learning models in dynamic systems modelling. Considering the evolution of a dynamic system:

$$\mathcal{X}(t_{s+\delta t}) = \mathcal{M}(\mathcal{X}(t_s)) + \mathcal{E}(t_s) \tag{1.1}$$

where $\mathcal{M}$ represents the dynamic model, $\mathcal{X}(t_s)$ is the state variable at time t_s, and $\mathcal{E}(t_s)$ is the model noise. Data assimilation methods are used for reducing the error $\mathcal{E}$ by the inclusion of observational data into the dynamic model $\mathcal{M}$, thus improving forecast accuracy (referred to as $\mathcal{M}_{physical} + \mathcal{E}_{DA}$ *mode*). While it is fairly new and limited, a hybrid data assimilation (DA) and machine learning (ML) model can be generally classified as:

(1) $\mathcal{X}(t_{s+\delta t}) = \mathcal{M}_{physical}(\mathcal{X}(t_s)) + \mathcal{E}_{DA+ML}(t_s)$: The dynamic model is physical-informed modelling, where the dynamic processes are represented by the governing equations. The physical models can thus be used for complex nonlinear dynamic simulations, but require a high computational cost. Machine learning techniques are here used for efficient data assimilation in real-time forecast. Machine learn-

ing models learn the assimilation process (error term $\mathcal{E}$) and improve the predictive accuracy of physical models (Wu et al., 2021; Pawar et al., 2020; Bao et al., 2020; Farchi et al., 2021; Brajard et al., 2020; Wikner et al., 2021; Buizza et al., 2021). For example, Bao et al. (2020) coupled the EnKF with a Generative Adversarial Networks (GAN) to update the latent variable by assimilating dynamic data into the groundwater model.

(2) $\mathcal{X}(t_{s+\delta t}) = \mathcal{M}_{physical+parameterized_{DA/ML}}(\mathcal{X}(t_s)) + \mathcal{E}(t_s)$: Data assimilation and machine learning techniques can be used for parameterization in physical models (Pawar and San, 2021; Brajard et al., 2021; Legler and Janjić, 2021). For example, Pawar and San (2021) used a set of neural network architectures to learn the correlation between resolved flow variables and the parametrizations of unresolved flow dynamics, where an EnKF data assimilation approach was used to correct the error in the trained neural networks.

(3) $\mathcal{X}(t_{s+\delta t}) = \mathcal{M}_{ML+DA}(\mathcal{X}(t_s))+\mathcal{E}(t_s)$: Instead of physical models, a machine learning model is used as a forward surrogate dynamic model $\mathcal{M}$, where the computational cost is significantly reduced in comparison to physical models. Here data assimilation techniques are used for training the machine learning models. For example, the data was assimilated for training the parameters (weights) within the network's hidden layers (Peyron et al., 2021; Buizza et al., 2021; Kang et al., 2021). Machine learning models (e.g. autoencoders) exploit the latent structure and data assimilation corrects model errors in the latent space. Note that data assimilation techniques are used only during the training period, whereas discussed above, the predictive accuracy in machine learning modelling will decay during long lead-time forecast.

(4) $\mathcal{X}(t_{s+\delta t}) = \mathcal{M}_{ML}(\mathcal{X}(t_s)) + \mathcal{E}_{DA+ML}(t_s)$: Data assimilation is used not only for training machine learning models, but also correcting the error term $\mathcal{E}$ from machine learning models, thus improving predictive accuracy. In current works (e.g.

Gottwald and Reich (2021)), data assimilation techniques are used for updating the outputs in machine learning modelling just once, not for long term real-time forecasting.

Generally speaking, the data assimilation improves the predictive capability of physic-based-data-driven and pure data-driven models, and the hybrid frameworks hold immense potential in physical sciences (Boucher et al., 2020). Despite these advanced trends, most of the hybrid frameworks either rely on the physical models as a forward model with high computation burden or use machine learning models with limited forecasting capability, to make short term forecasts. Thus, they are not applicable to efficiently make long lead-time forecasts of nonlinear dynamic systems. In this thesis, I have proposed a hybrid data assimilation and machine learning-based long lead-time forecast framework ($\mathcal{M}_{ML}$ + $\mathcal{E}_{DA+ML}$ *mode*). The ConvLSTM and DCGAN, acting as the dynamic surrogate model $\mathcal{M}_{ML}$, are used for efficient calculation of ensembles as well as long term forecasts.

1.3 Original contributions

The novel contributions in this thesis are followings:

(1) *First development of a deep convolutional generative adversarial network (DCGAN)* for predicting spatiotemporal flow distributions. This is the first attempt to use the DCGAN for predictive analysis in fluid flow problems. Its capabilities have been validated using two test cases of Hokkaido tsunami with different incoming waves along the coastal line in Chapter 2 and further realistic applications in Chapters 4 to 6.

(2) *Development of a hybrid deep adversarial autoencoder (VAE-GAN) model* for large data-driven nonlinear fluid flow prediction. The introduction of variational autoencoder into GAN is for reducing the dimensionality of large datasets. The capability

of the hybrid VAE-GAN is illustrated by varying inputs on a flow past a cylinder test case as well as the second case of water column collapse in Chapter 3 and further realistic applications in Chapters 4 to 6.

(3) *Daily and monthly streamflow forecasts in Thailand* using an artificial neural network (ANN) and a long short term memory (LSTM). The ANN and LSTM models are trained and validated for daily and monthly streamflow forecasting using the rainfall and runoff datasets collected from the Nan River Basin and Ping River Basin, Thailand (in Chapter 4).

(4) *Spatial-temporal flooding forecasting in Denmark* using a real-time DCGAN framework. The proposed methodology consists of a two-stage process: 1) dynamic flow learning and 2) real-time forecasting. In dynamic flow learning, deep convolutional neural networks are trained to capture the underlying flow patterns of spatiotemporal flow fields. In real-time forecasting, the DCGAN adopts a cascade predictive procedure. The last one-timestep-ahead forecast from the DCGAN can act as a new input for the next timestep-ahead forecast, which forms a long lead-time forecast in a recursive way. The model capability has been assessed using a 100-year return period extreme flood event occurred in Greve, Denmark (in Chapter 5).

(5) *Real-time spatiotemporal ozone forecast in China*. Three new machine learning models have been developed using a high spatial resolution ($15km \times 15km$) ozone datasets: a ConvLSTM and a DCGAN for hourly and daily spatiotemporal ozone prediction, and a hybrid VAE-GAN for providing ozone forecasts for a long lead-time beyond the training period (in Chapter 6).

(6) *An ensemble Kalman filter (EnKF) for GAN and ConvLSTM (GAN-ConvLSTM)* based long lead-time forecasting system has been proposed. The two-hybrid models yield long lead-time forecasts of dynamic states based on an iterative Multi-Input Multi-Output (MIMO) algorithm. The EnKF embedded in hybrid models is capable of reducing the uncertainty of long lead-time forecasts of chaotic dynamic systems.

The performance of the hybrid models has been demonstrated by a Lorenz 96 model (in Chapter 7).

1.4 Thesis outline

The aim of this research is to develop generative adversarial networks for nonlinear fluid flow modelling and forecasting in spatial and temporal scales as shown in Fig. 1.1. This thesis is organised as follows:

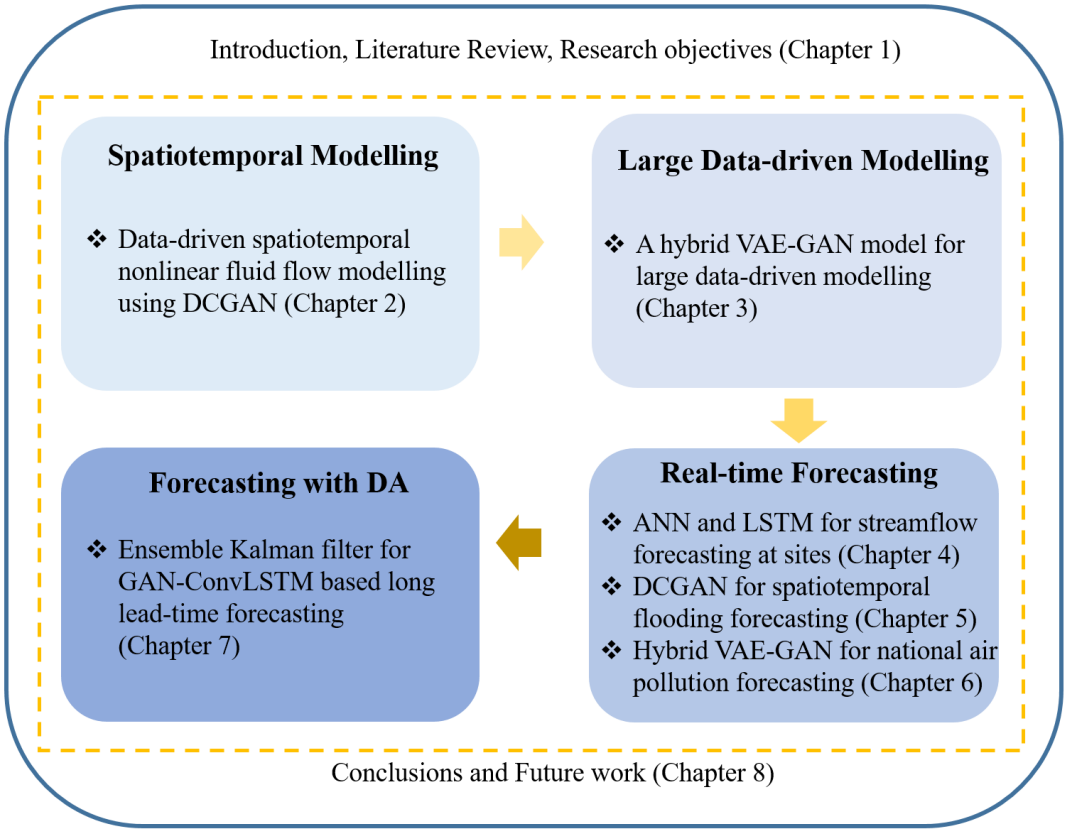


Fig. 1.1. Thesis structure and inter-relationships between individual chapters.

Chapter 1 is a brief introduction, including the objectives, literature review, original contributions and the outline of the thesis.

Chapter 2 proposed a general deep convolutional generative adversarial network (DC-

GAN) for nonlinear spatiotemporal fluid flow modelling. This Chapter is derived from:
Cheng, M., Fang, F., Pain, C.C. and Navon, I.M., 2020. Data-driven modelling of nonlinear spatiotemporal fluid flows using a deep convolutional generative adversarial network. Computer Methods in Applied Mechanics and Engineering, 365, p.113000.

Chapter 3 presents a hybrid deep adversarial autoencoder (VAE-GAN) to integrate generative adversarial network (GAN) and variational autoencoder (VAE) for predicting parameterized nonlinear fluid flows in spatial and temporal dimensions. This Chapter is derived from:
Cheng, M., Fang, F., Pain, C.C. and Navon, I.M., 2020. An advanced hybrid deep adversarial autoencoder for parameterized nonlinear fluid flow modelling, Computer Methods in Applied Mechanics and Engineering, 372, p. 113375.

Chapter 4 proposes an artificial neural network (ANN) and a long short term memory (LSTM), a powerful tool for learning long-term temporal dependencies and capturing nonlinear relationships, to forecast streamflow at daily and monthly scales for a long lead-time period. This Chapter is derived from:
Cheng, M., Fang, F., Kinouchi, T., Navon, I.M. and Pain, C.C., 2020. Long lead-time daily and monthly streamflow forecasting using machine learning methods. Journal of Hydrology, 590, p.125376.

Chapter 5 develops a real-time predictive deep convolutional generative adversarial network (DCGAN) for flooding forecasting. This Chapter is derived from:
Cheng, M., Fang, F., Navon, I.M. and Pain, C.C., 2021. A real-time flow forecasting with deep convolutional generative adversarial network: Application to flooding event in Denmark. Physics of Fluids, 33(5), p.056602.

Chapter 6 demonstrates the prediction and forecasting performances of VAE-GAN and ConvLSTM in hourly and daily spatiotemporal ozone forecasts over China. This Chapter is derived from and expands upon::
Cheng, M., Fang, F., Navon, I.M., Zheng, J., Tang, X., Zhu, J. and Pain, C., Spatiotem-

poral Hourly and Daily Ozone Forecasting in China Using a Hybrid Machine Learning Model: Autoencoder and Generative Adversarial Networks. Journal of Advances in Modeling Earth Systems, p.e2021MS002806.

Cheng, M., Fang, F., Navon, I.M., Zheng, J., Zhu, J. and Pain, C.C., 2022. Assessing uncertainty and heterogeneity in machine learning-based spatiotemporal ozone prediction in China. To be submitted.

Chapter 7 exploits an ensemble Kalman filter (EnKF) for generative adversarial network and convolutional long short term memory (GAN-ConvLSTM) based long lead-time forecasting system. This Chapter is derived from:
Cheng, M., Fang, F., Navon, I.M., and Pain, C., Ensemble Kalman filter for GAN-ConvLSTM based long lead-time forecasting. Submitted.

Chapter 8 presents the conclusion of this research and makes recommendations for future research.

CHAPTER 2

Development of a deep convolutional generative adversarial network for nonlinear spatiotemporal fluid flows

Deep learning techniques for improving fluid flow modelling have gained significant attention in recent years. Advanced deep learning techniques achieve great progress in rapidly predicting fluid flows without prior knowledge of the underlying physical relationships. However, most of existing researches focused mainly on either sequence learning or spatial learning, rarely on both spatial and temporal dynamics of fluid flows (Reichstein et al., 2019). In this work, an Artificial Intelligence (AI) fluid model based on a general deep convolutional generative adversarial network (DCGAN) has been developed for predicting spatiotemporal flow distributions. In deep convolutional networks, the high-dimensional flows can be converted into the low-dimensional "latent" representations. The complex features of flow dynamics can be captured by the adversarial networks. The above DCGAN fluid model enables us to provide reasonable predictive accuracy of flow fields while maintaining a high computational efficiency. The performance of the DCGAN is illustrated for two test cases of Hokkaido tsunami with different incoming waves along the coastal line. It is demonstrated that the results from the DCGAN are compa-

rable with those from the original high fidelity model (Fluidity). The spatiotemporal flow features have been represented as the flow evolves, especially, the wave phases and flow peaks can be captured accurately. In addition, the results illustrate that the online CPU cost is reduced by five orders of magnitude compared to the original high fidelity model simulations. The promising results show that the DCGAN can provide rapid and reliable spatiotemporal prediction for nonlinear fluid flows.

This chapter is derived from and expands upon Cheng et al. (2020c).

2.1 Introduction

In this work, I have developed a deep convolutional generative adversarial network (DCGAN) (Radford et al., 2015) for nonlinear unsteady fluid flow problems. This is the first attempt to use the DCGAN for predictive analysis in fluid flow problems. As is well known, both spatially and temporally flow dynamic properties are important in prediction. Classic machine learning approaches exploit dependencies in time or space, rarely both spatiotemporal dependencies (Reichstein et al., 2019). Here, the DCGAN is first used to extract the spatiotemporal coherence of the original high-dimensional fluid fields. The high-dimensional nonlinear fluid flow features can be converted into low-dimensional "latent" representations. With discriminating the real fluid fields and their realizations in the low-dimensional latent space, the training process herein is fast. This alleviates the problem of large data-driven modelling. In addition, GAN also has the key advantage that, in principle, any point in the latent space of inputs will be associated with a realistic looking solution.

The DCGAN is a general and efficient numerical tool for rapidly predicting nonlinear fluid flows for given inputs. In this work, I have successfully applied the DCGAN to a tsunami case in the southwestern Hokkaido Island, Japan. The performance of the DCGAN has been evaluated by comparing the results obtained with those from an unstructured mesh

finite element model (Fluidity) (Pain et al., 2001).

The structure of this article is organized as follows. Section 2.2 presents the governing equation for nonlinear fluid flows in dynamic systems. The architecture of the DCGAN is briefly introduced in section 2.3. The detailed process for the large data-driven modelling is described in section 2.4. Section 2.5 illustrates the capability of the DCGAN in the tsunami case, which entails the snapshot-based datasets generation with the different wave characteristics and the DCGAN training. Results of cases and the model efficiency are then presented in section 2.6. Conclusions are drawn in section 2.7.

2.2 Governing equations for nonlinear fluid flow prediction problems

Nonlinear fluid flows are usually simulated by solving the complex non-linear equations (for example, Navier-Stokes (NS) equations, Lorenz equations and shallow water equations) in the computational fluid dynamic models. In a general way, the governing equation of dynamic flows can be expressed as:

$$\frac{d\phi}{dt} = f(\phi, \mu, s, \mathbf{x}, t), \tag{2.1}$$

where ϕ is the state variable to be predicted (for example, water depth and velocity), μ represents the inputs (for example, initial and boundary conditions, modelling parameters) , $\mathbf{x} = (x, y, z)$ denotes the spatial coordinate system, s is the source term, and t is the time. f represents the nonlinear forcing factors which can result in a change in variable ϕ, for example, forces acting on the fluid element, diffusion, source and absorption terms.

This work focuses on the spatiotemporal fluid simulations that are driven by the inputs (for example, initial and boundary conditions, modelling parameters). The nonlinear relationship between the fluid flow solutions ϕ and the inputs μ in Eq.(2.1) can be re-written

in a general way:

$$\phi = \mathcal{P}(\mu), \tag{2.2}$$

where $\mathcal{P}$ represents a dynamic model. In this work, the dynamic model $\mathcal{P}$ will be represented by the DCGAN introduced in section 2.4. Based on a series of inputs μ, the main objective of this work is to develop a new DCGAN tool for simulating the spatiotemporal distribution of nonlinear fluid flows ϕ.

2.3 Adversarial network architecture

A generative adversarial network comprises a generator module (G) and a discriminator module (D), as shown in Fig. 2.1. The role of the generator is to map a random dataset μ (taking as inputs) into a desired output dataset ϕ. In traditional GANs, the random dataset μ is typically a Gaussian or uniform noise ranging from zero to one. The discriminator is used for distinguishing the real datasets ϕ_d (the targeted outputs) from the generated outputs ϕ. In the discriminator, $D(\phi_d) = 1$ if the targeted outputs are accepted while $D(G(\mu)) = 0$ if they are rejected.

In general, the training process in GAN can be considered as a minimization-maximization problem based on a cross entropy loss function $\mathbf{J}(D, G)$):

$$\min_G \max_D E_{\phi_d \sim p_{data}(\phi_d)}[\log D(\phi_d)] + E_{\mu \sim p_\mu(\mu)}[\log(1 - D(G(\mu)))], \tag{2.3}$$

where $p_\mu(\mu)$ is a prior distribution for the random dataset μ, and $p_{data}(\phi_d)$ is the corresponding probability data distribution for the targeted outputs ϕ_d.

In particular, the min-max process consists of two steps:

- Given a fixed generator, updating the discriminator by maximizing the discriminator function $\max_{\theta^{(D)}} \mathbf{J}^{(D)}\left(\theta^{(D)}, \theta^{(G)}\right)$ (where $\theta^{(D)}$ represents the parameters in the cost

functions of the discriminator, and the function $\mathbf{J}^{(D)}$ is differentiable.).

$$\mathbf{J}^{(D)}\left(\theta^{(D)},\theta^{(G)}\right) = E_{\phi_d\sim p_{data}(\phi_d)}[\log D(\phi_d,\theta^{(D)})]+E_{\mu\sim p_\mu(\mu)}[\log(1-D(G(\mu),\theta^{(G)},\theta^{(D)})]; \tag{2.4}$$

- Updating the generator by minimizing the generator function, i.e. $\min_{\theta^{(G)}} \mathbf{J}^{(G)}\left(\theta^{(D)},\theta^{(G)}\right)$ (where $\theta^{(G)}$ represents the parameter in the cost functions of the generator, and the function $\mathbf{J}^{(G)}$ is differentiable.)

$$\mathbf{J}^{(G)}\left(\theta^{(D)},\theta^{(G)}\right) = E_{\mu\sim p_\mu(\mu)}[\log(1-D(G(\mu)))]. \tag{2.5}$$

Following Goodfellow et al. (2014), the min-max game has a global optimum for $p_g(\phi) = p_{data}(\phi_d)$ (where p_g is a probability distribution of the generated output ϕ when $\mu \sim p_\mu(\mu)$), if D and G have enough capacity. The optimal discriminator for the $\mathbf{J}^{(G)}(G,D)$ can be characterized by:

$$D^*(\phi_d) = \frac{p_{data}(\phi_d)}{p_{data}(\phi_d)+p_g(\phi)} \tag{2.6}$$

The minimax game in Eq.(2.3) can be rewritten as

$$\max_D \mathbf{J}^{(G)}(G,D) = E_{\phi_d\sim p_{data}}[\log\frac{p_{data}(\phi_d)}{p_{data}(\phi_d)+p_g(\phi)}] + E_{\phi\sim p_g}[\log\frac{p_g(\phi)}{p_{data}(\phi_d)+p_g(\phi)}] \tag{2.7}$$

When the $D^*(\phi_d) = \frac{1}{2}$, the generator can use any random dataset μ to synthesis the generated outputs ϕ, and the discriminator loses the ability to distinguish between the real dataset ϕ_d and generated ones ϕ.

Gradient-based algorithms can be used for optimization of parameters $\theta^{(G)},\theta^{(D)}$ in the generator and discriminator. For details see sections below.

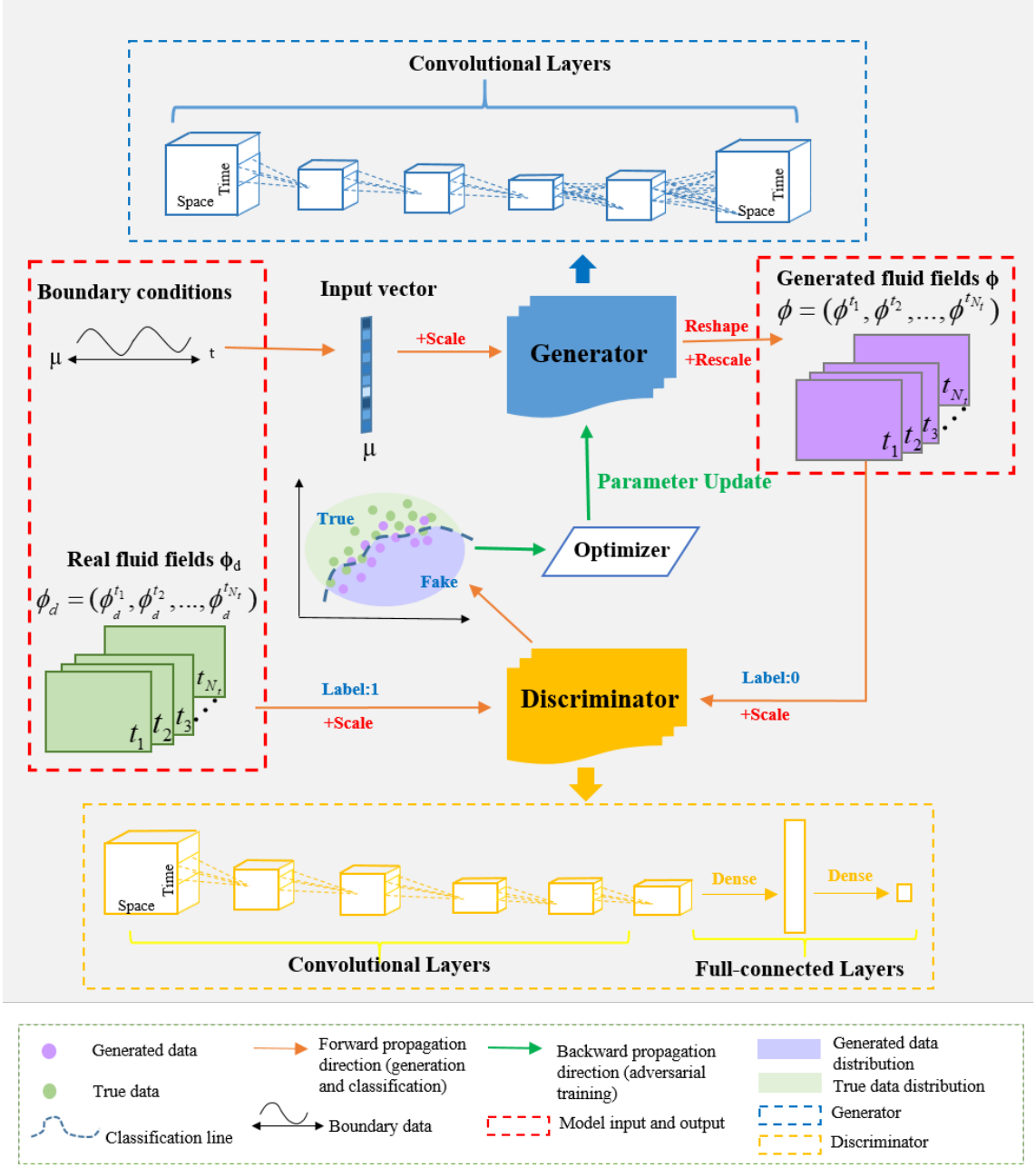


Fig. 2.1. Deep convolutional generative adversarial network architecture. The ϕ and ϕ_d can be represented by fluid flow solutions (e.g. water depth).

2.4 Deep convolutional GAN for spatiotemporal data-driven modelling

In this section, the details of constructing a predictive DCGAN for modelling nonlinear fluid flows are provided. First, a number of inputs and solution snapshots from a high fidelity model (as training datasets) are scaled for the deep convolutional neural networks. The DCGAN is then trained in the forward and backward propagations. Finally for any given new inputs, the spatiotemporal distributions of results ϕ are obtained by the predictive DCGAN.

2.4.1 Spatiotemporal datasets for deep convolutional networks

2.4.1.1 Input and output datasets

For spatiotemporal simulations, the output dataset in Eq.(2.1) can be expressed:

$$\phi = \begin{bmatrix} \phi_1^{t_1} & \phi_2^{t_1} & \dots & \phi_{N_x}^{t_1} \\ \phi_1^{t_2} & \phi_2^{t_2} & \dots & \phi_{N_x}^{t_2} \\ \vdots & \vdots & & \vdots \\ \phi_1^{t_{N_t}} & \phi_2^{t_{N_t}} & \dots & \phi_{N_x}^{t_{N_t}} \end{bmatrix} = (\phi^{t_1}, \phi^{t_2}, \dots, \phi^{t_{N_t}}), \tag{2.8}$$

while the input dataset

$$\mu = (\mu^{t_1}, \mu^{t_2}, \dots, \mu^{t_{N_t}}) \tag{2.9}$$

where the variables: $\phi, \mu \in \mathbb{R}^{N_x \times N_t}$, where N_t is the number of timesteps and N_x denotes the number of points in grids of the computational domain Ω. For the inputs μ, N_x represents the points along the boundary, such as the left boundary in Fig. 2.4. In a

DCGAN, the outputs ϕ can be re-constructed in a tensor form:

$$\phi, \mu : \phi, \mu \in \mathbb{R}^{N_x \times N_t} \rightarrow \mathbb{R}^{H \times W \times D}, \tag{2.10}$$

where the H represents the number of timesteps N_t, W is set to one, D denotes the number of points N_x in the computational domain Ω. This is similar to traditional image processing by convolutional neural networks, while D is represented with one or three invariant RGB color values at each pixel in rectangular images. In this way, the feature extraction process of fluid fields with unstructured mesh is to capture complicated non-linear fluid patterns in all timesteps at every point. Considering the temporal feature extraction by the square filters, the zero-padding operator is used to preserve the original input size.

2.4.1.2 Scaled datasets for deep convolutional networks

For numerical stabilization during the training process, all the inputs μ as well as the outputs ϕ are $scaled$ into a range between zero and one, as recommended by Lee and Carlberg (2018). The $scale$ operator on the inputs μ and outputs ϕ at time level t_j ($t_j \in [0, t_{N_t}]$) and at point n ($n \in [0, N_x]$), is given by:

$$\widetilde{\mu}_n^{t_j} = \frac{\mu_n^{t_j} - \mu_{min}}{\mu_{max} - \mu_{min}}, \quad \widetilde{\phi}_n^{t_j} = \frac{\phi_n^{t_j} - \phi_{min}}{\phi_{max} - \phi_{min}}, \tag{2.11}$$

where all quantities with a widetilde denote the scaled values, and the subscripts max and min represent the maximum and minimum values of μ and ϕ, respectively.

After obtaining results ($\widetilde{\phi}_n$) from the final convolutional layer, the generated data will be $rescaled$ back to the original scale ϕ by:

$$\phi_n^{t_j} = \widetilde{\phi}_n^{t_j}(\phi_{max} - \phi_{min}) + \phi_{min}. \tag{2.12}$$

2.4.2 Forward propagation

The DCGAN architecture is shown in Fig. 2.1. Here convolutional neural networks are used for extracting the spatiotemporal features. The CNNs define the relationships between the input and the output feature maps. In CNNs, the scaled values are used here. For simplicity, the widetilde in equations is dropped out afterwards.

2.4.2.1 Convolutional neural network in simulations of spatiotemporal flows

A CNN consists of convolutional layers and pooling layers. To preserve the size of the feature maps, zeros are padded around input feature maps. The filters with a size of $N_p \times N_p$ are used with a stride in CNNs .

The convolutional operation between the m^{th} input feature map in the $(l-1)^{th}$ layer and the n^{th} output feature map in the l^{th} layer can be expressed as follows:

$$\phi_n^{t_j^l} = \boldsymbol{\sigma}\left(b_n + \sum_{m=1}^{N_f^{l-1}} \sum_{p=1}^{N_p \times N_p} w_{m,n}^{p} \phi_{m,n}^{t_j^l,p}\right), \tag{2.13}$$

and the variables:

- $\phi_m^{t_j^l,p}, m \in (1,\ldots,N_f^{l-1})$ represents the value of a neuron (at temporal position t_j^l and at spatial position p) of the m^{th} input feature map in the $(l-1)^{th}$ layer;
- $\phi_n^{t_j^l,p}, n \in (1,\ldots,N_f^{l})$ represents the value of neurons (at temporal position t_j^l and at spatial position p) of the n^{th} output feature map at convolutional layer l;
- N_f^{l-1} and N_f^l are the number of feature maps in convolutional layers $l-1$ and l respectively;
- $N_p \times N_p$ size of filters with a stride.

- $\boldsymbol{\sigma}$ represents the activation function;
- w_m^p is the weight at position p of the m^{th} input feature map in the $(l-1)^{th}$ layer;
- b_n is the bias of the n^{th} output feature map in the l^{th} layer.

The operation between the convolutional layers $l-1$ and l in Eq.(2.13) can be re-written in a general way:

$$\phi_l = \mathcal{M}_l(\phi_{l-1}), \tag{2.14}$$

where

$$\phi_l = (\phi^1, \ldots, \phi^{t_j}, \ldots, \phi^{N_t^l})^T, \;\; \phi_{l-1} = (\phi^1, \ldots, \phi^{t_j^l}, \ldots, \phi^{N_t^{l-1}})^T \tag{2.15}$$

where N_{l-1} and N_l are the number of temporal positions connected to spatial feature maps in convolutional layers $l-1$ and l respectively; and

$$\phi^{t_j^l} = (\phi_1^{t_j^l}, \ldots, \phi_n^{t_j^l}, \ldots, \phi_{N_x^l}^{t_j^l}), \;\; \phi_{l-1} = (\phi_1^{t_1^{l-1}}, \ldots, \phi_m^{t_j^{l-1}}, \ldots, \phi_{N_x^{l-1}}^{N_t^{l-1}}), \tag{2.16}$$

where N_x^{l-1} and N_x^l are the number of spatial feature maps in convolutional layers $l-1$ and l respectively.

2.4.2.2 CNNs in the generator and discriminator

Given a set of input feature maps $\mu_i, i \in (1, N_f)$ defined in Eq.(2.9), the convolutional operation for the generator can be expressed:

$$\phi = \mathcal{G}(\mu) = \mathcal{M}_{L^{(G)}}^{(G)} \ldots \mathcal{M}_2^{(G)} \mathcal{M}_1^{(G)}(\mu), \tag{2.17}$$

while the discriminator is used for distinguishing the real datasets ϕ_d (the targeted outputs) from the generated outputs $\phi = \mathcal{G}(\mu)$

$$\mathcal{D}\left(\phi_d, \mathcal{G}(\mu)\right) = \mathcal{M}_{L^{(D)}}^{(D)} \ldots \mathcal{M}_2^{(D)} \mathcal{M}_1^{(D)}(\phi_d, \mathcal{G}\left(\mu\right)), \tag{2.18}$$

where $L^{(G)}$ and $L^{(D)}$ are the number of convolutional layers used in the generator and discriminator respectively. For the hidden layers, the rectifier non-linearity (ReLU) activation function ($f(x) = \max(0, x)$) is chosen as the activation function. For the output layer, the hyperbolic tangent function ($f(x) = \tanh(x)$) is adopted in the generator, while the sigmoid function ($f(x) = 1/(1 + \exp^{-x})$) is employed in the discriminator (Mosser et al., 2017; Lee and You, 2018).

2.4.3 Backward propagation

Let $\theta^{(D)} = (w^{(D)}, b^{(D)})$ and $\theta^{(G)} = (w^{(G)}, b^{(G)})$ represent the weights and biases in the cost functions of the discriminator and generator, respectively. To minimax the cost function $\mathbf{J}(D, G)$, the parameters $\theta^{(G)}$ and $\theta^{(D)}$ are updated by the surrogate loss (Metz et al., 2016) at the backward propagation:

$$\theta^{(G)} \leftarrow \theta^{(G)} - \eta\left(\frac{\partial \mathbf{J}^{(G)}(\theta^{(G)}, \theta_P^{(D)}(\theta^{(G)}, \theta^{(D)}))}{\partial \theta^{(G)}} + \frac{\partial \mathbf{J}^{(G)}(\theta^{(G)}, \theta_P^{(D)}(\theta^{(G)}, \theta^{(D)}))}{\partial \theta_P^{(D)}(\theta^{(G)}, \theta^{(D)})} \frac{d\theta_P^{(D)}(\theta^{(G)}, \theta^{(D)})}{d\theta^{(G)}}\right), \tag{2.19}$$

$$\theta^{(D)} \leftarrow \theta^{(D)} + \eta\left(\frac{\partial \mathbf{J}^{(D)}(\theta^{(D)}, \theta^{(G)}(\theta^{(G)}, \theta^{(D)}))}{\partial \theta^{(D)}} + \frac{\partial \mathbf{J}^{(D)}(\theta^{(D)}, \theta^{(G)}(\theta^{(G)}, \theta^{(D)}))}{\partial \theta^{(G)}(\theta^{(G)}, \theta^{(D)})} \frac{d\theta^{(G)}(\theta^{(G)}, \theta^{(D)})}{d\theta^{(D)}}\right), \tag{2.20}$$

where P is the iterated steps, as $P \to \infty$, the $\theta_P^{(D)}$ achieves a local optimum, where $\frac{\partial \mathbf{J}^{(D)}}{\partial \theta_P^{(D)}} = 0$.

The adaptive Moment Estimation (Adam) optimizer is adopted (Kingma and Ba, 2014) to optimize the values of parameters $\theta^{(D)}$ and $\theta^{(G)}$, with a learning rate of 0.0002. The learning rate is chosen by a trial and error method. If the learning rate is low, then the generator is hard to fool the discriminator and optimization will take a lot of time because steps towards the minimum of the loss function are tiny. If the learning rate is high, then training may not converge or even diverge. The discriminator loses its power to

distinguish which is true or fake very quickly. The processes of forward and backward propagations in the DCGAN are summarized in algorithm 1.

Algorithm 2.1 Forward and backward propagations in the DCGAN.

- Sample $\{\mu^{t_1}, \mu^{t_2}, \ldots, \mu^{t_{N_t}}\}$ from the inputs .
- Sample $\{\phi_d^{t_1}, \phi_d^{t_2}, \ldots, \phi_d^{t_{N_t}}\}$ from the real datasets.
- Scale the datasets (μ, ϕ_d)

for K = 1 to N_K (the number of training iterations) **do**

for P = 1 to N_P (iterations for updating the discriminator) **do**

- Compute the outputs in the generator: $\phi = \mathcal{G}(\mu, \theta^{(G)})$
- Compute the probability in the discriminator: $\mathcal{D}\left(\phi_d, \mathcal{G}(\mu, \theta^{(G)}), \theta^{(D)}\right)$
- Update the $\theta^{(D)}$ in the discriminator by ascending its stochastic gradient:

$$\theta^{(D)} \leftarrow \theta^{(D)} + \eta \frac{dJ^{(D)}(\theta^{(G)}, \theta^{(D)})}{d\theta^{(D)}}$$

end for

- Update the $\theta^{(G)}$ in the generator by descending its stochastic gradient:

$$\theta^{(G)} \leftarrow \theta^{(G)} - \eta \frac{dJ^{(G)}(\theta^{(G)}, \theta^{(D)})}{d\theta^{(G)}}$$

end for

- Rescale the outputs (ϕ)

2.4.4 Training and predictive processes

Offline training and validation processes⇒ Given N pairs of input-output $(\mu, \phi_d) \in (\vartheta_{tr}, \phi_{tr})$ during the training and validation processes (as shown in Fig. 2.2(a)), the input datasets $\mu_s \in \vartheta_{tr}$ are:

$$\vartheta_{tr} : \mu_s \in \mathbb{R}^{N_x \times N_t}, s \in \{1, ..., N\}, \tag{2.21}$$

while the corresponding output datasets $\phi_{d,s} \in \phi_{tr}$ are:

$$\phi_{tr} : \phi_{d,s} \in \mathbb{R}^{N_x \times N_t}, s \in \{1, ..., N\}, \tag{2.22}$$

where ϑ_{tr} and ϕ_{tr} denote the training inputs and the targeted outputs, respectively. A predictive DCGAN can be obtained after the training process (for details, see algorithm 1).

Online prediction process⇒ Given a new series of predicted inputs $\mu \in (\vartheta \backslash \vartheta_{tr})$ (where ϑ is the whole inputs, ϑ_{tr} is the training and validated inputs), the spatiotemporal distributions of results ϕ are obtained by the predictive DCGAN (as shown in Fig. 2.2(b)).

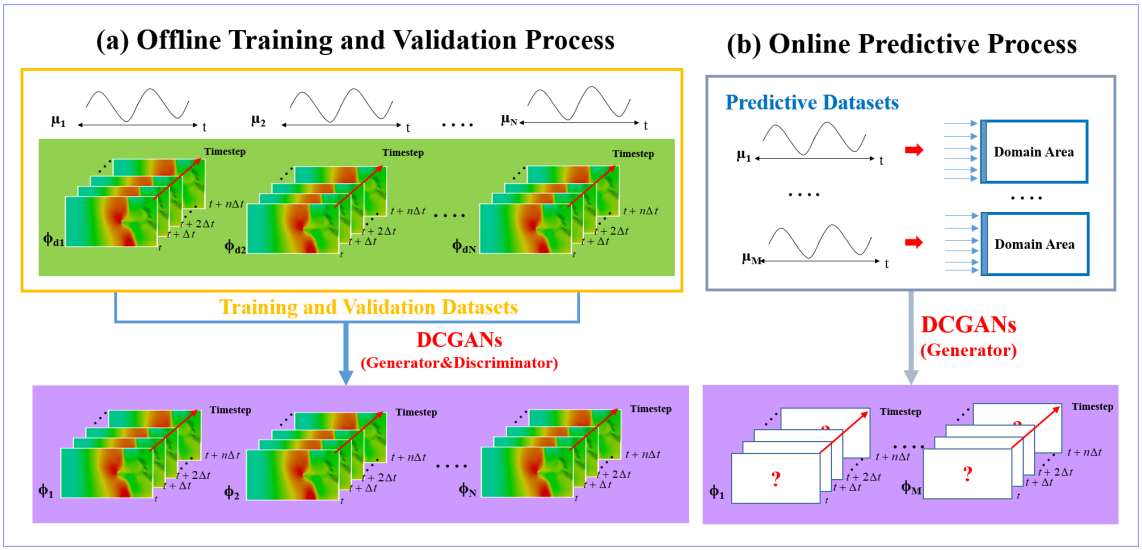


Fig. 2.2. Overview of the DCGAN in offline training process and online predictive process of constructing fluid fields.

2.5 Numerical examples

In this section, two flooding examples are presented to illustrate the capabilities of the DCGAN in resolving nonlinear fluid flow problem governed by the NS equations. In these examples, the inputs are a series of incoming waves along the coastal line while the corresponding outputs are the flooding solutions (e.g. water depth).

2.5.1 Descriptions of study cases

To demonstrate the predictive capability of the DCGAN for dynamic fluid flows, it has been applied to a case of flooding in Hokkaido, Japan. The flooding happened in 1993, caused by the Hokkaido Nansei-Oki earthquake offshore of southwestern Hokkaido Island, Japan. Incoming waves along the coastal line have been identified as an important

driver for flood hazards in coastal areas. The increased demand to improve flood prediction requires better understanding of uncertainties associated with flood hazards due to the variability of waves (Hu et al., 2018b, 2019b). In our flooding study cases, the characteristics of waves including the wave phase and wave height were considered. The phase variation represents the lags in timing of simulated flows (Prakash et al., 2014; Chang and Tsai, 2016), and the wave height variation indicates the magnitude of the flow peak (Kasiviswanathan et al., 2016; Chang and Tsai, 2016).

To investigate the danger caused by the wave run-up and flooding, two series of incoming waves $(H^{(e)}, e = (1, 2))$ are given and included in the inputs μ in two study cases, where each series of waves contains 13 types of waves, as shown in Fig. 2.3. The wave series can be expressed as followings:

Case 1: variation of wave phases

$$H^{(1)} = 0.01 \sin(t + \frac{\pi}{12}s), 0 \leq t \leq 50, s = 0, 1, 2, \ldots, 12, \tag{2.23}$$

Case 2: variation of wave heights

$$H^{(2)} = 0.01(0.4 + 0.1s) \sin(t), 0 \leq t \leq 50, s = 0, 1, 2, \ldots, 11, 26, \tag{2.24}$$

where $H^{(1)}$ and $H^{(2)}$ are the wave height (unit: m); t denotes the time (unit: s); and $\frac{\pi}{12}s$ represents the wave phase. Two cases are an initial try to use the DCGAN for spatiotemporal fluid flow prediction. Therefore, I only change the wave phase and wave height with the constant wave shape in both cases. In Chapter 5, the DCGAN is proven to make flood predictions using varied shapes of waves in Denmark.

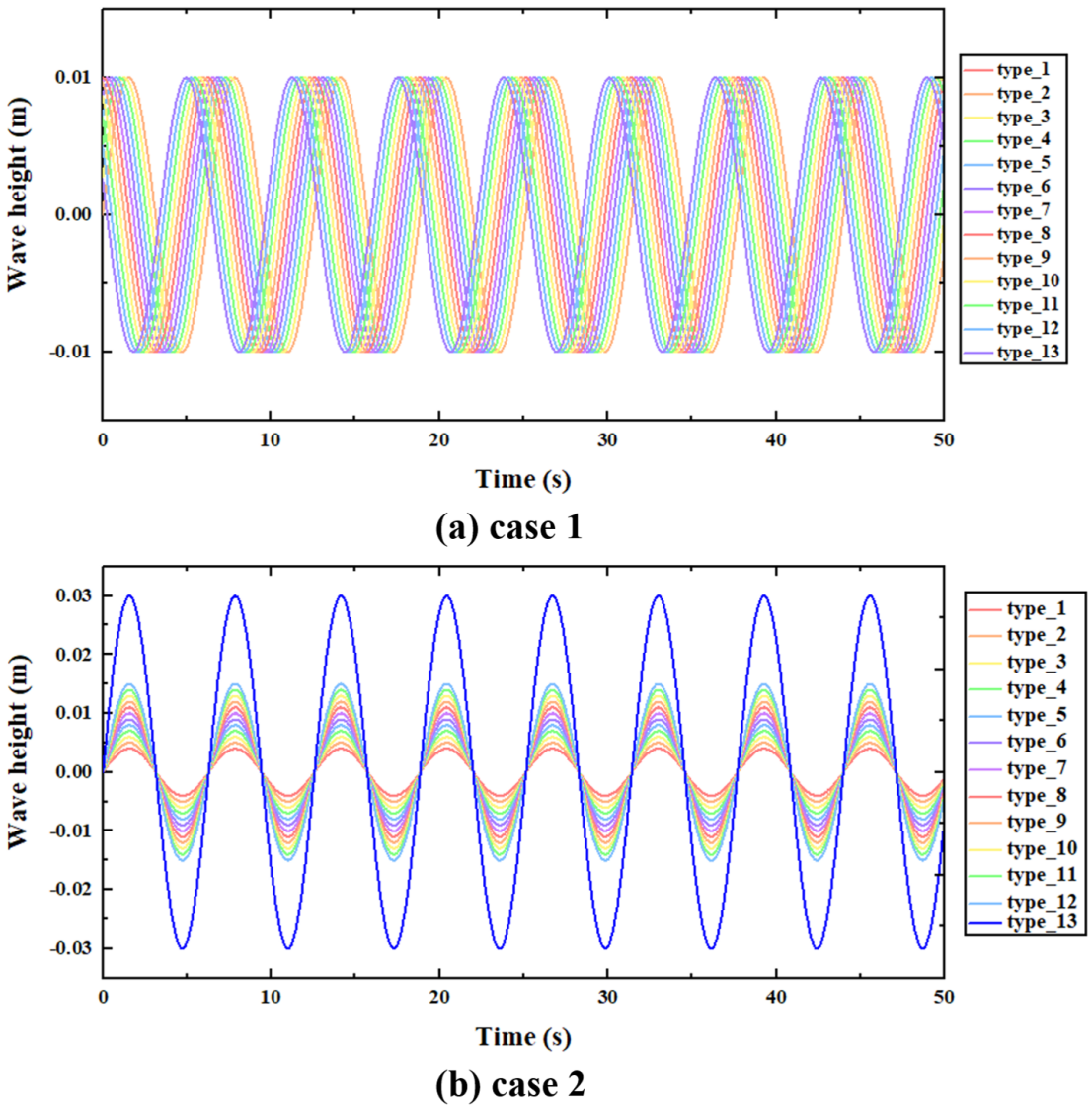


Fig. 2.3. Input varied waves for two cases: (a) representing the wave phase variation and (b) representing the wave height variation.

To simulate this tsunami, Fluidity (referred as the original high fidelity model, developed at Imperial College London, for details see Pain et al. (2001)) is employed to generate the training and validation datasets by solving the NS equations on unstructured finite element meshes. The computational domain is $5.448m \times 3.402m$ in horizontal, and one element deep is applied in vertical. The detailed bathymetry and coastal topography in model setup are referred to Funke et al. (2011). As shown in Fig. 2.4, the free-surface elevation on the left boundary was set to the incoming wave height, and no normal flow boundary condition was given to the remaining boundaries. The wetting and drying threshold value

was chosen to be $0.5mm$. In addition, the gravity magnitude was set to be $9.81ms^{-2}$, and $0.01m^2s^{-1}$ was adopted for the isotropic kinematic viscosity. A Manning-Strickler drag at the bottom is 0.002 $sm^{-\frac{1}{3}}$. The unstructured mesh with 7314 points (shown in Fig. 2.4) was generated by Gmsh, which is a free 3D finite element mesh generator with a built-in CAD engine and post-processor (Geuzaine and Remacle, 2009). The nonlinear fluid flow simulation starts with a static state. The whole simulation period was $50s$ with a timestep of $0.1s$.

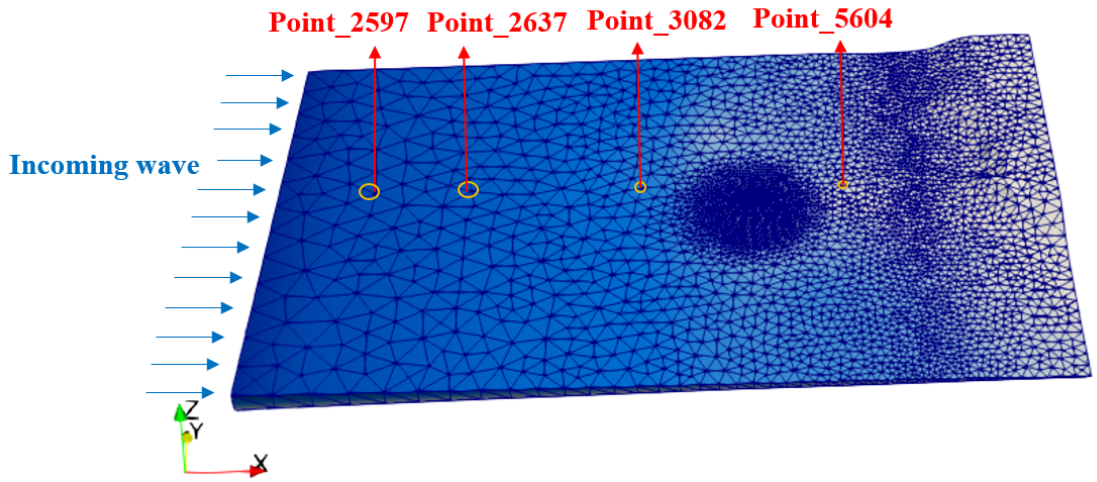


Fig. 2.4. Computational domain area and unstructured mesh (the incoming wave is the boundary condition along the costal line, and the simulation starts with a static state).

2.5.2 Data collections and predictive analysis

Collection of solution snapshots obtained by the original high fidelity model: Given a set of incoming waves for cases 1 and 2 ($e = 1, 2$) (defined in Eq.(2.23)-Eq.(2.24) and Fig. 2.3), which are re-written in a time discretized form:

$$\mu_s^e = (\mu_s^{e,t_1}, \mu_s^{e,t_2}, ..., \mu_s^{e,t_{N_t}}), (e \in [1, 2], s \in [0, 12], N_t = 501), \tag{2.25}$$

the corresponding targeted outputs (solution snapshots) are generated by running the orig-

inal high fidelity model:

$$\phi^{e}_{d,s} = \begin{bmatrix} \phi^{e,t_1}_{d,s,1} & \phi^{e,t_1}_{d,s,2} & \cdots & \phi^{e,t_1}_{d,s,N_x} \\ \phi^{e,t_2}_{d,s,1} & \phi^{e,t_2}_{d,s,2} & \cdots & \phi^{e,t_2}_{d,s,N_x} \\ \vdots & \vdots & & \vdots \\ \phi^{e,t_{N_t}}_{d,s,1} & \phi^{e,t_{N_t}}_{d,s,2} & \cdots & \phi^{e,t_{N_t}}_{d,s,N_x} \end{bmatrix}, (e \in [1,2], s \in [0,12], N_x = 7314, N_t = 501). \tag{2.26}$$

Training and validation processes: Given the training and validated input-output pairs, a predictive DCGAN is constructed. The first ten types of incoming waves in Eq.(2.23)-Eq.(2.24) are referred as the training inputs:

$$\vartheta_{tr} : \mu^{e}_{s} \in \mathbb{R}^{N_t}, (e \in [1,2], s \in [0,9], N_t = 501), \tag{2.27}$$

while the corresponding training outputs are the collection of the solution snapshots for each type of incoming waves in each case:

$$\phi_{tr} : \phi^{e}_{d,s} \in \mathbb{R}^{N_x \times N_t}, (e \in [1,2], s \in [0,9], N_x = 7314, N_t = 501). \tag{2.28}$$

Predictive analysis: Given a set of new inputs $(\vartheta \backslash \vartheta_{tr})$, the predictive analysis is undertaken by the DCGAN. Here the inputs $(\vartheta \backslash \vartheta_{tr})$ are the last three types of incoming waves defined in Eq.(2.23)-Eq.(2.24):

$$(\vartheta \backslash \vartheta_{tr}) : \mu^{e}_{s} \in \mathbb{R}^{N_t}, (e \in [1,2], s \in [10,12], N_t = 501). \tag{2.29}$$

2.5.3 Performance assessment

To assess the performance of the DCGAN in the predictive process, the errors of fluid fields between the generated outputs ϕ and the targeted outputs ϕ_d are evaluated at each point. The correlation coefficient (R) and the root mean squared error (RMSE) are pro-

vided. The calculation formulas are given below:

$$R^{n_t} = \frac{\sum_{i=1}^{N_x} (\phi_i^{n_t} - \overline{\phi}^{n_t})(\phi_{d,i}^{n_t} - \overline{\phi}_d^{\,n_t})}{\sqrt{\sum_{i=1}^{N_x} (\phi_i^{n_t} - \overline{\phi}^{n_t})^2} \sqrt{\sum_{i=1}^{N_x} (\phi_{d,i}^{n_t} - \overline{\phi}_d^{\,n_t})^2}}, \tag{2.30}$$

$$RMSE^{n_t} = \sqrt{\frac{\sum_{i=1}^{N_x} (\phi_i^{n_t} - \phi_{d,i}^{n_t})^2}{N_x}}, \tag{2.31}$$

where n_t is the current timestep, $n_t \in [0, N_t]$, $\phi_i^{n_t}$ and $\phi_{d,i}^{n_t}$ are generated and targeted water depths at point i ($i \in [0, N_x]$) respectively, while $\overline{\phi}^{n_t}$ and $\overline{\phi}_d^{\,n_t}$ denote the corresponding average water depths over all points at current timestep n_t.

2.6 Results

2.6.1 Case 1: Wave phase variation

In this case, the numerical illustration of the DCGAN was set up with a series of waves where the wave phase varies in the Hokkaido tsunami case. Given a new input $\mu \in (\vartheta \backslash \vartheta_{tr})$ (here, μ_{12}^1 defined in Eq.(2.29)), a comparison between the DCGAN and the original high fidelity model results is carried out.

Fig. 2.5 presents the comparison of water depth obtained from the DCGAN and the original high fidelity model at timesteps $60, 90, 216, 441$. It is worth noting that the results from the DCGAN attain a good agreement to those from the original high fidelity model at each timestep. The differences of the water depth between the DCGAN and the original high fidelity model are illustrated in Fig. 2.6. The differences are quite small over the whole domain, which suggests the DCGAN is able to obtain reasonable and accurate solutions for nonlinear fluid flows.

As shown in Fig. 2.7, the temporal variation of water depth is demonstrated at four points (locations shown in Fig. 2.4). It can be observed that the curves of water depth from the DCGAN have a slight difference from that of the original high fidelity model. In addition, the arriving time lags at four points can be noticed. The arrival time of flooding at point ID = 5604 is at timestep 50, where it is the farthest from the coastal boundary among the four points.

To further estimate the performance of the DCGAN, the correlation coefficient and RMSE are further illustrated in Fig. 2.8. It is evident that the values of RMSE are between $0.002m$ to $0.004m$ while the correlation coefficients are above 99%. These results demonstrate that the DCGAN performs very well and the predicted fluid fields are in good agreement with the true fluid fields.

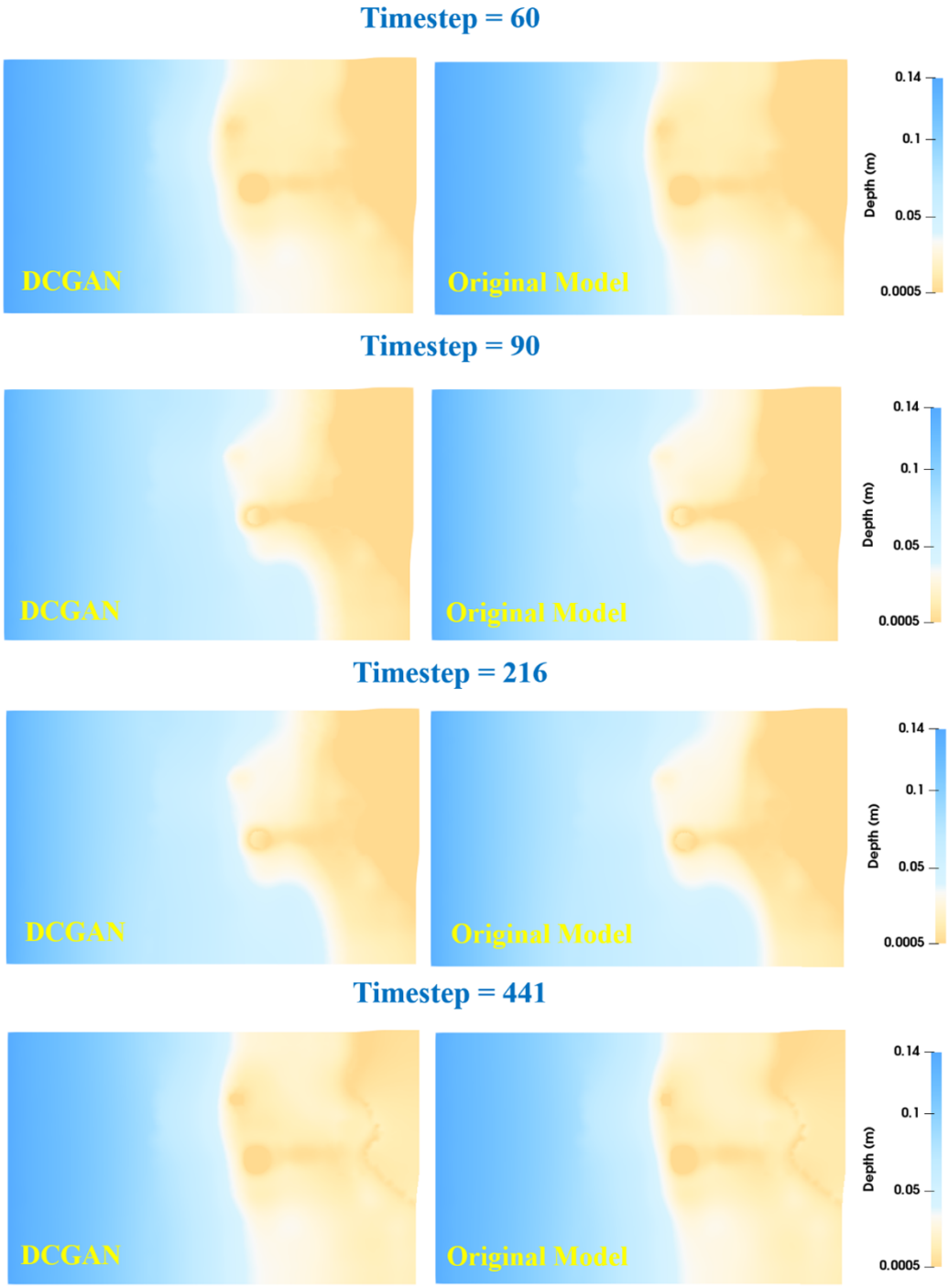


Fig. 2.5. Comparison of the spatial distribution of water depth fields obtained from the DCGAN (left) and the original high fidelity model (right) at timesteps $60, 90, 216, 441$.

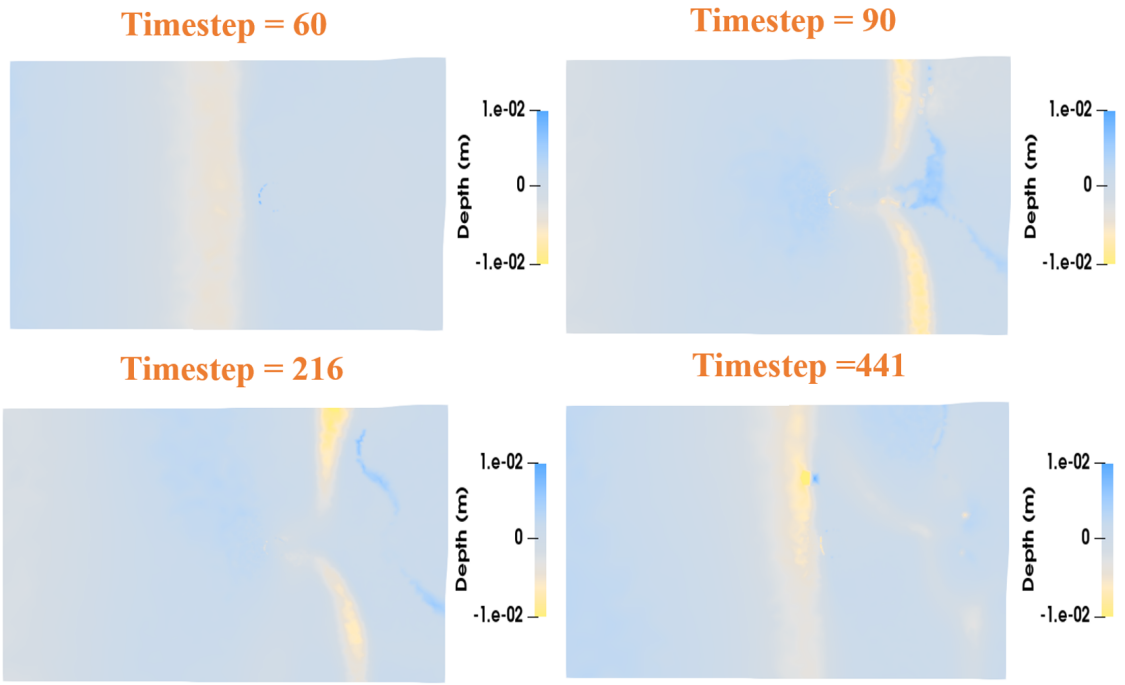


Fig. 2.6. Differences of water depth fields between the DCGAN and the original high fidelity model at timesteps $60, 90, 216, 441$.

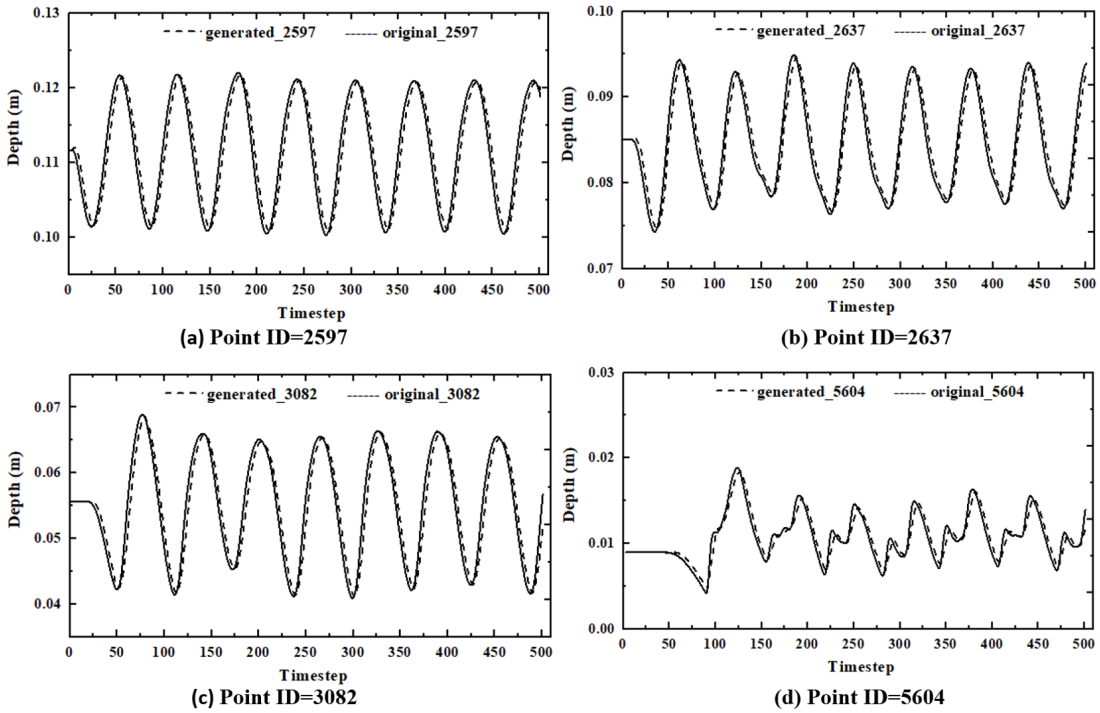
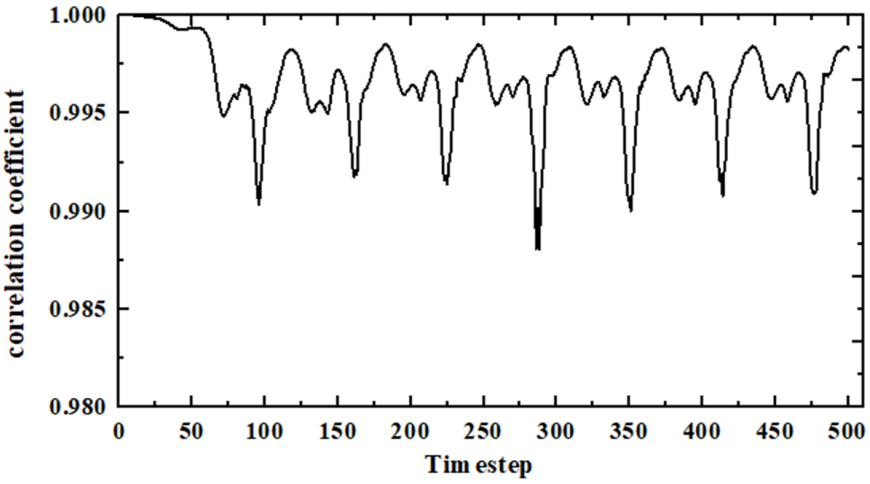
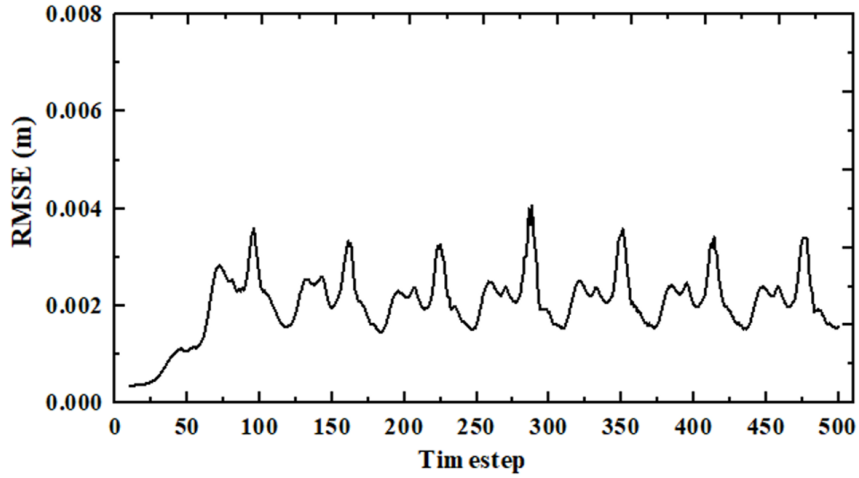


Fig. 2.7. Comparison of the temporal variation of water depth at points ID = 2597, 2637, 3082, 5604.



(a) Correlation coefficient



(b) RMSE

Fig. 2.8. The RMSE and correlation coefficients of water depth between the generated fields and original high fidelity fields during whole simulational period.

2.6.2 Case 2: Wave peak variation

In this case, the numerical illustration of the DCGAN was set up with a series of waves where the wave height varies. For a given input $\mu \in (\vartheta \backslash \vartheta_{tr})$ (here, μ_{12}^2 defined in Eq.(2.29)), Fig. 2.9 shows the comparison of predictive results between the DCGAN and the original high fidelity model. It is observed that most of water depth features are captured compared to the original fluid fields. In addition, the correlation coefficient of results at timesteps $50, 95, 205, 493$ achieves values beyond 97% shown in Fig. 2.10. These results highlight

the good performances of the DCGAN in predictive analysis of fluid fields.

The temporal variations of water depth at four points are presented as the box plots in Fig. 2.11. The middle line represents the mean value 50%, the upper represents the maximum (1 box) and the lower line denotes the minimum (-1 box). As regards to the mean value, the differences of mean values are less than $0.002m$ at points ID = 2597, 2637, 3082 and 5604. The depth peak predicted by the DCGAN is $0.1439m$ at the point ID = 2597, which is comparable to that $(0.1352m)$ from the original high fidelity model. A good performance of the DCGAN in predicting depth peak is achieved at points ID = 2637, 3082, and 5604. As for the minimum depth at the four points, the differences are less than $0.01m$ as depicted in Fig. 2.11.

The error analysis of water depth solutions has been further carried out using the correlation coefficient and RMSE during the simulation period, as shown in Fig. 2.12. The values of RMSE fluctuate at different timesteps and are less than $0.012m$. The correlation coefficient of results between the DCGAN and the high fidelity model exceeds 90%.

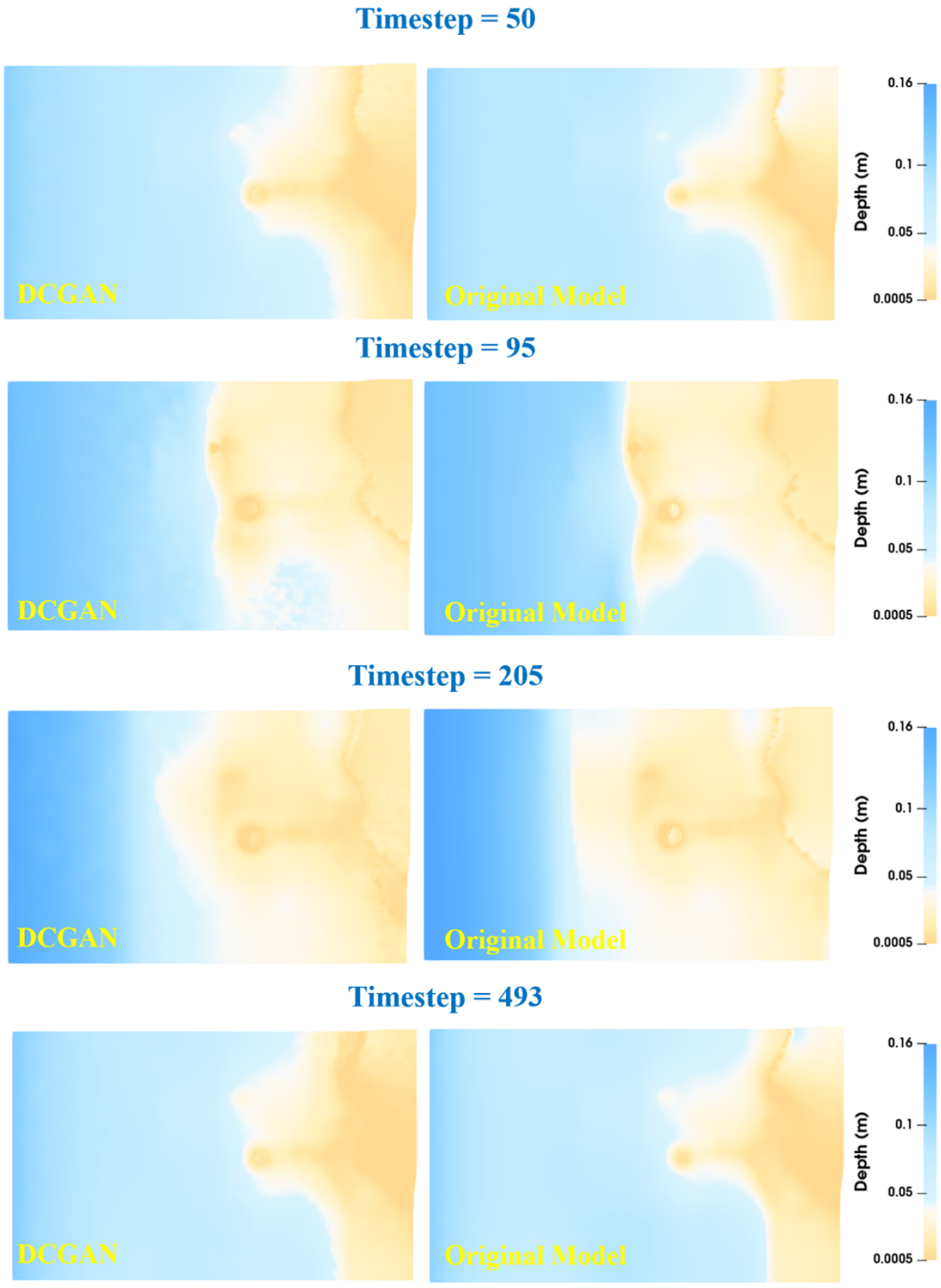


Fig. 2.9. Comparison of spatial distribution of water depth fields obtained from the DC-GAN (left) and the original high fidelity model (right) at timesteps $50, 95, 205, 493$.

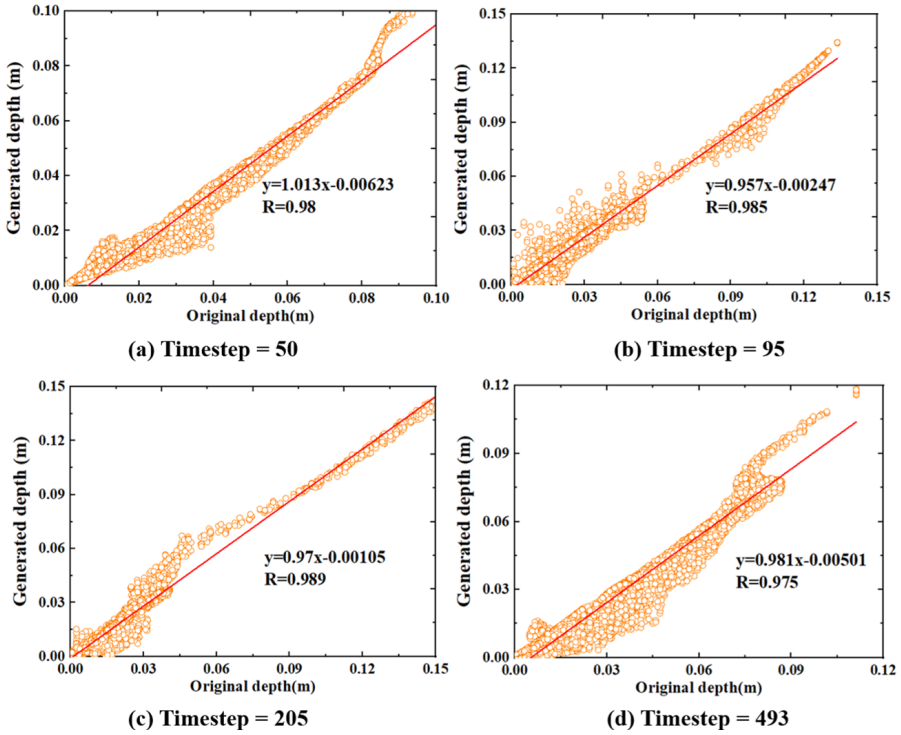


Fig. 2.10. The correlation coefficients of water depth solutions between the DCGAN and the original high fidelity model at timesteps $50, 95, 205, 493$.

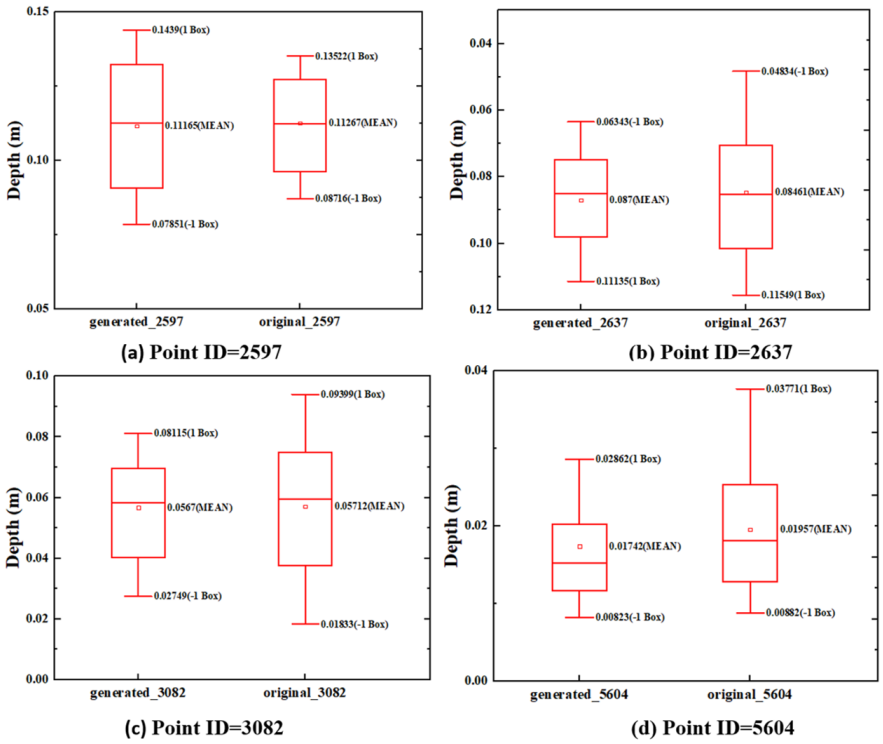


Fig. 2.11. Comparison of water depth solutions between the DCGAN and the original high fidelity model at four points during whole simulational period.

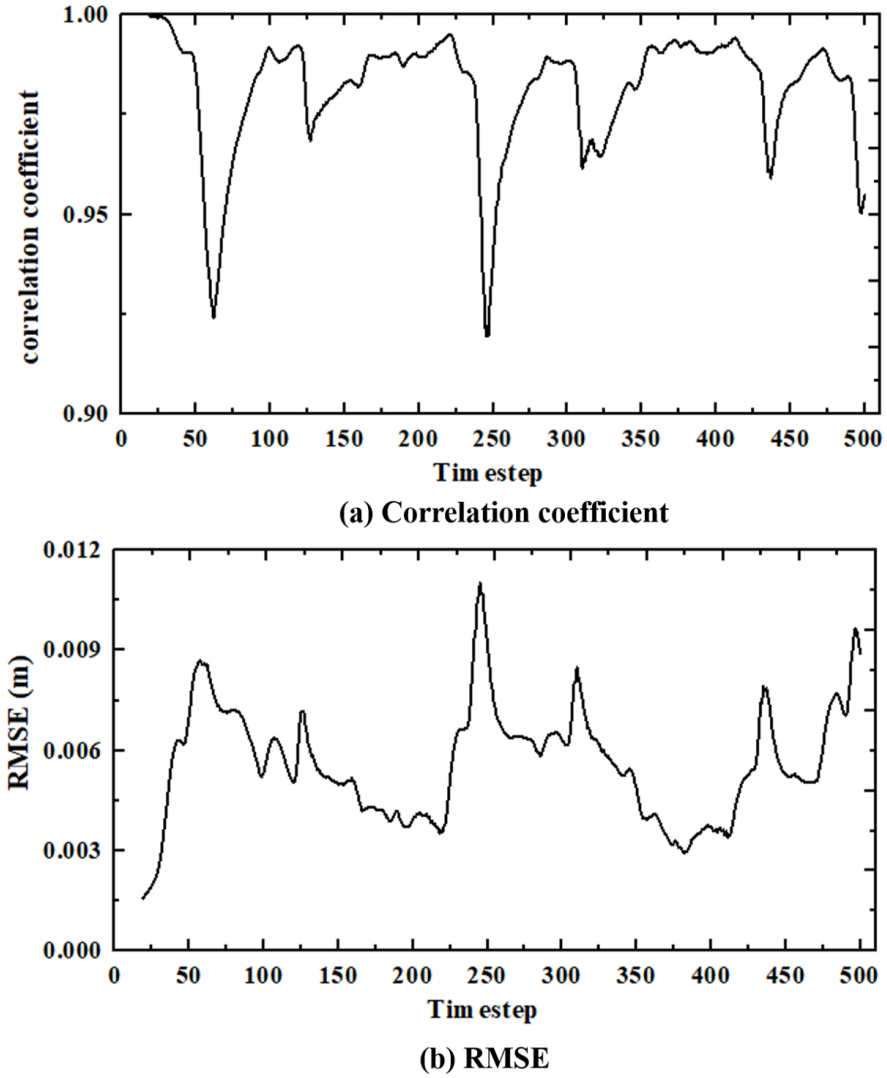


Fig. 2.12. The RMSE and correlation coefficients of water depth solutions between the DCGAN and the original high fidelity model during whole simulational period.

2.6.3 Computational efficiency

The DCGAN and the original high fidelity model were run on a computer (Intel Xeon(R) CPU@ 3.60GHz). The offline training time is $1h34min22s$. In terms of computation efficiency, it is worth mentioning that the online CPU cost used for running the DCGAN is less than $0.5s$, while $5676s$ for running the original high fidelity model. It can be seen that using the DCGAN, the CPU time is reduced by five orders of magnitude in comparison to the original high fidelity model.

To summarize, the DCGAN has successfully captured fluid flow features and provides the high-resolution fluid fields, especially for the nonlinear fluid flows in time and space. This serves as a technologically important motivation for using the DCGAN to efficiently forecast complicated fluid flows for realistic applications.

2.7 Conclusion

In this work, a DCGAN has been proposed for nonlinear spatiotemporal fluid flow prediction. It is the first time that the DCGAN is applied to prediction of spatially-temporally-coherent and high-dimensional nonlinear fluid flows. This novel DCGAN enables rapid and reliable prediction and can be used for providing risk reduction strategies in emergencies.

The performance of the DCGAN has been illustrated in the case of Hokkaido tsunami in this work. Two study cases are set up with two series of waves with varying wave phase and wave height. To estimate the accuracy of the DCGAN, a comparison of nonlinear flow results obtained by the DCGAN and the original high fidelity model has been carried out. It is shown that the DCGAN is able to capture the complex flow features and exhibits an overall good agreement with the original high fidelity model. Error analysis has also been performed for further evaluation of the DCGAN through RMSE and correlation coefficient. In comparison to the original high fidelity model, the DCGAN achieves a reasonable accuracy in the whole simulation, while the online CPU cost is reduced by five orders of magnitude.

Overall, a general DCGAN for spatiotemporal fluid flow modelling is presented in this work. It will be highly promising to apply the above DCGAN to various modelling problems, for example, turbulence modelling, ocean modelling, air pollution, and urban flows. Future work will focus on optimization of uncertainties in models and explore a hybrid machine learning approach (auto-encoder and GAN) for improving the accuracy and ef-

ficiency in prediction for complex problems. In this work I focus on the simulation of spatiotemporal distribution during a fixed simulation period for given varied boundary conditions, where the training and validation simulation periods are the same. In future, I will also focus on the real-time forecast– using the DCGAN for the lead-time prediction of spatiotemporal fluid flows. In addition, for realistic applications, a large number of snapshots (historical datasets), over a long time period, are needed for model training. The next chapter is going to present the hybrid VAE-GAN model for large data-driven fluid flow modelling.

CHAPTER 3

Development of an advanced hybrid deep adversarial autoencoder for large data-driven nonlinear fluid flow modelling

Considering the high computation cost required in conventional computation fluid dynamic simulations, machine learning methods have been introduced to flow dynamic simulations in years, aiming on reducing CPU time. In this work, I propose a hybrid deep adversarial autoencoder (VAE-GAN) to integrate generative adversarial network (GAN) and variational autoencoder (VAE) for predicting parameterized nonlinear fluid flows in spatial and temporal dimensions. High-dimensional inputs are compressed into the low-dimensional representations by nonlinear functions in a convolutional encoder. In this way, the predictive fluid flows reconstructed in a convolutional decoder contain the dynamic fluid flow physics of high nonlinearity and chaotic nature. In addition, the low-dimensional representations are applied to the adversarial network for model training and parameter optimization, which enables fast computation process. The capability of the hybrid VAE-GAN is illustrated by varying inputs on a flow past a cylinder test case as

well as a second case of water column collapse. Numerical results show that this hybrid VAE-GAN has successfully captured the spatiotemporal flow features with CPU speed-up of three orders of magnitude. These promising results suggest that the hybrid VAE-GAN can play a critical role in efficiently and accurately predicting complex flows in future research efforts.

This chapter is derived from and expands upon Cheng et al. (2020b).

3.1 Introduction

The hybrid VAE-GAN developed here is a robust and efficient numerical tool for accurate prediction of large data-driven nonlinear fluid flows. The advantages of the hybrid VAE-GAN include:

- The proposed method exploits spatial features by use of convolutional neural networks. It will be advantageous over the traditional reduced order models (ROMs) based on proper orthogonal decomposition (POD) (Fang et al., 2017; Xiao et al., 2019b), Galerkin regression (Loiseau and Brunton, 2018; Loiseau et al., 2018), cluster-based reduced order model (CROM) (Kaiser et al., 2014, 2018) and dynamic mode decomposition (DMD) (Kutz et al., 2016), which are used for model reduction of dynamical systems by linear combinations of the original snapshots. In convolutional neural networks, the high-dimensional datasets are compressed into the low-dimensional representations by nonlinearity functions in a convolutional encoder. In this way, the predictive fluid flows containing both high nonlinearity and chaotic nature can be represented by a convolutional decoder.

- The low-dimensional representations, several orders of magnitude smaller than the dimensional size of the original datasets, are applied into the adversarial network

for representation learning and parameter optimization, thus accelerating the computation process.

- With the trained hybrid VAE-GAN, for given unseen inputs, the spatiotemporal features can be automatically extracted in the encoder. Consequently, accurate predictive nonlinear fluid fields can be further obtained in the decoder in an efficient manner.

This is the first time that the hybrid VAE-GAN is adopted to address large data-driven nonlinear fluid flow problems. It is expected to make a breakthrough in predicting accurate nonlinear fluid flows with the high-speed computation.

The reminder of this chapter is organised as follows. Methodology of VAE is briefly introduced in section 3.3. The hybrid VAE-GAN for large data-driven nonlinear fluid fields is described in detail in section 3.4. section 3.5 demonstrates the performance of the hybrid VAE-GAN using a flow past a cylinder and water column collapse as test cases. Finally in section 3.6, conclusions are presented.

3.2 Governing equations

In the computational fluid dynamic models, nonlinear fluid flows are usually simulated by solving nonlinear partial differential equations, such as Navier-Stokes equations. In general, a parameterized partial differential equation for a spatiotemporal fluid flow problem can be written as:

$$\mathcal{M}(\hbar(x,\mu,t),x,\mu,t) = \mathcal{F}(\hbar(x,\mu,t),x,\mu,t), \tag{3.1}$$

where $\mathcal{M}$ denotes a nonlinear partial differential operator, $\hbar(x,\mu,t)$ is the state variable vector (for example, pressure, density, velocity, etc), x represents the spatial coordinate

system, μ denotes the parameter vector (for example, model input and boundary condition), t is the time and $\mathcal{F}$ is the source term.

3.3 Variational Autoencoder for model reduction

VAE was introduced by Kingma and Welling (2013), which combines Bayesian inference with deep learning. The VAE is a generative model which aims to produce the desired local variable $\hbar$ from the underlying latent variable ζ (as shown in Fig. 3.1). Mathematically, let $p(\zeta)$ be a prior distribution of ζ, and the probability of the local state variable $\hbar$ be modelled by

$$\hbar \sim p_\theta(\hbar|\zeta), \quad p_\theta(\hbar) = \int p_\theta(\hbar|\zeta)p_\theta(\zeta)d\zeta, \tag{3.2}$$

where $p_\theta(\hbar|\zeta)$ is the conditional distribution of the local state variable $\hbar$ given ζ, which is modelled by deep neural networks (called decoder) with parameters θ (for example, weights and bias in the decoder).

Since the true posterior $p_\theta(\zeta|\hbar) = p_\theta(\hbar|\zeta)p_\theta(\zeta)/p_\theta(\hbar)$ is intractable, the integral of the marginal likelihood of $p_\theta(\hbar)$ is also intractable (Kingma and Welling, 2013). To evaluate $p_\theta(\hbar)$, the distribution of the latent state vector $p_\theta(\zeta)$ is usually modelled by $q_\phi(\zeta|\hbar)$ as:

$$\zeta \sim q_\phi(\zeta|\hbar), \tag{3.3}$$

where $q_\phi(\zeta|\hbar)$ represents the conditional distribution of the ζ given local state variable $\hbar$ (as shown in Fig. 3.1), which is deep neural networks (called encoder) with parameters ϕ (for example, weights and bias in the encoder). The encoder enables model reduction for large-scale datasets in dynamic fluid systems (Lee and Carlberg, 2018).

To achieve the sample reconstruction, the reconstruction loss $\mathcal{L}_{rec}$ as the negative expected

log-likelihood of the samples needs to be maximized, as follows:

$$\mathcal{L}_{rec} = E_{q_\phi(\zeta|\hbar)}(\log p_\theta(\hbar|\zeta)). \tag{3.4}$$

The difference (called Kullback–Leibler divergence) between the distribution of $q_\phi(\zeta|\hbar)$ and the intractable true posterior $p_\theta(\zeta|\hbar)$ can be quantified as

$$\mathcal{L}_{KL} = D_{KL}(q_\phi(\zeta|\hbar)||p_\theta(\zeta|\hbar)). \tag{3.5}$$

The total loss $\mathcal{L}_{vae}$ contains the loss from the sample reconstruction (Eq.(3.4)) and the difference between the generated and true posteriors (Eq.(3.5)) ($\mathcal{L}_{vae} = \mathcal{L}_{rec} + \mathcal{L}_{KL}$) ($\mathcal{L}_{KL}$ is replaced by the adversarial loss $\mathcal{L}_{adv}$ in Eq.(3.24) of the hybrid VAE-GAN, see section 3.4.1.2), which can be minimized by gradient descent algorithms (Kingma and Welling, 2013). Since the KL divergence is non-negative, the total loss $\mathcal{L}_{vae}$ can be expressed as:

$$\mathcal{L}_{vae} = -D_{KL}(q_\phi(\zeta|\hbar)||p_\theta(\zeta|\hbar)) + E_{q_\phi(\zeta|\hbar)}(\log p_\theta(\hbar|\zeta)). \tag{3.6}$$

Correspondingly, the parameters of the encoder and decoder in VAE are optimized by maximizing the expectation of the lower bound of the log marginal likelihood as follows:

$$\max_\theta \max_\phi E_{p_{data}(\hbar)}[-D_{KL}(q_\phi(\zeta|\hbar)||p_\theta(\zeta|\hbar)) + E_{q_\phi(\zeta|\hbar)}(\log p_\theta(\hbar|\zeta))], \tag{3.7}$$

where $p_{data}(\hbar)$ is the prior distribution of $\hbar$.

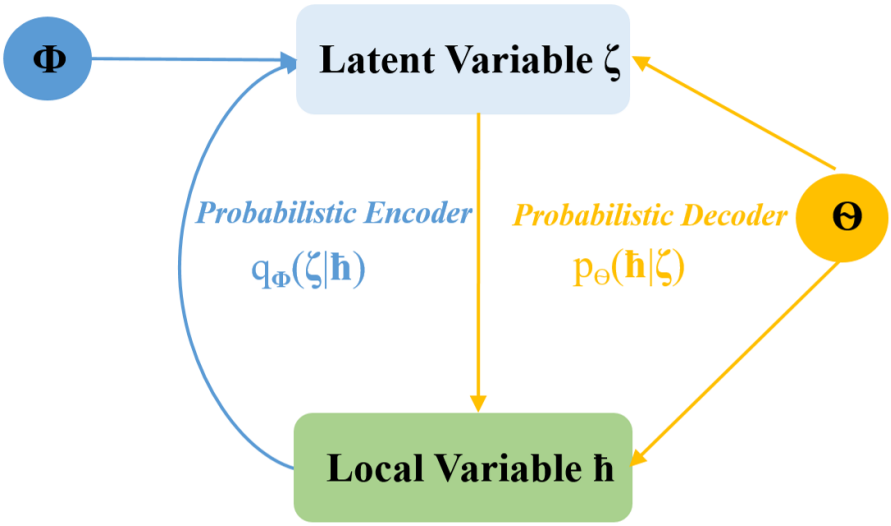


Fig. 3.1. Illustration of a VAE model. Yellow lines denotes the generative process in a probabilistic decoder $p_\theta(\hbar|\zeta)$, blue lines represent the variational approximation $q_\phi(\zeta|\hbar)$ in a probabilistic encoder to the intractable posterior $p_\theta(\hbar|\zeta)$. ϕ and θ are parameters which are learned jointly (see Kingma and Welling (2013)).

3.4 Hybrid deep adversarial autoencoder for nonlinear fluid flow modelling

In this work, for nonlinear fluid flow modeling, a hybrid deep learning fluid model based on deep adversarial autoencoder (VAE-GAN) is proposed by combining a VAE and a GAN.

In spatiotemporal fluid flow simulations, the state variable vector $\hbar$ represents the fluid flow distribution as

$$\hbar = \begin{bmatrix} \hbar_{t_1}^1 & \hbar_{t_1}^2 & \cdots & \hbar_{t_1}^{N_x} \\ \hbar_{t_2}^1 & \hbar_{t_2}^2 & \cdots & \hbar_{t_2}^{N_x} \\ \vdots & \vdots & & \vdots \\ \hbar_{t_{N_t}}^1 & \hbar_{t_{N_t}}^2 & \cdots & \hbar_{t_{N_t}}^{N_x} \end{bmatrix}, \tag{3.8}$$

and the state variable vector $\hbar$ in Eq.(3.8) can be rewritten as:

$$\hbar_{t_{n_t}} = (\hbar^1_{t_{n_t}}, ..., \hbar^{n_x}_{t_{n_t}}, ..., \hbar^{N_x}_{t_{n_t}}),\ \ \hbar^{n_x} = (\hbar^{n_x}_{t_1}, ..., \hbar^{n_x}_{t_{n_t}}, ..., \hbar^{n_x}_{t_{N_t}})^T, \tag{3.9}$$

where $\hbar_{t_{n_t}}$ is the state variable spatial solution at time level t_{n_t} ($t_{n_t} \in [0, t_{N_t}]$, N_t is the number of timesteps), $\hbar^{n_x}$ is the state variable spatial solution at the point n_x ($n_x \in [0, N_x]$, N_x denotes the number of points in grids of the computational domain Ω).

3.4.1 Hybrid deep adversarial autoencoder

When the dimension of $\hbar \in \Re^{N_t \times N_x}$ is large, efficiently predicting Eq.(3.1) for given inputs (e.g. boundary conditions) becomes challenging. Therefore, in the hybrid VAE-GAN (as described in Fig. 3.2(a)), the inputs μ and the real fluid fields $\hbar_d$ (the targeted outputs) are fed into the encoder, which transforms the high-dimensional datasets into low-dimensional representations ζ and ζ_d respectively. The discriminator can discriminate the low-dimensional representations ζ and ζ_d in an efficient way. The latent states ζ_d and ζ are then transformed into the reconstructed fluid fields $\hat{\hbar_d}$ (the reconstructed outputs) and predicted fluid fields $\hbar$ (the predicted outputs) respectively in the decoder. In the backward propagation process, the parameters in modules: encoder, decoder and discriminator are updated by minimizing the reconstruction loss and maximizing the adversarial loss.

As shown in Fig. 3.2(a), the encoder is acting as a generator in the hybrid VAE-GAN. In comparison to Eq.(3.3) in VAE, the generator here defines an aggregated posterior distribution of $q(\zeta)$ on the latent state as follows:

$$q(\zeta) = \int q_\phi(\zeta|\hbar) q_{data}(\hbar) d\hbar, \tag{3.10}$$

where $q_{data}(\hbar)$ is the data distribution for the datasets $\hbar$. The autoencoder in the hybrid VAE-GAN is regularized by matching the aggregated posterior $q(\zeta)$ to the prior $p(\zeta)$ in

Eq.(3.2).

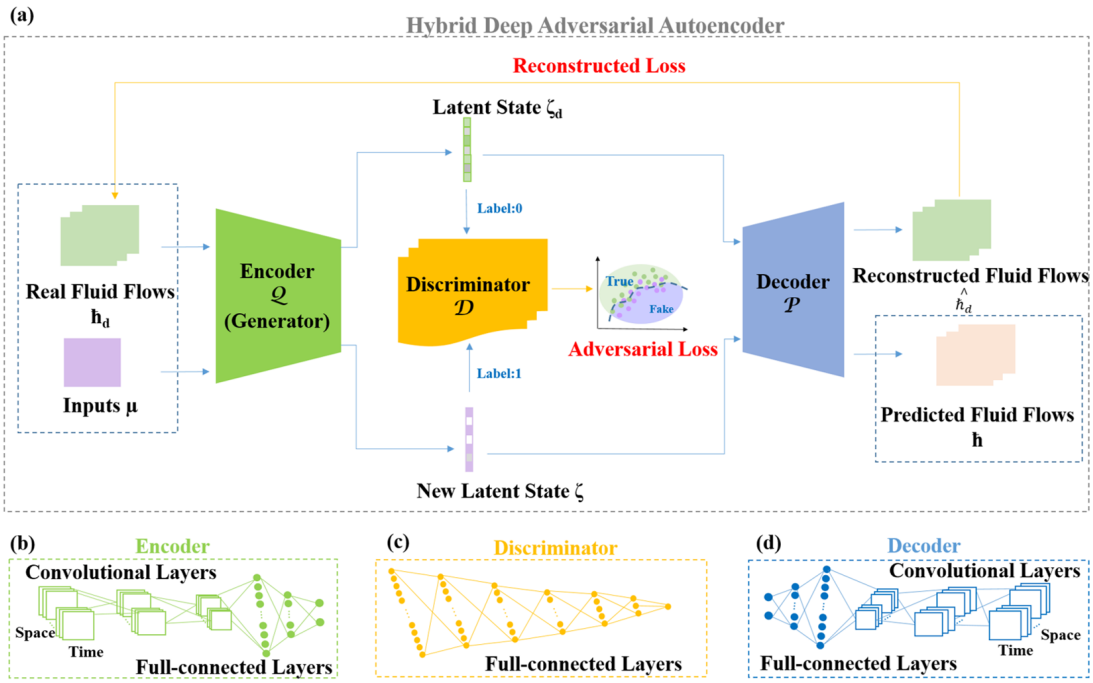


Fig. 3.2. (a) Illustration of a hybrid deep adversarial autoencoder (VAE-GAN), which consists of an encoder (acted as a generator) (b), a discriminator (c), and a decoder (d). The encoder and decoder compose of convolutional layers and full-connected layers. The discriminator is a stack of full-connected layers.

Given the parameter vector $\mu = [\mu^1, \mu^2, ..., \mu^{N_x}]$ as inputs and the state variable vector $\hbar_d = [\hbar_d^1, \hbar_d^2, ..., \hbar_d^{N_x}]$ as the targeted outputs, the model architecture, loss function and offline training and online predictive processes of the hybrid VAE-GAN are detailedly introduced as follows.

3.4.1.1 Model architecture

As shown in Fig. 3.2(b), (c) and (d), the modules: encoder, decoder and discriminator, consist of deep neural networks (containing convolutional layers and full-connected layers), which are used for extracting spatiotemporal features. Let h^{l-1} and h^l be the input

and output for the l^{th} layer, respectively. The output of a layer is computed as

$$h_j^l = \sigma^l(\sum_{i=1}^{N^{l-1}} h_i^{l-1} * w_{ij}^l + b_j^l), \tag{3.11}$$

where h_j^l denotes the j^{th} output-feature-map of layer l, σ represents the activation function, N^{l-1} is the number of input-feature-maps of layer $(l-1)$, w_{ij}^l is the weight linking the i^{th} input map to j^{th} output map of layer l, and b_j^l is the bias of j^{th} output map of layer l.

For the activation function σ, the rectifier non-linearity (ReLU) activation function ($\mathcal{I}(\alpha) = \max(0, \alpha)$) is normally recommended for the hidden layers, and the hyperbolic tangent function ($\mathcal{I}(\alpha) = \tanh(\alpha)$) and the sigmoid function ($\mathcal{I}(\alpha) = 1/(1 + \exp^{-\alpha})$) are chosen for the output layer (Mosser et al., 2017; Lee and You, 2018).

For simplifying the Eq.(3.11), it can be rewritten as:

$$h_l = \mathcal{H}(h_{l-1}, \sigma_l, \delta_l), \tag{3.12}$$

where $\delta_l \equiv (w_l, b_l)$ represents the parameters in the l^{th} layer.

Design of the Encoder: Mapping the targeted outputs $\hbar_d$ and inputs μ to the latent state ζ_d and ζ respectively as:

$$\mathcal{Q} : \hbar_d \in \Re^{N_{\hbar_d,el}} \mapsto \zeta_d \in \Re^{N_{\zeta_d,el}}, \mu \in \Re^{N_{\mu,el}} \mapsto \zeta \in \Re^{N_{\zeta,el}}, \tag{3.13}$$

where $\Re^{N_{\hbar_d,el}}$ and $\Re^{N_{\mu,el}}$ are the dimensions of the targeted outputs and inputs respectively, $\Re^{N_{\zeta_d,el}}$ and $\Re^{N_{\zeta,el}}$ are the dimensions of the latent states respectively.

The encoder $\mathcal{Q}$ forms the deep convolutional neural networks with N_e layers, producing the latent state ζ_d as:

$$\zeta_d = \mathcal{Q}(\hbar_d, \theta) = \mathcal{H}_{N_e}^{(E)}(\sigma_{N_e}^{(E)}; \delta_{N_e}^{(E)}) \circ \mathcal{H}_{N_e-1}^{(E)}(\sigma_{N_e-1}^{(E)}; \delta_{N_e-1}^{(E)}) \circ \cdots \circ \mathcal{H}_1^{(E)}(\hbar_d; \delta_1^{(E)}), \tag{3.14}$$

where $\mathcal{H}_l^{(E)}(\sigma_l^{(E)};\delta_l^{(E)})$ represents the layer l of the neural network, $l=1,...,N_e$, $\sigma_l^{(E)}$ represents the activation function at the layer l, $\delta_l^{(E)}$ denotes the weights and bias at the layer l, and $\theta \equiv (\delta_1^{(E)},...,\delta_{N_e}^{(E)})$.

Similarly, the latent state ζ is generated by the encoder as:

$$\zeta = \mathcal{Q}(\mu,\theta) = \mathcal{H}_{N_e}^{(E)}(\sigma_{N_e}^{(E)};\delta_{N_e}^{(E)}) \circ \ \mathcal{H}_{N_e-1}^{(E)}(\sigma_{N_e-1}^{(E)};\delta_{N_e-1}^{(E)}) \circ \cdots \circ \mathcal{H}_1^{(E)}(\mu;\delta_1^{(E)}). \quad (3.15)$$

Design of the Discriminator: The discriminator aims to judge the latent states ζ_d and ζ as:

$$\mathcal{D} : \zeta \in \Re^{N_{\zeta,el}} \mapsto p_\zeta, \zeta_d \in \Re^{N_{\zeta,el}} \mapsto q_{\zeta_d}. \quad (3.16)$$

The discriminator forms deep neural networks with M_d layers can be expressed as:

$$\mathcal{D}(\zeta,\zeta_d,\psi) = \mathcal{H}_{M_d}^{(M)}(\sigma_{M_d}^{(M)};\delta_{M_d}^{(M)}) \circ \ \mathcal{H}_{M_d-1}^{(M)}(\sigma_{M_d-1}^{(M)};\delta_{M_d-1}^{(M)}) \circ \cdots \circ \mathcal{H}_1^{(M)}(\zeta;\zeta_d;\delta_1^{(M)}), \quad (3.17)$$

where $\mathcal{H}_l^{(M)}(\sigma_l^{(M)};\delta_l^{(M)})$ represents the layer l of the neural network, $l=1,...,M_d$, $\sigma_l^{(M)}$ represents the activation function at the layer l, $\delta_l^{(M)}$ denotes the weights and bias at the layer l, and $\psi \equiv (\delta_1^{(M)},...,\delta_{M_d}^{(M)})$.

Design of the Decoder: Reconstructing the new latent state ζ_d to the targeted output $\hat{\hbar}_d$ and predicting the outputs $\hbar$ by latent state ζ as:

$$\mathcal{P} : \zeta_d \in \Re^{N_{\zeta_d,el}} \mapsto \hat{\hbar}_d \in \Re^{N_{\hat{\hbar}_d,el}}, \zeta \in \Re^{N_{\zeta,el}} \mapsto \hbar \in \Re^{N_{\hbar,el}}, \quad (3.18)$$

where $\Re^{N_{\hbar,el}}$ is the dimension of the outputs.

The decoder $\mathcal{P}$ with N_d layers takes the form:

$$\hat{\hbar_d} = \mathcal{P}(\zeta_d, \phi) = \mathcal{H}_{N_d}^{(D)}(\sigma_{N_d}^{(D)}; \delta_{N_d}^{(D)}) \circ \mathcal{H}_{N_d-1}^{(D)}(\sigma_{N_d-1}^{(D)}; \delta_{N_d-1}^{(D)}) \circ \cdots \circ \mathcal{H}_1^{(D)}(\zeta_d; \delta_1^{(D)}), \tag{3.19}$$

where $\mathcal{H}_l^{(D)}(\sigma_l^{(D)}; \delta_l^{(D)})$ represents the layer l of the neural network, $l = 1, ..., N_d$, $\sigma_l^{(D)}$ represents the activation function at the layer l, $\delta_l^{(D)}$ denotes the weights and bias at the layer l, and $\phi \equiv (\delta_1^{(D)}, ..., \delta_{N_d}^{(D)})$.

Similarly, the outputs $\hbar$ can be expressed as:

$$\hbar = \mathcal{P}(\zeta, \phi) = \mathcal{H}_{N_d}^{(D)}(\sigma_{N_d}^{(D)}; \delta_{N_d}^{(D)}) \circ \mathcal{H}_{N_d-1}^{(D)}(\sigma_{N_d-1}^{(D)}; \delta_{N_d-1}^{(D)}) \circ \cdots \circ \mathcal{H}_1^{(D)}(\zeta; \delta_1^{(D)}). \tag{3.20}$$

3.4.1.2 Loss function

As introduced by Makhzani et al. (2015), the encoder, decoder and discriminator of the hybrid VAE-GAN are trained jointly in the reconstruction phase and the regularization phase. In the reconstruction phase, the hybrid VAE-GAN updates the encoder $\mathcal{Q}$ and the decoder $\mathcal{P}$ to minimize the errors between the reconstructed targeted outputs $\hat{\hbar_d}$ and targeted outputs $\hbar_d$. In the regularization phase, the adversarial network updates the discriminator $\mathcal{D}$ to distinguish the true latent state ζ (as Eq.(3.15)) from the latent state ζ_d (as Eq.(3.14)) generated by encoder. The loss functions in two phases are as follows.

Reconstruction loss $\Rightarrow$ With jointing the encoder and the decoder, the autoencoder is trained using the reconstruction loss as:

$$\mathcal{L}_{rec}(\theta, \phi) = \left\| \hbar_d - \mathcal{P}(\mathcal{Q}(\hbar_d, \theta), \varphi) \right\|. \tag{3.21}$$

Adversarial loss ⇒ The discriminator is trained to distinguish between the latent states ζ and ζ_d based the loss function:

$$\mathcal{L}_d(\theta,\phi,\psi) = -E_{\zeta\sim p_\zeta(\zeta)}(\log \mathcal{D}(\mathcal{Q}(\mu))) - E_{\zeta_d\sim p_{\zeta_d}(\zeta_d)}(\log(1-\mathcal{D}(\mathcal{Q}(\hbar_d)))), \tag{3.22}$$

while the encoder (generator) tries to fool the discriminator using the following loss function:

$$\mathcal{L}_g(\theta,\phi,\psi) = -E_{\zeta_d\sim p_{\zeta_d}(\zeta_d)}(\log(1-\mathcal{D}(\mathcal{Q}(\hbar_d)))). \tag{3.23}$$

The adversarial loss function is:

$$\mathcal{L}_{adv}(\theta,\phi,\psi) = -\mathcal{L}_d(\theta,\phi,\psi) = E_{\zeta\sim p_\zeta(\zeta)}(\log \mathcal{D}(\mathcal{Q}(\mu)))+E_{\zeta_d\sim p_{\zeta_d}(\zeta_d)}(\log(1-\mathcal{D}(\mathcal{Q}(\hbar_d)))). \tag{3.24}$$

Hybrid loss ⇒ The hybrid VAE-GAN attempts to minimize the reconstruction loss and maximize the adversarial loss. The hybrid loss can be obtained as:

$$\min_{\theta,\phi}\max_{\psi} E_{\zeta\sim p_\zeta(\zeta)}(\log \mathcal{D}(\mathcal{Q}(\mu)))+E_{\zeta_d\sim p_{\zeta_d}(\zeta_d)}(\log(1-\mathcal{D}(\mathcal{Q}(\hbar_d))))+\left\|\hbar_d - \mathcal{P}(\mathcal{Q}(\hbar_d,\theta),\varphi)\right\|. \tag{3.25}$$

The above Eqs.(3.21) to (3.25) show that the hybrid VAE-GAN is intuitively similar to the VAE, except for the KL divergence penalty of the VAE being replaced by the adversarial loss. The adversarial training procedure makes the hybrid VAE-GAN more flexible in choice of the latent variable ζ, which has been discussed in the work of Makhzani et al. (2015). The hybrid model is trained using the stochastic gradient descent by performing alternative updates of each component as Algorithm 3.1.

Algorithm 3.1 Adversarial training of the hybrid deep adversarial autoencoder.

- Sample $\{\mu^1, \mu^2, \ldots, \mu^{N_x}\}$ from the inputs.
- Sample $\{\hbar_d^1, \hbar_d^2, \ldots, \hbar_d^{N_x}\}$ from the real datasets.

for P = 1 to N_P **do**

for K = 1 to N_K **do**

- Compute the latent state ζ_d in the encoder: $\zeta_d = \mathcal{Q}(\hbar_d, \theta)$.
- Compute the reconstructed outputs $\hbar_d$ in the decoder: $\hat{\hbar_d} = \mathcal{P}(\mathcal{Q}(\hbar_d, \theta), \phi)$.
- Compute the new latent state ζ in the encoder: $\zeta = \mathcal{Q}(\mu, \theta)$.
- Compute the outputs $\hbar$ in the decoder: $\hbar = \mathcal{P}(\mathcal{Q}(\mu, \theta), \phi)$.
- Compute the probability in the discriminator: $\mathcal{D}(\mathcal{Q}(\mu, \theta), \mathcal{Q}(\hbar_d, \theta), \psi)$
- Update ψ in the discriminator by performing its stochastic gradient ascent:

$$\psi \leftarrow \psi + \eta \cdot \nabla_\psi(\mathcal{L}_{rec}(\theta, \phi) + \mathcal{L}_{adv}(\theta, \phi, \psi))$$

end for

- Update θ in the encoder by performing its stochastic gradient descent:

$$\theta \leftarrow \theta - \eta \cdot \nabla_\theta(\mathcal{L}_{rec}(\theta, \phi) + \mathcal{L}_{adv}(\theta, \phi, \psi))$$

- Update ϕ in the decoder by performing its stochastic gradient descent:

$$\phi \leftarrow \phi - \eta \cdot \nabla_\phi(\mathcal{L}_{rec}(\theta, \phi) + \mathcal{L}_{adv}(\theta, \phi, \psi))$$

end for

3.4.1.3 Offline training and online predictive processes

The hybrid VAE-GAN applied for nonlinear fluid flow modelling consists of the following three processes (as shown in Fig. 3.3):

(1) Data preparation process: To develop an input-output relationship, the initial and boundary conditions $\{\mu^1, \mu^2, ..., \mu^{N_x}\}$ are chosen as the inputs and the spatiotemporal distributions of flow variables $\{\hbar_d^1, \hbar_d^2, ..., \hbar_d^{N_x}\}$ are acted as the outputs, where N_x represents the number of nodes in the domain area Ω. It is worth noting that the outputs are collected in the whole timesteps.

(2) Offline training and validation process: Using the input-output data pairs $(\mu, \hbar_d)$ collected in (1), the hybrid VAE-GAN is trained with the inputs ϑ_{tr} and corresponding outputs δ_{tr} (where ϑ_{tr} and δ_{tr} denote the training inputs and the outputs, respectively). Once

the training and validation processes are completed, the relationship between the input-output pair is determined.

(3) Online predictive process: Given a new series of predictive inputs $\mu \in (\vartheta \backslash \vartheta_{tr})$ (where ϑ is the whole inputs), the predictive outputs $\hbar$ are directly predicted by the hybrid VAE-GAN.

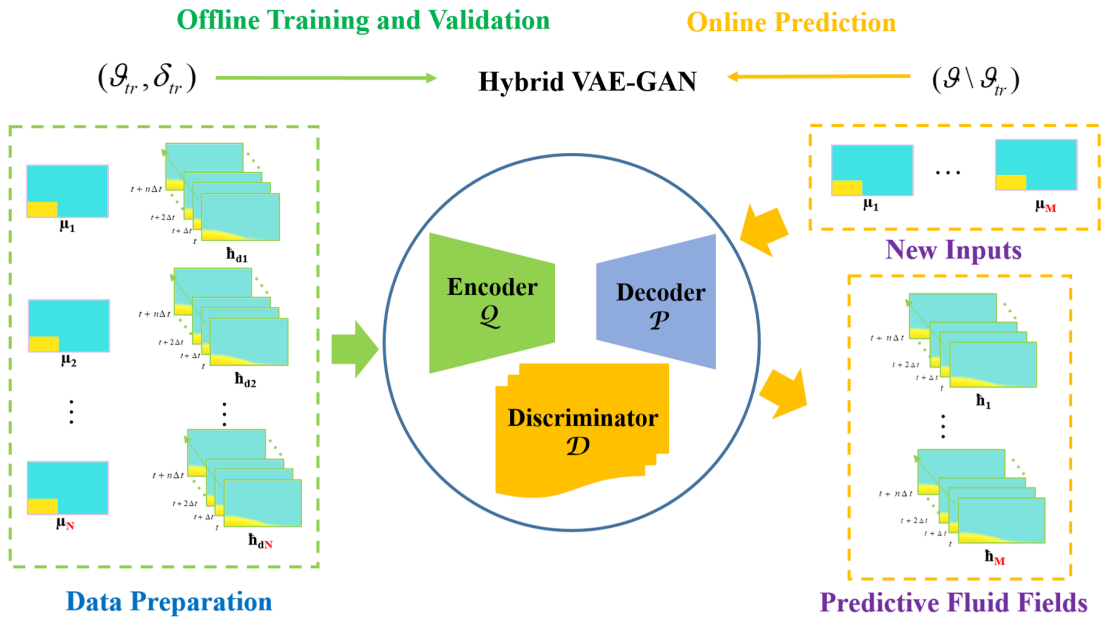


Fig. 3.3. The offline training and online predictive processes.

3.5 Numerical examples

In this section, two examples, flow past a cylinder and water column collapse, are used to illustrate the capabilities of the hybrid VAE-GAN in resolving nonlinear fluid flow problem governed by the Navier-Stokes equations. The input-output datasets in both examples were obtained by running the unstructured mesh finite element fluid model (*Fluidity*) (Pain et al., 2001) (referred as the original high fidelity model). In Table 3.1, the module architecture of the encoder, decoder and discriminator used for two numerical examples are summarized. The reason for choosing the different cases for VAE-GAN training and validation is that, the online CPU cost used for running the DCGAN is less than 0.5s in

test cases in Chapter 2, which is very fast enough. My aim in this chapter is to develop the hybrid VAE-GAN model for large data-driven and accelerate the computation, so I choose the water column collapse case which has larger datasets (nodes $\times$ timesteps $\times$ cases: 19097$\times$75$\times$45=64,452,375) than the tsumani case (7314$\times$501$\times$13=47,636,082) in Chapter 2.

The dimensional size of the latent state is set to 10 using the trial and error method. The Adam optimizer is employed to optimize the values of parameters (θ, ϕ, ψ), with a learning rate $\eta = 0.0002$. With the training and validated input-output pairs ϑ_{tr} and ϕ_{tr} in two examples, the hybrid VAE-GAN are trained using Algorithm 3.1.

Table 3.1: Architectures of $\mathcal{Q}$, $\mathcal{P}$, and $\mathcal{D}$

Module	Case 1: flow past a cylinder		Case 2: water column collapse	
Layer in Encoder $\mathcal{Q}$	Output Shape	Parameter number	Output Shape	Parameter number
InputLayer	(50,1,3213)	0	(75,1,19097)	0
$\mathcal{H}^{(E)}$ (leaky ReLU)	(50,1,128)	3,701,504	(75,1,128)	21,999,872
$\mathcal{H}^{(E)}$ (leaky ReLU)	(50,1,16)	18,448	(75,1,16)	18,448
$\mathcal{H}^{(E)}$ (leaky ReLU)	(800)	8,010	(1200)	12,010
OutputLayer	(10)	0	(10)	0
Layer in Decoder $\mathcal{P}$	Output Shape	Parameter number	Output Shape	Parameter number
InputLayer	(10)	0	(10)	0
$\mathcal{H}^{(D)}$ (leaky ReLU)	(50)	550	(75)	825
$\mathcal{H}^{(D)}$ (leaky ReLU)	(800)	40,800	(1200)	91,200
$\mathcal{H}^{(D)}$ (leaky ReLU)	(50,1,128)	18,560	(75,1,128)	18,560
OutputLayer(Tanh)	(50,1,3213)	3,704,589	(75,1,19097)	22,018,841
Layer in Discriminator $\mathcal{D}$	Output Shape	Parameter number	Output Shape	Parameter number
InputLayer	(10)	0	(10)	0
$\mathcal{H}^{(M)}$(leaky ReLU)	(512)	5,632	(512)	5,632
$\mathcal{H}^{(M)}$(leaky ReLU)	(256)	131,328	(256)	131,328
$\mathcal{H}^{(M)}$(leaky ReLU)	(64)	16,448	(64)	16,448
$\mathcal{H}^{(M)}$(leaky ReLU)	(32)	2,080	(32)	2,080
OutputLayer(Sigmoid)	(1)	33	(1)	33

3.5.1 Case 1: flow past a cylinder

The first case for numerical illustration of proposed methods is a two dimensional flow past a cylinder. In Fig. 3.4 shows a rectangle with the length = $2.2m$ and width = $0.41m$, and a cylinder with a radius of $0.05m$. The fluid flow simulations are driven by an inlet velocity from the left boundary of the domain area. The viscosity is 0.5×10^{-4}, and zero normal flow and no slip boundary conditions are applied to the cylinder and both lateral sides. The computational domain consists of 3213 nodes. The whole simulation period was $10s$ with a timestep of $0.05s$. The targeted outputs $\hbar_d$, 50 velocity solution snapshots, were obtained by running the original high fidelity model at the regularly spaced time intervals $0.2s$.

3.5.1.1 Sensitivity analysis

In this example, the inlet velocity is selected as the input parameter (as shown in Fig. 3.4). To estimate the sensitivity of results from the hybrid VAE-GAN with respect to the number of the input-output datasets, three scenarios have been set up, where 200, 100 and 40 input-output datasets are used for scenario_200, scenario_100 and scenario_40 respectively (shown in Fig. 3.4). In scenario_200 and scenario_100, the input parameter (inlet velocity) is ranged from 0.4 m/s to 0.6 m/s with the Gaussian distribution $\mathcal{N}(\bar{\mu}, \sigma^2)$, where the mean $\bar{\mu}$ is $0.5\,m/s$ and the variance of the distribution σ^2 is $0.001\,m/s$. In scenario_40, the input data is obtained from the linear distribution between $0.4\,m/s$ to $0.6\,m/s$. As the Reynolds number varies from 800 to 1200 with the velocity from 0.4 to 0.6 m/s, the flow changes from laminar flow into turbulent flow, which has fitted the test aim. However, the inlet velocity could also be set in a wider range if interested.

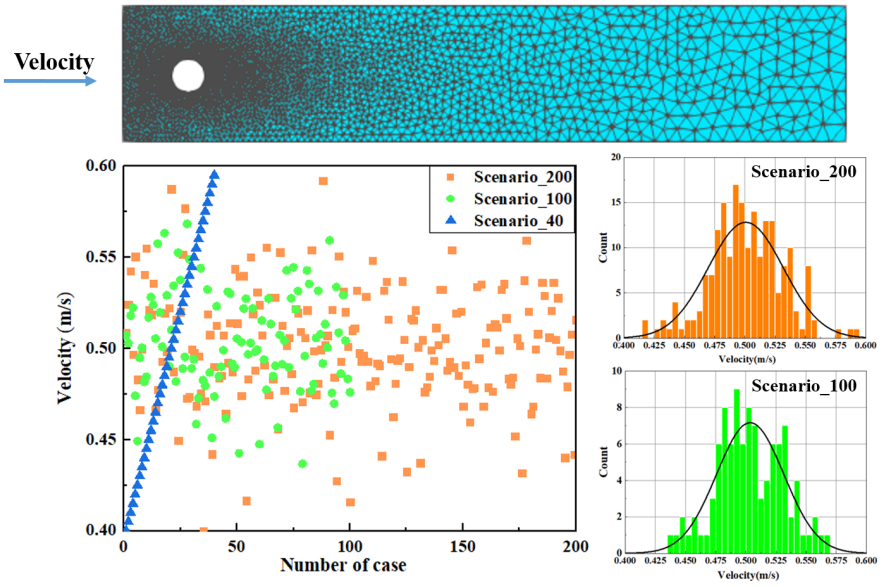


Fig. 3.4. The sketch of flow past a cylinder (mesh size ranges from 0.005 to 0.05) and the settings of boundary velocities.

N_a pairs of input and output datasets for training and validation were obtained by running the high fidelity model. The inputs and targeted outputs ϑ, ϕ can be re-written in a discretized form in space and time:

$$\vartheta = [\mu_a^1, \mu_a^2, ..., \mu_a^{N_{b,x}}], \tag{3.26}$$

$$\phi = \begin{bmatrix} \hbar_{a,t_1}^1 & \hbar_{a,t_1}^2 & \cdots & \hbar_{a,t_1}^{N_x} \\ \hbar_{a,t_2}^1 & \hbar_{a,t_2}^2 & \cdots & \hbar_{a,t_2}^{N_x} \\ \vdots & \vdots & & \vdots \\ \hbar_{a,t_{N_t}}^1 & \hbar_{a,t_{N_t}}^2 & \cdots & \hbar_{a,t_{N_t}}^{N_x} \end{bmatrix}, \tag{3.27}$$

where $\mu_a \in \mathbb{R}^{N_x}$ represents the input parameter (here, the inlet velocity) while the spatiotemporal distributions of flow variables $\hbar_a \in \mathbb{R}^{N_x}$ (here, the flow velocity) are acted as the outputs, the number of nodes over the domain $N_x = 3213$, the number of nodes at the inlet boundary $N_{b,x} = 39$, the number of time steps $N_t = 50$, and $a \in (1, \ldots, N_a)$, where N_a represents the total number of the input-output pairs used for training, valida-

tion and model prediction, $N_a = 200$, 100 and 40 pairs of input-output datasets are setup for scenario_200, scenario_100 and scenario_40 respectively.

Given a new inlet velocity $\mu = 0.52 \in (\vartheta \backslash \vartheta_{tr})$, I will evaluate the accuracy of the hybrid VAE-GAN generated with different number of training datasets. The RMSE of the predictive results from the hybrid VAE-GAN and the original high fidelity model is shown in Fig. 3.5. It can be noticed that the RMSE values in three scenarios are below $0.25m/s$ during the whole simulation period. The maximum RMSE values are $0.25m/s$, $0.21m/s$ and $0.14m/s$ for scenario_40, scenario_100 and scenario_200 respectively. The corresponding relative error values are shown in Fig. 3.6. It is noted that most of the relative errors are below 0.15 during the simulation period except for some moments $t = 2 - 3.5s$ and $t = 6 - 8s$. It is worth noting that the magnitude and trend of relative values in scenario_40 are comparable to those in scenario_100 and scenario_200 except for $t = 2 - 3.5s$, although the RMSE values in scenario_40 are higher than other two scenarios. This means that the hybrid VAE-GAN generated with a small number of training datasets could provide a reasonably accurate prediction of nonlinear fluid flows by using the hybrid VAE-GAN.

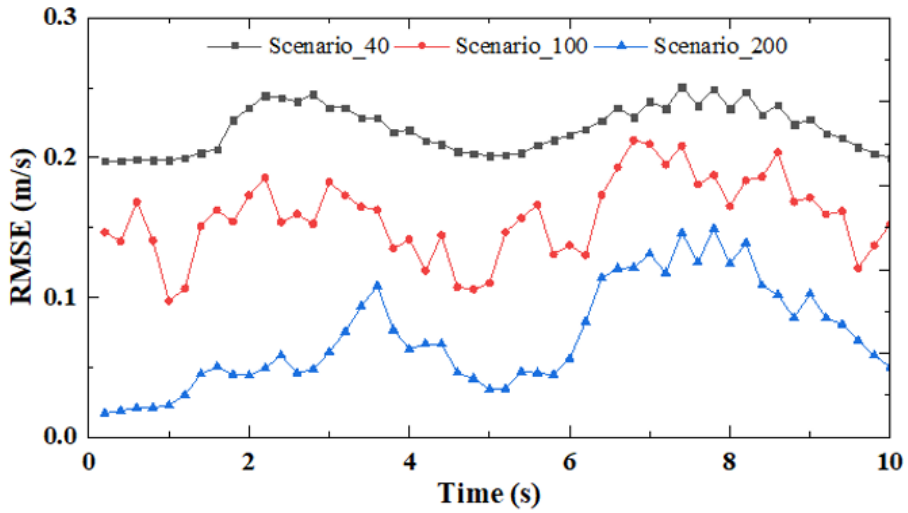


Fig. 3.5. The RMSE of velocity between the generated fields and original high fidelity fields during whole simulational period.

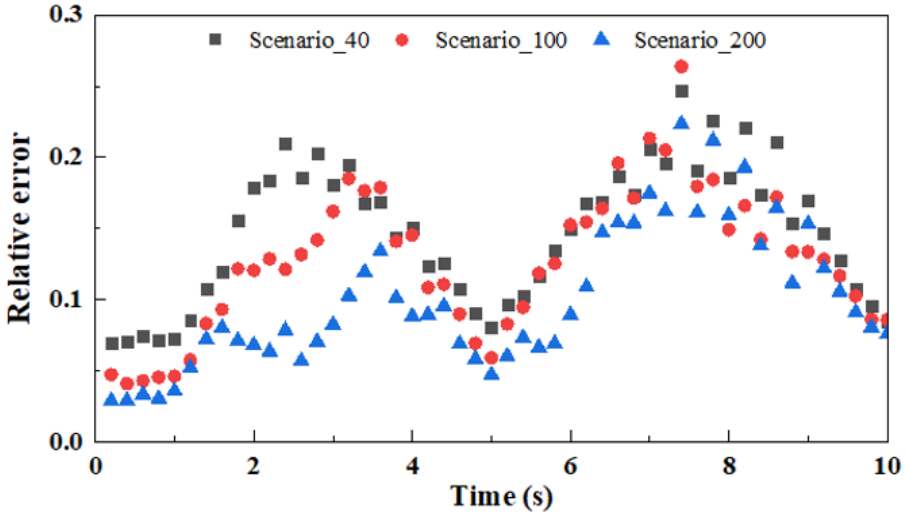


Fig. 3.6. The relative error of velocity solutions between the generated fields and original high fidelity fields during whole simulational period.

3.5.1.2 Model prediction

Fig. 3.7 displays the velocity solutions (given the inlet velocity $\mu = 0.52 \in (\vartheta \backslash \vartheta_{tr})$) obtained from the hybrid VAE-GAN in scenario_40 and high fidelity model at time levels $t = 1.8, 4, 5.6, 9.2s$. It is shown that there are small visual differences between the hybrid VAE-GAN and high fidelity model except for upper and lower edges of the domain area. The hybrid VAE-GAN can capture the details of the flow very well in comparison to the high fidelity model. In addition, the magnitude of the spatial velocity is also in good agreement to the high fidelity model solutions. However, the VAE-GAN fails to capture the velocity variation on the top and bottom boundaries, which may be explained by that the values of velocities on boundaries are very small so that the variation led by the inlet velocity is not captured by the VAE-GAN.

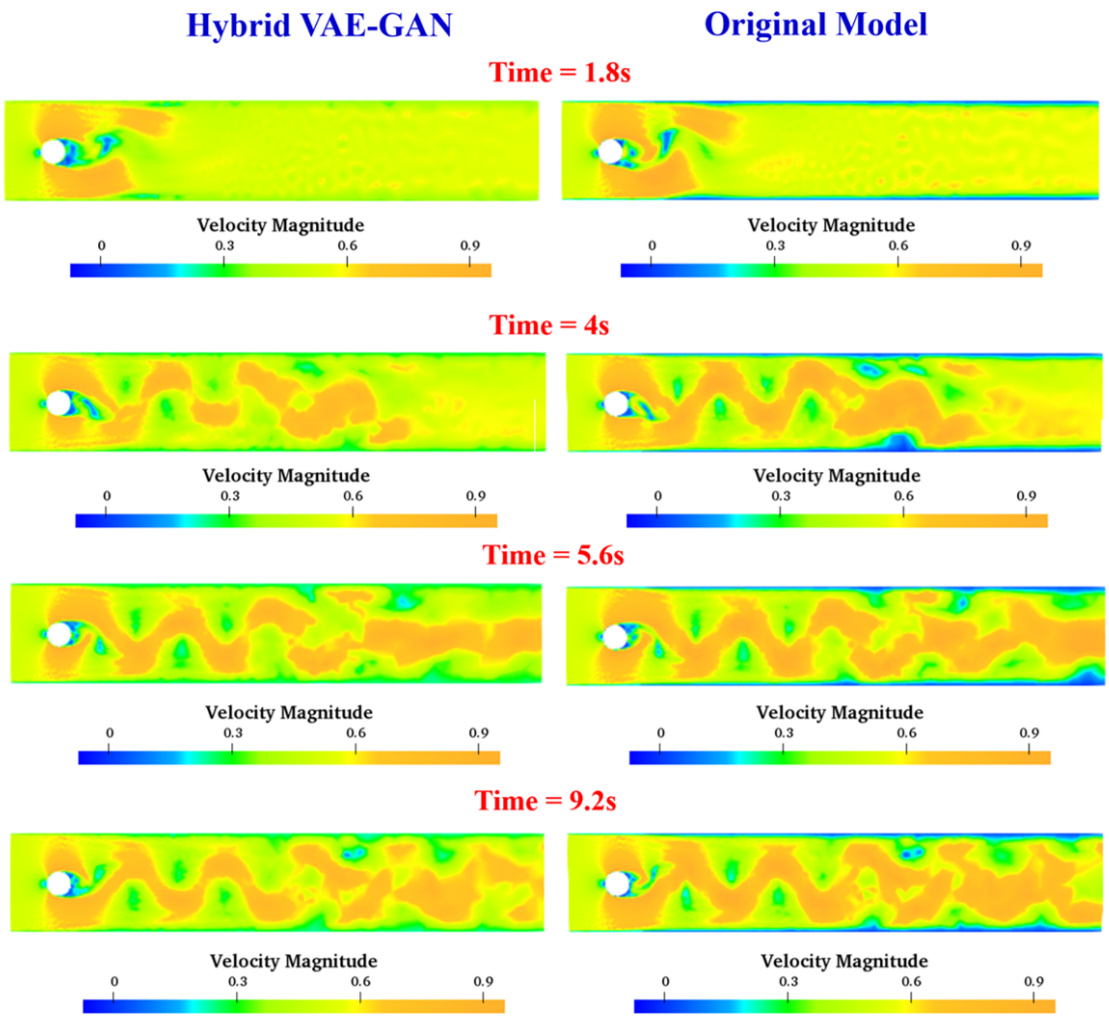


Fig. 3.7. Comparison of the spatial distribution of velocity fields obtained from the hybrid VAE-GAN (left) and the original high fidelity model (right) at time levels $t = 1.8, 4, 5.6, 9.2s$.

3.5.2 Case 2: water column collapse

The hybrid VAE-GAN has been further validated with a case of water column collapse. It is a benchmark test case for multi-material models, also known as a dam break problem. The dam-break flow problem has been of great importance in hydraulic engineering and hydropower generation. Dam-break flows are complex and generally discontinuous with abrupt variations of flow velocity and water depth (Seyedashraf et al., 2017). The flooding induced by the dam-break flows causes great loss of human life and property as well as

damaging the ecosystem in the downstream area.

In this example, the dam-break experiment is conducted in a tank with the length $3.22m$, the height $2m$, and the depth $1m$ (Zhou et al., 1999). The simulated reservoir of water is held behind a barrier at one end of the tank. When no variations are introduced in the third dimension, the experiment is reproduced in the horizontal and vertical dimensions within the domain area Ω. Fig. 3.8 shows the mesh in the domain area Ω. The densities of water and air are $1,000kgm^{-2}$ and $1kgm^{-2}$ respectively. The scalar fields a_k representing the volume fraction are introduced to distinguish the two materials. As shown in Fig. 3.8, the interface between the water (the yellow area, $a_k = 1$) and air (the blue area, $a_k = 0$) is delineated by contours at a_k of 0.025, 0.5 and 0.975. In this case, the simulation period is $[0, 1.85]s$, with a timestep = $0.025s$. Thus, the targeted outputs (here, pressure solutions in time and space) $\hbar_d$ were obtained by running the original high fidelity model, with a unstructured mesh of 19097 nodes.

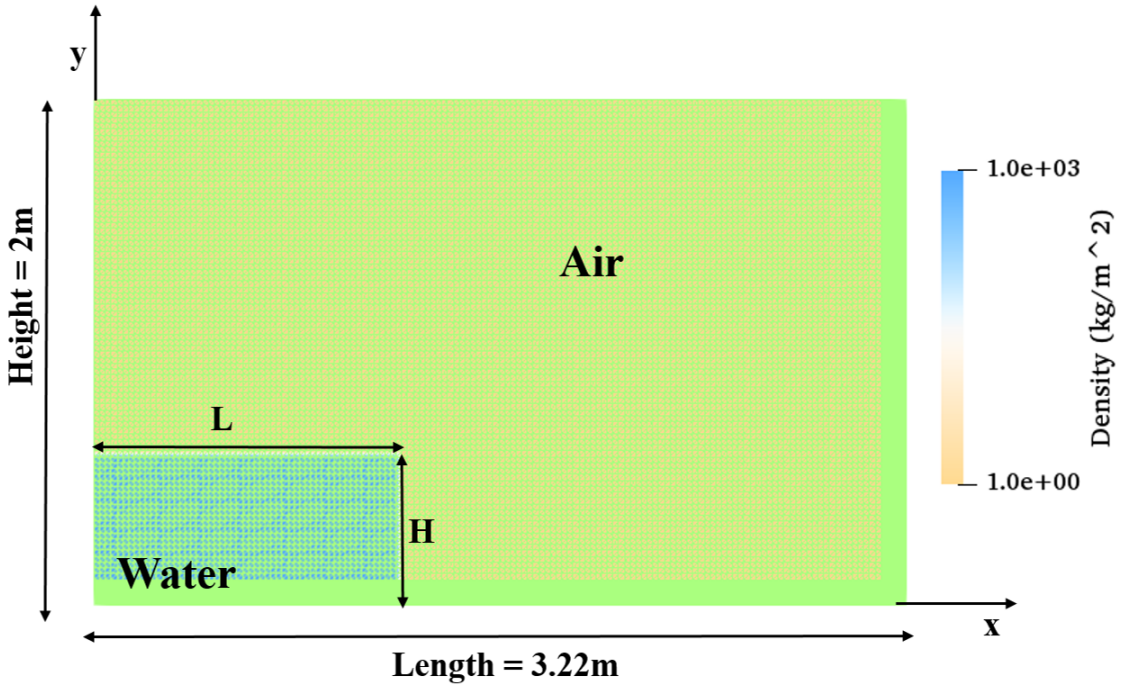


Fig. 3.8. The sketch of water collapse experiment, where bottom to up and right to left have denser meshes. The mesh size ranges from 0.003 to $0.2m$.

3.5.2.1 Model application

In this case, for training the hybrid VAE-GAN, the length L and height H of the tank are selected as the input parameters (as shown in Fig. 3.9). The length of the tank L is ranged from 1.2 m to 1.6 m, and the height H of the tank varies from 0.6 m to 1.0 m. A series of sizes (see Table 3.2) as inputs μ and the corresponding solution snapshots $\hbar_d$ (a total of 3375 snapshots of 45 pairs of input-output) were obtained by running the high fidelity model. The training and validated input-output pairs ϑ_{tr} and ϕ_{tr} (where $\mu_a \in \mathbb{R}^{N_x}, \hbar_{d,a} \in \mathbb{R}^{N_x \times N_t}$, $a \in [0, 39], N_{b,x} = N_x = 75, N_t = 19097$ in Eqs.(3.26) and (3.27) are selected for model training and parameter optimization, while the remained inputs $(\vartheta \backslash \vartheta_{tr})$ (where $\mu_a \in \mathbb{R}^{N_x}, a \in [0, 6], N_x = 19097$) are used for model prediction.

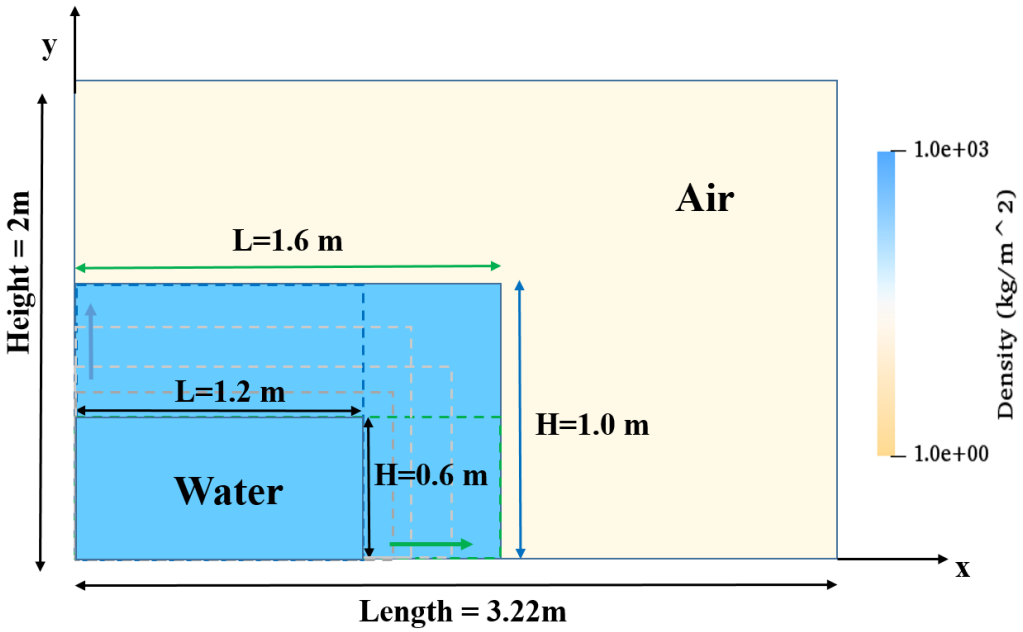


Fig. 3.9. The settings of length and height in water collapse experiment.

Table 3.2: The variation of length and height of the tank in cases

Cases	L(m)	H(m)	Cases	L(m)	H(m)	Cases	L(m)	H(m)
A1	1.2	0.6	A16	1.3	0.9	A31*	1.5	0.75
A2	1.2	0.65	A17	1.3	0.95	A32*	1.5	0.8
A3	1.2	0.7	A18	1.3	1.0	A33*	1.5	0.85
A4	1.2	0.75	A19	1.4	0.6	A34	1.5	0.9
A5	1.2	0.8	A20	1.4	0.65	A35	1.5	0.95
A6	1.2	0.85	A21	1.4	0.7	A36	1.5	1.0
A7	1.2	0.9	A22*	1.4	0.75	A37	1.6	0.6
A8	1.2	0.95	A23*	1.4	0.8	A38	1.6	0.65
A9	1.2	1.0	A24*	1.4	0.85	A39	1.6	0.7
A10	1.3	0.6	A25	1.4	0.9	A40	1.6	0.75
A11	1.3	0.65	A26	1.4	0.95	A41	1.6	0.8
A12	1.3	0.7	A27	1.4	1.0	A42	1.6	0.85
A13	1.3	0.75	A28	1.5	0.6	A43	1.6	0.9
A14	1.3	0.8	A29	1.5	0.65	A44	1.6	0.95
A15	1.3	0.85	A30	1.5	0.7	A45	1.6	1.0

Note: A* represents the validated datasets.

3.5.2.2 Model prediction

Prediction of spatial nonlinear fluid flows: To evaluate the predictive ability of the hybrid VAE-GAN, given the new input $\mu \in (\vartheta \backslash \vartheta_{tr})$, comparison of the predictive results from the hybrid VAE-GAN and the original high fidelity model are shown in Fig. 3.10. It can be seen that the hybrid VAE-GAN predicted the flow fields well, which captures the most pressure features at time levels $t = 0.2, 0.65, 1.275, 1.675s$. Visually, very little difference between the hybrid VAE-GAN and the original high fidelity model can be noticed. In order to compare the differences between the predictive and original fluid fields, the absolute error and correlation coefficient of pressure solutions within the computational domain area Ω are illustrated in Fig. 3.11. It is observed that the absolute errors are small over the whole domain area at different time levels, and the correlation coefficient between the hybrid VAE-GAN and the high fidelity model is higher than 0.99. The predictive results in spatial space suggest the hybrid VAE-GAN is able to obtain reasonable and accurate solutions in spatial space for nonlinear fluid flows.

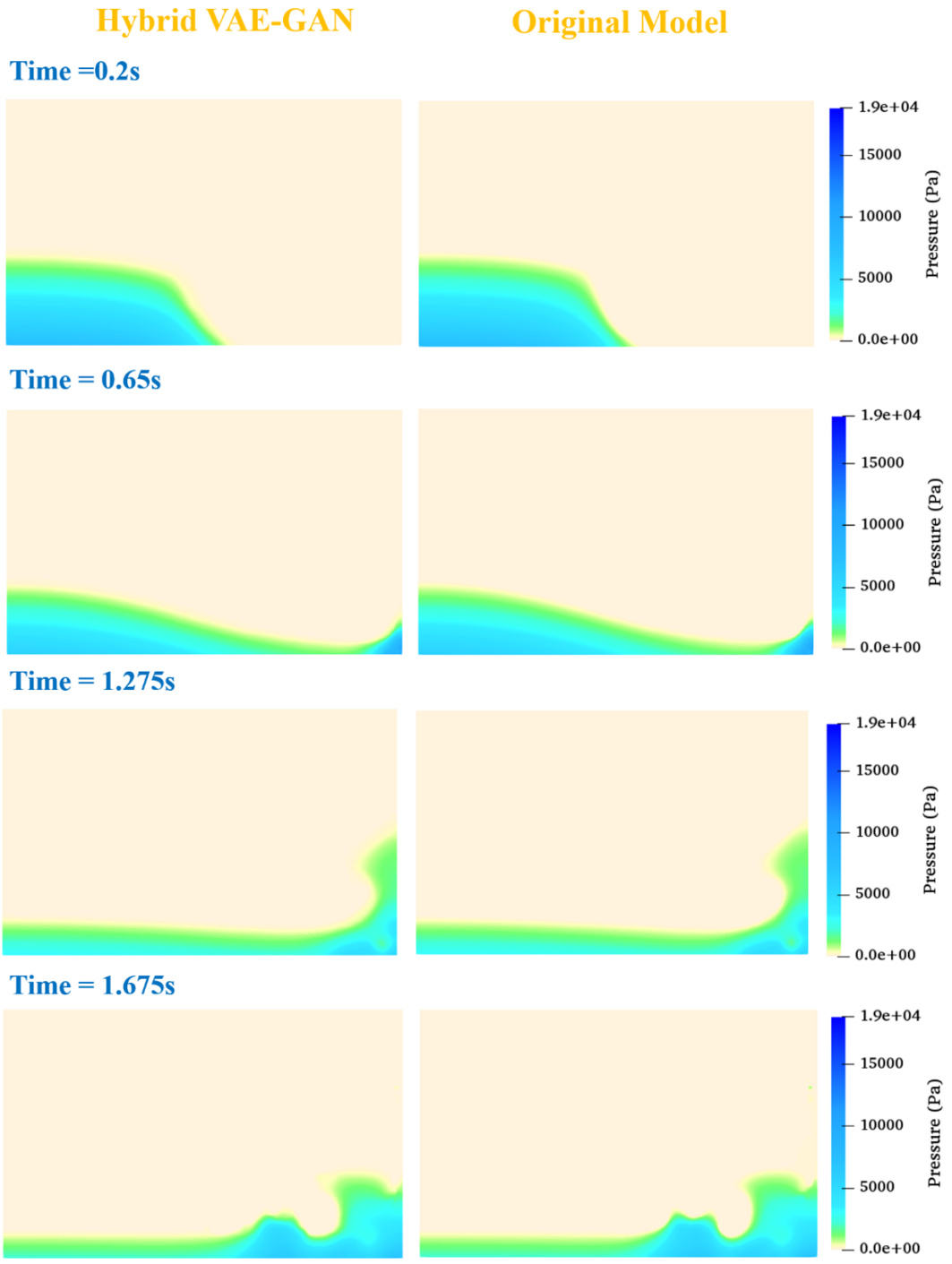


Fig. 3.10. Comparison of the spatial distribution of pressure fields obtained from the hybrid VAE-GAN (left) and the original high fidelity model (right) at time levels $t = 0.2, 0.65, 1.275, 1.675s$.

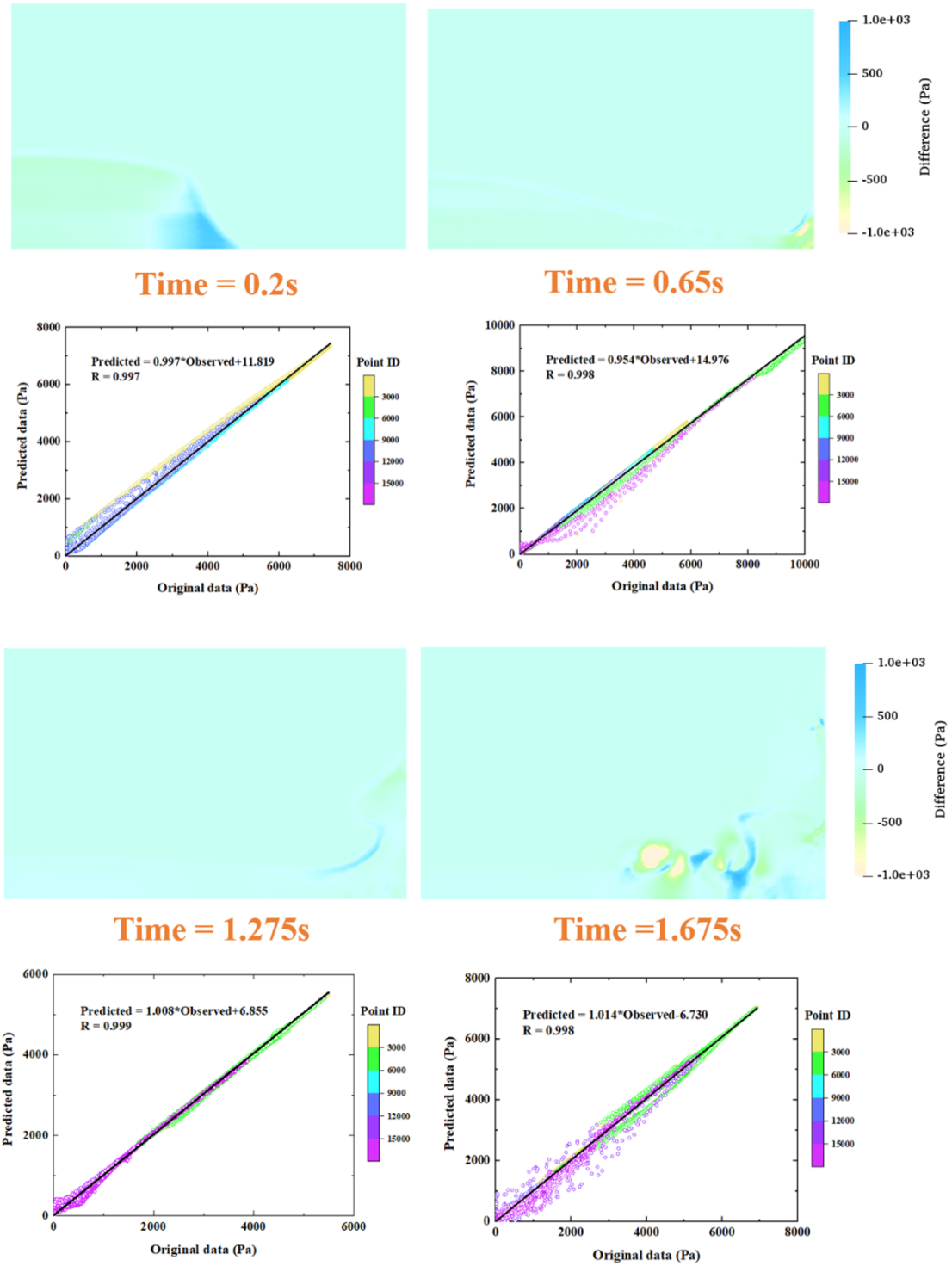


Fig. 3.11. Differences and correlation coefficients of pressure fields between the hybrid VAE-GAN and the original high fidelity model at time levels $t = 0.2, 0.65, 1.275, 1.675s$.

Prediction of temporal nonlinear fluid flows: Fig. 3.12 shows the temporal variation of pressure solution predicted by the two models. Before the water collapse happens, the

points P1, P2, P5 and P6 are located in the water area (as marked with a blue rectangle in Fig. 3.9), while other four points P3, P4, P7 and P8 are placed in the area filled with air. Model performance has been evaluated by comparison of pressure variation in these detector points. In Fig. 3.12(a), (b), (e) and (f), the pressure values at four points are slightly increased and then gradually decreased as the water collapse happens. As for detector P3, it is beyond the scale of water front motions before $t = 1.5s$, and the pressure trend in Fig. 3.12(c) is not so obvious. After $t = 1.5s$, influenced by the overturning water, the pressure at the detector P3 starts to fluctuate. Compared to detector P3, it can be noted that the pressure values at detector P4, P7 and P8 are uprising when the water drops down the horizon into rightward direction. The sequent water experiences dropping, traveling, rising up along the right vertical wall, and dropping again processes. It can be observed that the curves of pressure from the hybrid VAE-GAN achieve a good agreement with those of the original high fidelity model, except for the under-predictions of dropping again process after $t = 1.5s$. However, it is also a challenging problem for numerical simulation when the water hits the right wall and again meets the horizontal free surface (Park et al., 2009).

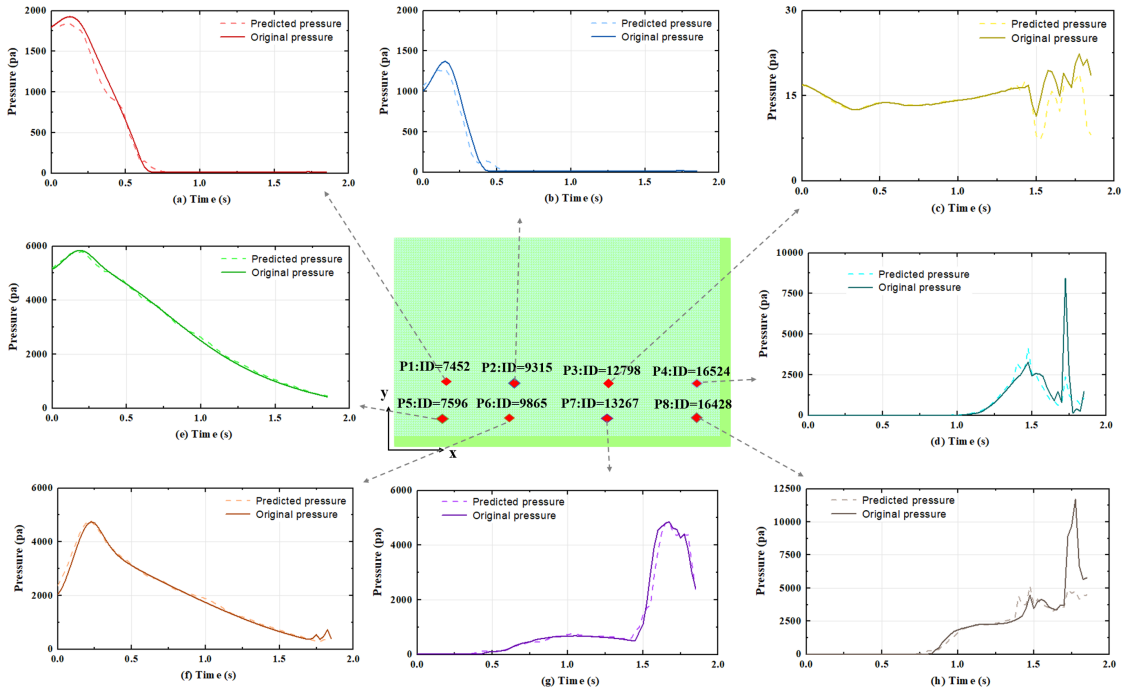
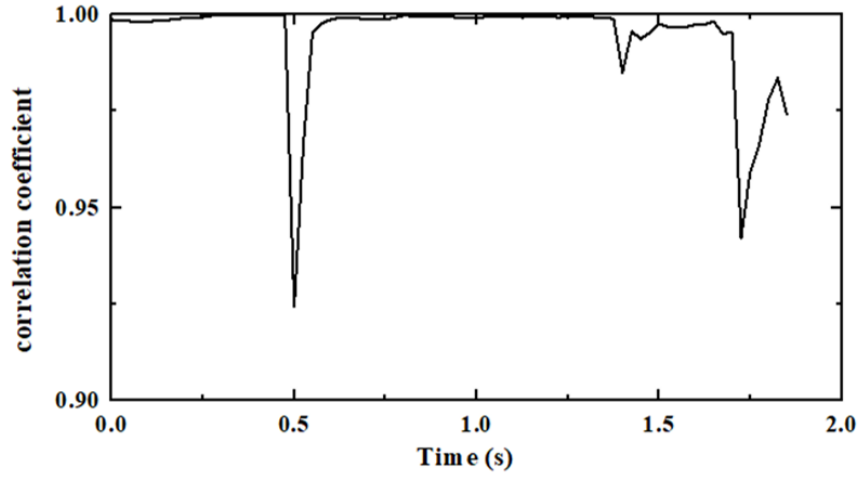
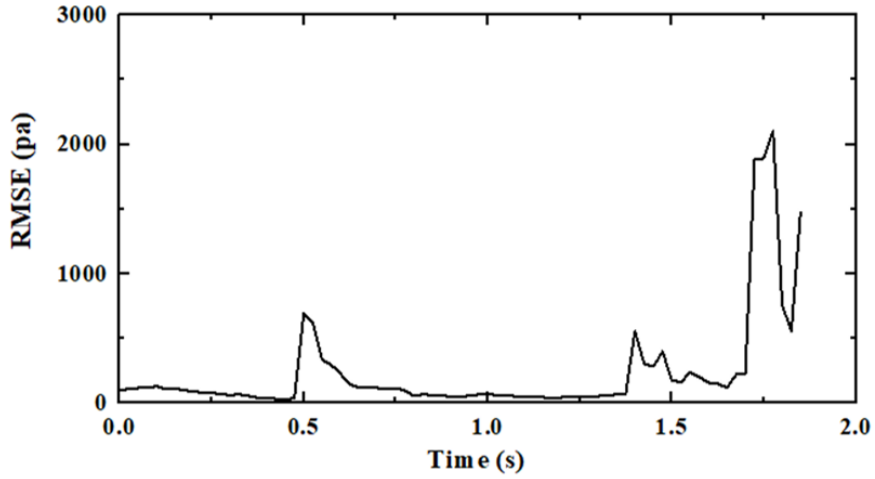


Fig. 3.12. Comparison of the temporal variation of pressure at points ID = 7452, 7596, 9315, 9865, 12798, 13267, 16428, 16524.

To further evaluate the performance of the hybrid VAE-GAN, the correlation coefficient and RMSE are calculated in the temporal-spatio space. The error analysis of pressure solutions in the temporal space is shown in Fig. 3.13. It is evident that most of RMSE values are between $0pa$ and $100pa$ and the correlation coefficients values are close to 99% during the whole simulation period except for some moments including $t = 0.5s$ and after $t = 1.5s$. The water front hits the right wall at $t = 0.5s$ and water drops again towards the left wall after $t = 1.5s$, which causes obvious errors as discussed above. The maximum value of RMSE is $2102pa$ at $t = 1.775s$, and the corresponding minimum value of correlation coefficient is 96%. In light of fluid field prediction in the temporal space, these results demonstrate that the hybrid VAE-GAN performs well and the predicted fluid fields are in good agreement with the true fluid fields during the whole process.



(a) Correlation Coefficient

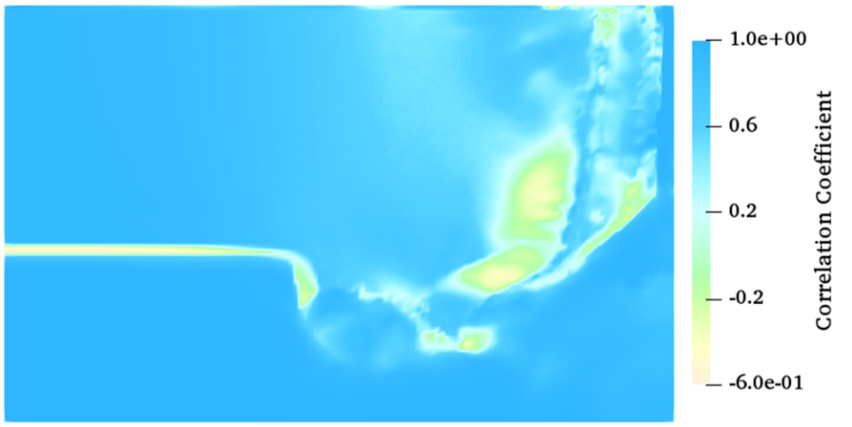


(b) RMSE

Fig. 3.13. The correlation coefficients and RMSE of pressure between the generated fields and original high fidelity fields during whole simulational period.

Fig. 3.14 illustrates the model performance in the spatial space. It can be noted that most values of RMSE are below $500pa$ in the most domain area Ω, except for the area close to the right wall, where the largest errors exist during the process of water rising up along the right vertical wall and maximum value of RMSE is $1940pa$. In addition, it is clear that the values of correlation coefficient in most area exceed 0.99, although the correlation coefficient is negative in some area influenced by the interface of multi-materials. This can be explained that when the water front strikes the right vertical wall, the water jet

causes a sudden rise of pressure and air entrainment (Park et al., 2009). The phenomena is shown in Fig. 3.12(c), where depicts the pressure fluctuation after $t = 1.5s$. The above results demonstrate the capability of hybrid VAE-GAN to predict accurate fluid flows.



(a) Correlation Coefficient



(b) RMSE

Fig. 3.14. The correlation coefficients and RMSE of pressure in time series in the computational domain area Ω.

3.5.3 Model accuracy and efficiency of hybrid VAE-GAN

In this section, the model accuracy and computational cost of the hybrid VAE-GAN are provided for all the case studies. Table 3.3 shows the online prediction accuracy of the hybrid VAE-GAN at different timesteps. It can be seen that the relative errors are between 0.03 to 0.16 in all the cases.

Table 3.3: The relative error of variables between the generated fields and original high fidelity fields during whole simulational period for all the cases.

Timestep	2	10	25	44
Case 1: flow past a cylinder (time interval = $0.2s$)				
scenario_200	0.03	0.1	0.04	0.11
scenario_100	0.04	0.14	0.05	0.13
scenario_40	0.07	0.14	0.08	0.15
Case 2: water column collapse (time interval = $0.025s$)	0.14	0.09	0.14	0.16

The simulations of the hybrid VAE-GAN and the original high fidelity model were preformed on Intel Xeon(R) CPU@ 3.60GHz. Table 3.4 lists the computation time required for the hybrid VAE-GAN and high fidelity model to simulate the flow past a cylinder and water column collapse in the online prediction process, where the CPU time required for the offline training is not included. It can be seen that the CPU time for running the hybrid VAE-GAN is reduced drastically by three orders of magnitude in comparison to the original high fidelity model in two examples.

Table 3.4: Online CPU time required for running the hybrid VAE-GAN and high fidelity model during the whole simulation period for two cases.

Case	Hybid VAE-GAN (Seconds)	High fidelity model (Seconds)
Case 1: flow past a cylinder	1	427
Case 2: water column collapse	22	8837

3.6 Conclusion

In this work, a hybrid VAE-GAN method has been, for the first time, used to predict nonlinear fluid flows in varying parameterized space. For any given input parameters μ, the hybrid VAE-GAN is capable of predicting accurately dynamic nonlinear fluid flows and remains highly efficient in simulations since it combines both advantages of VAE and GAN.

The performance of the hybrid VAE-GAN has been demonstrated by a flow past a cylin-

der test case and a second case of water column collapse. To evaluate the model accuracy, a detailed comparison between the original high fidelity model (Fluidity) and the hybrid VAE-GAN has been undertaken. The accuracy assessment has also been performed through the correlation coefficient, RMSE and relative error. The numerical simulations show that the hybrid VAE-GAN exhibits good agreement with the original high fidelity model, in both space and time. Additionally, a significant CPU speed-up has been achieved by the hybrid VAE-GAN for all the test cases.

The hybrid VAE-GAN is an efficient and robust tool for simulations of parameterized nonlinear fluid flows. It provides a wide range of applications, for instance, risk response management in emergencies and natural hazards (e.g. flooding, dam break, etc), real-time decision making. In the following chapters, more work will be focused on the predictive ability for a long lead-time prediction (beyond the trained period) by the hybrid VAE-GAN and other machine learning models.

CHAPTER 4

Long lead-time streamflow forecasting using machine learning methods

Long lead-time streamflow forecasting is of great significance for water resources planning and management in both the short and long terms. Despite of some studies using machine learning methods in streamflow forecasting, only few studies have been conducted to explore long lead-time forecasting capabilities of these methods, and gain an insight into systematic comparison of model forecasting performance in both the short and long terms. In this work, an artificial neural network (ANN) and a long short term memory (LSTM), a powerful tool for learning long-term temporal dependencies and capturing nonlinear relationship, have been adopted to forecast streamflow at daily and monthly scales for a long lead-time period. For long lead-time streamflow forecasting, a recursive forecasting procedure, which takes the last one-step-ahead forecast as a new input for the next-step-ahead forecast, is used in the ANN and LSTM forecasting systems. Two models are trained and validated for streamflow forecasting using the rainfall and runoff datasets collected from the Nan River Basin and Ping River Basin, Thailand, covering the period 1974 *to* 2014*. To further explore the impact of parameter settings on model performance, two parameters, i.e. the length of time lag and the number of maximum epochs, are examined in the ANN and LSTM models. The main findings are highlighted*

here. First, with an optimal setting up of model parameters, both the ANN and LSTM model can provide accurate daily forecasting (up to 20 days ahead). Second, in comparison to the ANN model, the LSTM model exhibits better model performance in long lead-time daily forecasting, but less satisfactory in multi-monthly forecasting due to lack of large monthly training dataset. Third, the selection of the length of the time lag and number of maximum epochs used in both ANN and LSTM modelling are the key for long lead-time streamflow forecasting at daily and monthly scales. These findings suggest that the LSTM could be advance in daily streamflow forecasting and thus would be helpful to assist in strategy decisions in water resource management.

This chapter is derived from and expands upon Cheng et al. (2020a).

4.1 Introduction

The purpose of this chapter is to identify the robust modelling approach from two popular machine methods, namely ANN and LSTM, especially for the long lead-time forecasting capability, through assessing both modelling accuracy and precision at daily and monthly scales. Two models are applied into two realistic cases, the Nan River Basin (NRB) and Ping River Basin (PRB), two of main subbasins of Chao Phraya River Basin (CPRB), the heart of Thailand. The main objectives of this chapter are: (1) to investigate the parameter settings on model forecasting performance at daily and monthly scales, and enable a screening of the best setting of parameters to attain an accurate model; (2) to explore the forecasting capabilities of machine learning methods for a long lead-time at daily and monthly scales; (3) to compare model forecasting results at daily and monthly scales, provide a deep insight into the quality of model forecast and explore the way in which various processes are represented.

The remainder of the chapter is organized as follows. In section 4.2, the governing equations for multi-step-ahead streamflow forecast strategy are briefly formulated, and the

daily and monthly forecast models (ANN and LSTM) are introduced in detail. Section 4.3 presents the study area-Nan River Basin and model development for streamflow forecasting. Forecasting performance of two models in daily and monthly scales is described in section 4.4. Finally in section 4.5, conclusions are presented.

4.2 Methodology

In this chapter, the long lead-time streamflow forecasting is conducted using both ANN and LSTM at daily and monthly scales. The ANN and LSTM models are developed for forecasting streamflow at several lead-times, such as 1-20 days and 1-12 months. In the model development, the model for one-step-ahead forecasting is first obtained in the training process. Then a recursive strategy is adopted in the trained models so that the multi-step-ahead forecast can be achieved. The following sections provide a detailed description of the one-step-ahead and multi-step-ahead daily/monthly streamflow forecasting strategy.

4.2.1 Multi-step-ahead daily and monthly streamflow forecast strategy

4.2.1.1 One-step-ahead daily and monthly streamflow forecast

In general, a streamflow forecasting model $\mathcal{F}$ is used to provide the predictive streamflow Q_{t+1} in a specified time scale, such as daily or monthly, based on historical climatic records. It is generally known that streamflow generation processes are influenced by many factors, including rainfall, evaporation, temperature, etc (Guo et al., 2011). As rainfall, a natural process that has a high degree of variability in both time and space, is the key factor for streamflow, this chapter focuses on developing a streamflow forecasting model depending on rainfall. Given a series of rainfall observation $\{R_{t-M+1}, ..., R_t, R_{t+1}\}$ (where

M is the length of time lag), an one-step-ahead streamflow forecast can be given as:

$$\widehat{Q}_{t+1} = \mathcal{F}(R_{t-M+1}, ..., R_t, R_{t+1}), \tag{4.1}$$

where $\widehat{Q}_{t+1}$ represents the predictive streamflow at time $(t+1)$, R denotes rainfall dataset, $\mathcal{F}$ is a forecasting model.

In order to obtain a satisfactory forecast accuracy, a key problem is how to choose the length of lagged inputs M in Eq.(4.1). Enough high-quality observed inputs enable the pattern detection between rainfall and runoff, however, the length of inputs is limited by availability of measurements in practice. In addition, as the lagged time length M increases, so too does the number of inputs and the complexity of the forecasting model developed (Bowden et al., 2005). In this chapter, an investigation of impacts of the time lag length on model forecasting performance is conducted in section 4.4.1.1 and section 4.4.2.1.

4.2.1.2 Multi-step-ahead daily and monthly streamflow forecast

The multi-step-ahead time series forecasting can be described as an estimation on future time series $\widehat{Q}_{t+h}$ (where $h = (1, .., H)$, H is the total of predictive steps, also called the lead-time). The most popular forecasting strategy is the recursive (also called iterated) strategy (Taieb et al., 2010; Bontempi et al., 2012). In this study, to achieve the multi-step-ahead streamflow forecast, a single forecasting model $\mathcal{F}$ is trained to perform an one-step-ahead forecast in section 4.2.1.1. After the learning process, the multi-step-ahead streamflow can be predicted using:

$$\widehat{Q}_{t+h} = \begin{cases} \mathcal{F}(R_{t-M+1}, ..., R_t, R_{t+1}), & (h = 1) \\ \mathcal{F}(R_{t-M+h}, ..., R_{t+1}, \widehat{Q}_{t+1}, ..., \widehat{Q}_{t+h-1}), & (h \in \{2, ..., M+1\}) \\ \mathcal{F}(\widehat{Q}_{t-M+h-1}, ..., \widehat{Q}_{t+h-1}), & (h \in \{M+2, ..., H\}) \end{cases}, \tag{4.2}$$

where H is the total of predictive steps, also the length of lead-time, M is the length of time lag. The last equation in Eq.(4.2) is used for streamflow forecasting when the rainfall data is not available during the predictive period. In this chapter, the multi-step-ahead streamflow forecast $\widehat{Q}_{t+h}$ at daily and monthly scales is obtained as following steps:

(1) Splitting a series of rainfall and runoff datasets into training datasets $(\mathcal{X}_{training}, \mathcal{Y}_{training}) \in$ the training period (T_0, T_p) and validation datasets $(\mathcal{X}_{validated}, \mathcal{Y}_{validated}) \in$ the validation period (T_p, T_N). Let $\mathcal{X}_{training} = \{R_{t-M+1}, ..., R_t, R_{t+1}\}$ $((t+1) \in (T_0, T_p)$) as inputs, and $\mathcal{Y}_{training} = Q_{t+1}$ as the targeted outputs.

(2) Selecting the time lag length (M, the number of lagged rainfall records to use as inputs) to create a combination of inputs and outputs.

(3) Training the forecasting model $\mathcal{F}$ by different combination parameters, e.g. the length of time lag and the maximum epoch number, and selecting the best parameter combination for one-step-ahead streamflow forecasting.

(4) Selecting the length of lead-time H in Eq.(4.2).

(5) Using the $\mathcal{X}_{validated} = \{R_{t-M+1}, ..., R_t, R_{t+1}\}$, $((t-M) \in (T_p, T_N))$ as new inputs, to forecast the next-step-ahead streamflow by the trained ANN model in step (3).

(6) Combining the one-step-ahead streamflow forecast in step (5) and the available rainfall records as new inputs, and repeating the step (5) to obtain the second-step-ahead streamflow forecast.

(7) Repeating the steps (6) and (5) to forecast H-step-ahead streamflow in Eq.(4.2) during (T_p, T_N).

In this work, the ANN and LSTM models are selected as the forecasting model $\mathcal{F}$ in Eqs. (4.1) and (4.2) for daily and monthly streamflow forecasting in a recursive strategy (as

shown in Fig. 4.1, details in Algorithm 4.1), which is introduced in detail in section 4.2.2 and section 4.2.3 respectively.

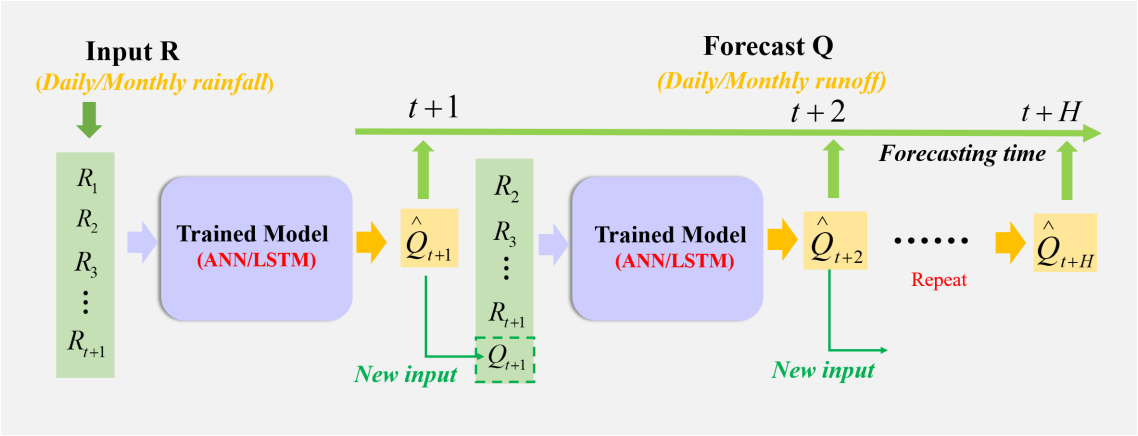


Fig. 4.1. Model forecasting strategy using a recursive way at daily and monthly scales.

Algorithm 4.1 Multi-step-ahead forecast using ANN and LSTM at daily and monthly scales. ANN and LSTM are used for learning the input-output relationship $\mathcal{F}$ in Eq.(4.2)

(1) Parameter optimization in training process.
- Obtain the optimal parameters in training process.
- Select the optimal time lag M.
- Select the optimal maximum epoch E.

(2) Multi-step-ahead streamflow forecast process.
- Input $\mathcal{X}_{validated}$: Sample M+1 steps rainfall datasets $\{R_{t-M+1}, ..., R_t, R_{t+1}\}$.
- Output $\mathcal{Y}_{validated}$: Sample one step runoff dataset Q_{t+1}.
- Select the forecasting step H.

for h = 1 to H **do**
  if $h = 1$ **then**
    - Predict the next-step-ahead streamflow using the trained function $\mathcal{F}$ in Eq.(4.2):

$$\widehat{Q}_{t+h} = \mathcal{F}(R_{t-M+h}, ..., R_{t+1}).$$

  else if $h \leq M + 1$ **then**
    - Predict the next-step-ahead streamflow using the trained function $\mathcal{F}$ in Eq.(4.2):

$$\widehat{Q}_{t+h} = \mathcal{F}(R_{t-M+h}, ..., R_{t+1}, \widehat{Q}_{t+1}, ..., \widehat{Q}_{t+h-1}).$$

  else
    - Predict the next-step-ahead streamflow using the trained $\mathcal{F}$ in Eq.(4.2):

$$\widehat{Q}_{t+h} = \mathcal{F}(\widehat{Q}_{t-M+h-1}, ..., \widehat{Q}_{t+h-1}).$$

  end if
end for
- Obtain the length-H steps of forecast $\widetilde{\mathcal{Q}}$.

4.2.2 Artificial neural network for daily and monthly streamflow forecast

ANN, a mathematical model of biologically inspired system, provides a novel and appealing solution to the problem of relating input and output variables in complex systems (Basheer and Hajmeer, 2000). It requires no information about the underlying complex physical process, while constructing black-box models of complex and nonlinear relationships between the input and output variables. In general, an ANN model consists of three typical layers. Each layer processes a series of neurons, which are fully connected with

those of the following layer. As illustrated in Fig. A.1, the first layer is the input layer that receives the input data. The hidden layer is the information processing section, which uses weights to achieve nonlinear transformation between the connection links to determine the output. The output layer receives the processed information from the last hidden layer and then outputs the result. The ANN model can be mathematically formulated as:

$$\mathcal{Y}_k = \sigma_k[\sum_{j=1}^{m} w_{kj}\sigma_j(\sum_{i=1}^{n} w_{ij}\mathcal{X}_i + b_j) + b_k] \tag{4.3}$$

where $\mathcal{X}_i$ represents the input value at the neuron i, $\mathcal{Y}_k$ is the output value at the neuron k, σ_k and σ_j denote the activation function for the hidden and output layers respectively. m and n are the number of neurons at the input and hidden layers respectively, w_{ij} is the weight between the input neuron i and the hidden neuron j, w_{kj} is the weight between the hidden neuron j and the output neuron k, b_j and b_k are biases of the j^{th} neuron at the hidden layer and k^{th} neuron at the output layer, respectively.

To achieve appropriate performance of ANNs, the parameters, for example, the number of neurons, weights and biases between layers and the number of iterations, should be optimized. In this study, a four-layer ANN network that consists of one input layer, two hidden layers and one output layer is established. The sigmoid function ($f(a) = 1/(1 + \exp^{-a})$) and the hyperbolic tangent function ($f(a) = \tanh(a)$) are employed between the layers. Levenberg-Marquardt algorithm (Asadi et al., 2013; Alizadeh et al., 2017; Zhang et al., 2018) is selected as the optimization algorithm for training the ANN model, which is a way to adjust the parameters consisting of weights w and bias b between layers. Referring to multi-step-ahead forecast steps in section 4.2.1.2, the ANN forecasting approach at daily and monthly scales (as shown in Fig. 4.1) is summarized in Algorithm 4.1.

4.2.3 Long short-term memory model (LSTM) for daily and monthly streamflow forecast

The LSTM architecture is composed of special units (called memory blocks) in the recurrent hidden layers. Each memory block contains the self-connected memory cells and multiplicative units. The memory cells are used to store the temporal state of the networks. The multiplicative units, including the input, output and forget gates, controls the flow of information between the cells. The input gate is responsible for controlling the flow of inputs into the memory cell, while the output gate conducts the output flow of cell activations. The forget gate scales the internal state of the cell. Fig. A.2 shows the information flow through a LSTM cell.

The LSTM transition equations are written below:

$$f_s = \sigma(W_f[h_{s-1}, x_s] + b_f), \tag{4.4}$$

$$i_s = \sigma(W_i[h_{s-1}, x_s] + b_i), \tag{4.5}$$

$$\widetilde{C_s} = \tanh(W_c[h_{s-1}, x_s] + b_c), \tag{4.6}$$

$$C_s = f_s \odot C_{s-1} + i_s \odot \widetilde{C_s}, \tag{4.7}$$

$$O_s = \sigma(W_o[h_{s-1}, x_s] + b_o), \tag{4.8}$$

$$h_s = O_s \odot \tanh(C_s), \tag{4.9}$$

where i_s, O_s and f_s are the input, output and forget gates respectively, W_i, W_o, W_f and W_c represent the weights for each gate, b_i, b_o, b_f and b_c are the bias terms, σ denotes the logistic sigmoid function, $tanh$ is the hyperbolic tangent function, $\odot$ is the the scalar product of two vectors, C_s represents the cell state, $\widetilde{C_s}$ is the updated cell state, x_s and h_s are the cell input and output respectively.

In this study, the LSTM forecasting approach consists of one input layer, two hidden layers and one output layer. The LSTM is trained based on truncated Back Propagation

Through Time (BPTT), which uses a back propagation network to update the parameters in iterations (Werbos, 1990). Referring to multi-step-ahead forecast steps in section 4.2.1.2, the LSTM model forecasting at daily and monthly scales similar to that of the ANN forecasting approach, is described in Fig. 4.1 and Algorithm 4.1.

4.3 Application

4.3.1 Study area

The Chao Phraya River Basin (CPRB), the heart of Thailand, is the center of rice production and the region economy (Wichakul et al., 2013). Nan River Basin (NRB) and Ping River Basin (PRB) were selected as two study sites as shown in Fig. 4.2. The first study site NRB, as one of major subbasins of CPRB, covers an area of $11,950\,km^2$ (Kinouchi et al., 2018). The CPRB, located in the heart of Thailand, is the center of rice production and the region economy (Wichakul et al., 2013). Nan River Basin (NRB) as one of major subbasins of CPRB, covers the area of $11,950\,km^2$ (Kinouchi et al., 2018). This basin includes the third largest dam reservoir -Sirikit Dam Reservoir in Thailand as shown in Fig. 4.2. The function of Sirikit Dam Reservoir is to supply domestic water, irrigation and power generation. The river flow through the center of Nan province and drains into the reservoir. The basin is in a subtropical monsoon region, with relatively abundant rainfall and a humid climate from May to September. The average annual precipitation over the basin is 900-2000 mm. To determine the relationship between rainfall and runoff, streamflow into the Sirikit Dam Reservoir and rainfall data from gauges (as shown in Fig. 4.2) were collected at daily and monthly scales from 1974 to 2014. The introduction and results of streamflow forecasts at the second study site PRB, are provided in the section A.2.

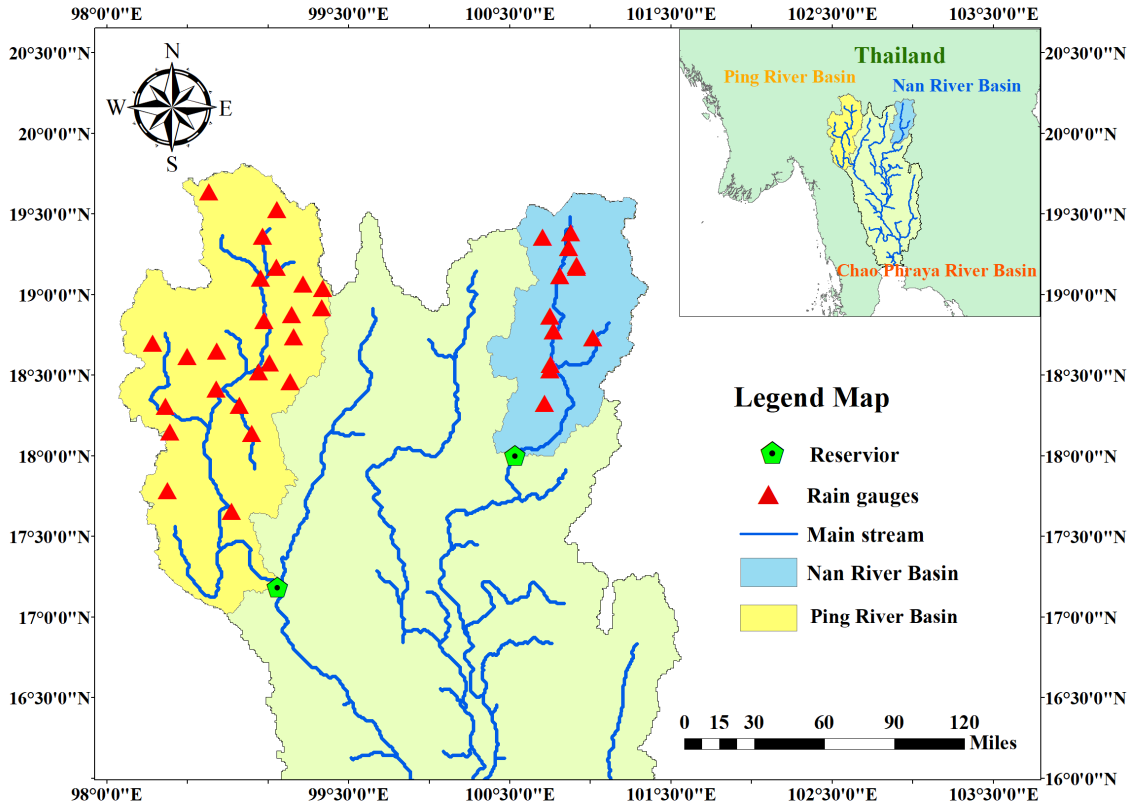


Fig. 4.2. Location of the Nan River Basin (NRB) and Ping River Basin (PRB).

4.3.2 Daily and monthly model development

In this study, the ANN and LSTM models are employed to forecast streamflow at daily and monthly scales. The daily and monthly dataset are split into the separations of training and validation datasets as shown in Table 4.1. The model inputs are the available observed rainfall, and predicted streamflow (output) from the previous time levels $(t-M+1,\ldots,t)$ (as shown in in Eq.(4.2)). The targeted output is the streamflow at the time level $t+1$. All datasets were scaled to the range of 0-1 to match the consistency of machine learning-based models, and rescaled back to the original values after the model simulation (Zhang et al., 2018).

Table 4.1: The training and validation datasets at daily and monthly scales.

	Training period	Validation period
Daily scale	1 January 1974 to 17 November 2007	18 November 2007 to 20 February 2011
Monthly scale	January 1974 to March 2004	April 2004 to December 2014

To evaluate the model parameter impact, different setups of model parameters are employed in the ANN and LSTM models during training period in both daily and monthly forecasting. The parameters settings used in two models are summarized in Table 4.2. To demonstrate the model convergence, an example is shown in Fig. A.3. In this case, the ANN model is applied for streamflow forecasting and the training dataset is 11500 days. The model is trained in 5000 epochs with time lag=10days. The loss between the predicted and true streamflow is calculated by mean squared error (MSE). In Fig. A.3, it can be noticed that the ANN model converges as its loss actually moves towards a minima with a decreasing trend in the whole training process.

Table 4.2: The parameter settings of ANN and LSTM models at daily and monthly forecast.

Time lag length	Daily forecast (day)										Monthly forecast (month)				
	1	3	5	10	15	20	25	30	35	40	1	2	3	4	5
Epochs (ANN)	100, 200, 300, 400, 500										100, 200, 300, 400, 500				
Epochs (LSTM)	10, 20, 30, 40, 50										10, 20, 30, 40, 50				

4.3.3 Model performance evaluation

The performance of the ANN and LSTM forecasting models developed in this study is assessed using various standard statistical performance evaluation criteria. The statistical measures used here are the root mean squared error (RMSE), the Nash-Sutcliffe efficiency coefficient (NSE) (Nash and Sutcliffe, 1970), the coefficient of correlation (CC) (Taylor, 1990), and the mean absolute error (MAE) (Legates and McCabe Jr, 1999). These indicators can be defined as follows:

(1) Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (Q_i^s - Q_i^o)}{n}}, \tag{4.10}$$

(2) Nash–Sutcliffe Efficiency Coefficient (NSE):

$$NSE = 1 - \left[\frac{\sum_{i=1}^{n} (Q_i^o - Q_i^s)^2}{\sum_{i=1}^{n} (Q_i^o - \overline{Q}^o)^2} \right], \tag{4.11}$$

(3) Coefficient of Correlation (CC):

$$CC = \frac{\sum_{i=1}^{n} (Q_i^s - \overline{Q}^s)(Q_i^o - \overline{Q}^o)}{\sqrt{\sum_{i=1}^{n} (Q_i^s - \overline{Q}^s)^2 \sum_{i=1}^{n} (Q_i^o - \overline{Q}^o)^2}}, \tag{4.12}$$

(4) Mean Absolute Error (MAE):

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |Q_i^s - Q_i^o|, \tag{4.13}$$

where Q_i^s and Q_i^o are the i^{th} forecasting and observed streamflow respectively, $\overline{Q}^s$ and $\overline{Q}^o$ are the average forecasting and observed streamflow, n is the number of datasets.

4.4 Results and Discussions

In this section, the observed streamflow in the Sirikit Dam Reservoir is compared with forecasting streamflow from the ANN and LSTM models, combined with different model parameters at daily and monthly scales over the validation period.

4.4.1 Daily streamflow forecast

4.4.1.1 Modelling parameter optimization in daily streamflow forecast

A number of ANN and LSTM modelling cases have been set up with varying parameters (the length of the time lag and the maximum epoch number). The impact of modelling parameters used in ANN and LSTM on results is demonstrated in Fig. 4.3 and Fig. 4.4. As shown in Fig. 4.3, the average RMSE in the boxes is gradually decreased when the length of the time lag increases from 1 day to 10 days and then remains almost the same ($220\,m^3/s$) when the time lag exceeds 10 days in ANN modelling, while the length of the time lag has a slight effect on results of LSTM modelling. Fig. 4.4 shows that the average values of RMSE in both the ANN and LSTM models exhibit a rising trend when the maximum epoch number increases.

The impact of the number of lagged days and epochs used in ANN and LSTM modelling is further estimated by the map of the RMSE of the forecasting daily streamflow results shown in Fig. A.4 (where the box is masked from green to blue in Fig. A.4 (a) and (b)). It is observed that the LSTM model achieves better forecast performance than ANN models. The lowest and highest RMSE (outlined in red) for ANN modelling are $149.25\,m^3/s$ and $271.51\,m^3/s$ respectively, while for LSTM modelling, $118.51\,m^3/s$ and $199.22\,m^3/s$ respectively. The optimal settings of the lagged days and the maximum epoch number are provided in Table 4.3. The best statistical result of ANN modelling in one-step-ahead forecast is achieved with the time lag = 10 days and epochs = 200. For LSTM modelling, the best statistical results are obtained for 25 days and 10 respectively.

Overall, the ANN and LSTM models exhibit different forecasting performance with varying parameter settings. It can be concluded that (1) an optimal time lag can be selected to improve the ANN performance, (2) the effects of the length of time lag for the LSTM model are uncertain in one-day-ahead forecast, and (3) the increase of the number of epochs deteriorates the ANN and LSTM performance in one-day-ahead forecast.

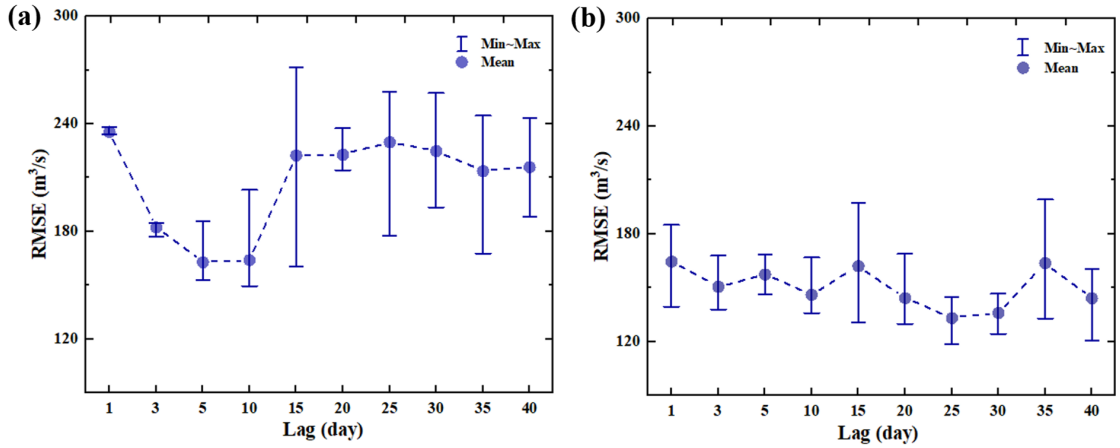


Fig. 4.3. Impact of the length of the time lag on streamflow daily forecasting, where the RMSE of streamflow results using (a) ANN, and (b) LSTM.

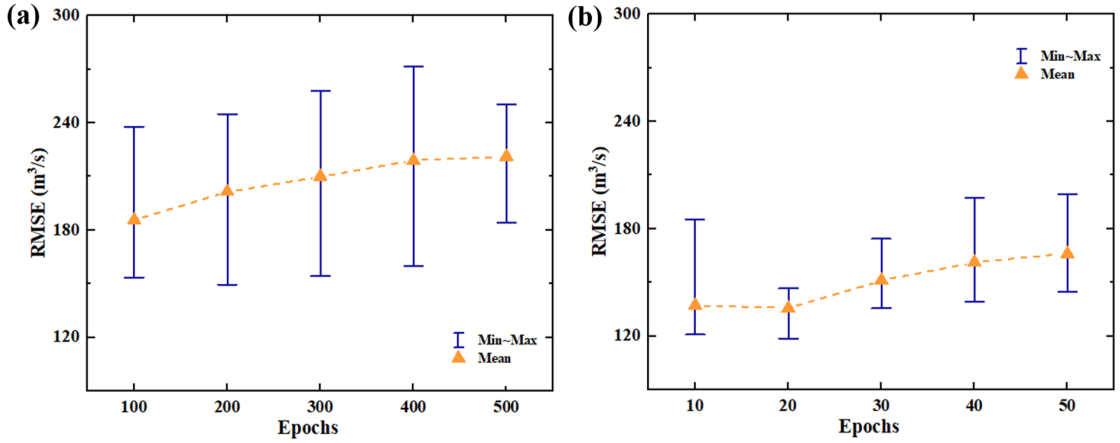


Fig. 4.4. Impact of the maximum number of epochs on streamflow daily forecasting, where the RMSE of streamflow results using (a) ANN, and (b) LSTM.

Table 4.3: The optimal parameter settings in daily forecasting.

	Time lag (days)	Epochs	RMSE (m^3/s)
ANN	10	200	149.56
LSTM	25	20	118.51

4.4.1.2 One-day-ahead streamflow forecast

Fig. 4.5 shows the daily hydrographs at Sirikit Dam Reservoir in one-day-ahead forecast from 2007 to 2010 using optimal parameters in Table 4.3. As shown in Fig. 4.5, it can

be noticed that the forecasting streamflow exhibits good agreement with the observed streamflow. The streamflow forecast using the ANN and LSTM models could capture the daily variability, but underestimates the streamflow in the dry season while overestimates it in the wet season. By using LSTM model, the predicted streamflow shows a reduction of overestimation in the wet season compared to that using the ANN model, and exhibits a good match with the observations in Fig. 4.5. A comparison between the ANN and LSTM models indicates that the LSTM model performs better than the ANN model in daily streamflow forecasting. For example, the R^2 is 0.86 for LSTM modelling while it is 0.76 in 2008 streamflow forecast for ANN modelling. In addition, it is noted that the number of epochs required for LSTM model convergence is much smaller than the ANN model while the number of the time lag has little effect on LSTM model results. This indicates that the LSTM model is more robust and can extract nonlinearity characteristics of data more efficiently (Zhang et al., 2018) in comparison to the ANN model.

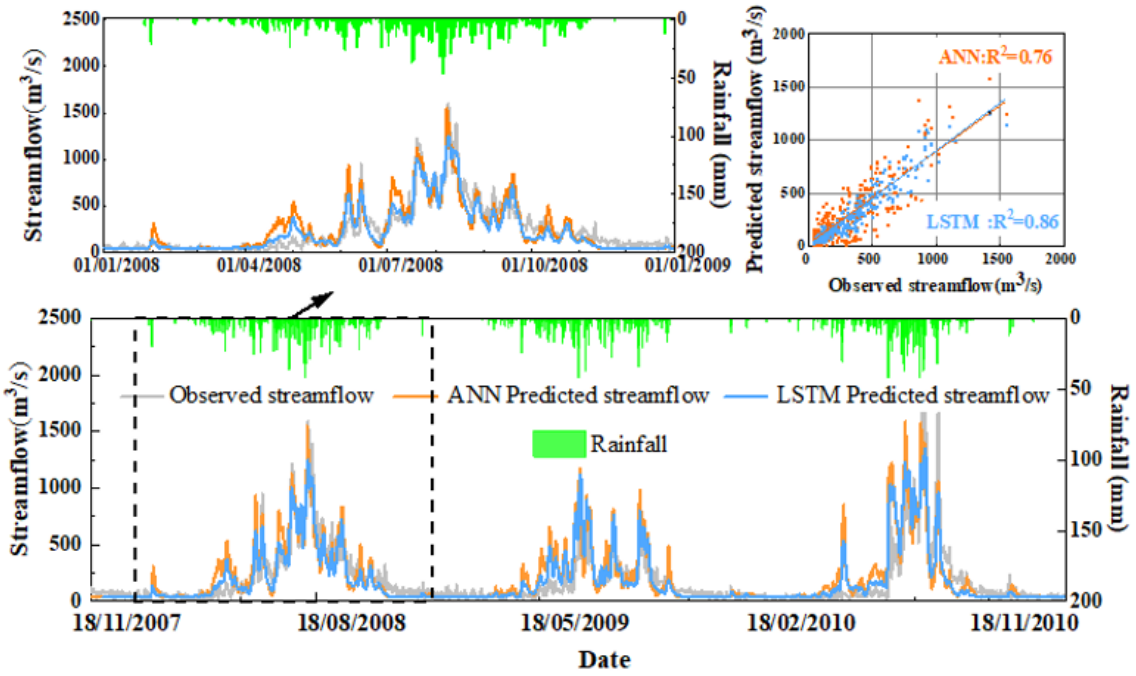


Fig. 4.5. Comparison of forecasting and observed daily streamflow using the ANN and LSTM models with the optimal parameters in Table 4.3.

4.4.1.3 Multi-day-ahead streamflow forecast

The ANN and LSTM models have been further applied to multi-day-ahead streamflow forecasting. The comparative plots of the forecasting results obtained from the ANN and LSTM models at the lead-time = 1 day, 2 day, 3 day, 4 day, and 5 day are shown in Fig. 4.6. It can be observed that a relatively good agreement between the observed and forecasting streamflow, especially the streamflow peak is achieved at the lead-time = 1 day and 2 day. With increasing the lead-time, the arrival time of forecasting streamflow peaks becomes slightly delayed, which is caused by the accumulation of forecast errors. Compared with low streamflow simulation of the LSTM model from 5 October 2008 to 26 October 2008, the streamflow peaks of the ANN model are generally higher than observed streamflow. In addition, the predicted streamflow exhibits abnormal fluctuations in the ANN simulation.

Four indicators, RMSE, NSE, CC, and MAE, are chosen to evaluate the multi-day-ahead forecast performance during the validation period (from 18 November 2007 to 20 February 2011). The corresponding evaluation of forecasting streamflow results during the lead-times of 1-20 days is illustrated in Fig. 4.7. It can be seen that the LSTM model generally outperforms the ANN model with the smaller RMSE and MAE (in Fig. 4.7 (a) and (d)), and higher NSE and CC especially at the longer lead-times (in Fig. 4.7 (b) and (c)). It is also visible that the predicted accuracy of the ANN and LSTM models decays as the forecasting horizon extends further in time (i.e. with the increasing lead-time length). As the lead-time increases from 1 to 20 days, the RMSE and MAE in ANN modelling increase approximately by 254% and 232% respectively, while in LSTM modelling, the RMSE and MAE increase approximately by 291% and 215% respectively. The NSE ranges between 0.1 and 0.68 for the ANN model, while between 0.33 and 0.78 for the LSTM model. The CC decreases from 0.83 to 0.59 for the ANN model and from 0.89 and 0.64 for the LSTM model as the lead-time increases.

The results suggest that the forecasting performance for the LSTM model is better than

that of the ANN model for long lead-time daily forecasting, which may be explained by the model structure. For LSTM model, it processes the time series datasets as a sequence and one element as input at a time, and the past temporal information is stored in the memory cell, which helps the LSTM model to capture datasets trend and to exhibit more powerful forecasting capability than the ANN model (Zhang et al., 2018; Kratzert et al., 2018; Le et al., 2019). This enables a memory mechanism in the LSTM model, where the network uses information about past calculations from a few past steps to inform the decision of whether or not this information should be passed along to the next iteration (Mouatadid et al., 2019). As the LSTM model processes the input data in many time steps, the input data are used to update a number of parameters in the LSTM internal memory cell states in every step during a training period. During the prediction period, the memory cell states depend only on the input at a specific time step and the states from the last time step (Kratzert et al., 2018). However, the ANN model does not have a temporal memory and the inputs in the model are assumed to be independent of each other, so that it is difficult to recognise temporal changes (de la Fuente et al., 2019). Therefore, memory cells in the model structure help the LSTM model capture datasets trend and exhibit more powerful forecasting capability than the ANN model.

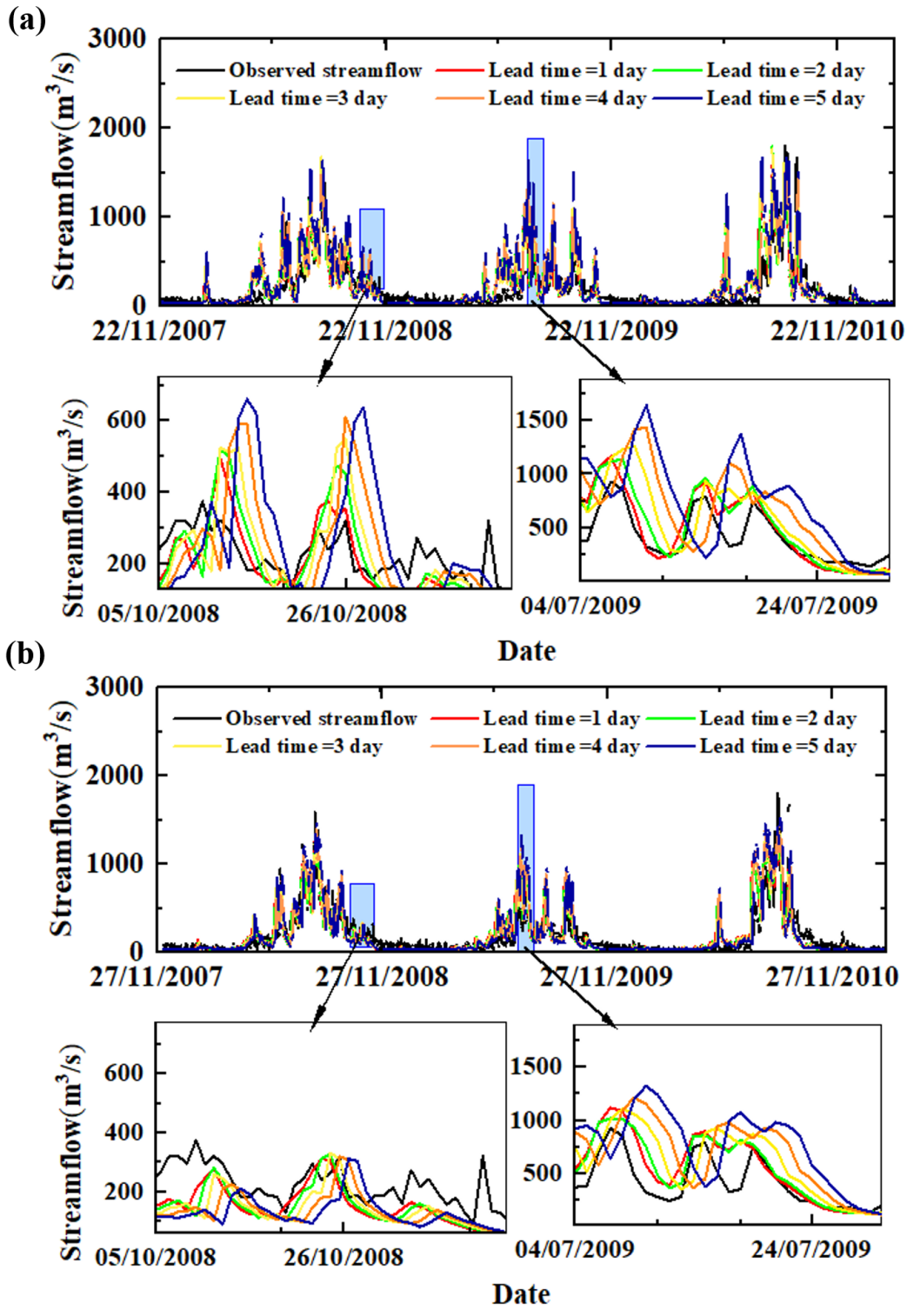


Fig. 4.6. Forecast performance at the lead-times 1, 2, 3, 4 and 5 days: (a) with the ANN model and (b) with the LSTM model.

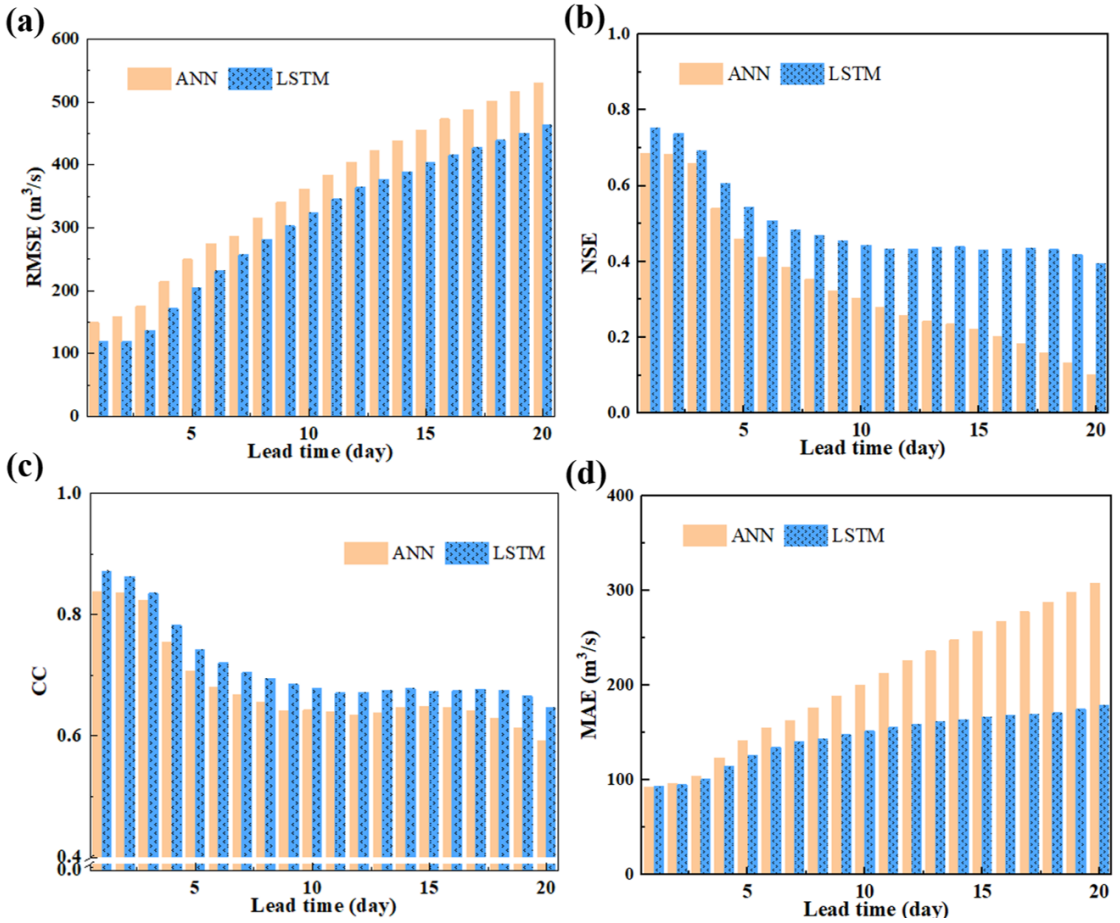


Fig. 4.7. Forecast performance with the ANN and LSTM models over the validation period of year 2007 to 2011 at the lead-times of 1-20 days: (a) RMSE, (b) NSE, (c) CC and (d) MAE.

4.4.2 Monthly streamflow forecast

4.4.2.1 Modelling parameter optimization in monthly streamflow forecast

For optimal setting up of modelling parameters used in ANN and LSTM, I have undertaken a number of test cases with varying the time lag length and epoch number. The error estimate of results with different modelling parameters has been plotted in Fig. 4.8 and Fig. 4.9. As shown in Fig. 4.8, it can be noticed that the average RMSE in the boxes is diminishing rapidly as the length of the time lag rises from 1 month to 5 months in ANN modelling, while the length of the time lag has no obvious impact on results in LSTM modelling. In contrast, the average values of RMSE exhibit a descending trend when the

maximum number of epochs increases in LSTM modelling, however, there is no obvious trend of epochs effecting on ANN modelling in Fig. 4.9.

Fig. A.5 shows the map of the RMSE of the forecasting daily streamflow results in ANN and LSTM modelling. By Comparing Fig. A.5 (a) and (b), it is obvious that the ANN model is sensitive to the time lag selection while the LSTM model is more susceptible to the maximum epoch number. For example, in ANN modelling, the RSME of results is increased from $3800\,m^3/s$ (blue box) to $1500\,m^3/s$ (yellow box) when the time lag rises from 1 to 5 months with a fixed maximum epoch number in Fig. A.5(a). Similarly, in Fig. A.5(b), the performance of the LSTM model becomes better when the maximum epoch number gradually increases from 10 to 50 with the time lag = 1 month. The lowest and highest RMSE (outlined in red) for the ANN model are $1660.16\,m^3/s$ and $3738.63\,m^3/s$ respectively, while those of the LSTM model are $1698.97\,m^3/s$ and $2535.57\,m^3/s$ respectively. The highest accuracy statistics of the ANN and LSTM models at monthly scale are given in Table C.1. The best statistical result of the ANN model in one-month-ahead forecast is achieved with the time lag = 4 months and epochs = 500. For the LSTM model, the time lag and maximum epoch numbers are 2 months and 50, respectively. The results indicate that the ANN model is comparable to the LSTM model in monthly streamflow forecast.

Overall, for the influence of different parameter settings on the ANN and LSTM model performance, it can be concluded that (1) increasing the time lag could improve the ANN model accuracy in one-month-ahead forecast while remain inconspicuous for the LSTM model, and (2) the increase of maximum epochs could improve the LSTM accuracy but remains uncertain for the ANN model in one-month-ahead forecast.

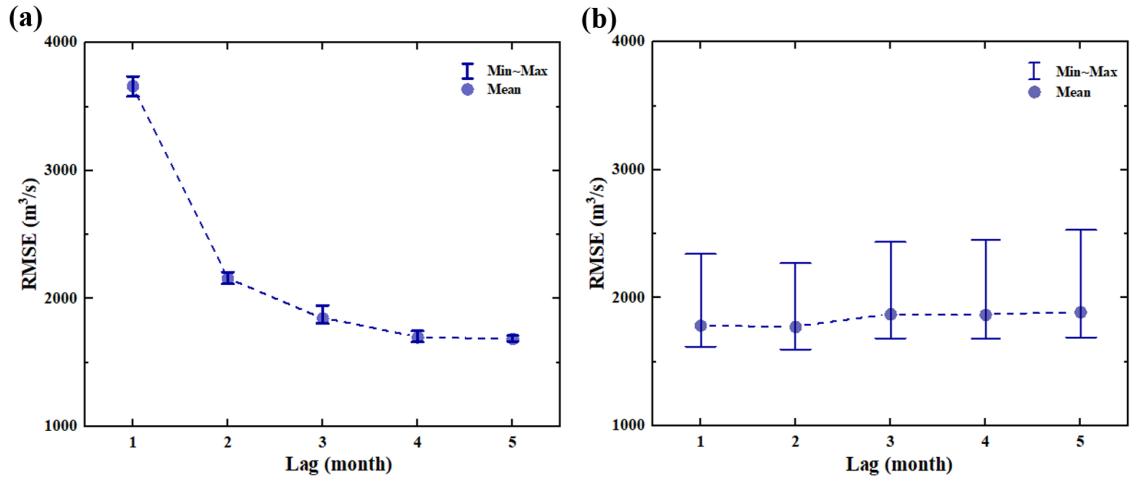


Fig. 4.8. Impact of the time lag on streamflow monthly forecasting, where the RMSE of streamflow results using (a) ANN, and (b) LSTM.

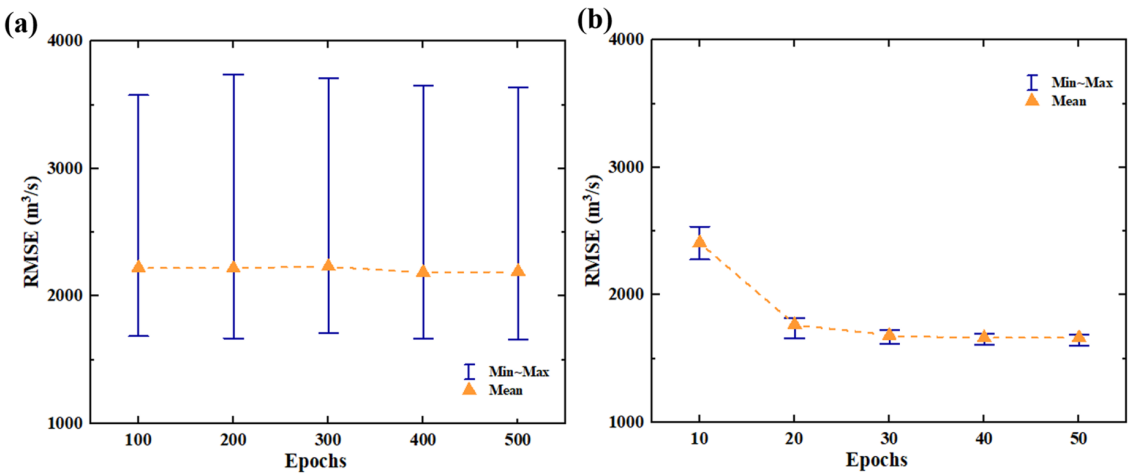


Fig. 4.9. Impact of the maximum number of epochs on streamflow monthly forecasting, where the RMSE of streamflow results using (a) ANN, and (b) LSTM.

Table 4.4: The optimal parameter settings in monthly forecasting.

	Time lag (months)	Epochs	RMSE (m^3/s)
ANN	4	500	1660.16
LSTM	2	50	1598.97

4.4.2.2 One-month-ahead streamflow forecast

Fig. 4.10 displays the monthly hydrographs at Sirikit Dam Reservoir in one-month-ahead forecast from year 2004 to 2012 using the optimal parameters in Table C.1. It is shown

in Fig. 4.10 that the forecasting streamflow exhibits good agreement with the observed streamflow. The ANN and LSTM models not only capture the monthly streamflow variability, but also successfully predict the low streamflow in dry season. During the validation period (from April of 2004 to December of 2013), the R^2 is 0.94 and 0.95 for the ANN and LSTM models, respectively.

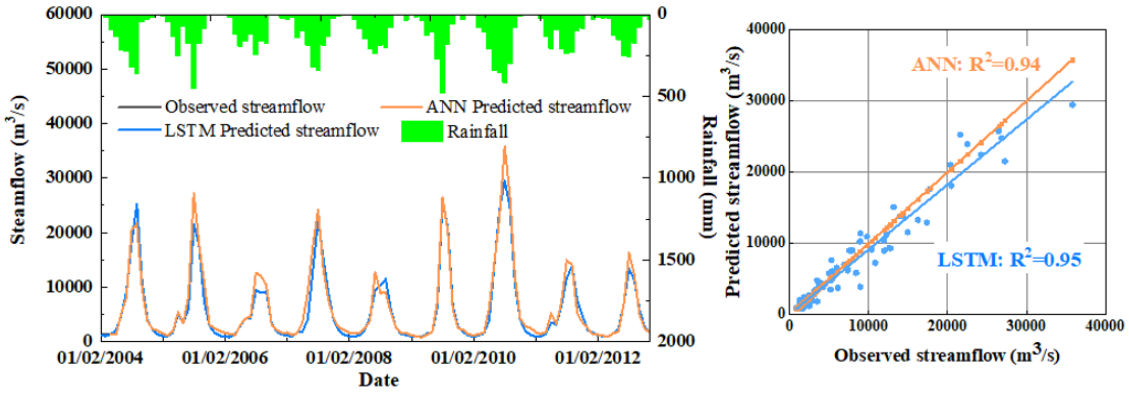


Fig. 4.10. Comparison of forecasting and observed monthly streamflow using the ANN and LSTM models with the optimal parameters in Table C.1.

4.4.2.3 Multi-month-ahead streamflow forecasting

The ANN and LSTM models have been further applied to multi-month-ahead streamflow forecasting. Fig. 4.11 gives the comparative plots of the forecasting results obtained from the ANN and LSTM models at the lead-time = 1 month, 2 month, and 3 month. One observes a relatively good match between the observed and forecasting streamflow from two models in the first and second month ahead forecast. Low streamflow is accurately forecasted, and high streamflow events are also properly captured although the arrival time of forecasting streamflow peaks is slightly delayed in multi-month-ahead forecasting.

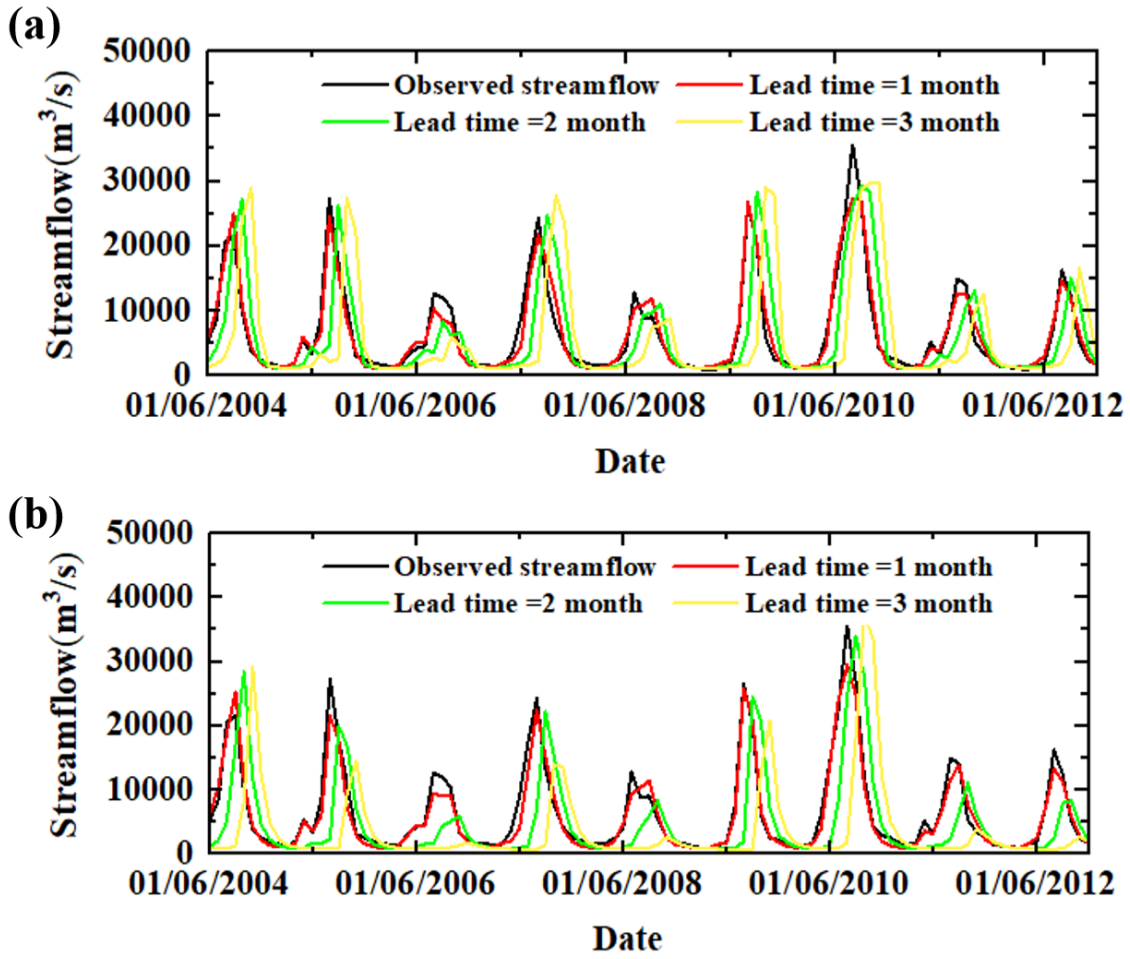


Fig. 4.11. Forecast performance at lead-times 1, 2 and 3 months: (a) with the ANN model and (b) with the LSTM model.

Again, four indicators are used to evaluate the multi-month-ahead forecast performance during the validation period (from April of 2004 to December of 2014). The results from these indicators for the forecast performance at the lead-times of 1-12 months are illustrated in Fig. 4.12. It is seen that as the lead-time length increases from 1 to 12 months, the RMSE and MAE of results in ANN modelling increase approximately by 499% and 457%, respectively, while in LSTM modelling, approximately by 442% and 425%, respectively. Bearing in mind that NSE ≤ 0 indicates the model forecasts are unreliable, It can be seen that the LSTM model can perform accurately streamflow forecast only for a one-month-lead period (afterwards, NSE ≤ 0), while the ANN model is able to forecasting streamflow up to 2 months ahead (Fig. 4.12(b)). The CC of forecasting streamflow

in both the ANN and LSTM models rapidly decreases from 0.97 to < 0.20 for the first 3-month predictive period. Both model performance and forecast reliability is diminished as the lead-time increases as shown in Fig. 4.12. The main reason for this could be the autocorrelation, which decreases in time series, making time series much less predictable with rising the lead-time (Liu et al., 2014).

It is noticed that the LSTM gradually fails to capture flow peaks in some occasions in the third month ahead forecast. In contrast to its good performance in long lead-time daily forecasting, the LSTM model fails in multi-month-ahead forecasting. The reason for this is that in this case study, the number of observed monthly datasets available is much smaller than that of daily datasets. For accurate prediction in LSTM, a large number of datasets is required for learning the long-term dependencies between the input and output datasets during the training process. This finding is similar to that of Kratzert et al. (2018), where the authors found that the data intensive nature of the LSTMs was a potential barrier for applying them in data-scarce problems.

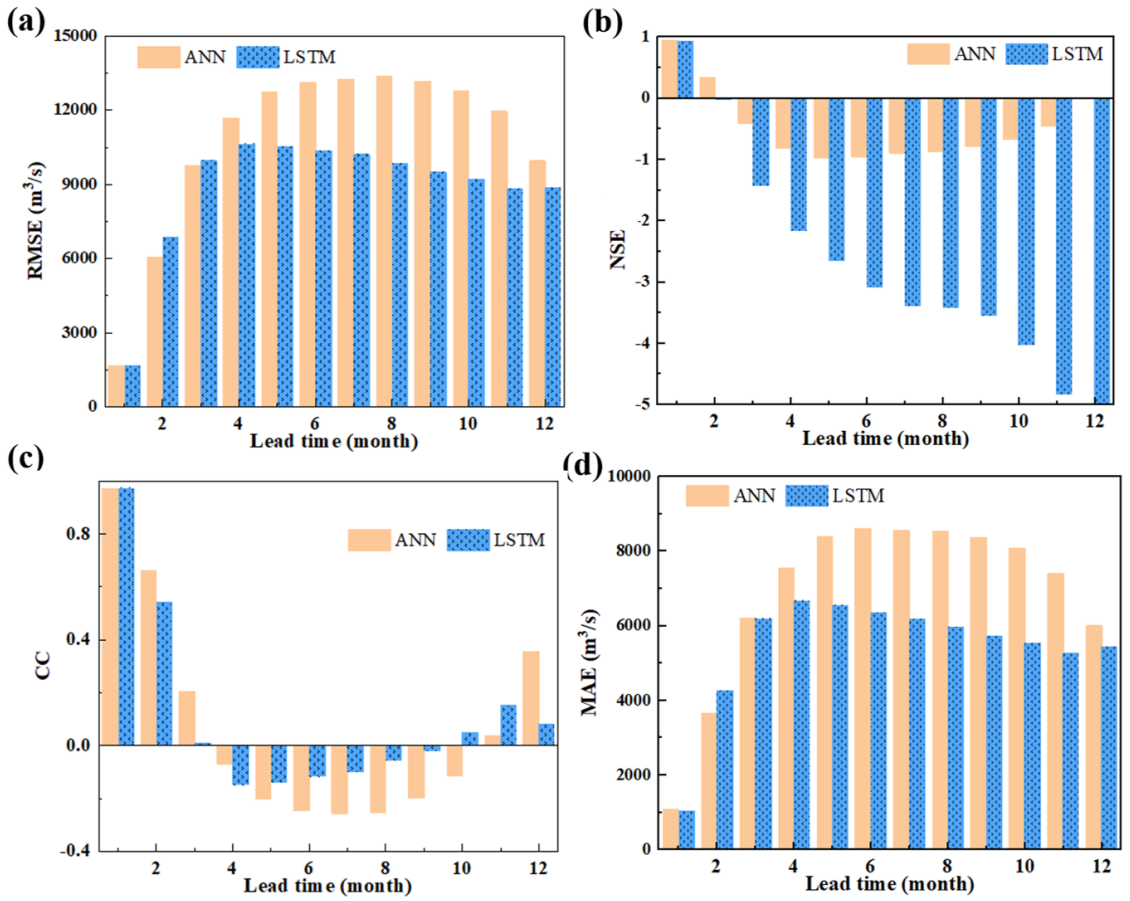


Fig. 4.12. Forecast performance with the ANN and LSTM models at lead-times of 1-12 months: (a) RMSE, (b) NSE, (c) CC and (d) MAE.

4.5 Conclusion

In this study, a recursive forecasting framework has been developed, to explore the long lead-time forecasting capabilities of the ANN and LSTM models at daily and monthly scales. The impact of the selection of parameters (the length of time lag and the number of maximum epochs) on model performance is also explored in ANN and LSTM modelling. The proposed models have been applied to a realistic case study, the Nan River Basin (NRB), as one of major subbasins of the Chao Phraya River Basin (CPRB) in the heart of Thailand. The main findings are listed as follows:

(1) In daily streamflow forecasting, the LSTM model outperforms the ANN model up

to 20 days ahead. The forecasting streamflow achieves a relatively good agreement with the observed streamflow, especially for the streamflow peak.

(2) In monthly streamflow forecasting, the situation becomes reverse in the LSTM model, where the forecasts of the ANN model turn out to be better with the increase of lead-time up to 12 months ahead. High streamflow events could be properly captured by the ANN model, except for a slight flow peak delayed.

(3) The length of time lag and the number of epochs exert great impacts on model forecasting performance at daily and monthly scales. This indicates that parameter optimization at different time scales could be effective to enhance the long lead-time forecasting accuracy of the machine learning models in the short and long terms.

Overall, our study shows that the LSTM method is superior to the ANN model in the daily streamflow forecasting for a long lead-time. This could provide an insight to water resource managers to infer daily river discharge ahead in realistic hydrological applications. While for monthly streamflow forecast, the forecasting accuracy of both models is still restricted over two or three months ahead. This could be a subject of future research efforts. Providing additional climatic factors into the presented models such as temperature and humidity, or developing hybrid models such as machine learning methods combined with data assimilation, would improve model accuracy in monthly streamflow forecast for a long lead-time. In the following chapters, the machine learning with multiple inputs and combined with data assimilation are going to be presented.

CHAPTER 5

A real-time flooding forecasting using deep convolutional GAN and application to Denmark

Real-time flood forecasting is crucial for supporting emergency responses to inundation-prone regions. Due to uncertainties in the future (e.g. meteorological conditions and model parameter inputs), it is challenging to make accurate forecasts of spatiotemporal floods. In this work, a real-time predictive deep convolutional generative adversarial network (DCGAN) is developed for flooding forecasting. The proposed methodology consists of a two-stage process: 1) dynamic flow learning and 2) real-time forecasting. In dynamic flow learning, the deep convolutional neural networks are trained to capture the underlying flow patterns of spatiotemporal flow fields. In real-time forecasting, the DCGAN adopts a cascade predictive procedure. The last one-timestep-ahead forecast from the DCGAN can act as a new input for the next timestep-ahead forecast, which forms a long lead-time forecast in a recursive way. The model capability is assessed using a 100-year return period extreme flood event occurred in Greve, Denmark. Results indicate that the predictive fluid flows from the DCGAN and the high fidelity model are in a good agreement (the correlation coefficient $\geq 97\%$ *and the mean absolute error* $\leq 0.008m$*) for*

a lead-900 timestep forecast. This is an important step towards real-time flow forecasting although further evaluation of the DCGAN performance is required in complex realistic cases in the future.

This chapter is derived from and expands upon Cheng et al. (2021).

5.1 Introduction

In this work, a deep convolutional generative adversarial network (DCGAN) is proposed to tackle challenges presented by the long lead-time flooding forecasting. GANs are capable of processing high-dimensional datasets and producing high-resolution images (Liu et al., 2019; Ledig et al., 2017; Choi et al., 2018; Isola et al., 2017). Recent progress in deep learning has shown that GAN-based techniques are promising for spatiotemporal-based applications such as trajectory prediction, events generation and time-series data imputation (Gao et al., 2022). Cheng et al. (2020c) first introduced DCGAN for predicting spatiotemporal flow distributions. This work mainly focused on simulations of parameterized nonlinear fluid flows for a training period. It was shown that the complex features of flow dynamics were captured by the adversarial networks.

In this work, further implementation of the DCGAN has been presented for long lead-time flood forecasting in the future (beyond the trained period). The capability of the DCGAN has been evaluated for a realistic flooding event in Denmark. The main features of the predictive DCGAN are:

- An efficient DCGAN architecture, consisting of deep convolutional neural networks, learns rich distributions from the high-dimensional spatiotemporal datasets and yields a low computation cost for predicting spatiotemporal floods.
- A cascade predictive implementation of the DCGAN, provides accurate predictive results for a long lead-time, which highlights the potential in complex and mas-

sive applications. During dynamic flow learning, the spatiotemporal flow features of historical high-dimensional flow fields are deeply exploited by the deep convolutional neural networks, thus capturing the underlying flow patterns. During real-time forecasting, the forecast process follows a cascade manner, which means that the last one-timestep-ahead predictive solution is used as a new input for the next step forecast.

- Having the capability of exploring the flood propagation process over a urban area.

The chapter is organized as follows. The governing equations for flood forecast problems are briefly formulated in section 5.2. The real-time DCGAN for flood forecasting is introduced in detail in section 5.3. Section 5.4 demonstrates the predictive performance of the DCGAN using flood events in Greve, Denmark as a realistic test case. Finally in section 5.5, conclusions are presented.

5.2 Spatiotemporal flood forecasting problem

In traditional computational dynamic models, the parameterized set of partial differential equations (for example, the Navier Stokes or shallow water equations) are used for resolving flood dynamics problems as

$$R(U, \mu, x, t) = S, \tag{5.1}$$

where R is the nonlinear model operator, U represents the variables to be predicted (for example, velocity, pressure, water depth etc.), x is the spatial coordinate system, μ denotes the uncertainty parameters, boundary and initial conditions, t is the time and S denotes the source term.

The spatiotemporal flood forecast process, is a way to learn from the historical dataset $U \in \Re^{t_N \times p_r}$ and predict the future flow solution $U \in \Re^{t_{N+\tau} \times p_r}$ on the spatial space (where

Table 5.1: The definition of variables used in this work.

Variable	Definition
p_r	Total number of nodes in a grid in the computation domain Ω.
X	Locations of nodes in the computation domain Ω, $X = (X_1, \ldots, X_{p_z}, \ldots, X_{p_r})$.
τ	Lead time length of the predictive variable solutions.
K	Previous time length of the historical variable solutions.
U	Variable solutions at temporal and spatial spaces.
$\widetilde{U}$	Variable forecast during the lead time period $(t_N, t_{N+\tau})$.
U_g	Variable solutions obtained by the generator during the training process.
U_d	Real dataset used to distinguish it from the generated data U_g during the training process.
I	Input dataset, containing previous length-K time series of variable solutions.
O	Output dataset, containing the one-timestep-ahead predictive variable solution.

τ is the lead timestep, N represents the number of timesteps, p_r is the total number of nodes in the computation domain Ω). Given the solutions/dataset at the previous length-K time series of variable solutions $\{U_{t_i} | i = 1, ..., K\}$, the forecast of length-τ time series of variable solutions $\{U_{t_i} | i = 1, ..., \tau\}$ can be expressed mathematically as:

$$U_{t_{n+1}}, ..., U_{t_{n+\tau}} = P\left(U_{t_{n-K+1}}, ..., U_{t_n}, \mu\right), \tag{5.2}$$

where P represents a dynamic forecast model for the fluid flows expressed in Eq.(5.1), U_{t_n} is the variable spatial solution at time step t_n, μ as the conditional input is used to improve the forecast accuracy. In this work, the dynamic predictive model P will be represented by a real-time predictive DCGAN, which will be introduced in section 5.3.

The variables used in the following sections are defined in Table 5.1.

5.3 Deep convolutional GAN for real-time flood forecasting

The main objective of this work is to develop a real-time predictive DCGAN tool for spatiotemporal flood modelling. To achieve this objective, a two-stage process is adopted

(shown in Fig. 5.1):

- *Dynamic flow learning*. During the training process, the dynamic model P in Eq.(5.2) is represented by the DCGAN which combines the generator G and discriminator D modules. Given the training input-output datasets (I^{tr}, O^{tr}), the spatiotemporal flow features and dynamic patterns of historical floods are deeply exploited by the DCGAN. Once the training process is finished, an optimal dynamic system P is obtained aiming to learn the flow features in time and space (the details of the training process, see section 5.3.1).

- *Real-time forecasting*. The real-time forecast adopts a cascade manner for predicting the flow evolution over time. The last one-timestep-ahead forecast from the DCGAN can act as a new input for the one-timestep-ahead forecast, which forms a long lead-time forecast in a recursive way (the details of the forecasting process, see section 5.3.2).

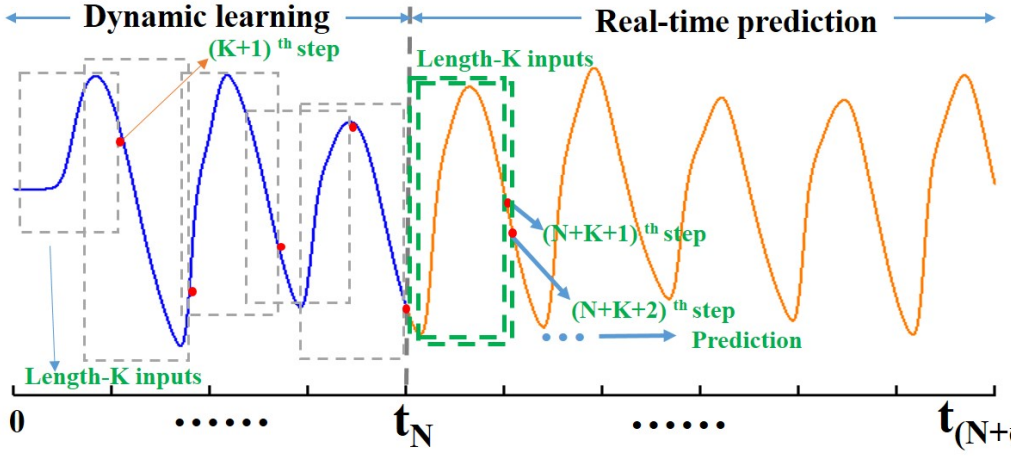


Fig. 5.1. The schematic diagram of the dynamic flow learning and forecast processes (taking the wave as an example)

5.3.1 Dynamic flow learning of DCGAN

A DCGAN consists of two modules, a generator G and a discriminator D (Goodfellow et al., 2014). As shown in Fig. 5.2, the generator is responsible for generating spatiotemporal distributions of data U_g from a random distribution I. The discriminator is responsible for distinguishing the real data U_d from the generated data U_g by making a binary decision. Therefore, the dynamic model P is represented by the two modules in the DCGAN ($P = (G, D)$).

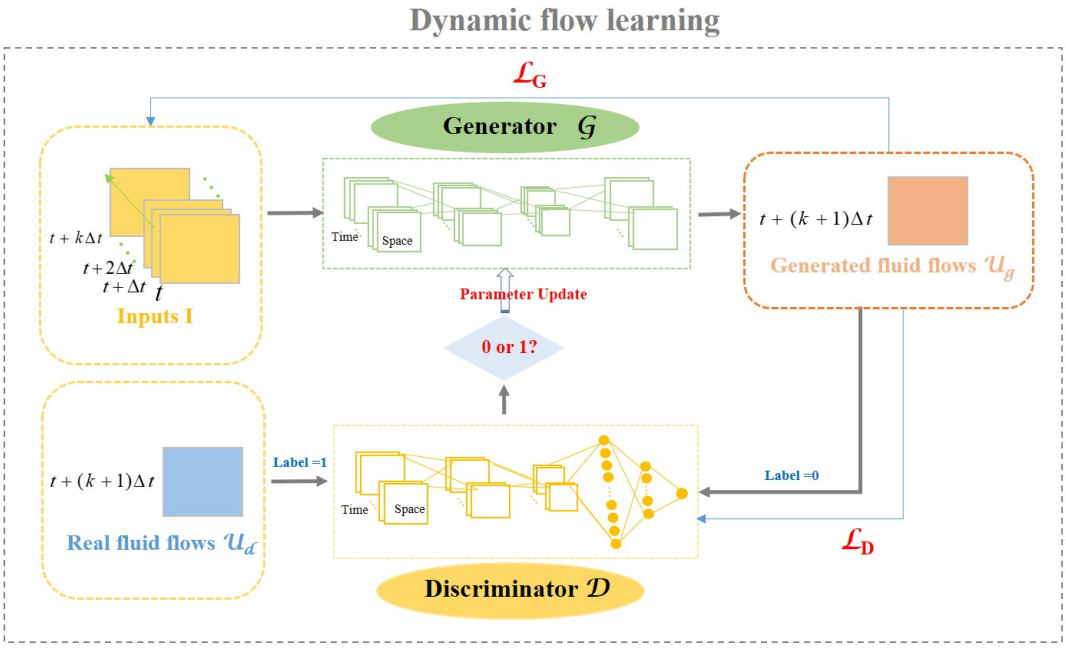


Fig. 5.2. Illustration of the dynamic flow learning system: a deep convolutional GAN (DCGAN), which consists of a generator and a discriminator. The generator and discriminator are composed of convolutional layers and full-connected layers. The combination of the loss functions in two modules forms the adversarial training.

Generator network: I use a deep neural network F_g that takes the inputs I with length-K timesteps $\{U_{t_{n-K+1}}(X), ..., U_{t_{n-i+1}}(X), ..., U_{t_n}(X), \mu\}$. As shown in Fig. 5.2, the generator consists of deep convolutional layers, and each hidden layer is followed by an activation function (e.g, the rectifier non-linearity (ReLU), the hyperbolic tangent function, and the sigmoid function) (Zhong et al., 2019). The convolutional layers are able to capture

the spatial and temporal dependencies in a flood field by using filters. Filters often extract certain types of features from the inputs, and can represent spatially localized interactions (Weyn et al., 2020). The batch normalization (Ioffe and Szegedy, 2015) is used which stabilizes learning by normalizing the output values retain a mean of 0 and standard deviation of 1. During the training process shown in Fig. 5.2, the parameters ψ (weights and bias in the neural network) in the generator are updated to minimize the loss function $\mathcal{L}_G$:

$$\mathcal{L}_G = E_{I\sim p_I(I)}[\log(1 - D(G(I)))], \tag{5.3}$$

where $p_I(I)$ is a prior distribution for the inputs I.

Discriminator network: The discriminator takes as its inputs a generated forecast U_g and a targeted output (i.e. real data) O ($O = U_d = U_{t_{n+1}}(X)$). The generated and targeted datasets in the discriminator are processed using deep convolutional layers and full-connected layers, and each hidden layer is also followed by an activation function and batch normalization similar to the architecture used in the generator. The output layer is densely connected to the final hidden layer which yields a number between 0 and 1. The output $D(U_d) = 1$ means that the targeted outputs U_d are accepted by the discriminator while $D(G(I)) = 0$ means that the generated forecasts U_g are rejected.

In the model training process, the parameters ζ (weights and bias in the neural network) in the discriminator are updated by maximizing $\mathcal{L}_D$ as:

$$\mathcal{L}_D = E_{U_d\sim p_{data}(U_d)}[\log D(U_d)] + E_{I\sim p_I(I)}[\log(1 - D(G(I)))], \tag{5.4}$$

where $p_{data}(U_d)$ is the probability data distribution for the targeted outputs U_d.

Combining the loss functions $\mathcal{L}_G$ in Eq.(5.3) and $\mathcal{L}_D$ in Eq.(5.4), the objective function for the DCGAN can be expressed as:

$$\min_G \max_D \mathcal{L}(G, D) = E_{U_d\sim p_{data}(U_d)}[\log D(U_d)] + E_{I\sim p_I(I)}[\log(1 - D(G(I)))]. \tag{5.5}$$

In practice, gradient-based methods, for example the adaptive Moment Estimation (Adam) optimizer and RMSprop (Tieleman and Hinton, 2012), are used to optimize the loss function $\mathcal{L}(G, D)$. The model architecture and parameters of the DCGAN are described in detailed in Table 5.2 applied to flood prediction problems in this work. After completing the dynamic learning process of the DCGAN, the parameters in the generator and discriminator are optimized. The flood fields U in the lead-time steps can then be predicted by the DCGAN.

Table 5.2: Architecture and model parameter setting of the DCGAN

Module	Layer
Generator G	Input
	Conv2D, leaky ReLU ($\alpha = 0.2$)
	Conv2D, leaky ReLU ($\alpha = 0.2$)
	Conv2D, leaky ReLU ($\alpha = 0.2$)
	Conv2D, leaky ReLU ($\alpha = 0.2$)
	Output, Tanh
Discriminator D	Input
	Conv2D, leaky ReLU ($\alpha = 0.2$)
	Conv2D, leaky ReLU ($\alpha = 0.2$)
	Conv2D, leaky ReLU ($\alpha = 0.2$)
	Conv2D, leaky ReLU ($\alpha = 0.2$)
	Full connected, leaky ReLU ($\alpha = 0.2$)
	Output (1), Sigmoid
Parameter	Setting
filter size	3×3
batch normalization	momentum=0.8
batch size	10
optimizer	Adam
learning rate; momentum	$2{\times}10^{-4}$, 0.5
epochs	500

5.3.2 Real-time forecast of DCGAN

Once the DCGAN used for representing the dynamic model P is generated during the dynamic learning process, the one-timestep-ahead forecast of fluid field $\widetilde{U}$ can be obtained.

For the one-timestep-ahead flood forecast in Eq.(5.2) can be rewritten as:

$$U_{t_{n+1}}(X) = G\left(U_{t_{n-K+1}}(X), ..., U_{t_n}(X), \mu\right). \tag{5.6}$$

To further achieve the long lead-time forecast of floods, a cascading deployment is used as shown in Fig. 5.3. It starts with the one-timestep-ahead forecast $\{\widetilde{U}_{t_n}(X)|n = K + 1\}$, given the inputs $I \in \Re^{t_K \times p_r}$ containing $K-$ timesteps historical data $\{U_{t_n}(X)|n = 1, .., K\}$. Afterwards, in the following one-timestep-ahead forecast $\{\widetilde{U}_{t_n}(X)|n = K + 2, \dots\}$, the previous predicted solution $\{\widetilde{U}_{t_n}(X)|n = K + 1, \dots\}$ will be included in the inputs. Given the previous length-K timesteps, the forecast of length-τ timesteps in Eq. (5.2) or Eq.(5.6) can be rewritten as:

$$\widetilde{U}_{t_{n+i}}(X) = \begin{cases} G\left(U_{t_{n-k+i}}(X), ..., U_{t_n}(X), \widetilde{U}_{t_{n+1}}(X), ..., \widetilde{U}_{t_{n+i-1}}(X), \mu\right), i \leq K \\ G\left(\widetilde{U}_{t_{n-K+i}}(X), ..., \widetilde{U}_{t_{n+i-2}}(X), \widetilde{U}_{t_{n+i-1}}(X), \mu\right), i > K \end{cases} \tag{5.7}$$

where $i = (1, ..., \tau)$, and the historical flow datasets $(U_{t_{n-k+i}}, ..., U_{t_n}(X))$ are used when $i < K$, while all the inputs originate from the predicted solution at the previous timesteps when $i > K$ (beyond the training period).

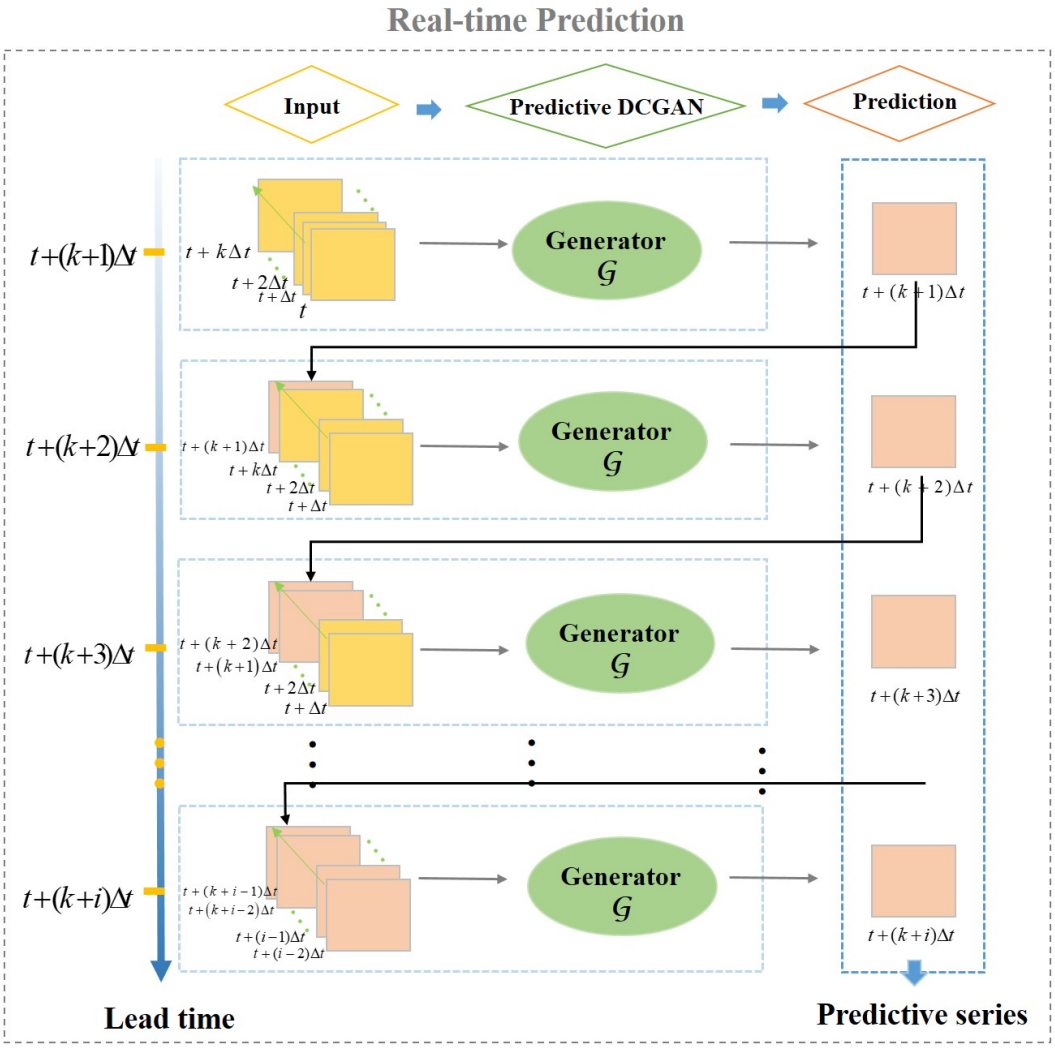


Fig. 5.3. Illustration of the real-time forecast of the DCGAN. The forecast of flood fields employs a cascade manner, i.e. the last one-timestep-ahead forecast from the DCGAN can act as a new input for the next step-ahead forecast, which forms a long lead-time forecast in a recursive way.

5.3.3 Model evaluation

To assess to performance of the DCGAN model developed in this study, different statistical evaluations including the root mean squared error (RMSE), the Nash-Sutcliffe efficiency coefficient (NSE), the coefficient of correlation (CC), the mean absolute error (MAE), the mean magnitude of relative error (MMRE), Nash-Sutcliffe efficiency ratio (NSER)), and the percentage of accuracy-precision (PAP) (Heidaryan, 2019) are used.

The Nash-Sutcliffe efficiency coefficient and percentage of accuracy-precision are defined as

$$NSE = 1 - \left[\frac{\sum_{i=1}^{n} \left(U_g^i - U_d^i\right)^2}{\sum_{i=1}^{n} \left(U_d^i - \overline{U_d^i}\right)^2} \right], \tag{5.8}$$

$$PAP = 100 \left(1 - \frac{\sqrt{2}}{2} \sqrt{ \left(\frac{1}{n} \sum_{i=1}^{n} \left| \frac{U_g^i - U_d^i}{U_d^i} \right| \right)^2 + \left[\frac{\sum_{i=1}^{n} \left(U_g^i - U_d^i\right)^2}{\sum_{i=1}^{n} \left(U_d^i - \overline{U_d^i}\right)^2} \right]^2 } \right), \tag{5.9}$$

where U_g^i and U_d^i are the i^{th} forecasting and observed flows respectively, and $\overline{U_d}$ is the average observed flow. The MMRE and NSER vary between 0 and $+\infty$, with acceptable levels of performance between 0 and 1. The PAP ranges from 0 to 100, while zero indicates both poor accuracy and precision (Heidaryan, 2019).

The forecasts are also analyzed in terms of their distributions by using the Q-Q plot, which is a visual method comparing the values of observations in the predictive distribution on the x-axis and the uniform distribution on the y-axis to examine the reliability of the forecasting distribution. If the plot follows a 1:1 line, it means that the flood forecasts are perfectly reliable. The advantage of Q-Q plot is that it can illustrate different aspects of forecasts including under- or over-estimated predictive uncertainty (Xu et al., 2017).

5.4 Application and results

5.4.1 Study area

To assess the long lead-time predictive capability of the DCGAN model, it has been applied to a flood event in the northeastern part of Greve, Denmark. As shown in Fig. 5.4, Greve is a coastal city connected to the Baltic Sea, with elevations between 12.76 to

$-2.1m$ in the study domain area. It is potentially affected by flood disasters induced by extreme sea-level events along its coast. For example, an extreme historical flood in Køge Bay occurred in 13th October 1760, with a maximum water level of $3.7m$ (Berbel Roman, 2014). Considering the impact of climate change in the next 100 years, future climate change conditions should be taken into account to estimate the future extreme sea-level (Hu et al., 2019b). Therefore, a 100-yr return period extreme water level event (2100 upper sea-level flood event in Berbel Roman (2014)) was used in this work, to predict the expected changes in future sea surges (as shown in Fig. 5.5).

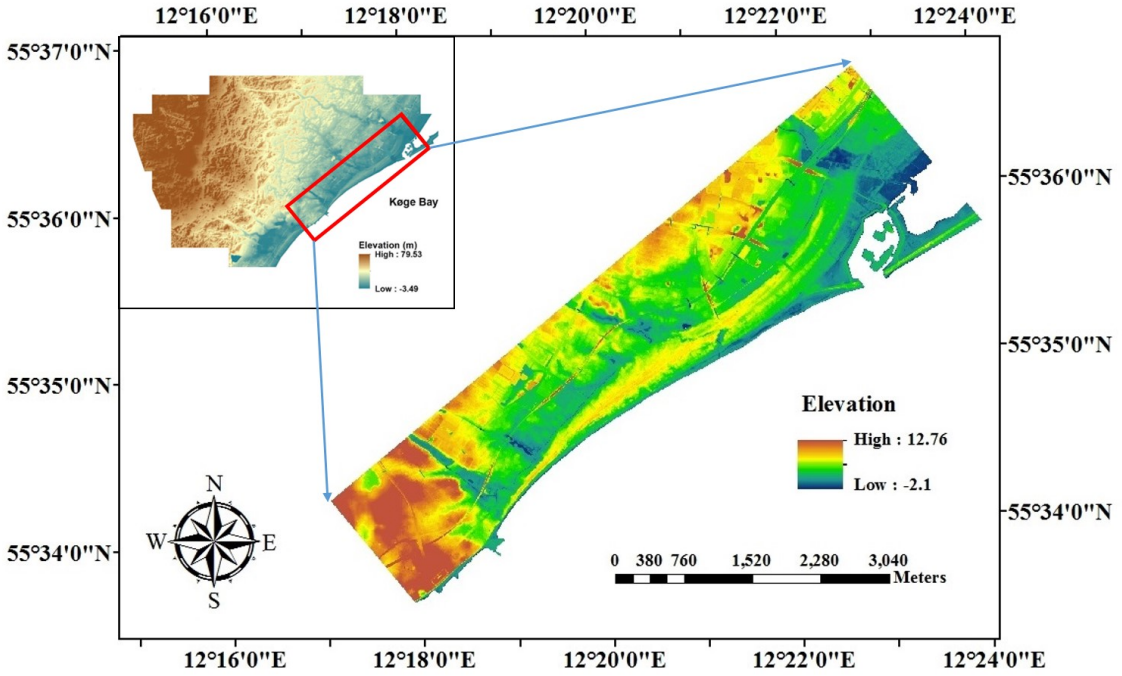


Fig. 5.4. Study area in Greve, Municipality of Denmark (left side see Hu et al. (2019b))

.

In this study, the training datasets and reference solutions (referred as the real/original water depths) are obtained from the simulation results by running the high fidelity model - MIKE 3 FM, which is a software tool for modelling unsteady three-dimensional free surface flows (MIKE21 and MIKE3 Flow Model, 2009). As shown in Fig. 5.5, the model

domain representing the coastal area in Greve (Denmark) has a size of $2.3 \times 7.5km$. The unstructured mesh consists of a total of 39290 points. An extreme seawater level event with a maximum value $3.08m$ is used as the input boundary condition along the coastal line. No normal flow boundary conditions were applied to other boundaries. Courant–Friedrichs–Lewy (CFL) number is 0.8 and drying and wetting depths are $0.01m$ and $0.03m$ respectively. The total simulation time was $24h$ with a timestep $\Delta t = 1s$. The simulations started from a 'static' state. To illustrate the DCGAN capability for uncertainty analysis and forecast, section 5.4.2 performs a sensitivity analysis by the DCGAN with respect to different incoming waves along the coastline, while section 5.4.3 demonstrates the forecasting capability of the DCGAN for a long lead-time.

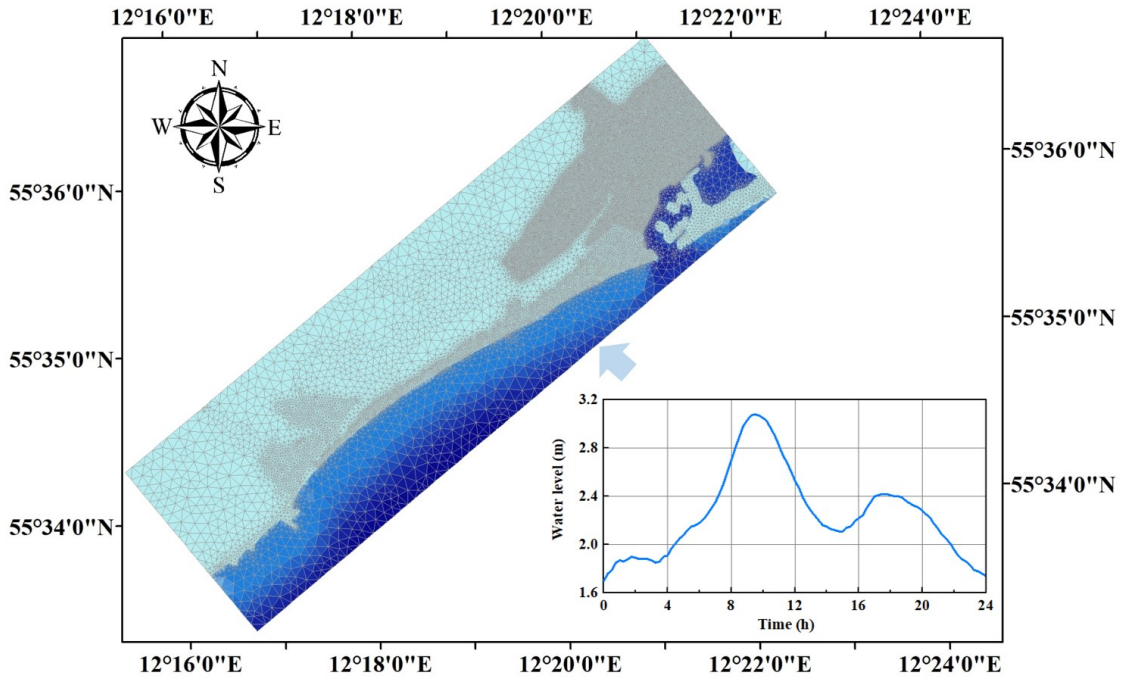


Fig. 5.5. Unstructured mesh of study area and extreme sea water level along the coastal line, where the mesh is composed by 25203 elements, 38437 faces, 13235 nodes and 2 sections (triangular and rectangular).

5.4.2 Uncertainty quantification

In this section, to evaluate model uncertainty in spatiotemporal flood modelling, I consider different flood events of observed and predicted floods using different incoming waves. As shown in Fig. 5.5, 15 types of input parameters (incoming waves) range from -30% to $+30\%$ of the water level in 2100 upper sea-level flood event are generated using a Gaussian distribution $N(\bar{\mu}, \sigma^2)$. For example, the mean $\bar{\mu}$ is $2.18\,m$ and the variance of the distribution σ^2 is $0.047\,m$ at $t = 6h$ in Fig. 5.6(b). At $t = 12.5h$ in Fig. 5.6(c), the water levels in 15 types of incoming waves are obtained from the Gaussian distribution between $1.632\,m$ to $3.147\,m$.

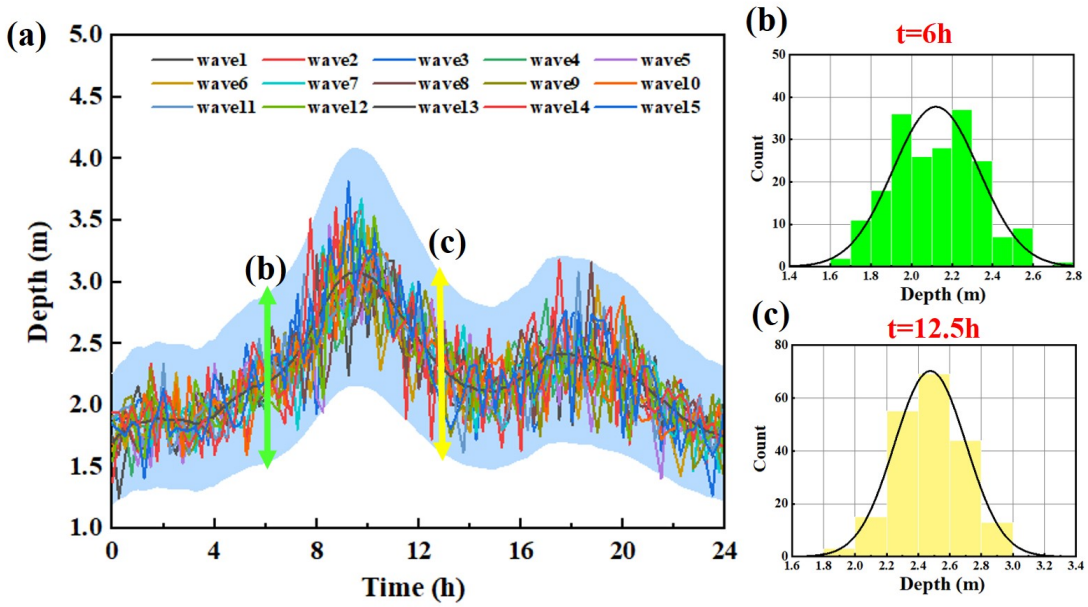


Fig. 5.6. Different types of incoming waves obtained from the Gaussian distribution of the extreme sea water level (the solid line representing the 2100 upper sea flood level in Fig. 5.5) from 0 to $24h$. (a) represents 15 types of incoming waves, and (b) and (c) are the Gaussian distribution of water depths at $t = 6h$ and $12.5h$, respectively.

Given a set of incoming waves (in Fig. 5.6), the corresponding solution snapshots U (a total of 145 snapshots sampled at an equal time interval of $\Delta t = 10min$ from 0 to $24h$, $N = 145, p_r = 39290$) are obtained by running the high fidelity model. In model training and validation processes, a predictive DCGAN is constructed by using the training and

validated input-output pairs. I use the first 12 types of incoming waves as the training inputs I^{tr}, while the corresponding training outputs O^{tr} are the collection of the solution snapshots for each type of incoming waves. During the online predictive process, given a set of new inputs I^{pre}, the predictive analysis is undertaken by the DCGAN. Here the inputs I^{pre} are the last three types of incoming waves (wave 13, 14 and 15) shown in Fig. 5.6.

For a given input $\mu \in I^{pre}$ (wave 13), Fig. 5.7 presents the comparison of water depth obtained from the DCGAN and the original high fidelity model at $t = 5, 9, 14, 18h$. It is worth noting that the predicted flow fields from the DCGAN attain a good agreement to those from the high fidelity model at different time levels. The basic flow physics is clearly captured, and the shape of the water depth is predicted quite well, especially in urban areas (as shown in the rectangles in Fig. 5.7). In addition, a relatively good correspondence is observed between the predicted depth of DCGAN and those of the original high fidelity model: the correlation coefficient of results at time levels $t = 5, 9, 14, 18h$ achieves values beyond 97% shown in Fig. 5.8. Table 5.3 further summarizes the statistical metrics for model performance. It can be seen that the values of MMRE and NSER at different time levels are below 0.25, and the PAP values are beyond 82.

The results indicate that the DCGAN is able to obtain accurate simulations of spatiotemporal distribution during a fixed simulation period for given varied boundary conditions. The spatiotemporal flow features have been faithfully represented as the flow evolves, and the magnitude of predicted water depth during the simulation period is in a good agreement with the reference water depth. The promising results obtained show that the DCGAN is able to detect the underlying functional mapping through pattern extraction, confirming results obtained in the work of (Leinonen et al., 2019; Zhong et al., 2019; Li et al., 2018). Leinonen et al. (2019) suggested that the DCGAN has potential to solve probability distribution problems in which the random fields have complex spatial structures.

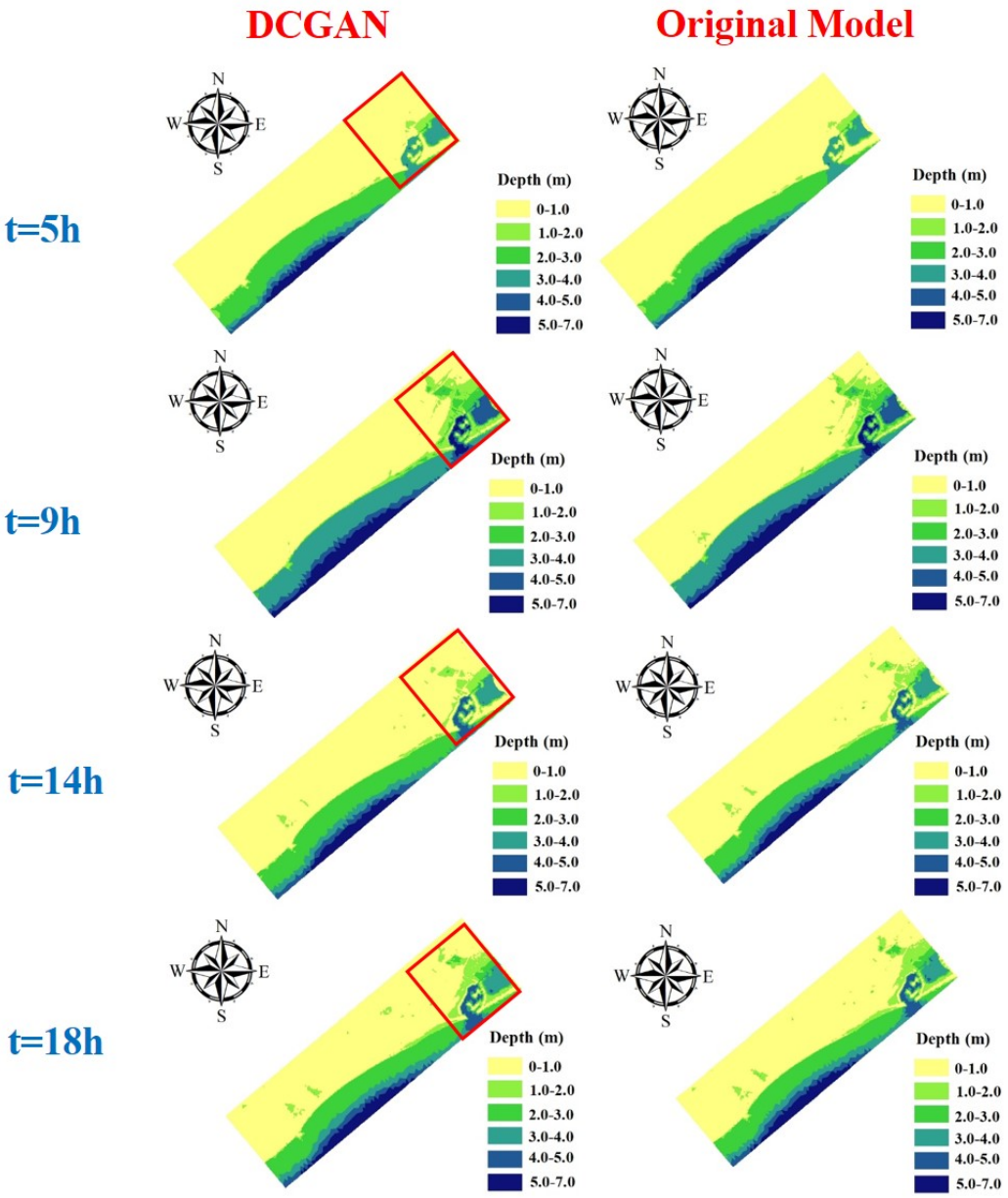


Fig. 5.7. Comparison of the spatial distribution of predictive water depth obtained from the DCGAN and the original high fidelity model at $t = 5, 8, 14, 18h$. Here the rectangles regions are the urban cities with dense buildings.

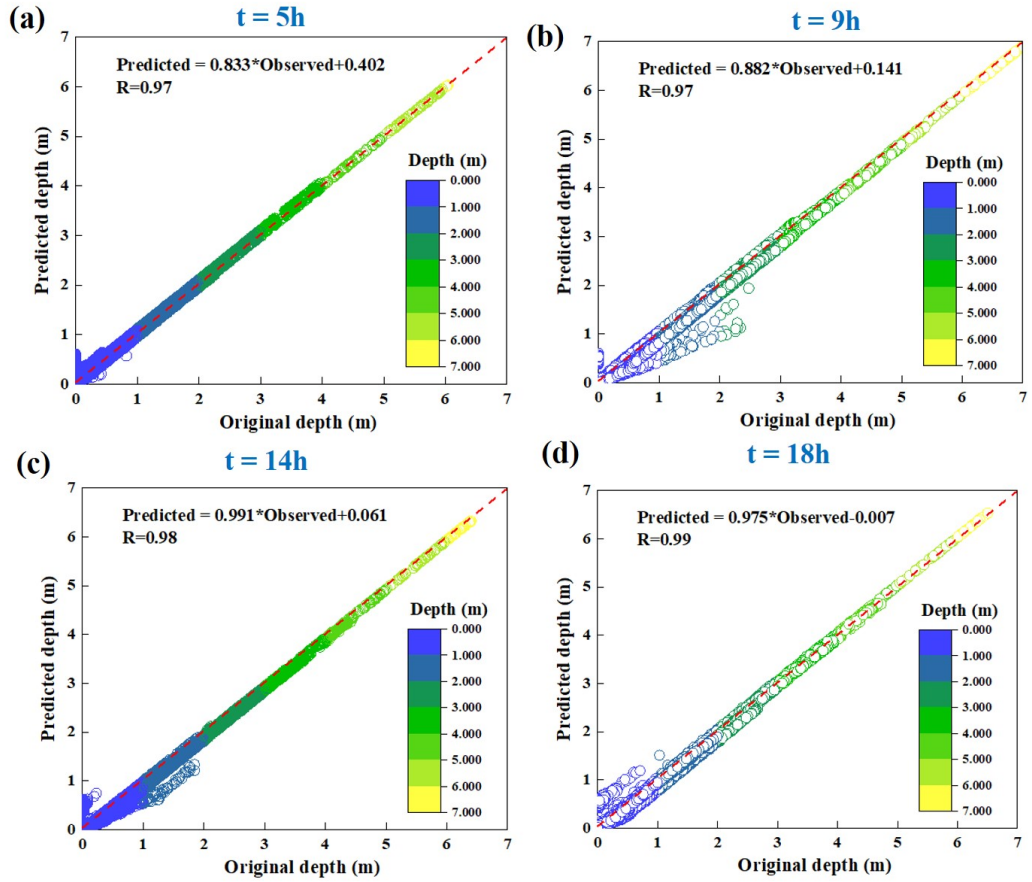


Fig. 5.8. The correlation coefficients of water depth solutions between the DCGAN and the original high fidelity model at different time levels: (a) $t = 5h$, (b) $t = 8h$, (c) $t = 14h$, and (d) $t = 18h$.

Table 5.3: Statistical metrics for water depth solutions between the DCGAN and the original high fidelity model at $t = 5, 8, 14, 18h$.

Time	CC	MMRE	NSER	PAP
t=5h	0.97	0.06	0.24	82.5
t=8h	0.97	0.14	0.05	89.4
t=14h	0.98	0.09	0.04	93
t=18h	0.99	0.21	0.01	85.1

5.4.3 Real-time flooding forecast

In this section, I further test the forecasting ability of DCGAN in future events that is beyond the training time. In this case, given the incoming waves (2100 upper sea-level

flood event shown in Fig. 5.5), the corresponding solution snapshots U (a total of 2881 snapshots sampled at an equal time interval of $\Delta t = 30s$ from 0 to $24h$) are obtained by running the high fidelity model. In the study scenario, the solutions at first 1800 timesteps from $0s$ to $15h$ are selected as the training datasets, while the remaining datasets from $t_{1801} = 15.01h$ to $t_{2881} = 24h$ are prepared for one-timestep-ahead forecast (as shown in Fig. 5.1, $t_N = t_{1800}$ and $t_{N+\tau} = t_{2700}$).

To evaluate the predictive ability of the DCGAN, a comparison between the predicted water depth of the DCGAN and the high fidelity model has been undertaken. Fig. 5.9 depicts the temporal variations of the water depth at eight points within the computational domain area Ω. It can be observed that a relatively good agreement exists between the predictive and original water depths from $t = 15.02h$ to $t = 23h$. The water depths at eight detectors have the same variation tendency as the incoming waves from $t = 15.02h$ to $t = 23h$ in Fig. 5.5. The DCGAN results of the water depth at detectors P5 and P6 along the coastline are very close to those from the high fidelity model although minor differences ($< 0.005m$) between results at peaks are noticed. It can be seen that the water depths at detector P1 obtained from the DCGAN have a slight difference from that of original high fidelity model after $t = 21h$. This due to the fact that the detector P1 is located at the channel, thus having a larger water depth even though the level of incoming wave reduces. In addition, there may exist an arrival time lag at detectors if the impact of buildings is taken into account in the flooding simulations (Hu et al., 2019b). The promising results obtained show that the proposed DCGAN can predict the time series of floods reasonably well in comparison to the high fidelity model.

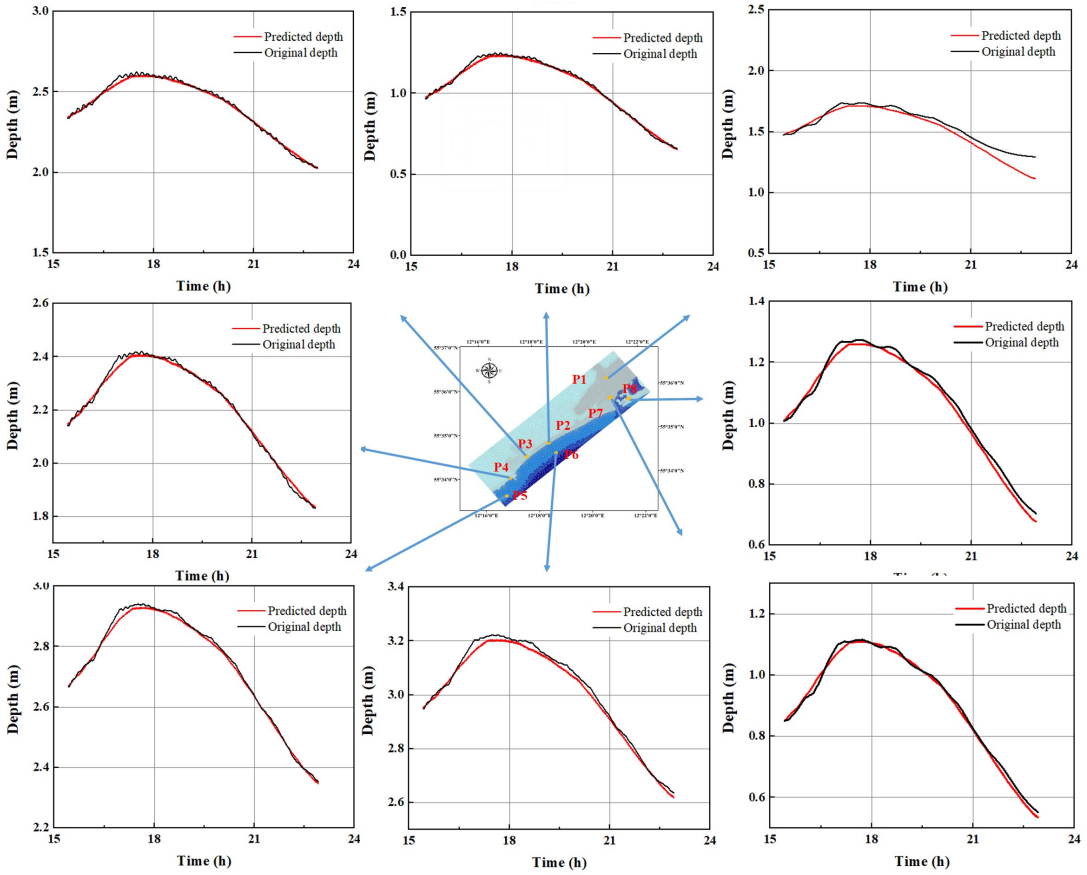


Fig. 5.9. Comparison of the temporal distribution of predictive water depth obtained from the DCGAN and the original high fidelity model at eight observed points, where the water depth from the high fidelity model is taken as the real (original) depth while the predictive depth is referred as the results from the DCGAN.

To depict the spatial distribution of nonlinear fluid fields during the forecasting period (from $15.01h$ to $23h$), the areas marked with rectangles in Fig. 5.7 are selected to show the visible changes as the flood propagates over the urban area. The spatial distribution of the predicted water depth at time $t = 15.5, 17.9, 22.8h$ is shown in Fig. 5.10. Again very little difference between the results from the DCGAN and the original high fidelity model can be visually noticed at these time steps. The underlying flow pattern is captured well by the predictive DCGAN, especially in complex topographic features, including buildings and channels. The absolute error of the water depth predicted by the DCGAN is also illustrated in Fig. 5.10, in which the high fidelity model is taken as the reference solution. It can be noticed that most of absolute errors are smaller than $0.01m$ over the whole

domain area at time steps $t = 15.5, 17.9, 22.8h$. The influence of buildings is observed that the absolute errors increase around buildings when the incoming wave rises from $t = 15.5h$ to $17.9h$. However, it relieves when the water depth declines from $t = 17.9h$ to $22.8h$.

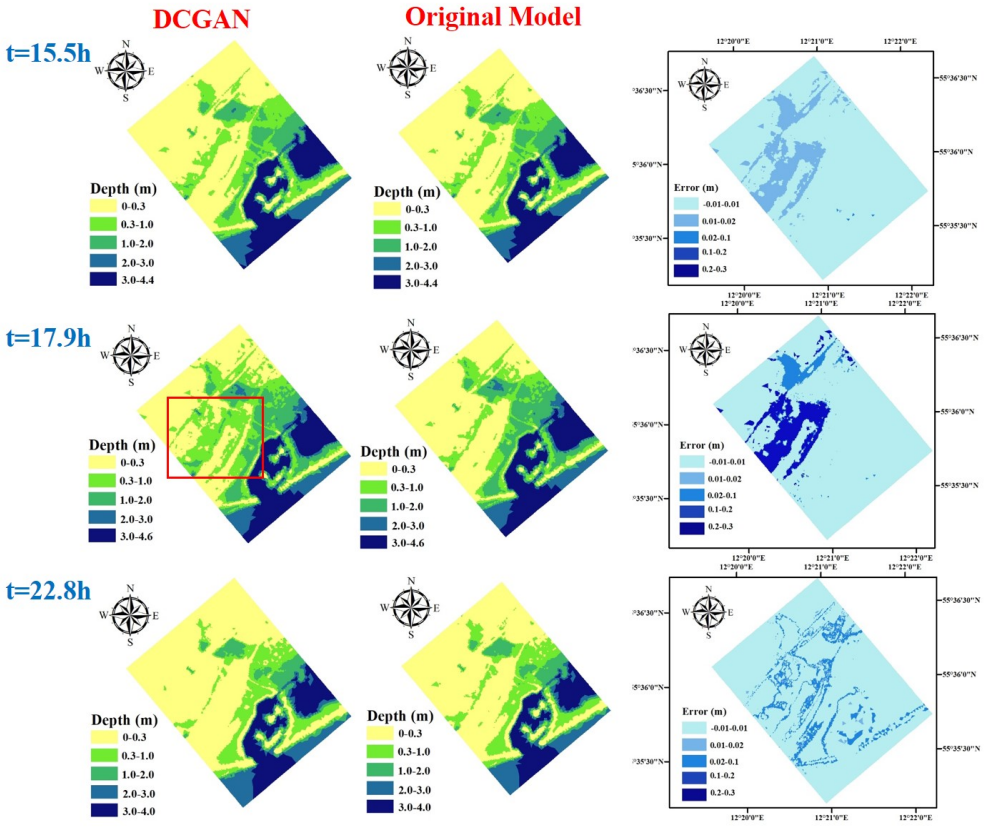


Fig. 5.10. Comparison of the spatial distribution of predictive water depth obtained from the DCGAN and the original high fidelity model at time steps $t = 15.5, 17.9, 22.8h$.

To further demonstrate the forecasting capability of the DCGAN, the error analysis is carried out by four indicators, i.e. Root Mean Squared Error (RMSE), Nash-Sutcliffe Efficiency Coefficient (NSE), Coefficient of Correlation (R), and Mean Absolute Error (MAE) for a long lead-time forecast. The corresponding evaluation of forecasting nonlinear flow results during the lead-times from $15.01h$ to $23h$ is illustrated in Fig. 5.11. It can be seen that the values of RMSE and MAE range from 0 to 0.15 and 0 to 0.08, respectively (in Fig. 5.11 (a) and (c)). It is worth to mention that values of NSE and R are beyond 0.97

and 0.98 respectively (in Fig. 5.11 (b) and (d)). The maximum values of RMSE and MAE, as well as the minimum values of NSE and R, occur when the water level of incoming wave attains its peak around $t = 18h$. This suggests the flood forecasting accuracy is highly correlated with the incoming waves, which exhibits a descending trend when the water level of the incoming wave increases. It is also noted that the forecast reliability diminishes after $t = 21h$. The reason for this is that the forecasting results have a smaller inundation as the incoming wave decays after $t = 21h$, and the water depth in most parts of study area decreases to some extent, and may even reach dry conditions. Again, it proves that the flood forecasting accuracy is sensitive to the boundary conditions, i.e. the incoming sea level (Hu et al., 2019b).

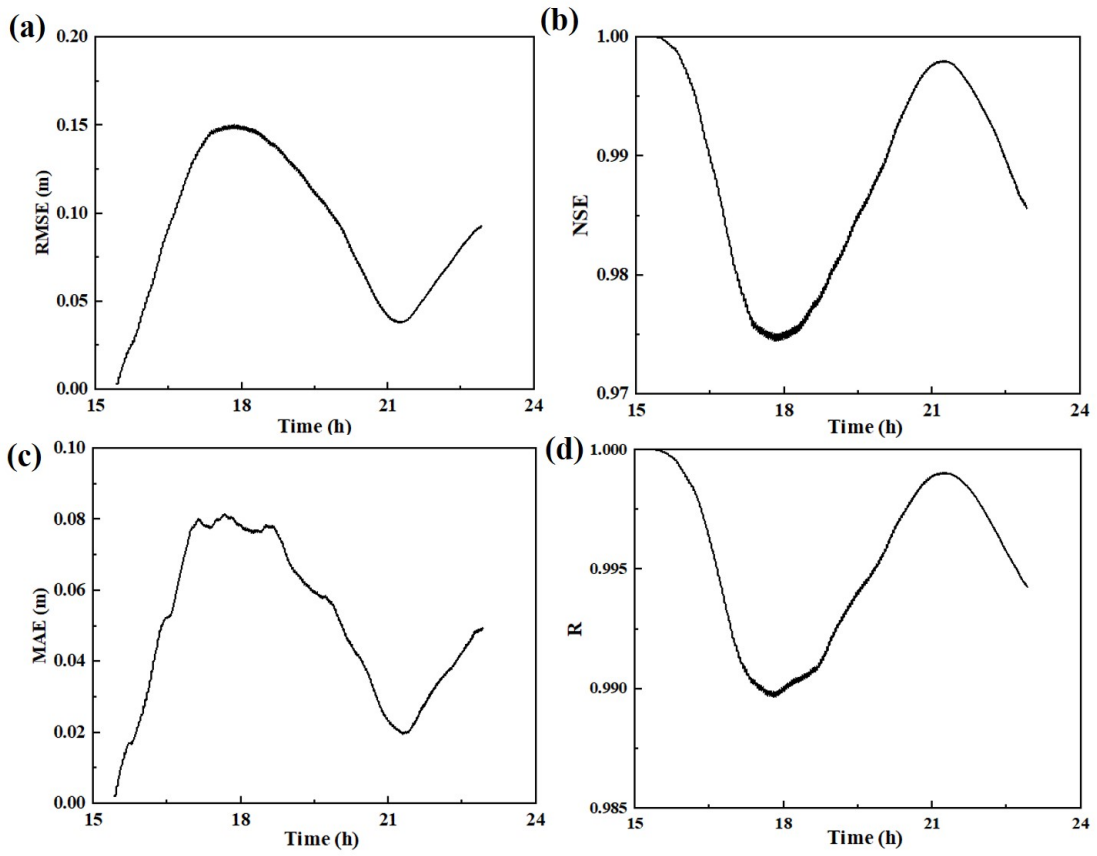


Fig. 5.11. Forecast performance with the DCGAN during the long lead-time of the computational domain area Ω: (a) RMSE, (b) NSE, (c) MAE and (d) R.

In Fig. 5.11, it can be seen that there is no obvious trend of increasing forecast error with

rising the lead-time while the statistical error analysis exhibits a close connection with incoming waves. One of the reasons for this is the inclusion of boundary conditions in the inputs at each predictive time step. The use of the boundary conditions (acting as the flood event driver) in the DCGAN, could improve the flood predictions at long lead-time series and mitigate the cumulated forecast error problem. This finding is similar to that of Chattopadhyay et al. (2020), where authors found that the driven variables included in weather forecast patterns could achieve higher accuracies and longer lead-time prediction with use of deep learning methods. While the inclusion of boundary conditions in deep learning leads to a better model performance (Boucher et al., 2020), there is often a lack of information on boundary conditions in a realistic application. To tackle this issue, inverse uncertainty quantification techniques can be used for estimation and calibration of boundary conditions. The boundary conditions for limited area simulations (LAS) are usually obtained from a global model or a large-scale model containing the area of LAS.

Overall, the results obtained suggest that the DCGAN is a powerful tool in flow pattern generalization and in nonlinear flow forecasting beyond the training period. In future research work, the trained DCGAN could also be incorporated into data assimilation techniques, which can effectively improve the predictive accuracy along with the real-time driving forcing (Irrgang et al., 2020).

5.5 Conclusion

Real-time forecasting is of great significance for better decision-making in flood disasters. In this work, a real-time DCGAN predictive modelling tool for flood forecasting has been developed. This tool has the capabilities of dynamic learning and real-time forecast. The DCGAN can be used to compress the high-dimensional flood fields into the low-dimensional features, and then recover these latent states into the original ones. This process extracts the fluid flow features and learns the dynamic flow patterns.

The proposed approach is assessed with two scenarios, one focuses on uncertainty quantification of inflow boundary conditions and the other on evaluation of long lead-time predictive performance of the DCGAN tool. It has been applied to flood events within Greve, Denmark. The results show that the DCGAN is able to capture the complex flow features and explore the flood propagation process over the urban area. It exhibits an overall good agreement with the original high fidelity model in simulations of spatiotemporal flow distribution for a lead-900 timestep forecast.

Overall, the proposed tool enables rapid and accurate long lead-time forecast of dynamic flows and has great potential for real-time forecasting and management, e.g. flood hazard mitigation and advanced management of policy responses to flood disaster, effective water resource management, and irrigation management decisions. The next chapter will present the ConvLSTM, DCGAN and VAE-GAN capabilities in hourly and daily ozone prediction in spatial and temporal spaces.

CHAPTER 6

Spatiotemporal hourly and daily ozone forecasting in China using GAN and ConvLSTM

Efficient and accurate real-time forecasting of national spatial ozone distribution is critical to the provision of effective early warning. Traditional numerical air quality models require a high computational cost associated with running large-scale numerical simulations. In this chapter, firstly, I systematically analyse the hourly and daily spatiotemporal predictive performances using convolutional long short term memory (ConvLSTM) and deep convolutional generative adversarial network (DCGAN) models over the Beijing-Tianjin-Hebei region, China. Secondly, I introduce a hybrid model (VAE-GAN) combining a generative adversarial network (GAN) with a variational autoencoder (VAE) to learn the dynamic ozone distributions in spatial and temporal spaces. The VAE-GAN model can not only decipher the complex nonlinear relationship between the inputs (the past states/ozone and meteorological factors) and outputs (ozone), but also provide ozone forecasts for a long lead-time beyond the training period. Results show that (1) the machine learning-based models achieve better spatiotemporal ozone concentration prediction performance with multiple meteorological conditions. The ConvLSTM model demonstrates

the practical feasibility of identifying high ozone concentration distribution and capturing spatiotemporal ozone variation patterns at a high spatial resolution (here $15km \times 15km$*). (2) VAE-GAN achieves a reasonable accuracy in the prediction of both the spatial and temporal evolution patterns of hourly and daily ozone fields, as compared to the Nested Air Quality Prediction Modelling System (commonly used in China), the reanalysis data and observations during the validation period. Thus, the VAE-GAN is a cost-effective tool for large data-driven predictions, which can potentially reinforce air pollution prediction efforts in providing risk assessment and management in a timely manner.*

This chapter is derived from and expands upon Cheng et al. (b) and Cheng et al. (c).

6.1 Introduction

In this work, firstly, I have quantified the uncertainty and heterogeneity of spatiotemporal ozone prediction using two machine learning models, i.e. convolutional LSTM (ConvLSTM) (Shi et al., 2015) and deep convolutional generative adversarial network (DCGAN) (Goodfellow et al., 2014) over the Beijing-Tianjin-Hebei (BTH) region, China. In the quantification framework, model sensitivity analysis is performed, which reveals the importance of selecting multiple meteorological factors and model architecture in spatiotemporal ozone prediction. I systematically explore the predictive capabilities of both machine learning-based methods for a whole year at hourly and daily scales, and conduct uncertainty analysis in spatial and temporal spaces. The comparisons are further conducted with physical modelling data and monitoring datasets at high spatial ($15km \times 15km$) and temporal (hourly and daily) resolutions.

Secondly, I have developed a spatiotemporal VAE-GAN for hourly and daily ozone forecasting. I use convolutional networks in the VAE-GAN model to improve its scalability for large data-driven modelling and implement a recursively forecasting strategy in the model architecture for achieving the long lead-time forecast. Overall, the VAE-GAN

model developed in this study can: 1) learn and classify the dynamic ozone features using high-dimensional datasets; 2) decipher the complex nonlinear relationship between the present states and future events among all training samples; 3) provide accurate hourly and daily ozone forecast in a long lead-time beyond the training period. By using high-resolution reanalysis data, it shows that the VAE-GAN can represent both the spatial and temporal variability of ozone concentration. To our best knowledge, this is the first proposal of the VAE-GAN for hourly and daily ozone forecasting using a combination of dynamic learning and real-time forecasting strategy.

In this work, the ConvLSTM, DCGAN and VAE-GAN model have been applied to hourly and daily forecasting of ozone concentrations with a high spatial resolution (15km $\times$ 15km) across China. The datasets are collected from the air quality reanalysis datasets during 2013-2019. The accuracy of the machine learning models have been evaluated by comparing the results with those obtained from both the physical air quality model (NAQPMS) and observations at monitoring stations. The remainder of this chapter is organized as follows. Section 6.2 provides the details of datasets used in this study. Section 6.3 introduces ConvLSTM and DCGAN architecture and prediction workflow, and the quantification of the uncertainty and heterogeneity of spatiotemporal ozone prediction. Section 6.4 presents the VAE-GAN architecture and forecasting workflow, and model performance analysis of ozone forecasting capabilities in spatial and temporal spaces. Finally in section 6.5, conclusions are presented.

6.2 Data

6.2.1 Reanalysis data

Ground-level ozone has become a severe pollutant in major urban areas of China (Zhan et al., 2018). Our objective is to develop an efficient and accurate machine learning tool

for providing an interactive ozone map over China, which shows real-time (hourly/daily) ozone concentrations with a high spatial resolution across China. However, there are only 1436 monitoring stations over China, which are sparse and heterogeneous. It is obvious that the monitoring data is not enough to represent the spatial distribution of ozone on a high spatial scale. Therefore, I select the 7-year Chinese air quality reanalysis (CAQRA) dataset for our training and validation purposes (Kong et al., 2021). The CAQRA dataset from 2013 to 2019 was produced by assimilating surface observations at 1436 monitoring stations into the Nested Air Quality Prediction Modeling System (NAQPMS). Such combination could be the optimal way by which physical modelling results from the NAQPMS can provide dynamic understanding from the governing equations, while monitoring data may find unexpected patterns not provided by physical modelling.

The CAQRA dataset contains six pollutants ($PM_{2.5}$, PM_{10}, SO_2, NO_2, CO and ozone) and meteorological factors (temperature, humidity and wind speed) at the high spatial (15km $\times$ 15km) and temporal (hourly) resolution, which are publicly accessible at Science Data Bank (ScienceDB). Each ozone map or meteorological map across China has a high spatial resolution with a total node number of $47,232$. The ozone concentration over China varies in both spatial and temporal spaces. For example, as shown in Fig. 6.1, it can be noted that the distributions of average hourly ozone concentration in June and December of 2018 are heterogeneous, exhibiting significant seasonal variations. This can be explained by the fact that ozone production is tightly correlated with meteorological conditions, such as ambient temperature, sunshine radiation and humidity, causing the variation of emitted intensity of atmospheric pollutants in different seasons (Bloomer et al., 2010). The high ozone concentration in June is closely related to the enhanced photochemistry reaction due to high temperature and strong solar radiance in Summer (Kong et al., 2021). To demonstrate the performance of our proposed model with seasonal variations, I choose the months of June and December as representing Summer and Winter while exhibiting hourly spatiotemporal ozone forecasts in the following section 6.4.1.

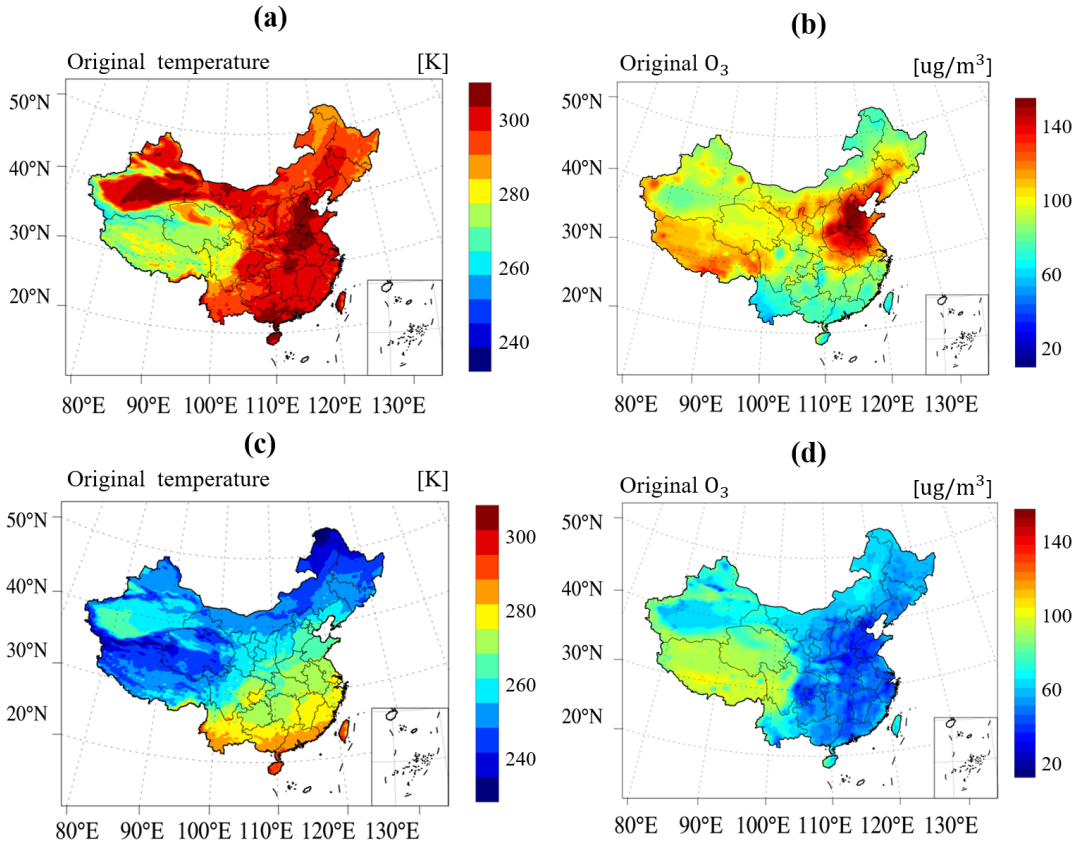


Fig. 6.1. Spatial distribution of the average hourly temperature and ozone concentration in China. (a) and (b) June; (c) and (d) December in 2018.

6.2.2 Monitoring and physical modelling data

The objective of this study is to develop a machine learning-based ozone forecasting model, which will be accurate and more efficient than existing air quality models. For comparison purposes, the Nested Air Quality Prediction Modelling System – NAQPMS (commonly used in China, here referred to as the physical model) is chosen. The NAQPMS model is driven by the hourly meteorological fields produced by the Weather Research and Forecasting (WRF) model, and the aqueous-phase chemistry and wet deposition are simulated based on the Regional Acid Deposition Model (RADM) mechanism in the Community Multi-scale Air Quality (CMAQ) model. The emissions of air pollutants considered in the NAQPMS model include the monthly anthropogenic, biomass burning, biogenic volatile organic compound (BVOC), marine VOC, soil NOx and lightning NOx emis-

sions. More details of the NAQPMS configuration are introduced in Kong et al. (2021). The hourly physical modeling dataset at the high spatial (15km $\times$ 15km) and temporal (hourly) resolutions is generated from the NAQPMS in June of 2018, which has the same spatial and temporal resolutions as the reanalysis data (CAQRA dataset). I compare the VAE-GAN results with those from the NAQPMS, where the monitoring measurements are used as the reference data. Hourly monitoring of ozone data is published by the China National Environmental Monitoring Centre (CNEMC). The hourly ozone observations are collected from 1436 stations in June of 2018 (Kong et al., 2021).

6.3 Spatiotemporal ozone simulation

6.3.1 Method

6.3.1.1 ConvLSTM

The ConvLSTM model can effectively encode spatial information while also retaining the ability to process temporal data. It has the CNN structures in both input-to-state and state-to-state transitions of a LSTM architecture. A traditional LSTM architecture is composed of special units (called memory blocks), and each memory block has the self-connected memory cells and multiplicative units. The memory cell is specially used to save temporal states of the network, and the units (including the input, output and forget gates) are responsible for updating or discarding historical information between cells. The input format in traditional LSTM model is a vector but not suitable for spatiotemporal data. The spatial information is lost in the process that feature map flattened into one dimension. For a ConvLSTM model, the use of CNN structure can process the spatial information through feature transformation and extraction in the convolution layer (Barzegar et al., 2020). Therefore, a ConvLSTM is very effective development of LSTM for processing

spatiotemporal predictive task. The ConvLSTM can be formulated as

$$f_s = \sigma(W_{xf} * \mathcal{X}_s + W_{hf} * \mathcal{H}_{s-1} + W_{cf} \circ \mathcal{C}_{s-1} + b_f), \tag{6.1}$$

$$i_s = \sigma(W_{xi} * \mathcal{X}_s + W_{hi} * \mathcal{H}_{s-1} + W_{ci} \circ \mathcal{C}_{s-1} + b_i), \tag{6.2}$$

$$\widetilde{\mathcal{C}}_s = \tanh(W_{xc} * \mathcal{X}_s + W_{hc} * \mathcal{H}_{s-1} + b_c), \tag{6.3}$$

$$\mathcal{C}_s = f_s \circ \mathcal{C}_{s-1} + i_s \circ \widetilde{\mathcal{C}}_s, \tag{6.4}$$

$$O_s = \sigma(W_{xo} * \mathcal{X}_s + W_{ho} * \mathcal{H}_{s-1} + W_{co} \circ \mathcal{C}_s + b_o), \tag{6.5}$$

$$\mathcal{H}_s = O_s \circ \tanh(\mathcal{C}_s), \tag{6.6}$$

where i_s, O_s and f_s are the input, output and forget gates respectively, W_i, W_o, W_f and W_c represent the weights for each gate, b_i, b_o, b_f and b_c are the bias terms, σ denotes the logistic sigmoid function, $tanh$ is the hyperbolic tangent function, $\mathcal{C}_s$ represents the cell state, $\widetilde{\mathcal{C}}_s$ is the updated cell state, $\mathcal{X}_s$ and $\mathcal{H}_s$ are the cell input and output respectively. The difference between the ConvLSTM and traditional LSTM is that all the inputs, cell states, hidden states, and gates in the ConvLSTM are 3D tensors. In addition, the operator '$*$' stays instead of the vector multiplication. '$*$' is the convolution operation and '$\circ$' denotes the Hadamard product.

In this study, the ConvLSTM predictive model is built consisting of one input layer, two convolutional LSTM layers and one convolutional layer. The Leaky rectified linear unit (Leaky ReLU) activation function ($f(x) = \max(ax, x)$) is employed between the layers. The network has 3 batch sizes and runs 20 epochs. The ConvLSTM network and our whole workflow were implemented in Keras (Gulli and Pal, 2017), an open-source machine learning framework.

6.3.1.2 DCGAN and LDCGAN

The original DCGAN consists of two modules, a generator $\mathcal{G}$ and a discriminator $\mathcal{D}$ (Goodfellow et al., 2014). As shown in Fig. 6.3, the discriminator $\mathcal{D}$, is fed with sam-

ples from "real" data $\mathcal{O}_d$ and "fake" data (the reconstructed samples) $\widehat{\mathcal{O}}$ created by the generator $\mathcal{G}$: $\widehat{\mathcal{O}} = \mathcal{G}(\mathcal{X})$. Here, the real samples $\mathcal{O}_d$ are a series of ozone maps (referred to as targeted outputs) and $\mathcal{X}$ is model inputs, i.e., meteorological factor maps. The discriminator $\mathcal{D}$ tries to distinguish the $\widehat{\mathcal{O}}$ from $\mathcal{O}_d$ by making a binary decision. In contrast, the generator $\mathcal{G}$ aims at fooling the discriminator $\mathcal{D}$ into labelling $\widehat{\mathcal{O}}$ as a sample $\mathcal{O}_d$. Mathematically, this translates into the following minimization-maximization loss function $\mathcal{L}_{DCGAN}$:

$$\min_{\mathcal{G}} \max_{\mathcal{D}} E_{\mathcal{O}_d \sim p_{data}(\mathcal{O}_d)}[\log \mathcal{D}(\mathcal{O}_d)] + E_{\mathcal{X} \sim p_{\mathcal{X}}(\mathcal{X})}[\log(1 - \mathcal{D}(\mathcal{G}(\mathcal{X})))], \tag{6.7}$$

where $p_{\mathcal{X}}(\mathcal{X})$ is a prior distribution for the input dataset $\mathcal{X}$, and $p_{data}(\mathcal{O}_d)$ is the corresponding probability data distribution for the targeted outputs $\mathcal{O}_d$.

In practice, the minimization-maximization loss function in Eq.(6.7) is solved in two steps. First one maximizes the loss function in the discriminator $\mathcal{D}$ as:

$$\mathcal{L}_D = E_{\mathcal{O}_d \sim p_{data}(\mathcal{O}_d)}[\log \mathcal{D}(\mathcal{O}_d)] + E_{\mathcal{X} \sim p_{\mathcal{X}}(\mathcal{X})}[\log(1 - \mathcal{D}(\mathcal{G}(\mathcal{X})))], \tag{6.8}$$

while keeping the parameters of $\mathcal{G}(\mathcal{X})$ fixed to current values.

Second, one minimizes the loss function in the generator $\mathcal{G}$:

$$\mathcal{L}_G = E_{\mathcal{X} \sim p_{\mathcal{X}}(\mathcal{X})}[\log(1 - \mathcal{D}(\mathcal{G}(\mathcal{X})))]. \tag{6.9}$$

In the LDCGAN model, the loss function in the generator $\mathcal{G}$ consists of two terms:

$$\mathcal{L}_G = E_{\mathcal{X} \sim p_{\mathcal{X}}(\mathcal{X})}[\log(1 - \mathcal{D}(\mathcal{G}(\mathcal{X})))] + \lambda E_{\mathcal{O}_d \sim p_{data}(\mathcal{O}_d)} \|\mathcal{G}(\mathcal{X}) - \mathcal{O}_d\|_2^2, \tag{6.10}$$

where the hyperparameter λ is the $L2$ objective weight.

The final objective of LDCGAN is:

$$\min_{\mathcal{G}} \max_{\mathcal{D}} \mathcal{L}_{DCGAN} + \lambda E_{\mathcal{O}_d \sim p_{data}(\mathcal{O}_d)} \|\mathcal{G}(\mathcal{X}) - \mathcal{O}_d\|_2^2. \tag{6.11}$$

In this study, both DCGAN and LDCGAN models are implemented by using the U-net architecture in the generator $\mathcal{G}$. The U-Net convolutional neural network consists of two parts: an encoder and a decoder. The encoder transforms the inputs $\mathcal{X}$ into a compact latent feature representation. The decoder adopts a sequential symmetric expanding path to increase the resolution of the generated ozone maps. The encoder process consists of the repeated application of 3x3 convolutions with a stride of 2 for downsampling, each followed by a batch normalization layer (Ioffe and Szegedy, 2015) and Leaky rectified linear unit (Leaky ReLU) activation function. The decoder performs a deconvolution process, which is formed by a 2$\times$ 2 upsampling and 3$\times$3 convolution with a stride of 2. For details of U-net architecture, see Jiang et al. (2021). The discriminator $\mathcal{D}$ uses five hidden convolutional layers, each using a densely connected layer with a Leaky ReLU activation function. The output layer is densely connected to the final hidden layer with a sigmoid activation function. Specifically, our networks are trained using the Adam optimizer and the weights in the networks are updated after computing the combined loss of each batch. The batch size of the networks is 3 and both models run 1000 epochs for training the network.

6.3.2 Results

6.3.2.1 Data processing

In this study, the Beijing-Tianjin-Hebei (BTH) region is chosen as our study area since it is a densely populated area and has high ozone concentrations. The hourly spatiotemporal ozone fields and meteorological conditions are extracted with a spatial resolution $15km \times 15km$ over the BTH region during the period between the years 2013-2018,

where the datasets in the periods of 2013 to 2017 and 2018 are used for model training and validation, respectively.

Meteorology variations and climate change influence the chemistry and transport pathway of ozone concentration (Lu et al., 2019). Here I choose three highly ozone-related factors as predictors, i.e. temperature, humidity and wind (Zhan et al., 2018). Temperature and humidity can alter the partitioning and efficiency of chemical reactions and dry deposition, and therefore modulate ozone production and loss. Wind speed and direction are contributed to the spatiotemporal patterns of transported ozone precursors. Here, I explore the impacts of meteorological factors (as inputs $\mathcal{X}$) on the spatiotemporal heterogeneity of ozone distribution $\mathcal{O}$: $\mathcal{O} = \mathcal{F}(\mathcal{X})$. Fig. 6.2 plots the importance variation of three meteorological factors using the random forest model (Breiman, 2001) in predicting the ozone concentrations at hourly and daily scales, respectively. It can be noted that ozone concentration is influenced notably by temperature. The temperature weights in hourly and daily prediction range from 0.7 to 0.9 in a spatial scale, which are higher than humidity and wind weights over the BTH region. In contrast to daily prediction, the humidity in hourly ozone prediction exhibits higher importance over the BTH region. It is found that in the north of the BTH region, the V wind velocity component (parallel to the latitude) takes a higher proportion than that of the U wind velocity component (parallel to the longitude), thus assuming higher-relative importance in ozone concentration prediction, which is in consistent with the findings in Ma et al. (2021b).

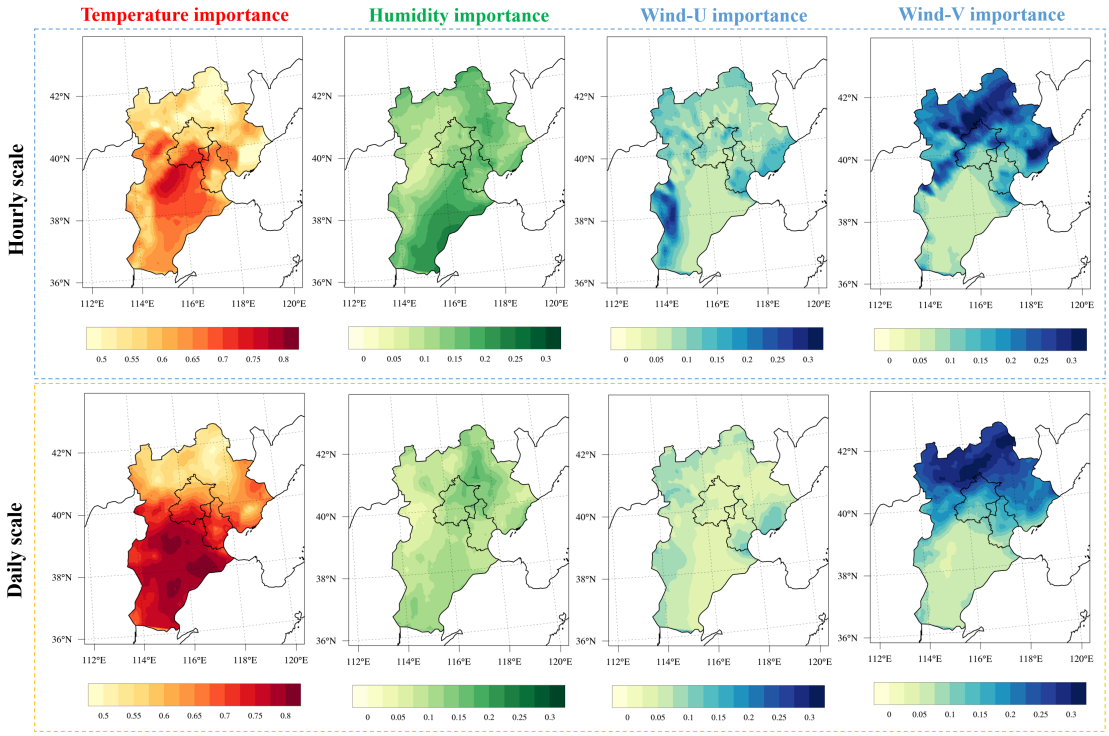


Fig. 6.2. Variable importance plot using the random forest model in predicting the ozone concentrations over the BTH region at hourly and daily scales, respectively.

A schematic diagram of the proposed machine learning framework for hourly and daily ozone forecasting using ConvLSTM, DCGAN and LDCGAN is shown in Fig. 6.3. These predictive models were assessed by computing the correlation coefficient (R), Root Mean Squared Error (RMSE) and the cumulative distribution formula (CDF) from the probability density function.

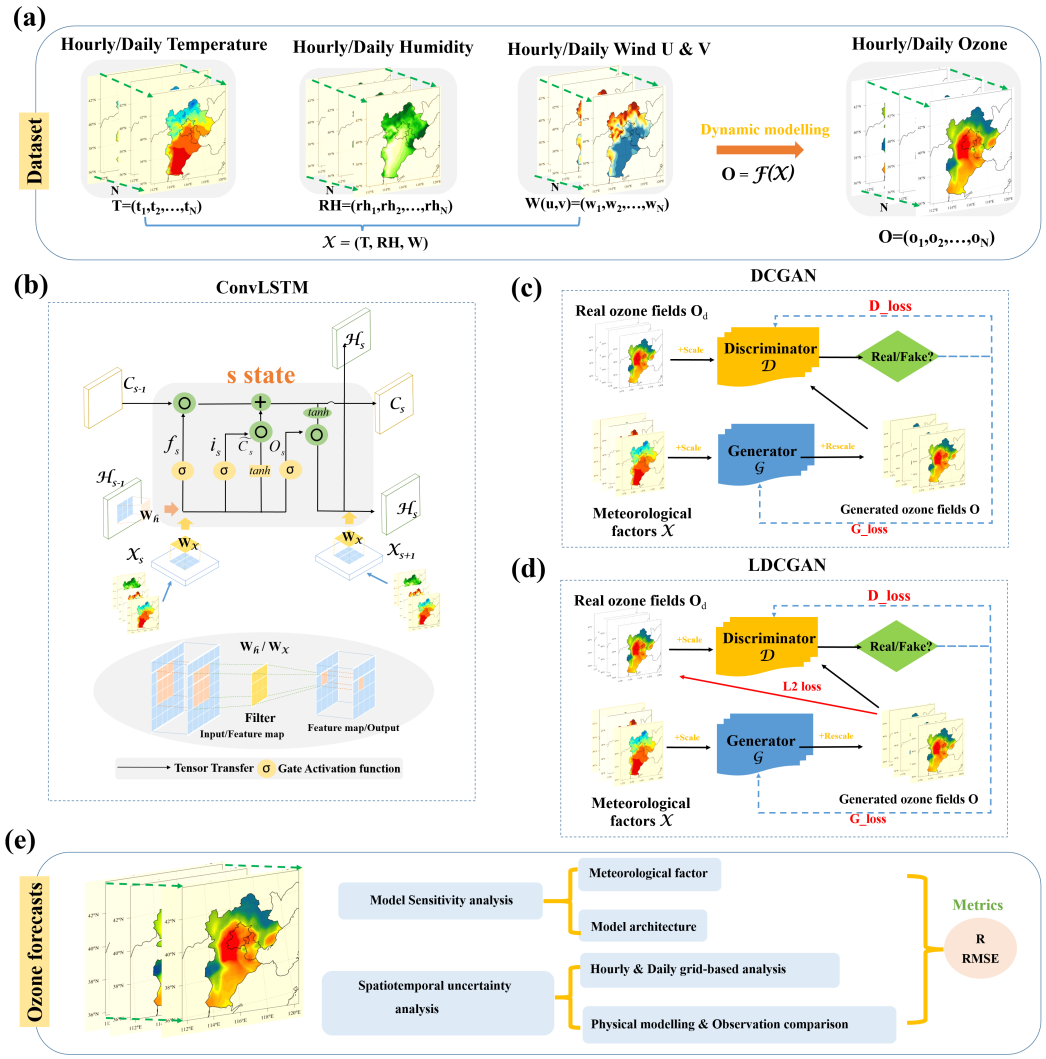


Fig. 6.3. Schematic of real-time ozone concentration predictive strategy by using ConvLSTM, DCGAN and LDCGAN. (a) Selection of the spatiotemporal meteorological variables (including temperature, humidity and wind velocity) over the BTH region as model inputs and ozone as model output. (b)(c)(d) Architectures of the ConvLSTM, DCGAN and LDCGAN models for ozone prediction. The LDCGAN model is an improved version of the DCGAN model with an additional loss function, details shown in section 6.3.1.2. (e) Uncertainty and heterogeneity analysis of spatiotemporal ozone prediction in the model validation period.

6.3.2.2 Model sensitivity analysis

A number of case studies have been set up for sensitivity analysis of spatiotemporal ozone prediction in terms of the choice of machine learning methods and meteorological condi-

tions. The corresponding results and error analysis are plotted in Fig. 6.4. The meteorological factors are considered in case 1 (temperature), case 2 (temperature and humidity) and case 3 (temperature, humidity and wind velocity), where ConvLSTM, DCGAN and LDCGAN are used. LDCGAN is an improved version of DCGAN with an extra loss functional term (L2 norm), which ensures that not only the probability distribution, but also the predicted results fit into the 'true' values.

I have first quantified the role of meteorological factors on spatiotemporal ozone concentration prediction. Our results reveal that temperature plays the dominant role in modulating the spatiotemporal variability of ozone concentrations in the BTH region. The humidity and wind velocity could also be attributable to larger spatiotemporal variations (shown in Fig. 6.4(c)). With adequate humidity as well as sunlight over the BTH region, the ozone photochemical chain reaction initiates easily while the variations in surface ozone loading could be regulated largely by wind (Ma et al., 2021a). It can also be seen that there is an overall decline trend of RMSE values with multiple meteorological inputs (Fig. 6.4(a)). The mean RMSEs decrease from $37.8ug/m^3$ and $36.2ug/m^3$ (case 1) to $31.2ug/m^3$ and $32.5ug/m^3$ (case 3) for the ConvLSTM and LDCGAN respectively while for the DCGAN the mean RMSEs are reduced by 5.7% when both the temperature and humidity are included in the model inputs.

Comparative performance of the ConvLSTM, DCGAN and LDCGAN models has been further evaluated. Among the three models, the ConvLSTM model yields the lowest RMSE of $31.2ug/m^3$ (Fig. 6.4(a)) and highest correlation coefficient of 0.9 (Fig. 6.4(b)). The highest correlation coefficient shown in the Taylor diagram (Fig. 6.4(b)), is 0.8 (case 2) in DCGAN and 0.82 (case 3) in LDCGAN respectively. Regarding the cumulative distribution of RMSEs (Fig. 6.4(c)), it is shown that the RMSEs of 80% points are smaller than $37ug/m^3$ (case 3) for ConvLSTM, $43ug/m^3$ (case 2) for DCGAN, and $41ug/m^3$ (case 3) for LDCGAN respectively. We can see that the LDCGAN (an improved version of DCGAN with an extra loss functional term) has improved the accuracy of predicted re-

sults in comparison to DCGAN. The RMSEs in the spatial distribution of hourly-averaged ozone concentrations (the left columns in Fig. 6.4(c)) have been reduced, thus better capturing the spatial characteristics at a high spatial resolution ($15km \times 15km$) as well as the temporal variation (Fig. 6.4(a)).

In general, our results reveal the predictive power of machine learning-based models increases with a combination of meteorological factors along with an improved model architecture. Both the ConvLSTM and LDCGAN models provide an improvement in the reduction of predictive errors by considering temperature, humidity and wind velocity as model inputs. This proves the importance of meteorological factors (Fig. 6.2) in the formation mechanism of ozone (Ma et al., 2021b; Wang et al., 2020c).

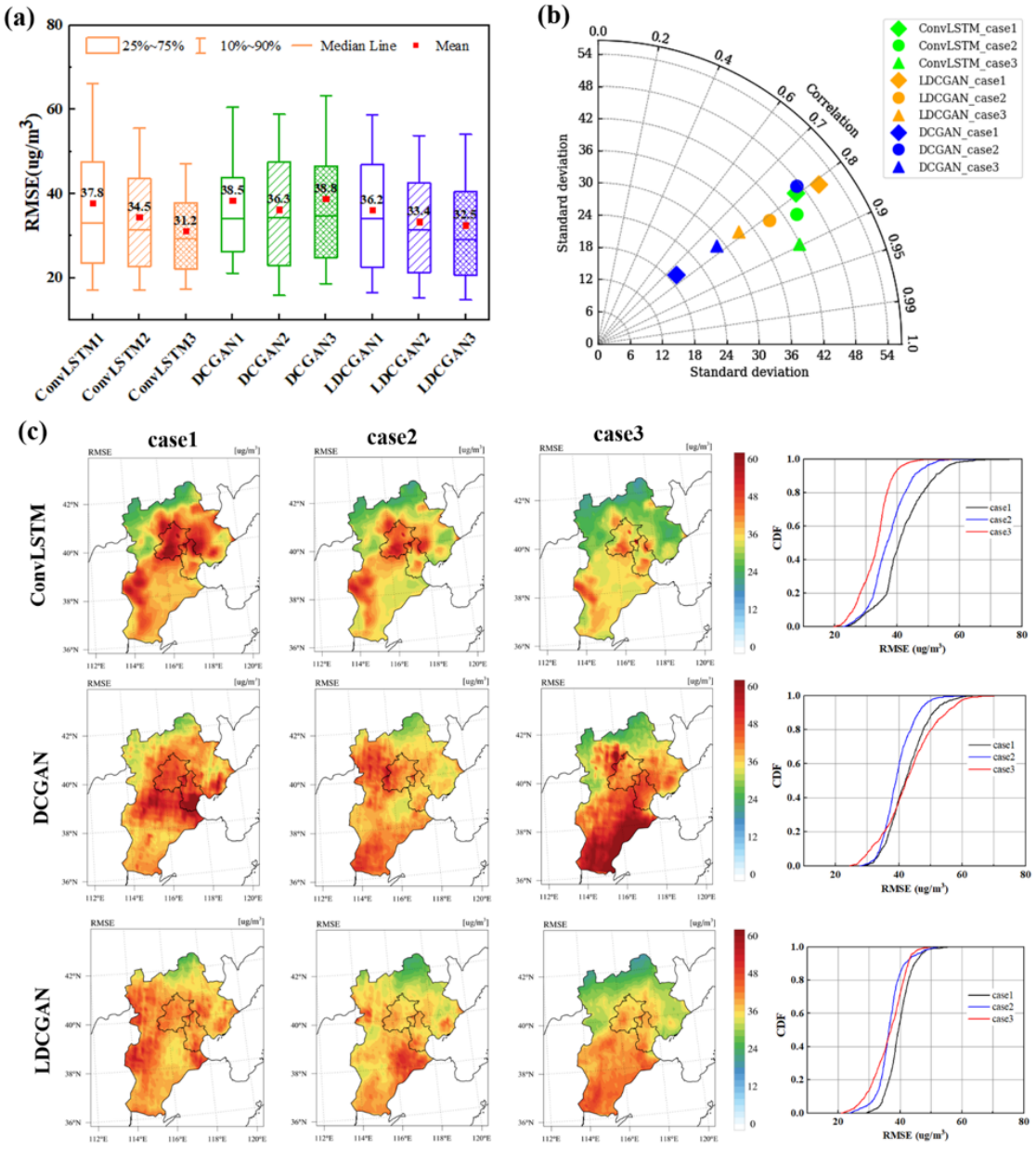


Fig. 6.4. Performance evaluation of the three machine learning models in: (a) and (b) temporal (hourly) variation of spatial-averaged predicted ozone concentrations, and (c) spatial distribution of hourly-averaged predicted ozone concentrations at a high spatial resolution of $15km \times 15km$, which are assessed by RMSE, the standard deviation, correlation in Taylor diagrams and cumulative density functions (CDF) for the RMSE, where multiple meteorological factors are considered in case 1 (temperature), case 2 (temperature and humidity) and case 3 (temperature, humidity and wind velocity).

6.3.2.3 Spatiotemporal ozone prediction analysis

6.3.2.3.1 Hourly ozone prediction analysis

Hourly variation of spatial-averaged ozone concentrations predicted by the ConvLSTM, DCGAN and LDCGAN models in 2018 are depicted in Fig. 6.5. It can be noticed that in Fig. 6.5(a), the ozone prediction using the ConvLSTM, DCGAN and LDCGAN models could capture hourly ozone variability and exhibit a good match with the reanalysis data. Among these machine learning-based models, the ConvLSTM model can better capture the peak of ozone concentrations (e.g. 22-30 of March). It can be also noticed that the LDCGAN achieves the promising predictive accuracy of the mean ozone concentrations, but has the limitation in capturing high ozone concentration variations.

The error analysis has been undertaken and the corresponding results are plotted in Fig. 6.5(b) and (c). It further indicates that the ConvLSTM model performs better than the DCGAN and LDCGAN models in hourly ozone prediction. For example, the correlation coefficient is 0.89 for ConvLSTM modelling while it is 0.78 for both DCGAN and LDCGAN modelling in hourly ozone prediction in 2018. Fig. 6.5(c) shows that the range of bias (from $-15ug/m^3$ (10^{th} percentile) to $34ug/m^3$ (90^{th} percentile)) in the ConvLSTM model is smaller, compared to those predicted by the DCGAN and LDCGAN models, which have the bias range of $-43ug/m^3$ and $-41ug/m^3$ (10^{th} percentile) to $36ug/m^3$ and $26ug/m^3$ (90^{th} percentile) respectively. However, the ConvLSTM model has a higher median value of bias ($12ug/m^3$) than the DCGAN and LDCGAN models. This is because an overestimation of low ozone concentrations predicted by the ConvLSTM model leads to a high average bias.

Spatial heterogeneity patterns of temporal-averaged predictive hourly ozone concentrations in four seasons of 2018 are depicted at a high spatial resolution of $15km \times 15km$ in Fig. 6.6. It can be seen that the three machine learning-based models could largely capture the season characteristics as shown in Fig. 6.6(a). Photochemical production of

ozone in Summer (April to June) and Autumn (July to September) is much stronger (Wang et al., 2020a) than that in Winter. This can clearly be seen from the ozone concentrations predicted by all models which are high in Summer and Autumn and low in Spring and Winter. Compared to the reanalysis data, the predicted ozone concentrations exhibit comparable spatial variations. Fig. 6.6(b) presents the quantitative analysis between the ozone concentrations predicted by the machine learning-based models and reanalysis data from the spatial and heterogeneity perspectives. It is visible that the relative errors of results in four seasons from the ConvLSTM model are slightly lower in comparison to the other two machine learning-based models. The relative error of results from the ConvLSTM model mostly ranges from -10% to 10% except for some peak values in the central areas of the BTH region. In general, the ConvLSTM exhibits the best performance, especially in Summer and Spring (having a small relative error, see Fig. 6.6(b) while the LDCGAN can predict better in terms of the yearly-averaged ozone.

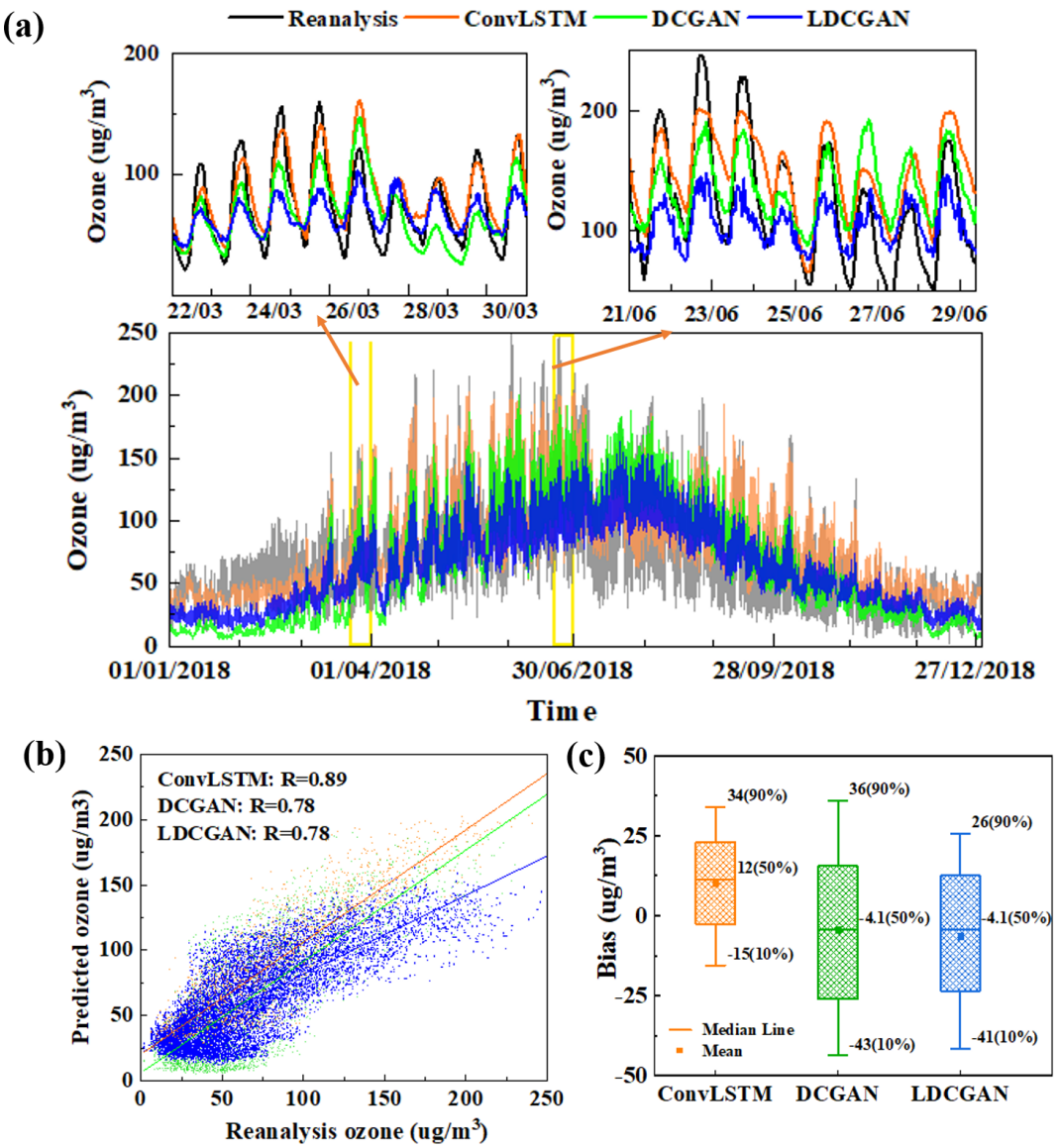


Fig. 6.5. Temporal variation of hourly predictive ozone concentration and uncertainty analysis in 2018 by the ConvLSTM, DCGAN and LDCGAN models. (a) Temporal variation of spatial-averaged variation hourly ozone concentration over the BTH region using three machine learning-based models and reanalysis data (as reference). (b) Correlation coefficients of predictive results from the three machine learning-based models compared to the hourly reanalysis data in 2018. (c) Biases of predictive results from the three machine learning-based models compared to the hourly reanalysis data in 2018.

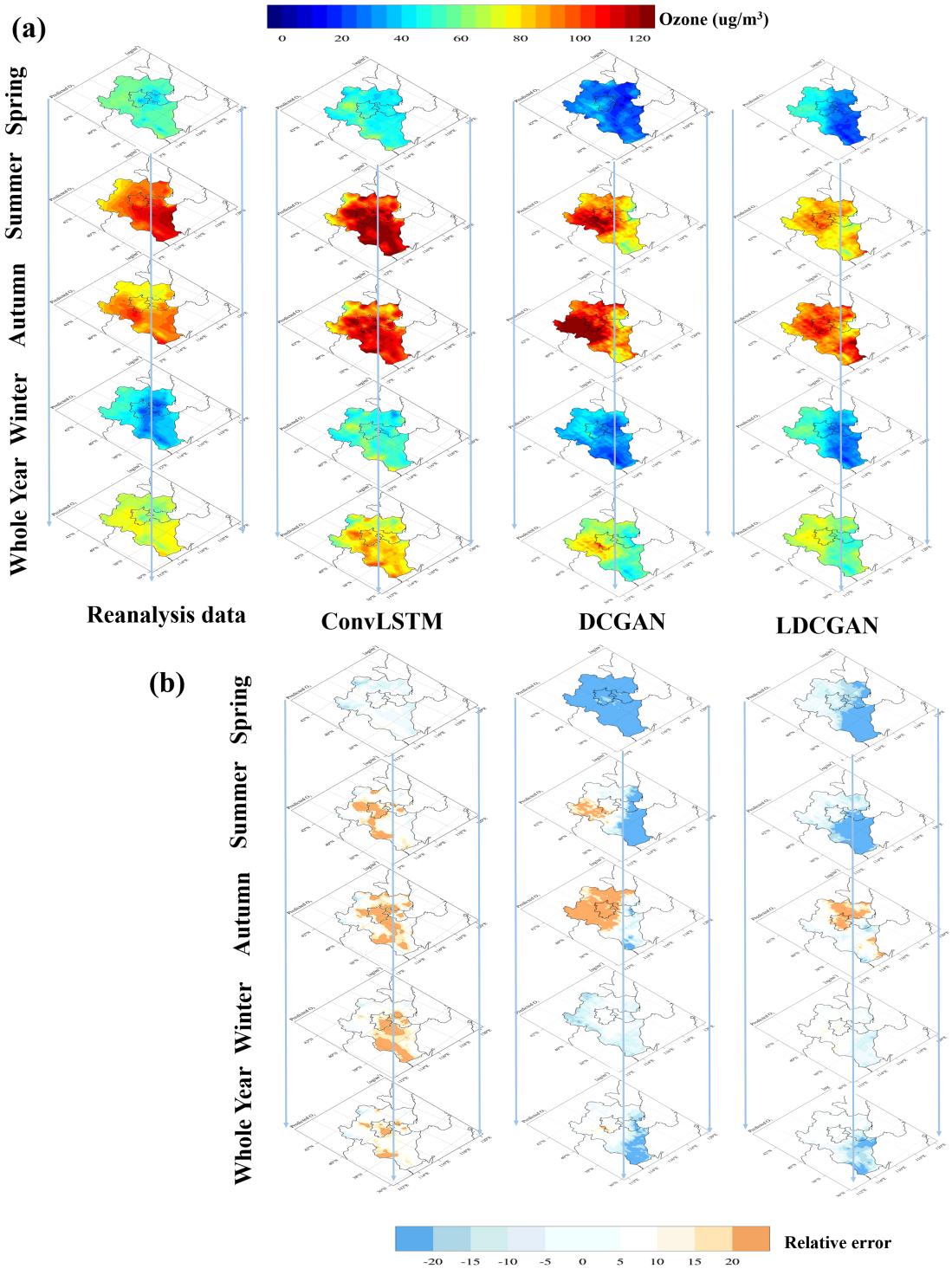


Fig. 6.6. Spatial variation of hourly predictive ozone concentrations and error analysis by the ConvLSTM, DCGAN and LDCGAN models. (a) Spatial distribution of temporal-averaged variation hourly ozone concentration over the BTH region in four seasons and the whole year of 2018 by three models. (b) Relative errors of temporal-averaged predictive results from the three machine learning-based models compared to the reanalysis data in four seasons and the whole year. Note that the spatial resolution is $15km \times 15km$.

6.3.2.3.2 Daily ozone prediction analysis

Fig. 6.7 gives the comparative plots of the daily ozone prediction results obtained from the ConvLSTM, DCGAN and LDCGAN models. One can see a relatively good match between the observed and predictive ozone from the three models. In comparison to the DCGAN and LDCGAN model, the ConvLSTM models not only capture the daily ozone concentration variability, but also can better predict the high ozone concentration, especially in Summer (e.g. the red line match well with the black line from 19 June to 27 June in Fig. 6.7(a)). During the validation period of 2018, the correlation coefficient is 0.91, 0.65 and 0.73 (Fig. 6.7(b)) for the ConvLSTM, DCGAN and LDCGAN models, respectively. In terms of result bias, the ConvLSTM model achieves the smallest median value of $0.91ug/m^3$ (Fig. 6.7(c)). Fig. 6.8 presents the spatial distribution of daily ozone prediction in the four seasons as well as the whole year. As discussed above, all machine learning-based models could capture the season characteristics especially in Summer and Autumn in Fig. 6.8(a), Again, among these models, the ConvLSTM model has the smallest residuals in four seasons over the BTH region, except for an underestimation of ozone concentrations in the south of the BTH region in Summer. It is worth mentioning that the relative errors in Fig. 6.8(b) from the three models mostly range from -5% to 5% throughout the whole year.

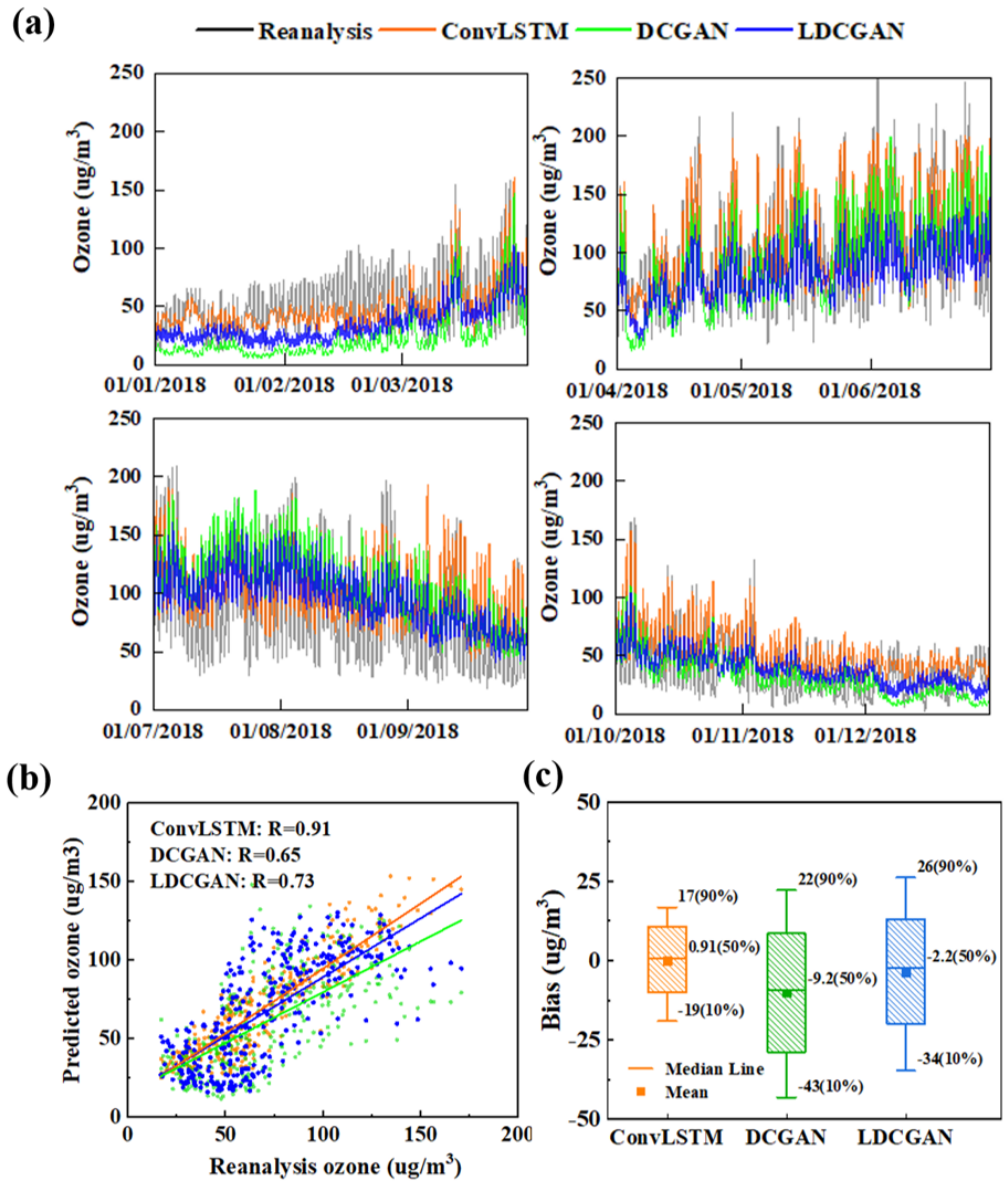


Fig. 6.7. Same as Fig. 6.5, except for daily ozone concentration prediction.

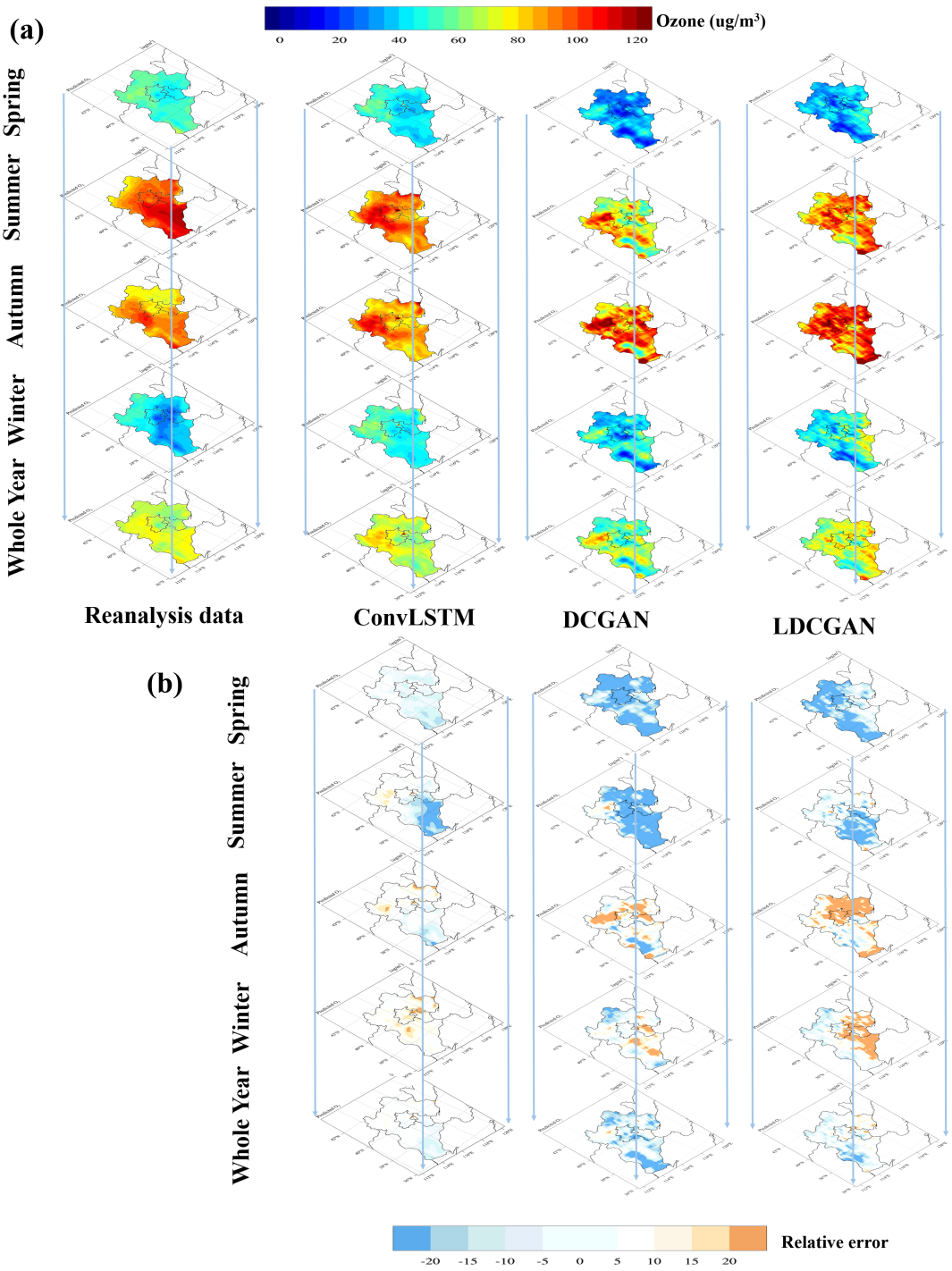


Fig. 6.8. Same as Fig. 6.6, except for daily ozone concentration prediction.

6.3.2.4 Machine learning-based ozone prediction in comparison to physical air pollution modelling and observations

6.3.2.4.1 Comparison with physical air pollution modelling in spatial and temporal spaces

To further analyse the spatiotemporal uncertainty of ozone prediction, the predicted ozone concentrations in June of 2018 from the ConvLSTM, DCGAN and LDCGAN models are compared to those from the Nested Air Quality Prediction Modelling System (referred to as the physical model) without data assimilation (Kong et al., 2021). Again, the Chinese air quality reanalysis (CAQRA) dataset is used as the 'true' reference in this section.

Fig. 6.9(a) shows the RMSEs of spatial-averaged predicted ozone concentrations in the whole month of June in 2018. It can be noticed that the results from the physical model (grey line) have higher RMSEs (highest $83ug/m^3$) in the day time than that of the ConvLSTM models. This means that the ConvLSTM model exhibits the better predictive performance of spatiotemporal ozone concentrations than the physical model over the BTH region in June of 2018. To further evaluate the model performance at the spatial scale, the spatial distribution of predicted ozone concentration over the BTH region at 12:00 on day 13 of June in 2018 is plotted in Fig. 6.9 (b). It can be noted that the ozone concentrations are higher in the south region, while a lower concentration is in the northwest region. The physical model, as well as DCGAN and LDCGAN models, fail to capture the high ozone concentration distribution in the BTH region. By contrast, the ozone spatial distribution from the ConvLSTM model is in good agreement with that of the reanalysis data, which captures the heterogeneity of ozone distribution over the BTH region.

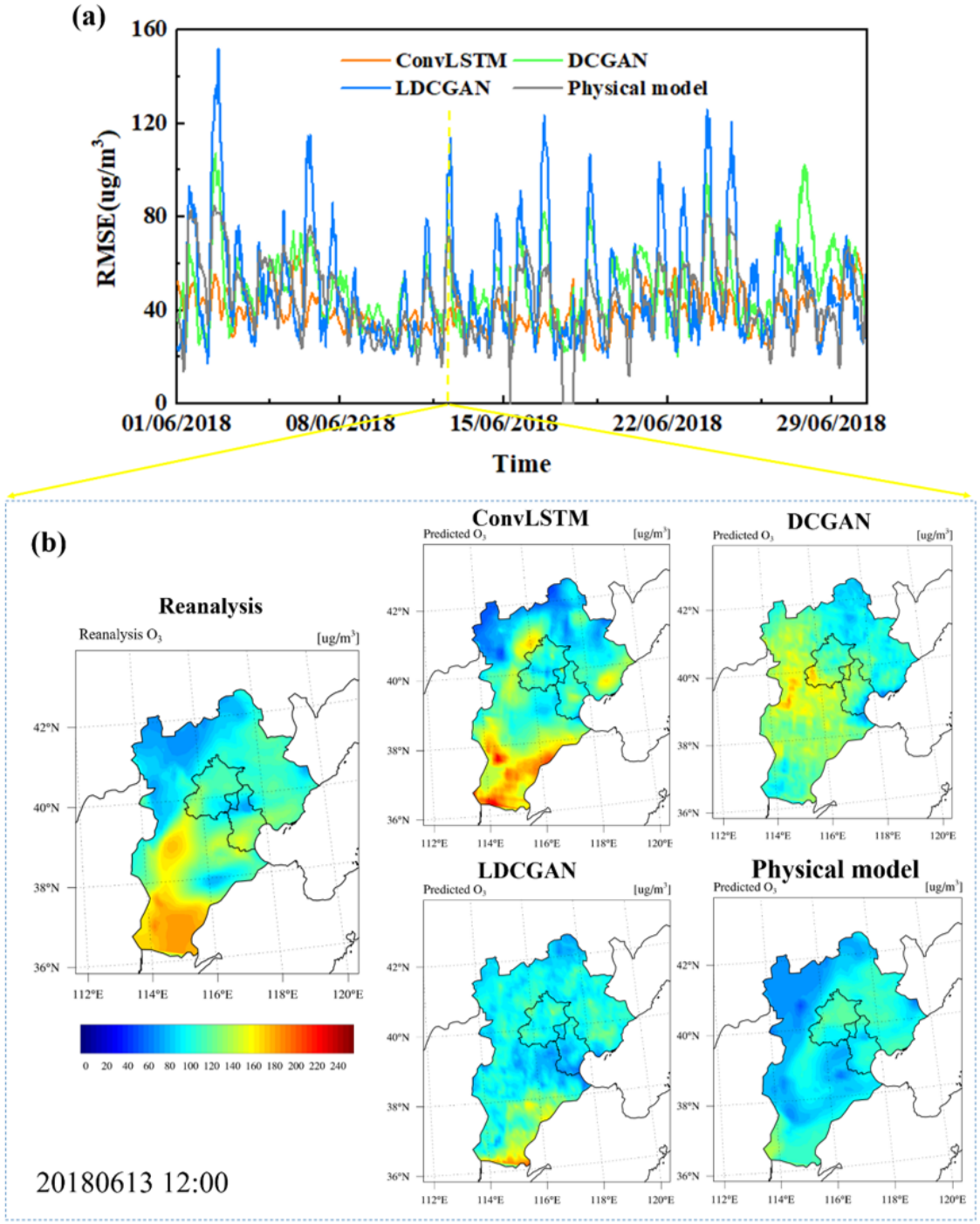


Fig. 6.9. The spatial distribution and RMSE of predictive results from the ConvLSTM, DCGAN and LDCGAN and physical model compared to the reanalysis data in the predictive period of June in 2018. (a) RMSE of predictive results from the three machine learning-based models and physical model compared to the hourly reanalysis data in June of 2018; (b) The predictive ozone distributions using ConvLSTM, DCGAN, LDCGAN and physical model compared to the ozone field from reanalysis data on 12:00 13 June of 2018. Note that the spatial resolution is $15km \times 15km$.

6.3.2.4.2 Comparison with physical air pollution modelling and observations at monitoring stations

The performance of the ConvLSTM, DCGAN and LDCGAN models in spatiotemporal prediction is further evaluated by comparing the predicted ozone concentrations with the monitoring data over the BTH region. Here, the monitoring data is used as 'true' reference in error estimation in this section. Fig. 6.10 shows the time series of hourly predicted and monitoring ozone concentrations in June of 2018 in the main cities over the BTH region. It can be seen that the predicted ozone by the ConvLSTM model is largely in good agreement with observations during the predictive period in June of 2018. It is noticed that the physical model underestimates the hourly ozone concentrations in June in most cities.

The error analysis of ozone concentrations has been carried out using the correlation coefficient and RMSE indicators shown in Fig. 6.11, where the wind rose maps are used to present the overall model performance on daily ozone prediction in June for each city. It can be seen that the reanalysis data achieve the best performance among all the results due to that it has combined the observations with physical modelling results by data assimilation techniques. By comparing modelling results, it is noted that the RMSEs of predicted results from the physical model are much higher than that of the machine learning-based models (Fig. 6.11(b)) although higher correlation coefficients in physical modelling are achieved in most cities except for Beijing, Chengde and Tangshan (Fig. 6.11(a)). Among the three machine learning-based models, our results show that the performance of the ConvLSTM model in the eleven cities is better than that of the DCGAN and LDCGAN models. Again, it is found that a well-trained machine learning-based model has captured the spatiotemporal characteristics of the ozone concentration variation and is able to provide accurate prediction.

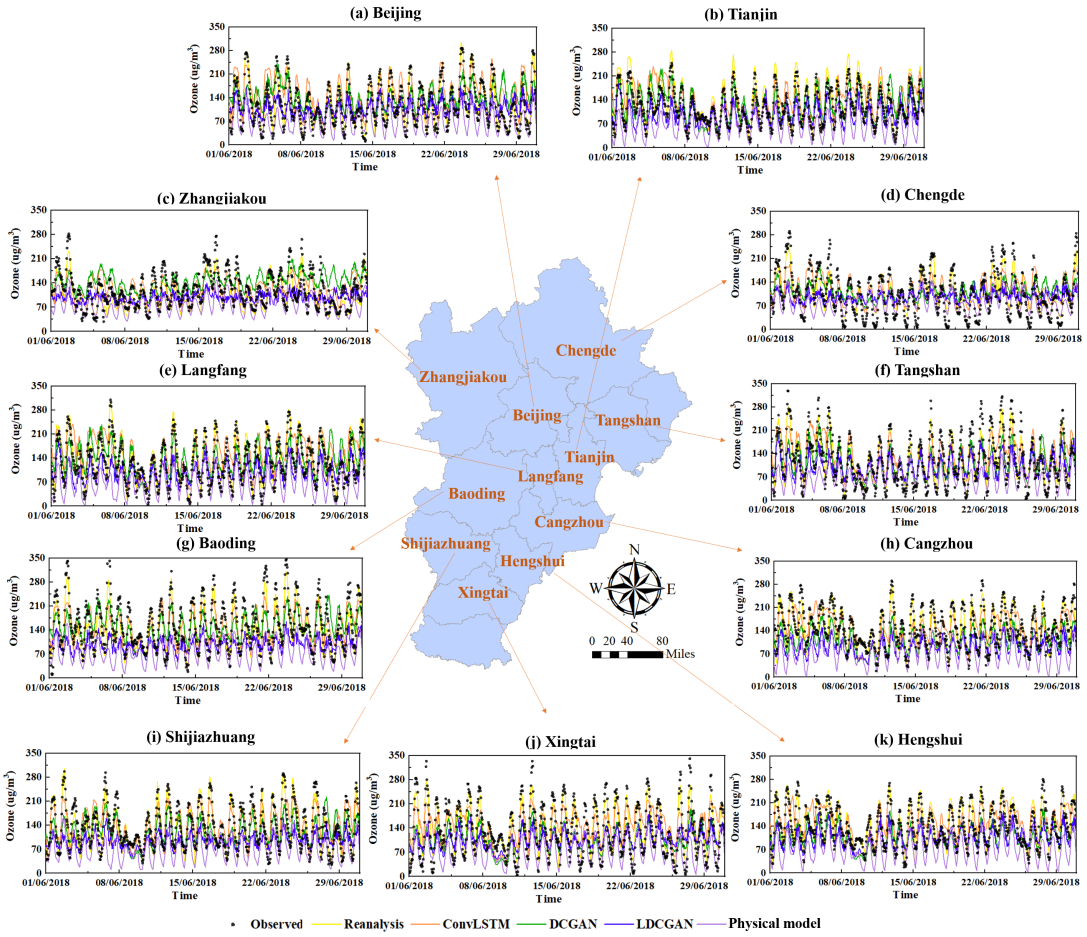


Fig. 6.10. Ozone time series comparison at the eleven main cities (Beijing, Tianjin, Zhangjiakou, Chengde, Langfang, Tangshan, Baoding, Cangzhou, Shijiazhuang, Xingtai and Hengshui) over the BTH region in the predictive period of June 2018, where the station observations at the eleven main cities are used as the 'true' references.

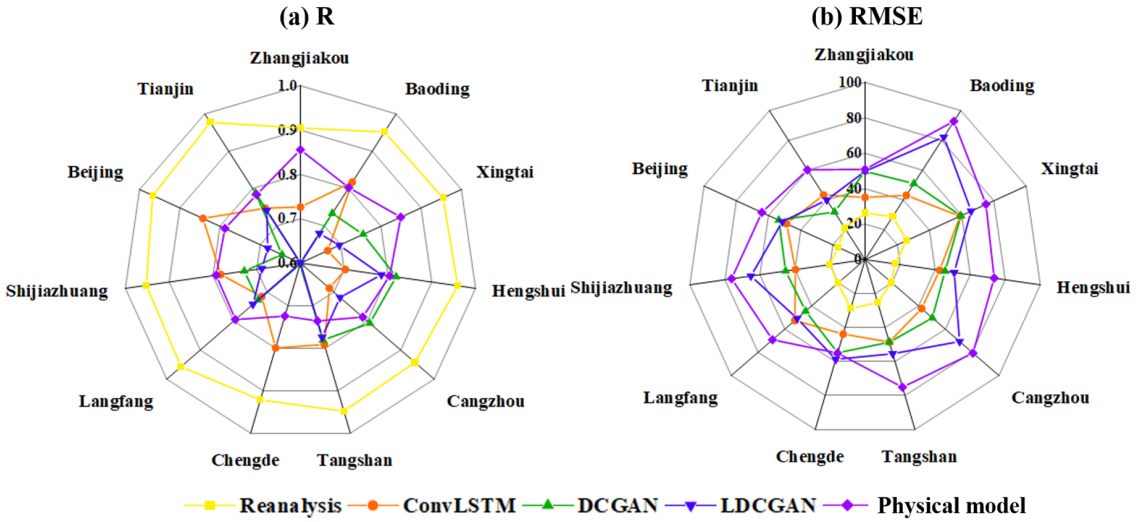


Fig. 6.11. Error analysis of results from the reanalysis data, physical model, ConvLSTM, DCGAN and LDCGAN models, where the observations in June of 2018 are taken as references (true values). (a) Correlation coefficient and (b) RMSE of predictive results from reanalysis data, physical model, ConvLSTM, DCGAN and LDCGAN models compared to the monitoring observations in eleven cities over the BTH region.

6.4 Real-time forecasting

6.4.1 Methods

Ozone forecasting involves nonlinear, strong coupling and heterogeneous problems (Su et al., 2020). As investigated by Zhan et al. (2018), the key ozone-related meteorological variables are the temperature, relative humidity, wind speed, evaporation and sunshine duration. Among these meteorological conditions, the daytime surface temperature is an important driver of ozone episodes. A high temperature can accelerate the chemical kinetic reaction and emissions of the natural components of ozone increase (Bloomer et al., 2009; Yin et al., 2019; Gu et al., 2020). This work thus focuses on developing an ozone forecasting model depending on temperature. The impact of other meteorological factors (such as humidity and wind speed) is also explored in section 6.4.2.

Given a series of ozone observations $\{O_{t-p}, ..., O_t\}$ (where p is the length of historical time steps) and the temperature data T_{t+1} at the predictive time level $(t+1)$, an one-step-

ahead ozone forecast can be expressed mathematically as:

$$\widehat{O}_{t+1} = \mathcal{F}(O_{t-p}, ..., O_t, T_{t+1}), \tag{6.12}$$

where $\widehat{O}_{t+1} = (\widehat{O}^1_{t+1}, ..., \widehat{O}^k_{t+1}, ..., \widehat{O}^{M_k}_{t+1})$ ($k \in (1, M_k)$, M_k is the number of points in the study area) represents the predictive ozone concentration at time level $(t+1)$ and $\mathcal{F}$ is a forecasting model. In this work, the VAE-GAN-based ozone forecasting model is selected as the forecasting model $\mathcal{F}$. A schematic diagram of the proposed VAE-GAN framework is presented in Fig. 6.12.

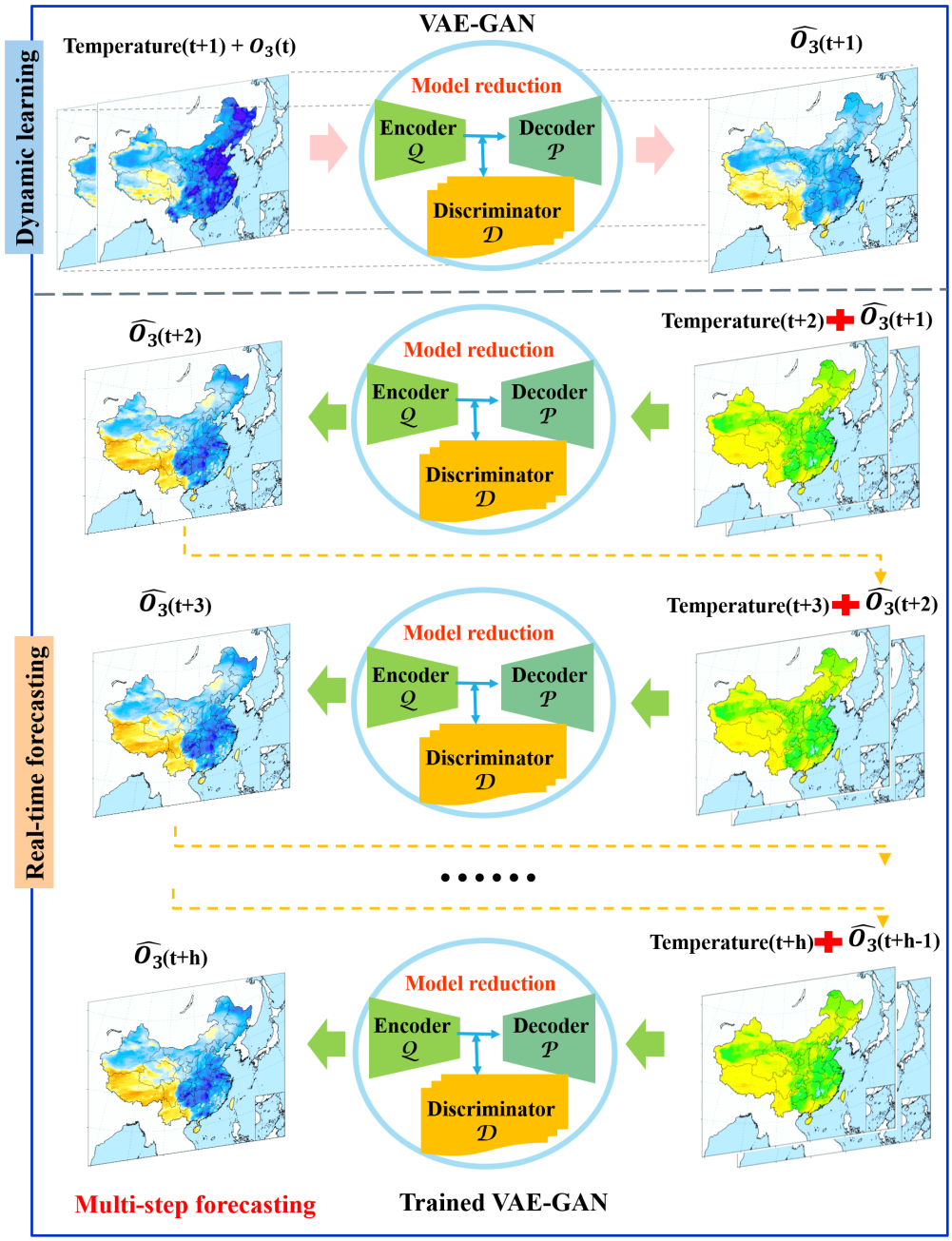


Fig. 6.12. Schematic of real-time ozone forecasting. The architecture consists of two processes: dynamic learning and real-time forecasting. In real-time forecasting process, one-step ozone forecast can be obtained by a direct prediction for one-step (hourly/daily) lead-time. To get forecasts for a long lead-time, a recursive strategy is adopted, which uses the last one-step ozone forecast as a new input to predict the next-step ozone concentration.

6.4.1.1 Dynamic learning

Given a large historical dataset $O = \{O_{t-p}, ..., O_t\} \in \Re^{(p+1)\times M_k}$ (where $p+1$ is the historical time length and M_k is the number of nodes) over the whole China area, it is challenging to efficiently predict the ozone concentration in spatial and temporal spaces. Here the variational autoencoder (VAE) machine learning technique, used in a wide variety of applications (Gonzalez and Balajewicz, 2018a), is introduced with the scope of dimensionality reduction. The architecture of VAE (including an encoder $\mathcal{Q}$ and a decoder $\mathcal{P}$) in the VAE-GAN model is used to identify low-dimensional representations of ozone concentration maps as they evolve in time. In comparison to traditional reduced order models (ROMs) using proper orthogonal decomposition (POD) (Xiao et al., 2019b), the VAE has a nonlinear activation function for its nonlinearly weighted input, thus better representing the nonlinear physical processes (Cheng et al., 2020b).

A GAN consists of two modules: a generator $\mathcal{G}$ and a discriminator $\mathcal{D}$. The generator is designed to produce solutions from a random dataset while the discriminator is tasked to discriminate the real samples from the generated solutions. The VAE-GAN is a probabilistic autoencoder that uses the GAN framework as a variational inference algorithm (Makhzani et al., 2015). In the hybrid VAE-GAN (as shown in the middle column of Fig. 6.12), the inputs μ and the real solutions O_d (the targeted outputs) are fed into the encoder $\mathcal{Q}$, which transforms the high-dimensional datasets into low-dimensional representations ζ and ζ_d respectively. The discriminator $\mathcal{D}$ then discriminates the low-dimensional representations ζ and ζ_d in an efficient way. The latent states ζ_d and ζ are then transformed into the reconstructed samples $\widehat{O_d}$ (the reconstructed outputs) and real solutions $\widehat{O}$ (the predicted outputs) respectively in the decoder $\mathcal{P}$.

For the encoder, I use a deep convolutional neural network (CNN) where the model inputs are the meteorological conditions (temperature T_{t+1} at the predictive time level) and historical ozone dataset O while the targeted outputs are the ozone concentrations O_{t+1} at

the predictive time level. The CNNs exhibit superior performance in extracting the high nonlinearity and chaotic nature from the inputs with a two-dimensional structure (Bolton and Zanna, 2019). The encoder has four convolutional layers followed by one densely connected layer. Each hidden layer is followed by a Leaky rectified linear unit (Leaky ReLU) activation function to extract feature maps from inputs. To avoid dimensionality loss of the edges of images, the padding operation is applied in convolutional layers. After the hidden layers, the fully connected layer is applied to further reduce the dimension of features from its previous layer and find the most task-relevant features for inference. This hierarchical convolutional feature learning acts as nonlinear model reduction, which finally transfers the inputs μ and targeted output O_d into the low dimensional latent states ζ and ζ_d respectively. It is shown in Table 6.1 that the size of datasets on ozone spatial distributions is reduced from 47, 232 to 64 in the latent space, i.e. a decrease of three orders of magnitude.

The discriminator is then used for distinguishing the real latent state ζ_d from the generated latent state ζ at the reduced space. The latent states ζ and ζ_d are processed using five hidden layers, each using a densely connected layer with a Leaky ReLU activation function. The output layer is densely connected to the final hidden layer with a sigmoid activation function (Mosser et al., 2017; Lee and You, 2019), to yield a number between 0 and 1 representing the probability that the ζ is a fake sample created from inputs μ (as opposed to a targeted output O_d).

For the decoder, it is a reverse process for reconstructing the latent states ζ and ζ_d to the high-dimensional states $\widehat{O}$ and $\widehat{O_d}$. The decoder has four hidden layers including one densely connected layer and three convolutional layers, each layer followed by a Leaky ReLU activation function. The final layer uses a tanh activation (Mosser et al., 2017; Lee and You, 2019) with outputs $\widehat{O}$ and $\widehat{O_d}$ shown in Table 6.1.

Table 6.1: Architecture of encoder, decoder and discriminator in the VAE-GAN.

Module	Layer
Encoder $\mathcal{Q}$	Input (47232)
	Conv2D, leaky ReLU ($\alpha = 1$)
	Conv2D, leaky ReLU ($\alpha = 1$)
	Conv2D, leaky ReLU ($\alpha = 1$)
	Full connected
	Output (64)
Decoder $\mathcal{P}$	Input (64)
	Full connected, leaky ReLU ($\alpha = 1$)
	Conv2D, leaky ReLU ($\alpha = 1$)
	Conv2D, leaky ReLU ($\alpha = 1$)
	Conv2D, leaky ReLU ($\alpha = 1$)
	Conv2D, Tanh
	Output (47232)
Discriminator $\mathcal{D}$	Input (64)
	Full connected, leaky ReLU ($\alpha = 1$)
	Full connected, leaky ReLU ($\alpha = 1$)
	Full connected, leaky ReLU ($\alpha = 1$)
	Full connected, leaky ReLU ($\alpha = 1$)
	Full connected, leaky ReLU ($\alpha = 1$)
	Full connected, Sigmoid
	Output (1)

Now the question is how can we train a forecasting model? The VAE-GAN is intuitively similar to variational autoencoders, with the key difference that the VAE-GAN replaces the Kullback-Leibler divergence penalty of VAEs with the adversarial loss described below (Makhzani et al., 2015). The training process involves the minimization of reconstruction loss between the reconstructed outputs $\widehat{O}$ and targeted ones O_d, as well as the maximization of adversarial loss in distinguishing between the latent states ζ and ζ_d. The reconstruction loss jointing the encoder and the decoder is expressed as:

$$\mathcal{L}_{rec}(\theta, \phi) = \left\| O_d - \mathcal{P}(\mathcal{Q}(O_d, \theta), \phi) \right\|, \tag{6.13}$$

and the adversarial loss jointing the encoder, decoder and discriminator is calculated as:

$$\mathcal{L}_{adv}(\theta, \phi, \psi) = E_{\zeta \sim p_\zeta(\zeta)}(\log \mathcal{D}(\mathcal{Q}(\mu))) + E_{\zeta_d \sim p_{\zeta_d}(\zeta_d)}(\log(1 - \mathcal{D}(\mathcal{Q}(O_d)))), \tag{6.14}$$

where θ, ψ, ϕ represent the parameters in the encoder $\mathcal{Q}$, discriminator $\mathcal{D}$ and decoder $\mathcal{P}$, respectively.

Typically, the VAE-GAN uses the hybrid loss ($\mathcal{L}_{rec} + \mathcal{L}_{adv}$) as the objective minimization-maximization function as follows:

$$\min_{\theta,\phi} \max_{\psi} E_{\zeta \sim p_\zeta(\zeta)}(\log \mathcal{D}(\mathcal{Q}(\mu))) + E_{\zeta_d \sim p_{\zeta_d}(\zeta_d)}(\log(1 - \mathcal{D}(\mathcal{Q}(O_d)))) + \left\| O_d - \mathcal{P}(\mathcal{Q}(O_d, \theta), \phi) \right\|. \tag{6.15}$$

The model can be minimized and maximized by stochastic gradient descent with back-propagation. This means that the loss function derivation is propagated throughout the network using the chain rule to update the model parameters θ, ϕ, ψ. In this work, I choose the adaptive moment estimation (Adam) as an adaptive stochastic gradient descent (SGD) algorithm, which has been proven to be efficient for different types of deep networks. Once the training process is completed, the model parameters (including weights and bias) are saved for real-time forecasting.

6.4.1.2 Real-time forecasting

The multi-step-ahead time series forecasting can be described as an estimation of future time series $\{O_{t+1}, ..., O_{t+H}\}$. Referring to the work of Cheng et al. (2021), the multi-step-ahead ozone forecasting process can be summarized as consisting of the following algorithmic steps:

(1) The forecasting model of VAE-GAN is trained to perform a new one-step-ahead forecast $\widehat{O}_{t+1}$ during the predictive period ($t + 1 \in (t_f, t_N)$);

(2) Selecting the new input temperature T_{t+2} at predictive time level $(t + 2)$;

(3) Using the one-step predicted $\widehat{O}_{t+1}$ and temperature T_{t+2} as new inputs, to forecast the next-step-ahead ozone concentration at predictive time level $(t + 2)$ by the trained model;

(4) Repeating steps (1-3) for H times at the multi-step forecasting strategy as shown in Fig. 6.12;

(5) Finally, the length-H timesteps of forecasts at point k ($k \in [0, M_k]$) are obtained as

$$\widehat{O}^k_{t+h} = \begin{cases} \{\mathcal{Q}, \mathcal{P}\}(O^k_{t-p+h-1}, ..., \widehat{O}^k_{t+1}, ..., \widehat{O}^k_{t+h-1}, T^k_{t+h}), & (h \in \{2, ..., p+1\}) \\ \{\mathcal{Q}, \mathcal{P}\}(\widehat{O}^k_{t-p+h-1}, ..., \widehat{O}^k_{t+h-1}, T^k_{t+h}), & (h \in \{p+2, ..., H\}) \end{cases}.$$

I use the Python library Keras (Gulli and Pal, 2017) with the TensorFlow 1.1.0 (Abadi, 2016) backend for all experiments of the proposed model. The performance of the VAE-GAN model developed in this study is assessed using the root mean squared error (RMSE), coefficient of correlation R, normalized mean bias (NMB) and normalized mean error (NME) (Emery et al., 2017) which are defined as follows:

$$RMSE = \sqrt{\frac{\sum_{j=1}^{N} (\widehat{O}_j - O_{d,j})^2}{N}}, \tag{6.16}$$

$$R = \frac{\sum_{j=1}^{N} (\widehat{O}_j - \overline{\widehat{O}})(O_{d,j} - \overline{O}_d)}{\sqrt{\sum_{j=1}^{N} (\widehat{O}_j - \overline{\widehat{O}})^2} \sqrt{\sum_{j=1}^{N} (O_{d,j} - \overline{O}_d)^2}}, \tag{6.17}$$

$$NMB = \frac{\sum_{j=1}^{N} (\widehat{O}_j - O_{d,j})}{\sum_{j=1}^{N} O_{d,j}} \times 100, \tag{6.18}$$

$$NME = \frac{\sum_{j=1}^{N} |\widehat{O}_j - O_{d,j}|}{\sum_{j=1}^{N} O_{d,j}} \times 100, \tag{6.19}$$

where the subscript j represents the pairing of N targeted ozone concentration O_d and

predictions $\widehat{O}$ by points ($[0, M_k]$) and time ($[t_f, t_N]$), and the overbars signify means over points and/or time.

6.4.2 Results

6.4.2.1 Data processing

During the training and validation processes, the input and output datasets are pre-processed by a Standard Scaler normalization method (Buitinck et al., 2013). The ozone concentration and meteorological factors are scaled to a specific interval prior to training the VAE-GAN model. In this study, the description of the training and validation datasets in hourly and daily ozone forecasting is provided in Table 6.2.

As shown in Table 6.2, I select the reanalysis datasets in 2017 for model training and 2018 for validation in hourly ozone forecasting (in section 6.4.2.2 and section 6.4.2.3). During the validation period, the reanalysis data is used for initializing ozone forecasting. I compare the forecasting results with the reanalysis data, monitoring observations and results from the physical model-NAQPMS in the six regions of whole China (North, Northeast, Southwest, Southeast, Northwest, and Central) as well as highly-populated regions (Jing-Jin-Ji, Yangtze River Delta and Pearl River Delta) (in section 6.4.2.2 and section 6.4.2.3). In daily ozone forecasting, I use the reanalysis data from the year 2013 to 2017 for model training and the reanalysis data from the year 2018 to 2019 for model validation (in section 6.4.2.4 and section 6.4.2.5).

Table 6.2: Summary of training and validation datasets used in hourly and daily ozone forecasting (where the model inputs are the meteorological conditions (temperature, humidity and wind speed) and the ozone concentrations at the last time level while the targeted outputs are the ozone concentrations at the predictive time level.)

Scale	Training dataset	Validation dataset
Hourly forecasting	Reanalysis data (June and December of 2017)	Reanalysis data (June and December of 2018)
		Monitoring data (June of 2018)
		Physical modelling data (June of 2018)
Daily forecasting	Reanalysis data (2013-2017)	Reanalysis data (2018-2019)

6.4.2.2 Hourly ozone forecasting analysis over China in summer and winter

In this section, I demonstrate the trained VAE-GAN model has the capability of capturing the details of the spatial distribution of ozone concentrations over China. For this purpose, the post-processed reanalysis dataset with a high spatial resolution (15km $\times$ 15km) is thus chosen as the 'true' reference during the validation period (see Table 6.2). The trained VAE-GAN is used for hourly ozone forecasting in June and December 2018. A comparison between the predicted ozone concentrations and reanalysis data as well as the corresponding error estimation are carried out in the followings.

6.4.2.2.1 Spatiotemporal hourly forecasting of ozone concentration

The spatial distribution of hourly ozone concentrations in June and December in 2018 is shown in Fig. 6.13. It can be seen that the VAE-GAN model captures a large part of the spatial distribution of ozone concentration at lead-times $t = 30, 60h$ in June and December, respectively. It is noted that the ozone concentration in June is higher than that in December. This can be explained by the fact that a high temperature leads to increased convection, unstable atmosphere stratification, and increased precipitation, which are beneficial to the diffusion and deposition of ozone (Yan et al., 2021). By contrast, the low temperature in December causes a surface inversion, and the height of the atmo-

spheric mixed layer is low, which is not conductive to vertical convection. Compared to the reanalysis data, in June, the predicted ozone concentration is somewhat underestimated over the central of China at different lead-times, while slightly overestimated over the North-east and South-east area close to the North Pacific Ocean.

The last column of Fig. 6.13 shows the residual maps between the ozone distributions predicted by the VAE-GAN and reanalysis data. Table 6.3 indicates that the mean residual varies from $-12.4ug/m^3$ (June) and $-6.91ug/m^3$ (December) at the lead-time of $t = 30h$ to $-5.88ug/m^3$ (June) and $-1.34ug/m^3$ (December) at the lead-time of $t = 60h$ respectively. It is found that 10% and 90% residuals in June are higher than those in December, that is, the predicted results using the VAE-GAN have lower accuracy in June in comparison to that in December. This may be caused by the complex and high ozone heterogeneity in the spatial distribution in June. Overall, the trained VAE-GAN has captured the spatial characteristics of ozone concentrations for given new inputs at a lead-time beyond the training period. Although a discrepancy between the predicted ozone concentrations and reanalysis data, the VAE-GAN is able to provide the same level of accuracy as that of the physical model-NAQPMS. To improve the predicted accuracy during the high ozone episode ($\geq 160ug/m^3$), the convolutional long short-term memory (ConvLSTM) approach is introduced to the VAE-GAN model. For details, see B.1.

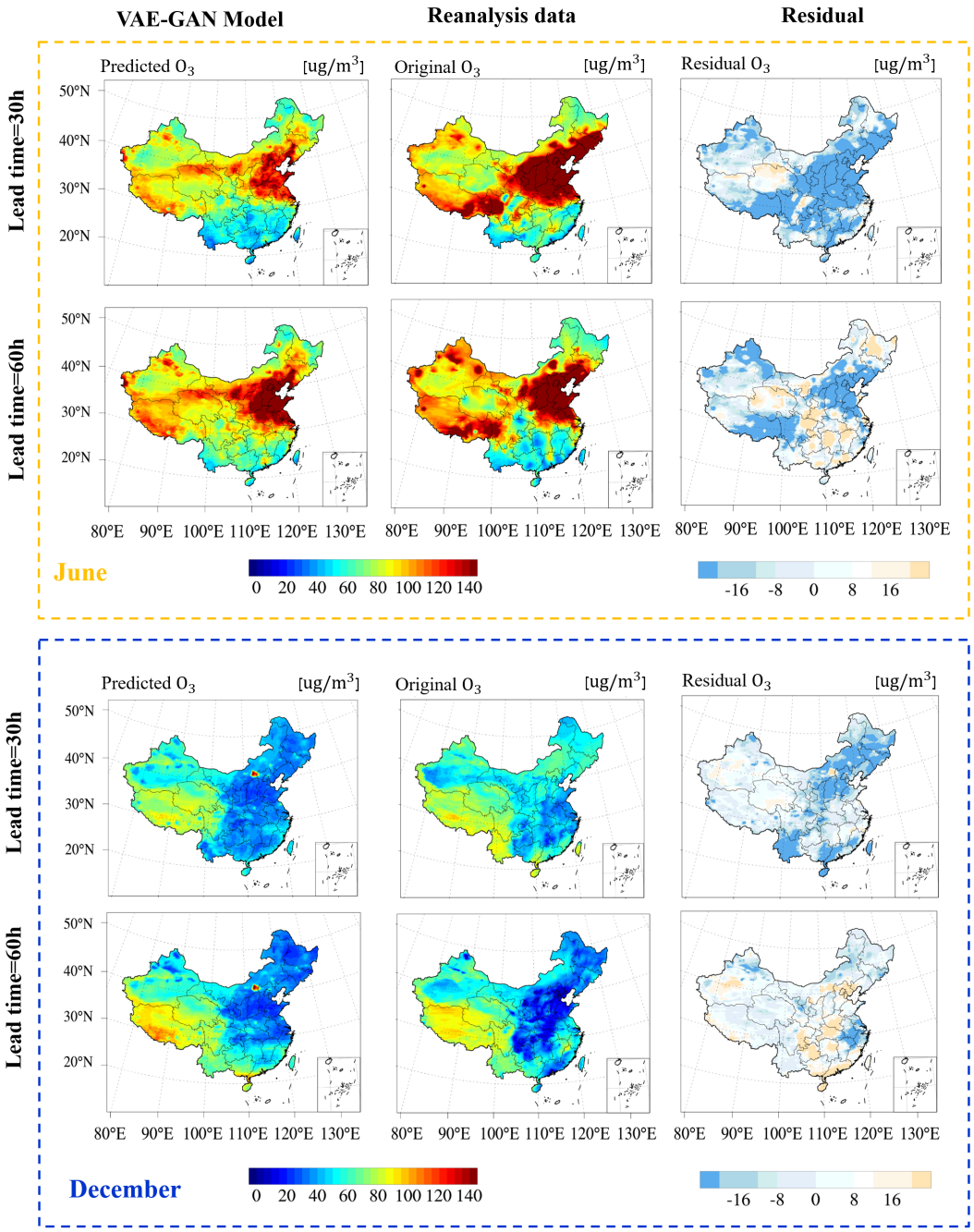


Fig. 6.13. Comparison of the spatial distribution of ozone fields obtained from the VAE-GAN model (left) and the reanalysis data (middle) at lead-times of $30h$ and $60h$ in June and December in 2018.

Table 6.3: Residuals of ozone fields between the VAE-GAN model and the reanalysis data at the lead-times $t = 30, 60h$ in June and December in 2018.

Indicators	June		December	
	Lead-time = 30h	Lead-time = 60h	Lead-time = 30h	Lead-time = 60h
Mean residual (ug/m^3)	-12.4	-5.88	-6.91	-1.34
90th percentile (ug/m^3)	13.95	23.44	8.92	15.09
10th percentile (ug/m^3)	-40.81	-32.30	-24.58	-18.39

6.4.2.2.2 Error analysis of hourly ozone forecasting

Further error analysis of hourly ozone forecasting is carried out through the RMSE and correlation coefficient. The temporal variations of spatial-averaged RMSEs and correlation coefficient of hourly predictive results from the VAE-GAN are shown in Fig. 6.14. It can be noticed that the spatial-averaged RMSE values in June and December in 2018 are generally around $22ug/m^3$ and $13ug/m^3$ during the entire forecasting period. The corresponding correlation coefficient values are shown in Fig. 6.14(b). It can be seen that the ozone concentration predicted by the VAE-GAN model shows a good correlation with the reanalysis data over a long lead-time. Most of correlation coefficients are around or beyond 0.6 during the forecasting period in June. In addition, it is worth noting that the magnitude of correlation coefficients in December is close to 0.9. This indicates that the VAE-GAN performs better in hourly ozone forecasting in December.

Fig. 6.15 illustrates the spatial distribution of temporal-averaged RMSEs. It can be noted that most of the temporal-averaged RMSEs are below $20ug/m^3$ in June and $16ug/m^3$ in December, except for the North-east areas where the largest errors exist due to complex factors including human activities, meteorology and industries influencing the ozone concentration. From the cumulative distribution of temporal-averaged RMSEs in both months (Fig. 6.15(c)), it is shown that the temporal-averaged RMSEs of 80% points are smaller than $25ug/m^3$. In addition, it is clear that the mean values of temporal-averaged RMSEs are $17.23ug/m^3$ and $9.88ug/m^3$ for June and December respectively.

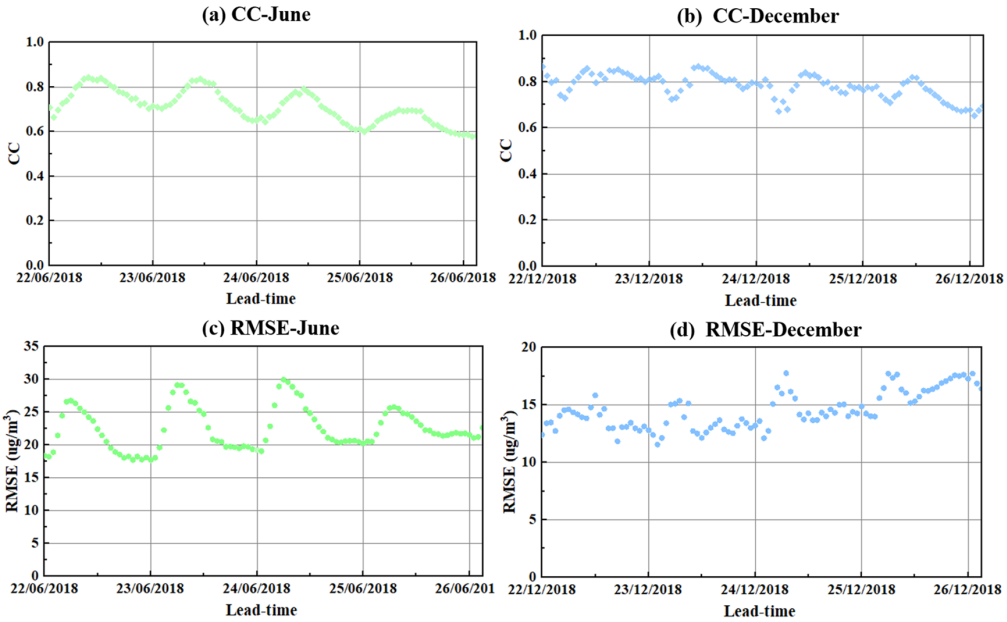


Fig. 6.14. The spatial-averaged RMSEs and correlation coefficients of ozone concentration between the VAE-GAN and the reanalysis data in June and December in 2018.

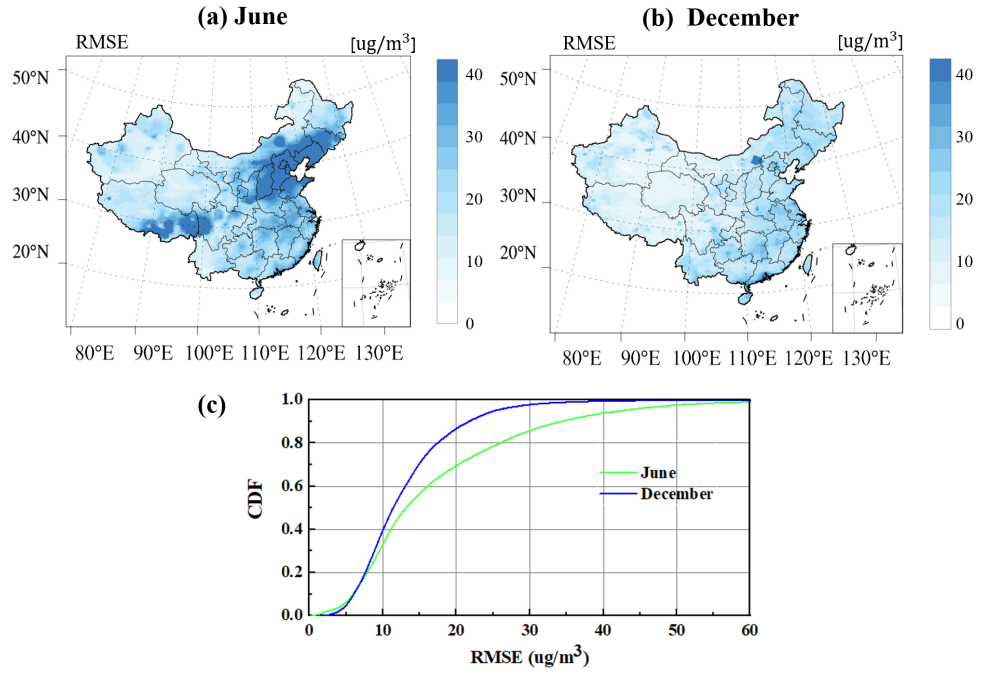


Fig. 6.15. The spatial distribution of temporal-averaged RMSEs and the comparison of the cumulative density functions (CDF) for the temporal-averaged RMSEs in June and December in 2018.

6.4.2.3 Comparison of hourly ozone forecasting between VAE-GAN and the physical model-NAQPMS

In this section, to further demonstrate the forecasting performance of the VAE-GAN, the predictive accuracy of the VAE-GAN is evaluated with an existing air quality physical model, that is, the Nested Air Quality Prediction Modelling System-NAQPMS, which is commonly used for real-time pollution forecasting in China. The comparison of ozone results obtained from the VAE-GAN and NAQPMS has been undertaken through (1) the spatial distribution and temporal variation of ozone concentrations over China, where the reanalysis data is considered as the 'true' reference; and (2) ozone concentration at monitoring stations, where the monitoring measurement is used as the 'true' reference.

6.4.2.3.1 Spatiotemporal distribution of ozone forecasting compared to physical modelling

The spatial distribution of hourly ozone forecasting results obtained from the VAE-GAN and physical model-NAQPMS at lead-times $t = 40, 60h$ is plotted in Fig. 6.16. It can be seen that the VAE-GAN and physical model-NAQPMS exhibit almost the same level of acceptable performance for hourly ozone forecasting. Compared with the physical model-NAQPMS, the VAE-GAN can better capture the spatial characteristics of high ozone concentrations in densely populated areas in China.

From Fig. 6.16, it can be noted that the spatiotemporal distributions of ozone concentrations have a high spatial heterogeneity across China. For further analysis of temporal variation, the whole country is thus divided into six regions: the North China Plain (NCP), Northeast China (NE), Southwest China (SW), Southeast China (SE), Northwest China (NW), and Central China (shown in Fig. 6.17). Fig. 6.17 shows the spatial-averaged RMSEs of predicted results in six regions during the predictive period of June in 2018. It is noted that the predicted results from both the physical model and VAE-GAN have a

low value of RMSE (below $30ug/m^3$) in the NW region. The highest RMSE of predicted ozone concentration occurs in the NCP region (Fig. 6.17(c)), which is consistent with the findings in Zhan et al. (2018). The spatial heterogeneity of ozone concentrations in summer is associated with the spatial pattern of humidity conditions. The drier condition in the NCP region leads to a high-level accumulation of ambient ozone. The predicted ozone concentrations from both the physical model-NAQPMS and VAE-GAN are underestimated in the NCP, thus resulting in the highest RMSE. However, the VAE-GAN model generally performs better than the physical model-NAQPMS in the NCP region, which can be seen in Fig. 6.16. In other regions (including NE, Central, SE and SW regions), the performance of the VAE-GAN model is comparable with the physical model-NAQPMS.

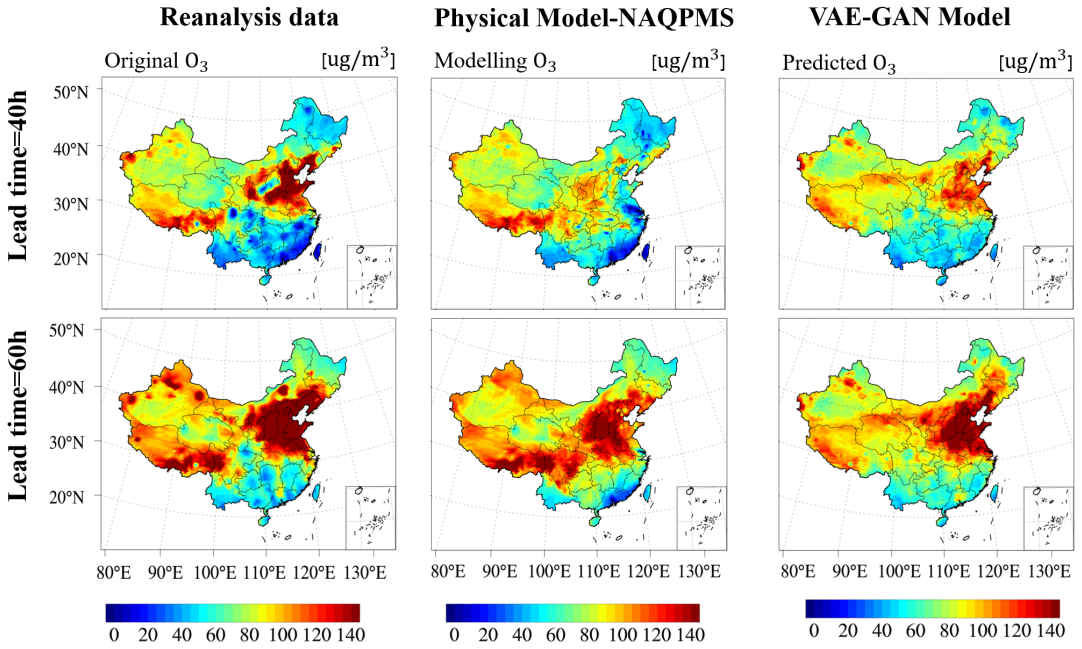


Fig. 6.16. Comparison of the spatial distribution of ozone fields obtained from the VAE-GAN model (right), physical model-NAQPMS (middle) and the reanalysis data (left) at lead-times $t = 40, 60h$ in June of 2018 respectively.

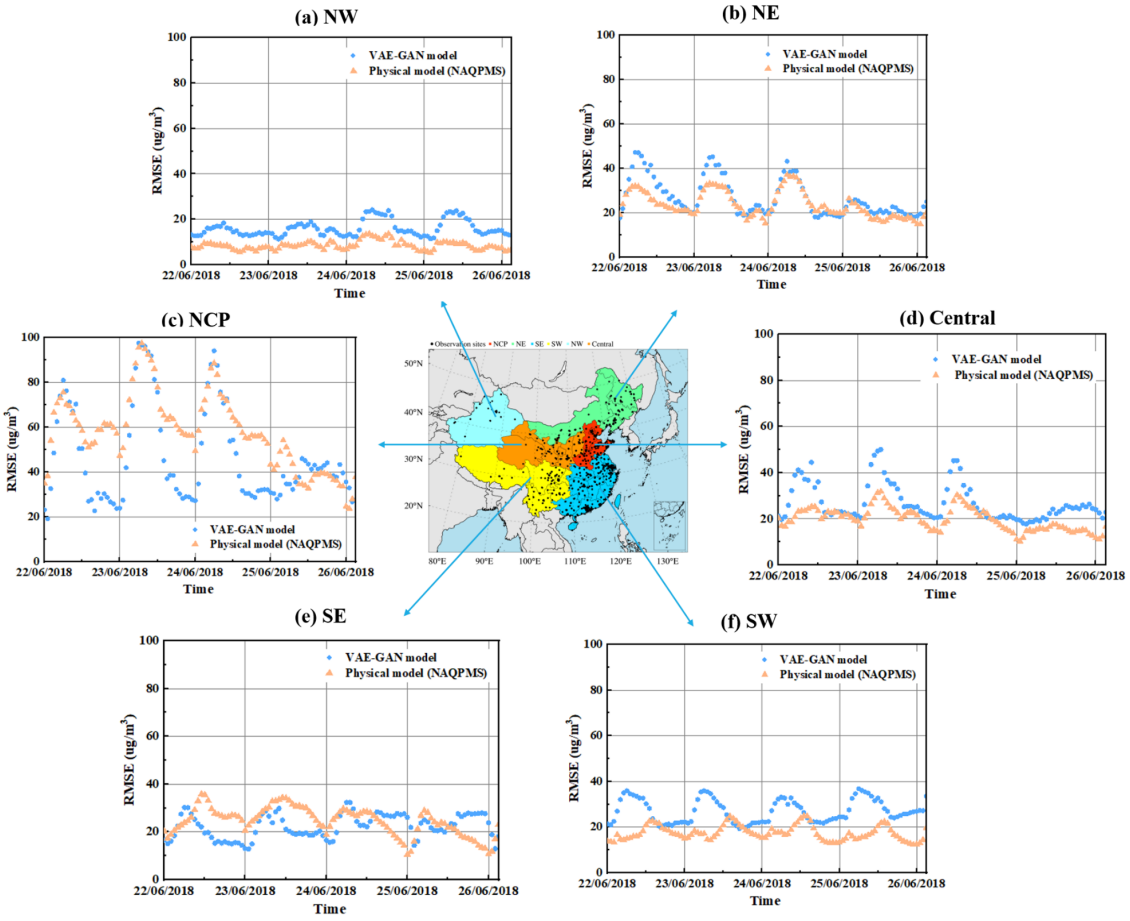


Fig. 6.17. The spatial-averaged RMSE of predictive results from the VAE-GAN and physical model-NAQPMS during predictive period of June in 2018. There are six regions: the North China Plain (NCP), Northeast China (NE), Southwest China (SW), Southeast China (SE), Northwest China (NW), and Central China (refer to Kong et al. (2021)). The reanalysis data is used as 'true' reference.

6.4.2.3.2 VAE-GAN-based ozone forecasting compared to physical modelling and observations at monitoring stations

The performance of the VAE-GAN in spatiotemporal forecasting is further evaluated by comparing the predicted ozone concentrations with those from the physical model-NAQPMS and the observations at monitoring stations. Here I choose three study regions in China: Jing-Jin-Ji (JJJ), Yangtze River Delta (YRD) and Pearl River Delta region (PRD) (shown in Fig. 6.18), which are densely populated regions in China. Fig. 6.18 shows the time series of hourly predicted and monitoring ozone concentrations during

22-30 June of 2018 in the major cities located in the three regions. It is noted that the ozone concentration is gradually increasing in the morning and reaches a peak at noon, and has a low value during the night time. This can be explained as follows. The formation of ozone is involved in the processes of photochemical reactions, advection, entrainment, and deposition. The entrainment of ozone into the atmospheric boundary layer greatly contributes to the increase of ozone concentration at day time (Freire et al., 2017), while due to the stable nocturnal boundary layer caused by radiative cooling at night time, the surface ozone concentrations typically reach a minimum level by nitrogen oxide titration and dry deposition (Liu et al., 2020a).

It can be seen that the predicted ozone by the VAE-GAN model is largely in good agreement with observations during the predictive period (22-30 June in 2018). It is noticed the VAE-GAN performs better than the physical model-NAQPMS in Shanghai, Nanjing, Hangzhou, Hefei in the YRD region. The discrepancy between the VAE-GAN predicted results and observations occurs when the peak values of ozone drastically increase at day time due to large uncertainties in traffic, emission inventory and meteorological conditions. To better capture the peak, the VAE-GAN-LSTM model is introduced in B.1. In Fig. B.1, the predicted ozone shows a reduction of underestimation in the ozone peaks compared to that using the VAE-GAN model, and exhibits a good match with the observations in cities with high ozone concentrations, such as Beijing, Tianjin, Shijiazhuang, Nanjing and Hefei.

The performance of the machine learning-based modelling and the physical model has further been evaluated using the statistical error indicators and the corresponding results are shown in Table 6.4. The ozone concentrations predicted by the VAE-based models mostly have a smaller value of RMSE, NMB and NME than that from the physical model-NAQPMS in most of regions in China. The NMB and NME values of predicted results in the JJJ and YRD regions largely lie within $\pm 22\%$ and 25% respectively for the VAE-GAN and VAE-GAN-LSTM models, while those are beyond $\pm 35\%$ (NMB) and 35% (NME)

for the physical model-NAQPMS. The performance of the VAE-GAN model is thus comparable or better than that of the physical model-NAQPMS in the JJJ and YRD regions. In the PRD region, the ozone concentration mostly has a low value of $< 100ug/m^3$ compared to that in the JJJ and YRD regions. The variation of ozone in the PRD region is linked to summer monsoon (Shao et al., 2009) which has not currently been considered in the machine learning-based models, but in physical modelling. This can explain why the performance of the physical model is better than the VAN-based models in the PRD region.

Generally speaking, the VAE-GAN model developed in this study is relatively reliable when used for ozone forecasting during a long lead-time. Our results suggest that the well-trained VAE-GAN has captured the spatiotemporal characteristics of the ozone concentration variation. In comparison to the physical model-NAQPMS, the VAE-GAN model can provide a high efficient ozone forecast while the same level of accuracy is achieved. This highlights the potential of the VAE-GAN model to be applied in conjunction with many real-time applications, for example, early warning for air pollution emergencies.

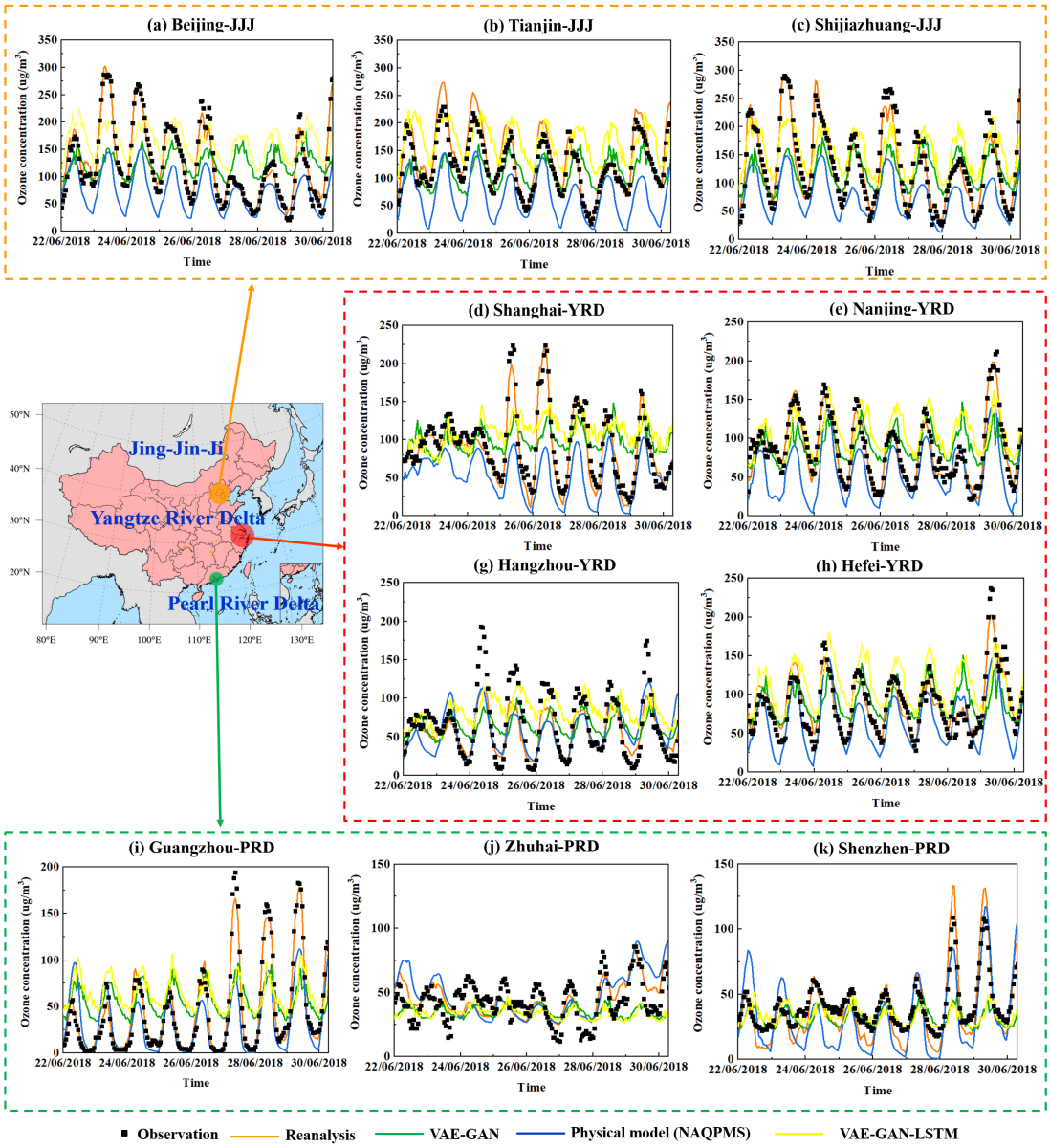


Fig. 6.18. Ozone time series at the cities in three densely populated regions (Jing-Jin-Ji, Yangtze River Delta, and Pearl River Delta) using the observations, reanalysis data, physical model-NAQPMS, VAE-GAN and VAE-GAN-LSTM models.

Table 6.4: Error analysis of results from the reanalysis data, the physical model-NAQPMS, VAE-GAN and VAE-GAN-LSTM models where the observations in June of 2018 are taken as references (true values).

Zone		Correlation Coefficient				RMSE (ug/m^3)			
		Reanalysis	Physical model (NAQPMS)	VAE-GAN	VAE-GAN -LSTM	Reanalysis	Physical model (NAQPMS)	VAE-GAN	VAE-GAN -LSTM
Jing-Jin-Ji	Beijing	0.98	0.83	0.52	0.57	20.02	63.38	43.34	57.98
	Tianjin	0.97	0.91	0.68	0.76	18.62	72.33	56.93	53.55
	Shijiazhuang	0.97	0.75	0.50	0.34	13.09	50.39	40.81	46.28
Yangtze River Delta	Shanghai	0.97	0.83	0.55	0.56	9.68	38.56	34.97	39.52
	Nanjing	0.95	0.74	0.58	0.37	22.12	29.64	35.89	42.68
	Hangzhou	0.95	0.79	0.54	0.58	13.24	32.41	33.05	40.29
	Hefei	0.99	0.83	0.58	0.48	6.98	30.21	42.25	46.38
Pearl River Delta	Guangzhou	0.98	0.82	0.42	0.36	10.90	18.18	18.66	18.64
	Shenzhen	0.68	0.46	0.27	0.23	11.82	17.73	17.05	17.54
	Zhuhai	0.49	0.37	0.33	0.52	42.13	52.11	21.59	19.31
Zone		NMB (%)				NME (%)			
		Reanalysis	Physical model (NAQPMS)	VAE-GAN	VAE-GAN -LSTM	Reanalysis	Physical model (NAQPMS)	VAE-GAN	VAE-GAN -LSTM
Jing-Jin-Ji	Beijing	12.18	-45.58	-18.30	23.29	12.88	45.58	23.95	29.23
	Tianjin	-5.65	-46.85	-22.62	0.87	9.38	46.85	28.29	22.76
	Shijiazhuang	-6.23	-44.04	-18.88	-11.96	8.69	44.04	21.96	19.25
Yangtze River Delta	Shanghai	-3.70	-34.30	-18.68	-3.07	6.94	34.69	23.86	19.60
	Nanjing	-26.95	-29.85	-37.45	-22.40	26.95	30.44	37.78	25.66
	Hangzhou	-3.75	-24.21	-14.98	7.24	10.09	26.27	21.77	24.10
	Hefei	-4.87	-42.93	-48.18	-46.01	5.45	42.93	48.18	46.01
Pearl River Delta	Guangzhou	21.88	-4.26	-59.48	-60.34	21.88	14.50	59.48	60.34
	Shenzhen	-2.91	0.52	-56.91	-63.20	2.91	2.42	56.91	63.20
	Zhuhai	22.42	27.09	-13.13	-7.46	23.99	28.83	14.63	11.82

6.4.2.4 Daily ozone forecasting analysis over China

In this section, the capability of the VAE-GAN is further demonstrated in daily ozone forecasting of the year 2018 in JJJ, YRD and PRD regions. The reanalysis datasets are split into the training dataset from the year 2013 to 2017 and the predictive dataset in 2018.

Fig. 6.19 shows the temporal variation of spatial-average daily ozone concentrations over the JJJ, YRD, and PRD regions during the 2018 predictive period. As shown in Fig. 6.19, it can be noticed that the predicted ozone concentration exhibits good agreement with the

reanalysis data in the JJJ and YPR regions. In the PRD region, the VAE-GAN model captures the daily variability, but overestimates the ozone in the summer. This could be explained by the responses between the meteorological conditions and ozone. In the regions of JJJ and YRD, the daily variation of ozone concentration shows the seasonal variation (an inverse U-shaped trend, which is highly related to the temperature) (Yang et al., 2020), where the highest and lowest ozone concentrations occur in Summer and Winter respectively. However, the trend of the ozone seasonal variation is weak in South China, including the PRD region. This is due to the warmer temperatures throughout the year and the Asian Summer monsoon, which brings cloudy and raining weather, marine air and strong deep convection and other unfavorable factors for ozone formation and accumulation (Lu et al., 2018; Yang et al., 2020; Ma and Yin, 2021). It is a challenge in existing air quality models when taking into account these complex meteorological factors. This can be seen in Table 4 that a poor correlation-prevails (average 40%) between the predicted results and monitoring data in the PRD exists for both the proposed VAE-GAN and physical model-NAQPMS models. Coupling of the complex meteorological conditions (e.g. monsoon) into the VAE-GAN training process will be explored to improve predictive accuracy in future studies.

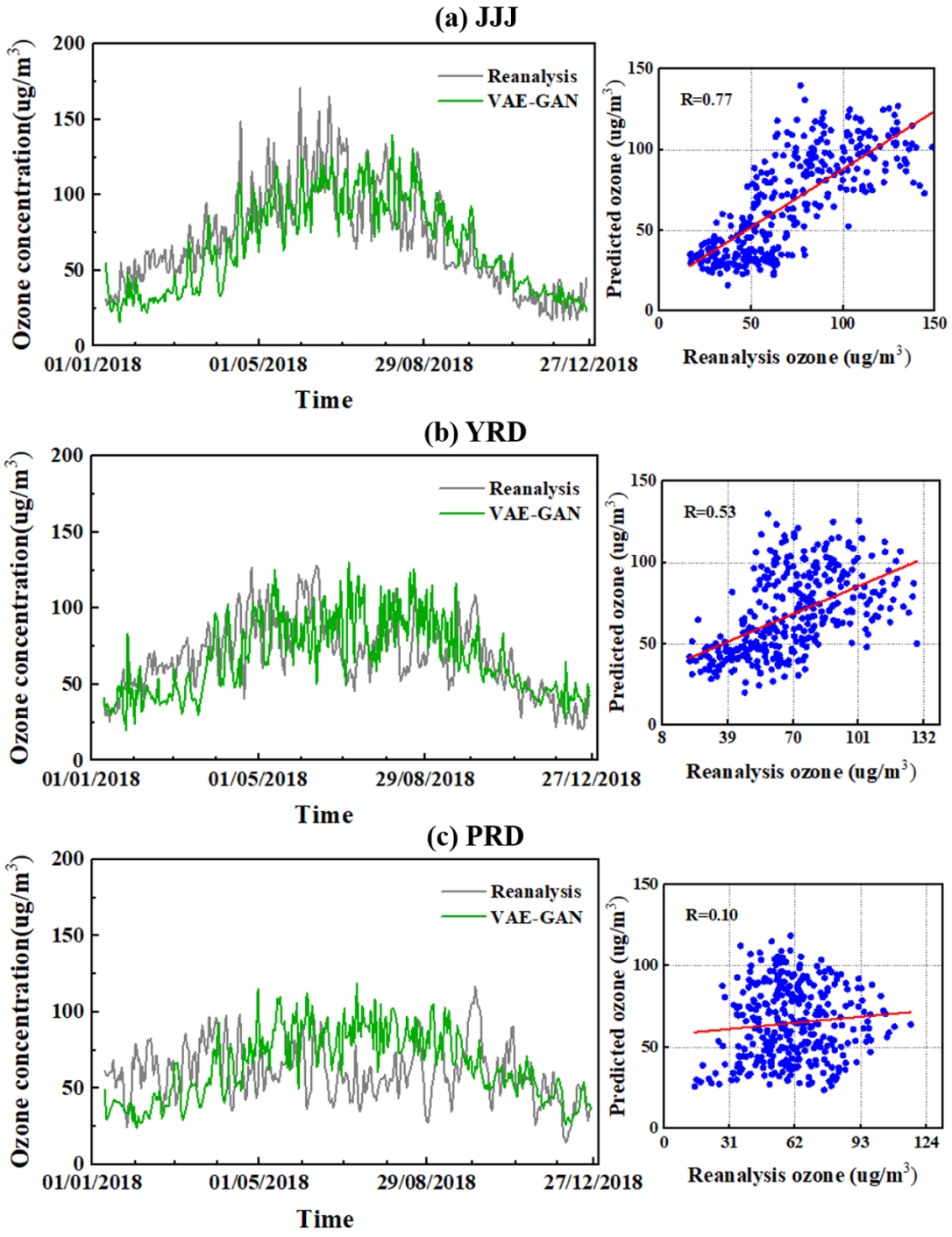


Fig. 6.19. A comparison of the temporal variation of spatial-average daily ozone concentrations from the VAE-GAN model and the reanalysis data in 2018.

Fig. 6.20 shows the spatial distribution of temporal-average daily ozone concentrations over the predictive period (the whole year of 2018). The predicted results from the VAE-GAN are generally comparable to the reanalysis data in JJJ, YRD and PRD regions. It

can be seen that the VAE-GAN model captures a large part of the spatial distribution of ozone concentrations. To further evaluate the VAE-GAN results, the relative errors of the predicted results over the three regions are presented in Fig. 6.21. Reasonable predictive accuracy is largely achieved with a low relative error of around $\pm 10\%$. Overall, the VAE-GAN model skilfully captures the spatial and temporal variation of ozone concentration during the predictive period.

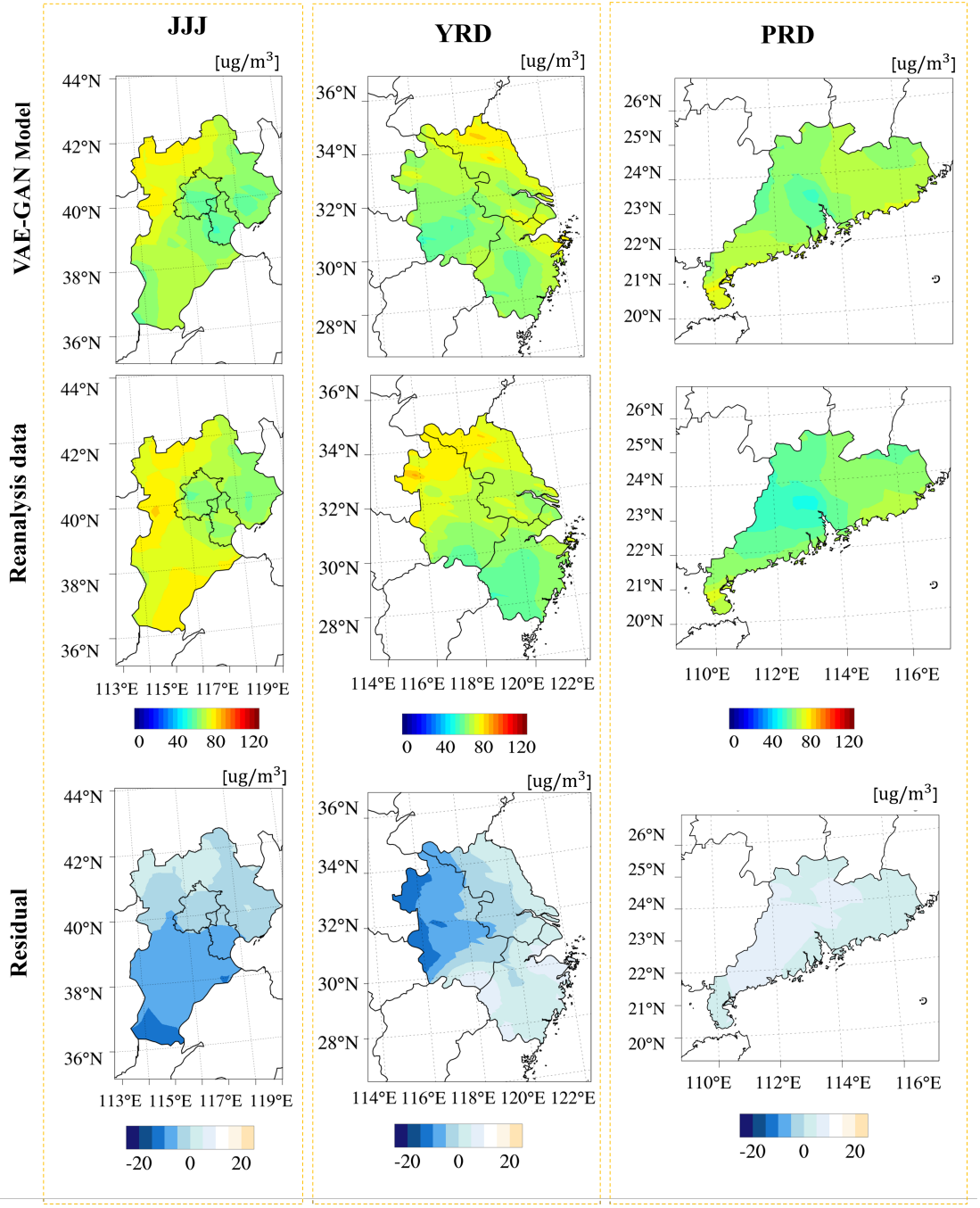


Fig. 6.20. A comparison of the spatial distribution of temporal-average daily ozone concentrations from the VAE-GAN model and the reanalysis data in the JJJ, YRD and PRD regions.

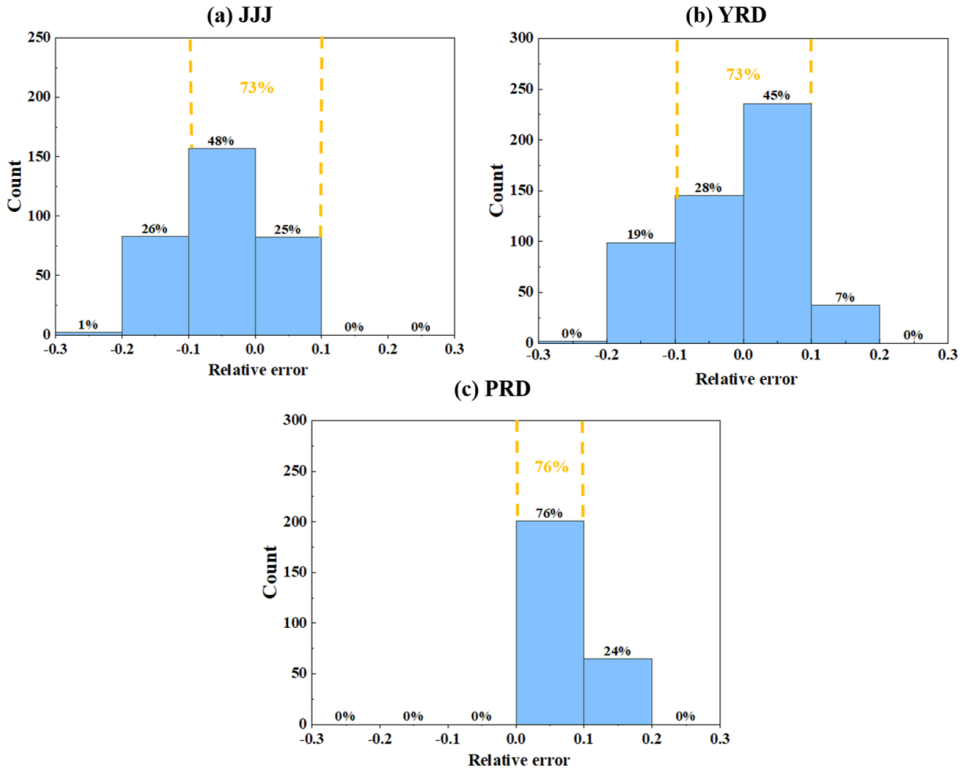


Fig. 6.21. A relative error histograms of the temporal-average daily ozone concentrations between the VAE-GAN model and the reanalysis data over the whole year of 2018. The blue bins aggregate the respective grid points over three regions. Numbers on top of the bins are percentage values with respect to the total number of grid points. Yellow numbers indicate the area proportion of the globe, where values of relative error are within the $\pm 10\%$ interval.

6.4.2.5 Impact of meteorological factors

In our study cases described above, the temperature is considered as one of model inputs since it is the top ozone-related meteorological condition. As known, other meteorological factors (relative humidity and wind speed) are also important in ozone formation and accumulation. For example, low humidity causes a reduction of cloud cover which contributes to photochemical reaction (Chen et al., 2020b). A high wind speed has a negative impact on ozone concentration by transporting and cleaning air pollutants out of the study areas (Wang et al., 2020c). In this section, two deep learning models (the VAE-GAN and the deep convolutional GAN (DCGAN)) are used for quantifying the impact of the meteorological factors on daily ozone forecasting during the validation period of 2018 and

2019. The model inputs are the meteorological conditions (temperature, relative humidity and wind speed) at the predictive time level and historical ozone concentrations while the targeted output is the ozone concentration at the predictive time level. The reanalysis datasets are used for training and validation purposes: the training datasets compassing the time period between years 2013 and 2017 and validation datasets from 2018 to 2019. The densely populated JJJ region in China is selected as the study area. I estimate the performance of the VAE-GAN and DCGAN models in daily forecasting with consideration of: (1) no meteorological factors (no met); (2) temperature (temp); (3) temp + relative humidity (rh), and (4) temp + rh + wind).

The spatial-averaged RMSEs of predicted ozone concentrations are plotted in Fig. 6.22(a) while the predicted daily ozone concentrations are shown in Fig. 6.22(b). It is shown in Fig. 6.22(b) that without the introduction of the meteorological factors, both of the VAE-GAN and DCGAN models fail to capture the temporal variation of daily ozone concentrations from the year 2018 and 2019.

In DCGAN modelling, it is clearly seen in Fig. 6.22(a) that the more meteorological factors are included in the model inputs, the smaller the spatial-averaged RMSEs of daily ozone results. The DCGAN reaches its best performance when the impact of temperature, relative humidity and wind speed is considered.

In VAE-GAN modelling, the introduction of temperature plays a crucial role in improving the accuracy of ozone forecasting. The VAE-GAN with the consideration of temperature performs best among all the study cases. The VAE-GAN achieves the lowest RMSE value of $23ug/m^3$ in daily ozone concentration prediction in the JJJ region from the year 2018 to 2019. In Fig. 6.22(b), it can be noted that by comparing with the reanalysis data, the VAE-GAN model can better represent the peaks of ozone concentrations than the DCGAN model. However, it is also noticed that in contrast to the DCGAN model, the RMSE of predicted results is decreased when more meteorological factors (relative humidity and wind speed) are considered in the model inputs. This is due to the limitation of the VAE-

GAN model. For multiple variable inputs, the quality of generated outputs suffers from the generator (decoder $\mathcal{P}$) attempting to cover all data manifolds in the data space, thus leading to a poor model performance (Pandeva and Schubert, 2019). Overall, with only one input for prediction, the VAE-GAN performs better than the DCGAN. However, with multi-inputs, the DCGAN could perform better than one input and the VAE-GAN exhibit a descending trend with more inputs. Therefore, the choice of the model could depend on the categories of inputs. Coupling of multiple meteorological inputs with the VAE-GAN will be investigated in our future work.

Overall, the temperature or other meteorological factors play a dominant role in improving the accuracy of ozone forecasting, e.g. reducing uncertainties and accumulative error with every step of ozone forecasting.

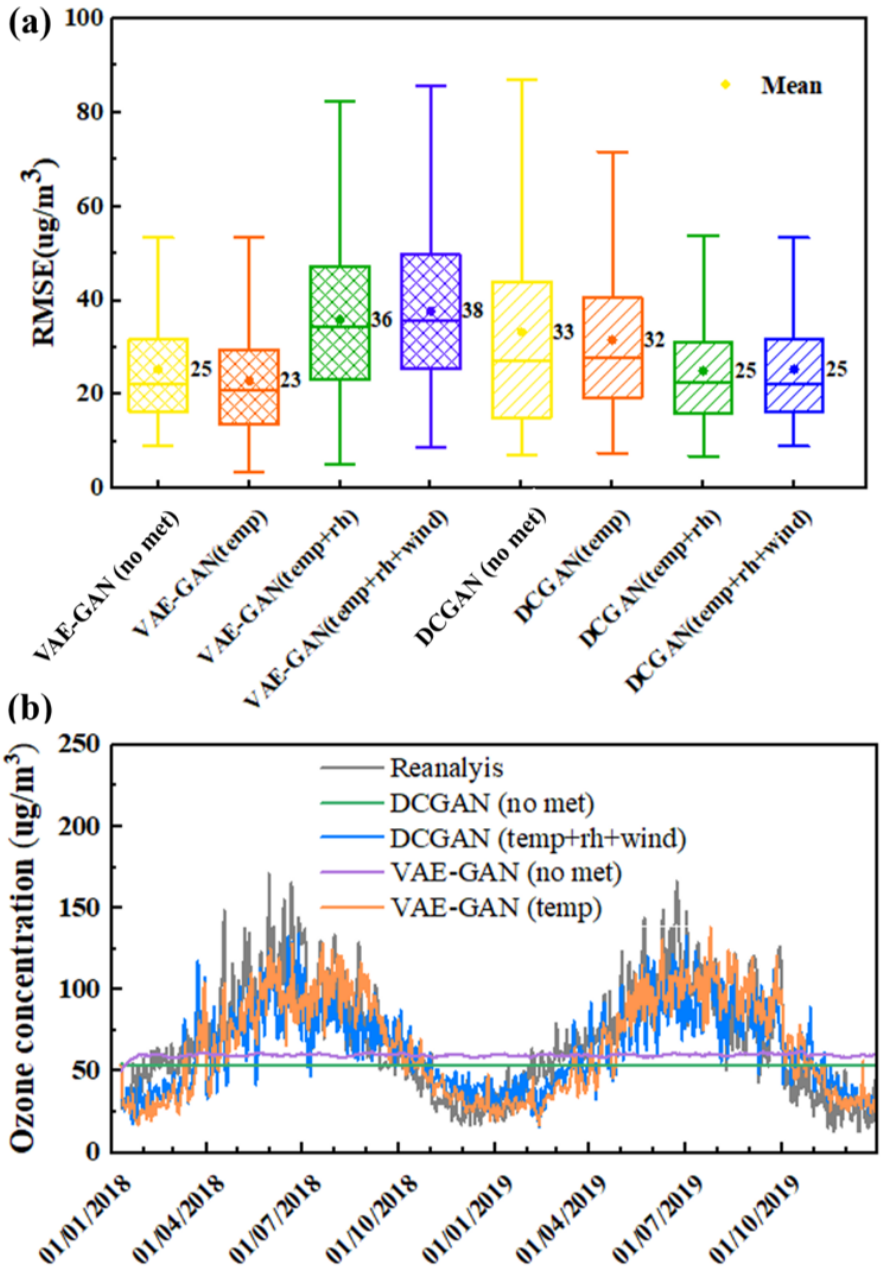


Fig. 6.22. Predictive daily results from the cases of VAE-GAN and DCGAN model with/without the introduction of the meteorological factors into the model inputs, where the densely populated JJJ region in China is selected as the study area and the predictive period is from the year 2018 to 2019. Note that in the case of no meteorological factors the models only take the last-timestep ozone as the model input.

6.5 Conclusions

In this work, I systematically analyse the hourly and daily performances of the ConvLSTM, DCGAN and LDCGAN modelling. In addition, a machine learning-based forecasting framework is implemented for efficient and accurate real-time (hourly and daily) ozone forecasting in China. A hybrid deep learning VAE-GAN approach is proposed to better explore the spatial and temporal features of ozone concentrations, where GAN has the capability of learning hierarchical feature representations of ozone concentrations while VAE is used for dimensionality reduction and avoiding model collapse. The hybrid VAE-GAN model consists of both adversarial training and real-time forecasting processes, which narrow the gaps between the generative and targeted ozone fields from the past to future states and provide accurate ozone forecasts for a long lead-time beyond the training period in a recursive way.

In our ConvLSTM, DCGAN and hybrid VAE-GAN ozone modelling, the training and validation datasets consist of hourly reanalysis datasets at a high spatial resolution of 15km $\times$ 15km across China from 2013 to 2019. Our results illustrate that the machine learning-based prediction exhibits excellent spatiotemporal performance over most of the BTH region in different seasons. In particular, the VAE-GAN model has captured the spatial and temporal evolution patterns of ozone concentrations during the predictive period of 2018 and 2019 in comparison to the reanalysis data, observations and physical model-NAQPMS. The VAE-GAN model can provide a high efficient ozone forecast while the same level of accuracy is achieved compared to the physical model-NAQPMS. However, it is also noticed that a low predictive accuracy of ozone concentrations occurs in some local regions where the daily variation of ozone concentrations exhibits volatility. In the following chapter, we will present a more comprehensive forecasting scheme to improve the predictive accuracy by combining the ConvLSTM and DCGAN with data assimilation techniques for uncertainty reduction.

CHAPTER 7

Ensemble Kalman filter for GAN-ConvLSTM based long lead-time forecasting

Data-driven machine learning techniques have been increasingly utilized in accelerating nonlinear dynamic system prediction. However, machine learning-based models for long lead-time forecasts remain a significant challenge due to the accumulation of uncertainty along the time dimension in online deployment. To tackle this issue, ensemble Kalman filter (EnKF) has been introduced to machine learning-based long term forecast models to reduce the uncertainty of long lead-time forecasts of chaotic dynamic systems. Both the deep convolutional generative adversarial network (DCGAN) and convolutional long short term memory (ConvLSTM) are used for learning the complex nonlinear relationships between the past and future states of dynamic systems. Using an iterative Multi-Input Multi-Output (MIMO) algorithm, the two-hybrid forecast models (DCGAN-EnKF and ConvLSTM-EnKF) are able to yield long lead-time forecasts of dynamic states. The performance of the hybrid models has been demonstrated by a Lorenz 96 model. Our results show that the use of EnKF in ConvLSTM and DCGAN models successfully corrects online model errors and significantly improves the real-time forecasting of dynamic

systems for a long lead-time.

This chapter is derived from and expands upon Cheng et al. (a).

7.1 Introduction

In this work, I propose a hybrid data assimilation and machine learning-based long lead-time forecast framework ($\mathcal{M}_{ML} + \mathcal{E}_{DA+ML}$ *mode*). The convolutional LSTM (ConvLSTM, Shi et al. (2015)) and deep convolutional GAN (DCGAN, Goodfellow et al. (2014)), acting as the dynamic surrogate model $\mathcal{M}_{ML}$, are used for efficient calculation of ensembles as well as long term forecasts. The EnKF will be introduced for reducing the model error $\mathcal{E}_{DA+ML}$. The main contributions of the work can be summarized as follows:

- A Multi-Input Multi-Output (MIMO) forecasting architecture is built in the DCGAN-ConvLSTM framework, which can make multi-step forecasts of temporal/dynamic information or prediction of any target variable.
- ConvLSTM and DCGAN both consist of convolutional LSTM layers, which can learn long term dependency by LSTM and extract time-invariant features using convolutional layers. Thus, the two algorithms are powerful enough to efficiently and accurately exploit the intrinsic dynamics of the high-dimensional dynamic system.
- A recursively MIMO forecasting strategy is developed to make long lead-time forecasts of any target variable.
- Importantly, the EnKF based DCGAN-ConvLSTM framework enables real-time assimilation of observations into the machine learning-based forecast models, which allows us to minimize model errors and determine uncertainties, thus improving the accuracy of forecasts.

This chapter is organized as follows. In section 7.2, the details of an EnKF are briefly introduced. The EnKF based GAN-ConvLSTM forecasting architecture for long lead-time forecasting is presented in detail in section 7.3. Detailed comparisons between model forecasts appear in section 7.4. Finally, the conclusion is provided in section 7.5.

7.2 Ensemble Kalman filter

EnKF is a Monte Carlo implementation of the Bayesian update problem, which uses ensembles of stochastic realizations (state vectors) for approximating the states of a dynamic system. State matrix consists of all the state vectors in the ensemble as:

$$\mathcal{X}(t) = (\mathcal{X}_1(t), \ldots, \mathcal{X}_k(t), \ldots, \mathcal{X}_{N_e}(t)) \in \mathbb{R}^{N_x \times N_e}, \tag{7.1}$$

where $\mathcal{X}_k(t)$ is the state vector of the dynamic system, the subscript k is the number of the state vector in the ensemble, the subscript N_x represents the length of the state vector, the subscript N_e is the size of the ensemble.

EnKF-based data assimilation consists of two steps, prediction and update. In the prediction step, the forecast model is performed on each ensemble member independently:

$$\mathcal{X}_k^f(t_s) = \mathcal{M}(\mathcal{X}_k^b(t_{s-1})) + \mathcal{E}_k(t_{s-1}), \tag{7.2}$$

where $\mathcal{M}$ is a forecast model representing the dynamic system, $\mathcal{X}_k^b(t_{s-1}) \in \mathbb{R}^{N_x}$ is the background vectors of all model states for the k^{th} ensemble member, and $\mathcal{E}_k(t_s)$ denotes the white noises for the forecast model at time t_s. The introduction of noises $\mathcal{E}_k(t_s)$ enables EnKF to correctly propagate the uncertainty engaged in the system, which avoids a divergence of the systems due to a low variance with observation in propagation process (Burgers et al., 1998). The ensemble mean of forecasting vectors of all model states at

time t_s is defined as:

$$\overline{\mathcal{X}}_k^f(t_s) = \frac{1}{N_e}\sum_{k=1}^{N_e}\mathcal{X}_k^f(t_s). \tag{7.3}$$

An estimate of the model error covariance P^f is then computed as:

$$P^f(t_s) = \frac{1}{N_e-1}\sum_{k=1}^{N_e}[\mathcal{X}_k^f(t_s) - \overline{\mathcal{X}}^f(t_s)][\mathcal{X}_k^f(t_s) - \overline{\mathcal{X}}^f(t_s)]^T. \tag{7.4}$$

In the update step, given the observation $\mathcal{Z}(t_s)$ each ensemble is then updated by:

$$\mathcal{X}_k^a(t_s) = \mathcal{X}_k^f(t_s) + K(t_s)[\mathcal{Z}(t_s) - \mathcal{H}\mathcal{X}_k^f(t_s)], \tag{7.5}$$

where $\mathcal{H}$ is the operator that converts the model states to observation space (Clark et al., 2008), and the Kalman gain is calculated by:

$$K(t_s) = (\rho_s \circ P^f(t_s))\mathcal{H}^T[\mathcal{H}(\rho_s \circ P^f(t_s))\mathcal{H}^T + \mathcal{R}(t_s)]^{-1}, \tag{7.6}$$

where $\mathcal{R} = \sigma^o\mathcal{I}$ is the observation error covariance matrix, and $\circ$ denotes a Schur product (an element-by-element multiplication). Here, the localization techniques are applied to ensemble assimilations (Blum et al., 2009; Falchetti and Alvarez, 2018; Greybush et al., 2011; Petrie, 2008). The operation ρ_s is the correlation coefficient based on distance (the localization radius which is space and time dependent) so that remote observations are removed. The (posterior) error covariance matrix $P^a(t_s)$ is estimated by:

$$P^a(t_s) = [1 - K(t_s)\mathcal{H}]P^f(t_s). \tag{7.7}$$

7.3 Long lead-time forecasting using hybrid machine learning models and EnKF

In this work, I aim at developing a machine learning-based model for the long lead-time forecast. In current machine learning models, it is still challenging to keep the predictive accuracy beyond the training period, not to mention long lead-time forecasts. Here a hybrid machine learning-based EnKF (ML-EnKF) framework is introduced to reduce the cumulative error in a long term forecast. The main procedures are described below:

(1) The DCGAN and ConvLSTM models are first trained for efficient nonlinear dyncamical prediction (section 7.3.1, $\mathcal{M}_{ML}$ *mode*);

(2) The trained DCGAN and ConvLSTM models are then used for long lead-time forecasts by a recursively way, where the last ensemble multiple-time step-ahead forecast/output are used as new inputs for the next time step-ahead ensemble forecasts (in section 7.3.2, $\mathcal{M}_{ML}$ *mode*);

(3) The EnKF data assimilation approach is further incorporated into the DCGAN and ConvLSTM models for reducing forecasting uncertainty of dynamic features at future time instants (in section 7.3.3, $\mathcal{M}_{ML} + \mathcal{E}_{DA+ML}$ *mode*).

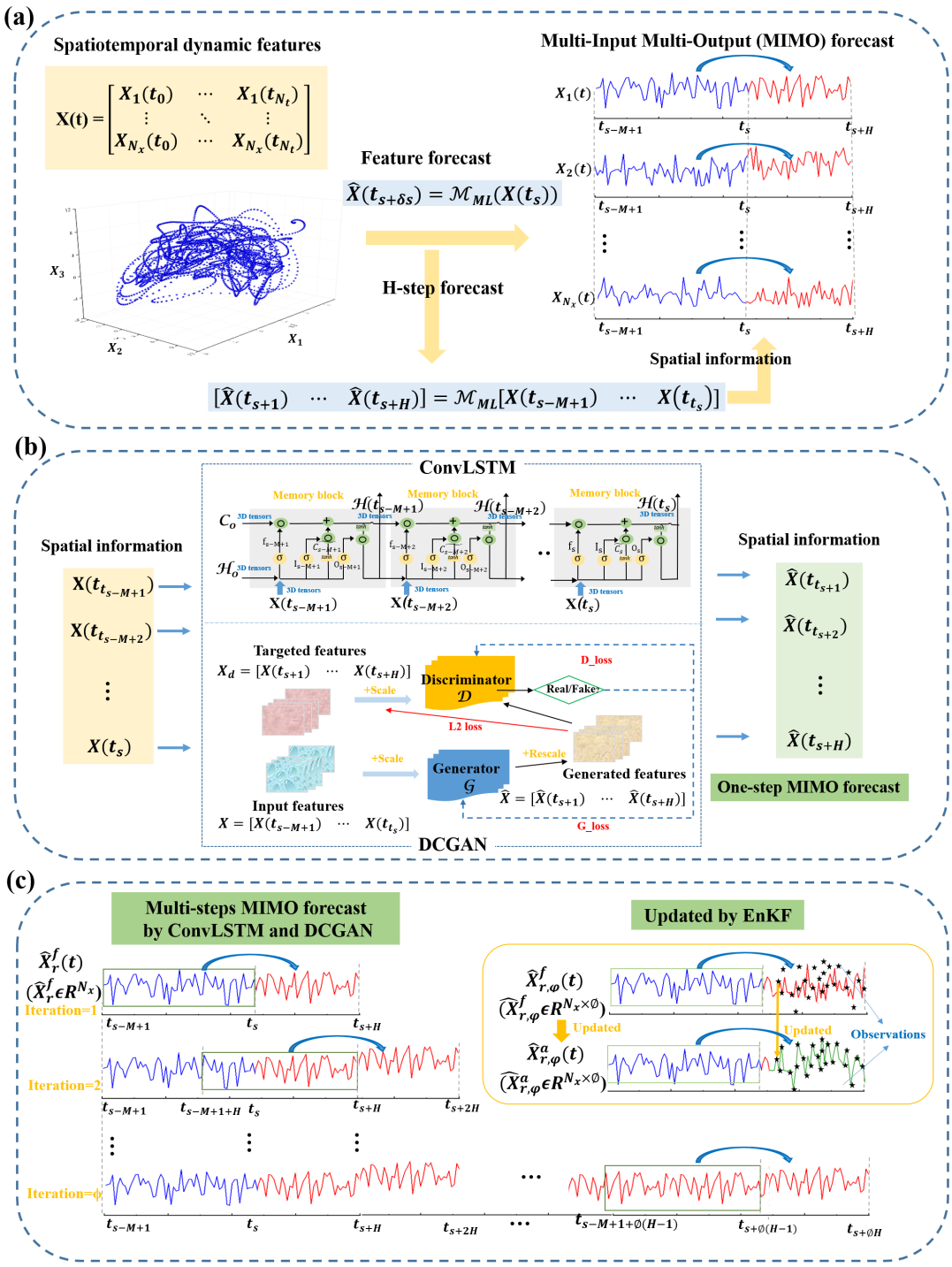


Fig. 7.1. Illustration of a hybrid model combining ensemble data assimilation and ConvLSTM and DCGAN models for long lead-time forecasting. (a) Spatiotemporal dynamic features are used for Multi-Input Multi-Output (MIMO) forecast. (b) Architectures of the ConvLSTM and DCGAN models for MIMO forecast. (c) Extension of forecasting horizon integrating ConvLSTM and DCGAN with EnKF in a recursively MIMO strategy.

7.3.1 Multi-Input Multi-Output (MIMO) forecast using ConvLSTM and DCGAN

In this sub-section, the DCGAN and ConvLSTM models are trained for representing the physical dynamics shown in Eq.(7.2) ($\mathcal{M}_{ML}$ *mode*). These machine learning models have replaced the physic models ($\mathcal{M}_{physical}$) as a forward surrogate model. For long lead-time forecasting, I adopt a Multi-Input Multi-Output (MIMO) strategy (Taieb et al., 2012) as shown in Fig. 7.1(a). The MIMO forecasting is an estimation on future time series:

$$[\hat{\mathcal{X}}(t_{s+1}), \ldots, \hat{\mathcal{X}}(t_{s+H})] = \mathcal{M}_{ML}[\mathcal{X}(t_{s-M+1}), \ldots, \mathcal{X}(t_s)], \tag{7.8}$$

where $\mathcal{M}_{ML}$ is a machine learning forecasting model, $\hat{\mathcal{X}}$ represents the state variables, the inputs are the state values from the previous time steps $(t_{s-M+1}, \ldots, t_s)$, M is the length of time lag while the outputs are the forecasts at the lead-time steps $((t_{s+1}, \ldots, t_{s+H}) \in (t_0, t_{N_t}))$, H denotes the total predictive time steps. Here the ConvLSTM and DCGAN models acting as forecasting model $\mathcal{M}_{ML}$ are used for learning the input-output relationship.

Given N_w pairs of multi-inputs and multi-outputs $(\mathcal{X}_I, \mathcal{X}_O) \in (\mathcal{I}_{tr}, \mathcal{O}_{tr})$ used for ConvLSTM and DCGAN training purpose, the training input data $\mathcal{X}_I$ is :

$$\mathcal{I}_{tr} : \mathcal{X}_I \in \mathbb{R}^{N_x \times M \times N_w} \tag{7.9}$$

while the corresponding output datasets $\mathcal{X}_O$ are:

$$\mathcal{O}_{tr} : \mathcal{X}_O \in \mathbb{R}^{N_x \times H \times N_w} \tag{7.10}$$

In this study, the ConvLSTM forecasting model consists of one input layer, two convolutional LSTM layers and two convolutional layers. The rectified linear unit (Leaky ReLU) activation function is employed between the layers. With respect to the DCGAN forecast-

ing model, the generator $\mathcal{G}$ has one input layer, one convolutional LSTM layer and two convolutional layers. The discriminator $\mathcal{D}$ uses three hidden convolutional layers, each using a densely connected layer. The output layer is densely connected to the final hidden layer with a sigmoid activation function. Specifically, both ConvLSTM and DCGAN models are trained using the Adam optimizer and the weights in the networks are updated after computing the combined loss of each batch.

7.3.2 Long lead-time forecast using ConvLSTM and DCGAN

In this sub-section, the long lead-time forecasts are obtained from the trained ConvLSTM and DCGAN in a recursively MIMO strategy ($\mathcal{M}_{ML}$ *mode*). Once the ConvLSTM and DCGAN training processes are completed, one can find the relationship between the input-output pairs $(\mathcal{X}_I, \mathcal{X}_O)$. By using the trained ConvLSTM and DCGAN models as shown in Fig. 7.1(b), the H-step predictive solutions beyond the training period $[t_0, t_{N_{tr}}]$ for any input data $\mathcal{X}_I \in (\mathcal{I}/\mathcal{I}_{tr})$ can be obtained in Eq.(7.8).

A recursively MIMO forecasting strategy (Cheng et al., 2020a) is here developed to make long lead-time forecasts, as shown in Fig. 7.1(c). The forecasts are given in Algorithm 7.1: where $\hat{\mathcal{M}}_{ML}$ represents the trained forecasting model, $\phi(\phi \in (1, \Phi))$ is the length of steps using the MIMO strategy.

Algorithm 7.1 Multistep MIMO forecasts using ConvLSTM and DCGAN.

(1) One MIMO step forecasting process.

- Set the length of time lag M and forecasting time step H.
- Obtain the optimal parameters in training process.
- Sample M steps datasets from the initial time t_s during the validation period $(t_{N_{tr}}, t_{N_t})$.
- Make H-step ensemble forecasts in one MIMO step strategy by ConvLSTM and DCGAN models.

(2) Long lead-time forecast process.

- Select the iteration length Φ.

for ϕ = 1 to Φ **do**

if $\phi = 1$ **then**

- The next multisteps-ahead is obtained by the trained function $\hat{\mathcal{M}}_{ML}$ as:

$$\hat{\mathcal{M}}_{ML}[\mathcal{X}(t_{s-M+1}), \ldots, \mathcal{X}(t_s)].$$

else if $H\phi \leq M$ **then**

- The next multisteps-ahead is obtained by the trained function $\hat{\mathcal{M}}_{ML}$ as:

$$\hat{\mathcal{M}}_{ML}[\mathcal{X}(t_{s-M-2+H(\phi-1)}), \ldots, \mathcal{X}(t_s), \hat{\mathcal{X}}(t_{s+1}), \ldots, \hat{\mathcal{X}}(t_{s+H(\phi-1)})].$$

else

- The next multisteps-ahead is obtained by the trained function $\hat{\mathcal{M}}_{ML}$ as:

$$\hat{\mathcal{M}}_{ML}[\hat{\mathcal{X}}(t_{s-M+1+H(\phi-1)}), \ldots, \hat{\mathcal{X}}(t_{s+H(\phi-1)})].$$

end if

end for

- Obtain the length-$\Phi \times H$ multisteps of ensemble forecasts $\hat{\mathcal{X}}$.

7.3.3 Extension of forecasting horizon integrating ConvLSTM and DCGAN with EnKF

In the sections above, the DCGAN and ConvLSTM models have been developed for efficient forecasts. However, the performance of the trained machine learning forecasting models decays as the predictive horizon Φ extends beyond a few time steps. To reduce

the cumulative error in long lead-time forecasts, I here introduce the data assimilation technique (the EnKF approach) to the ConvLSTM and DCGAN models in the online forecasting process ($\mathcal{M}_{ML} + \mathcal{E}_{DA+ML}$ *mode*), as shown in Fig. 7.1(c).

As stated in section 7.2, the EnKF is a sequential ensemble data assimilation technique. Given the background state $\mathcal{X}^b$ ($\mathcal{X}^b = [\mathcal{X}(t_{s-M+1}), \ldots, \mathcal{X}(t_s)]$), the initial ensembles at time t_s ($t_s \in (t_{N_{tr}}, t_{N_t})$) are generated by:

$$\mathcal{X}_k^b(t_s) = \mathcal{X}^b(t_s) + \epsilon, \tag{7.11}$$

where ϵ is randomly obtained from a Gaussian distribution $\mathcal{N}(\mathcal{X}^b, \sigma^2)$, $k \in (1, N_e)$ and N_e is the size of the ensemble.

The forecast matrix in Eq.(7.2) from trained machine learning-based models at time t_s is defined by

$$\hat{\mathcal{X}}_k^f(t_s) = [\hat{\mathcal{X}}_{k,1}^f(t_s), \ldots, \hat{\mathcal{X}}_{k,s}^f(t_s), \ldots, \hat{\mathcal{X}}^f{}_{k,N_x}(t_s)] \in \mathbb{R}^{N_x \times N_e}. \tag{7.12}$$

Given the observations $\mathcal{Z}$ and the forecast ensemble $\hat{\mathcal{X}}_k^f$, the forecast results $\hat{\mathcal{X}}_k^a$ can be updated by following the steps shown in Algorithm 7.2:

Algorithm 7.2 Multistep MIMO forecasts using ConvLSTM and DCGAN and EnKF.

Set the iteration length Φ.
Set the length of the time lag M and forecasting time step H in one MIMO step.
Set the ensemble sample size N_e.
(1) Inputs:
- Generate the ensembles $\mathcal{X}^b$ at the initial forecast time t_s.
- Set observations $\mathcal{Z}$.
- Set $\hat{\mathcal{X}}^f(t_s) = \mathcal{X}^b(t_s)$

for $\phi = 1$ to Φ **do**
(2) Prediction step:
- Get the dynamic features from the trained models in Eq.(7.2):

$$\hat{\mathcal{X}}^f(t_{N_\phi}) = \hat{\mathcal{M}}_{ML}(\hat{\mathcal{X}}^f(t_{N_{\phi-1}})).$$

- Obtain the model error covariance P^f in Eq.(7.4).

for $i = 1$ to H **do**
(3) Update step:
- Calculate the Kalman gain $K(t_i)$ in Eq.(7.6).
- Update the ensemble of states $\hat{\mathcal{X}}^a(t_i)$ in Eq.(7.5).
- Update the error covariance matrix $P^a(t_i)$ in Eq.(7.7).

end for
end for
- Obtain the updated length-$\Phi \times H$ multisteps of forecasts $\hat{\mathcal{X}}^a$.

7.4 Numerical examples

In this section, our two-hybrid ConvLSTM-EnKF and DCGAN- EnKF models are tested with a one-level variant of the Lorenz 96, which has been commonly used as a prototypical test case in data assimilation (Brajard et al., 2020; Pawar et al., 2020; Gagne et al., 2020; Rao and Sandu, 2016). The Lorenz 96 model is defined on a periodic one-dimensional domain by the following set of ordinary differential equations:

$$\frac{d\mathcal{X}_r}{dt} = (\mathcal{X}_{r+1} - \mathcal{X}_{r-2})\mathcal{X}_{r-1} - \mathcal{X}_r + F \tag{7.13}$$

where $\mathcal{X}_r$ is the scalar state variable measured along a circle of constant latitude of the earth divided into N_x equal sized sectors, $r \in [0, N_x]$, $\mathcal{X}_{N_x} = \mathcal{X}_0$, $\mathcal{X}_{N_x-1} = \mathcal{X}_1$, $\mathcal{X}_{N_x-2} = \mathcal{X}_2$, $N_x = 40$. The external forcing $F = 12$ is chosen as a strongly supercritical value to make the system sufficiently chaotic.

7.4.1 Model setup

A long "truth" run of the physical Lorenz 96 model (referred to as physical modelling-L96) is performed to generate the datasets for training and validating the ConvLSTM-EnKF and DCGAN-EnKF models. The "truth" run is integrated for 200 MTUs using a time step δt= 0.01 MTU. Outputs from the first 180 MTUs are used for training, and the remaining 20 MTUs are used for validation. Therefore, I assess the performance of the trained models for temporal integration during the period $[180, 200]$ MTUs.

The performances of two trained models developed in this study are assessed using the root mean squared error (RMSE), and correlation coefficient (R) which are defined as follows:

$$RMSE = \sqrt{\frac{\sum_{j=1}^{N} (\widehat{\mathcal{X}}_j - \mathcal{X}_{d,j})^2}{N}}, \tag{7.14}$$

$$R = \frac{\sum_{j=1}^{N} (\widehat{\mathcal{X}}_j - \overline{\widehat{\mathcal{X}}})(\mathcal{X}_{d,j} - \overline{\mathcal{X}}_d)}{\sqrt{\sum_{j=1}^{N} (\widehat{\mathcal{X}}_j - \overline{\widehat{\mathcal{X}}})^2} \sqrt{\sum_{j=1}^{N} (\mathcal{X}_{d,j} - \overline{\mathcal{X}}_d)^2}}, \tag{7.15}$$

where the subscript j represents the pairing of N targeted states $\mathcal{X}_d$ and forecasts $\widehat{\mathcal{X}}$ by points ($[0, N_x]$) and predictive period ($[t_{N_{tr}}, t_{N_t}]$), and the overbars signify means over points and/or time.

7.4.2 One MIMO step forecast

Fig. 7.2 illustrates a series of forecasts from the trained ConvLSTM and DCGAN models in one step MIMO modelling. Here I use the last 60-time-step variable values as known inputs and obtain 30-time-step-ahead forecast of all state variables ($N_x = 40$). The 'true' results or references are computed using the physical modelling-L96. The true values are obtained by physical modelling-L96. As shown in Fig. 7.2, the forecasts of state variables (X_2, X_8, X_{16} and X_{32}) from ConvLSTM and DCGAN models could capture the physical dynamics and match well with the true values at the lead-times ($180.6 - 180.9$ MTUs). It can be noted that the ConvLSTM model performs better than the DCGAN model in one step MIMO forecasting where the correlation coefficients of ConvLSTM results are higher than 0.97 except for X_{32}. This can be further confirmed in Fig. 7.3, which compares the forecasts of all state variables from both the trained machine learning models during the lead-times ($180.6 - 180.9$ MTUs). It can be noticed that the residuals (bottom panel) in Fig. 7.3 of the ConvLSTM are smaller than those of the DCGAN model.

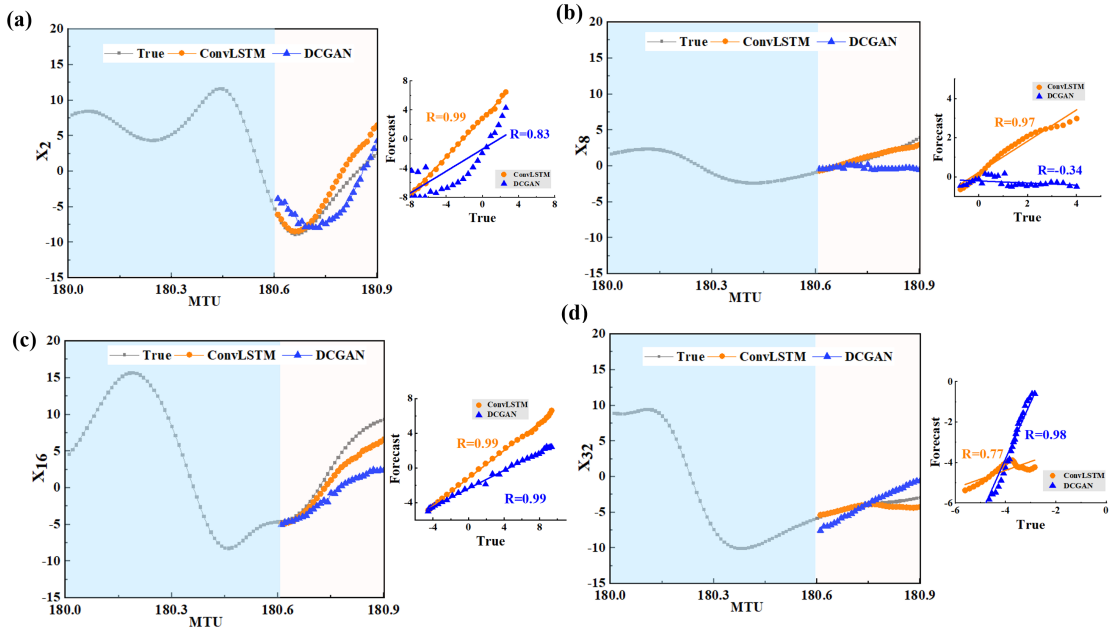


Fig. 7.2. One step MIMO forecasts of X_2, X_8, X_{16} and X_{32} state variables conducted by the ConvLSTM and DCGAN models, where the length of the known time series (for inputs) $M = 60$ (0.06 MTU) and the predicted time series $H = 30$ (0.03 MTU), i.e., 30-time-step-ahead forecasts.

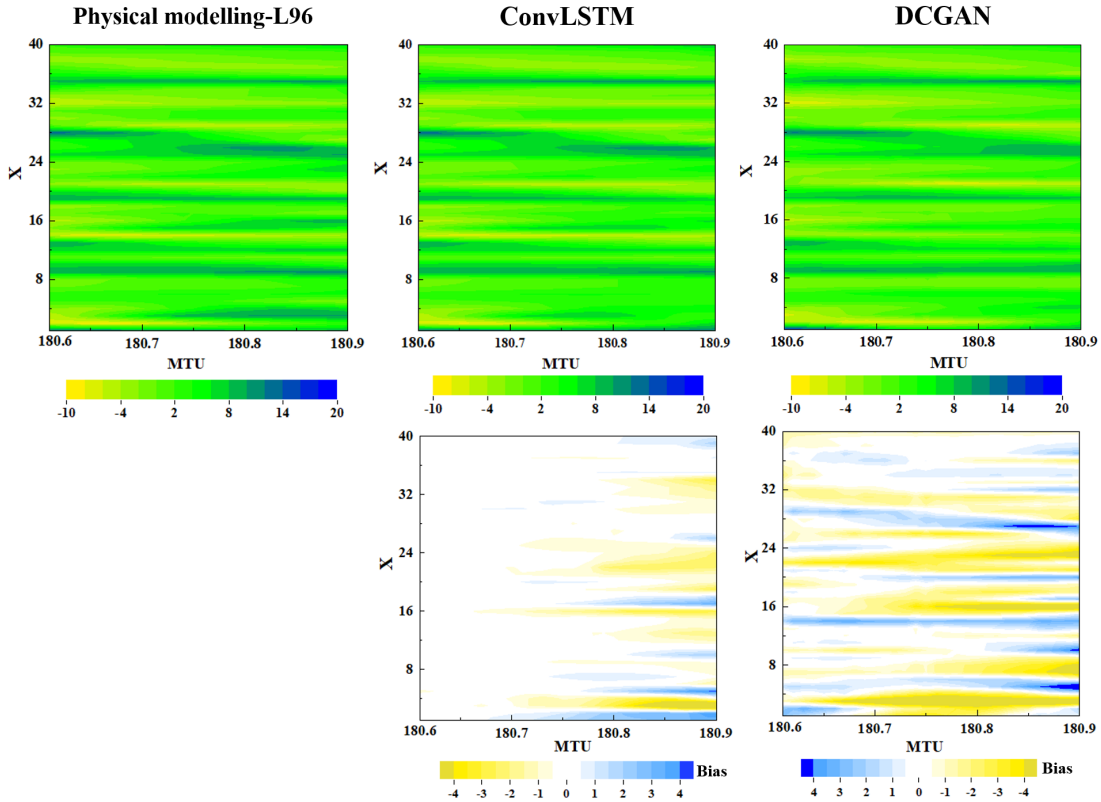


Fig. 7.3. Comparison of the trajectories (top panel) and residuals (bottom panel) of all state variables between the physical modeling-L96, the ConvLSTM and DCGAN models for the lead-times $180.6-180.9$ MTU, where the results from the physical modelling-L96 is considered as 'references'.

The accuracy of results from the ConvLSTM and DCGAN models has further been estimated during different predictive periods (shown in Fig. 7.4 and Fig. 7.5), where the references are computed using the physical modelling-L96. In Fig. 7.4, it can be seen that the residuals from both the ConvLSTM and DCGAN models are mostly smaller during these predictive periods. Fig. 7.5 shows the RMSEs of results from both the ConvLSTM and DCGAN models for four predictive periods. The box plots depict the median, the 25th and 75th percentiles and the minimum and maximum computed over four predictive periods. The bias of mean values is less than -0.16 and -0.56 for the ConvLSTM and DCGAN models respectively at different predictive periods. Again, the ConvLSTM model attains a good agreement with the results from the physical modeling-L96 than that of the DCGAN model. The ConvLSTM model can better detect the underlying functional

mapping through pattern extraction and long term dependency learning.

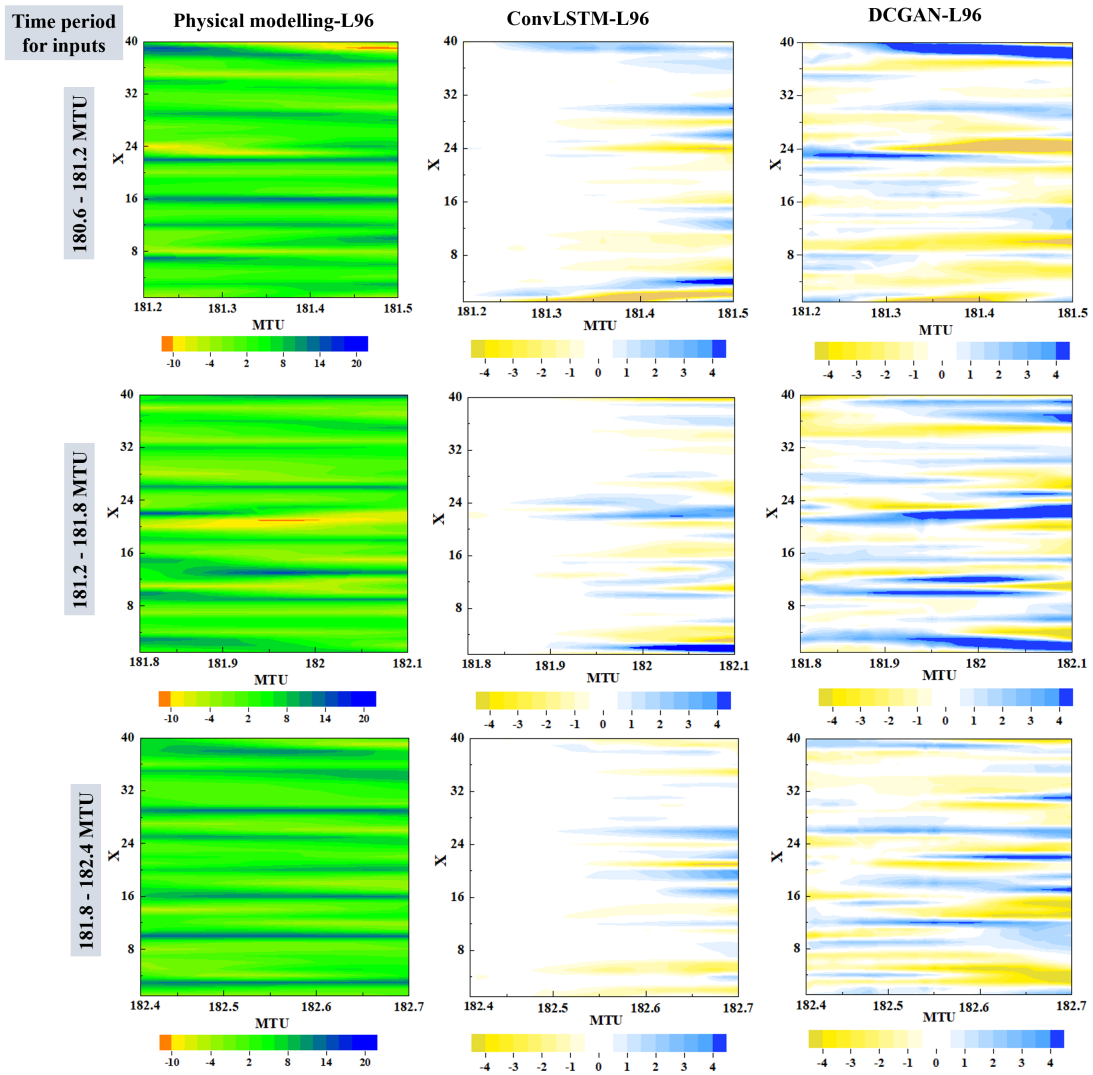


Fig. 7.4. Residuals of all states from the ConvLSTM and DCGAN models in comparison to those from the physical modeling-L96. Left column: full state trajectory from the physical modelling-L96; the difference of results between the machine learning model and physical modelling-L96: middle column-ConvLSTM and right column-DCGAN, where the data at the last-60-step time instances (shown in the left) as inputs while the corresponding forecast periods are $181.2-181.5$ MTU, $181.8-182.1$ MTU, and $182.4-182.7$ MTU.

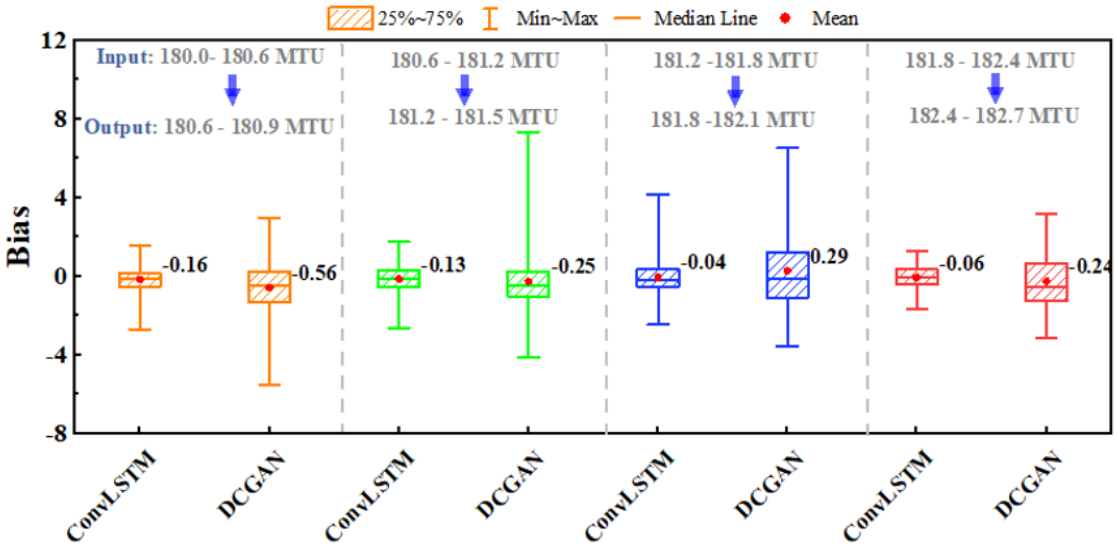


Fig. 7.5. The RMSE of predicted all states from the ConvLSTM and DCGAN models for the lead-time periods, $180.6-180.9$ MTU, $181.2-181.5$ MTU, $181.8-182.1$ MTU, and $182.4-182.7$ MTU.

7.4.3 Long lead-time forecasts with ML-EnKF

In this section, I further demonstrate the forecasting performance of the ConvLSTM and DCGAN models combined with EnKF in multi-step-MIMO forecasting. The EnKF is introduced for improving the accuracy in long term forecasting. The forecast period is $[180.6, 183.6]$ MTU. In MIMO forecast (in Eq.(7.8)), the total forecasting time steps in one iteration/step are set to $H = 30$ while the length of previous time steps for inputs is $M = 60$ (where the time step size is 0.01 MTU). The iteration number Φ of multi-step MIMO is 10, i.e. the long term forecast period (300-steps-ahead) is $10 \times (0.01 \times 30) = 3$ MTUs. The initial conditions $\mathcal{X}^b$ are the state variable values from 180 MTU (in Algorithm 7.2). Given a perturbation (randomly drawn from a Gaussian distribution with a zero mean) to the initial variable $\mathcal{X}^b$ (in Eq.(7.11)), the initial ensembles are generated, where the ensemble sample size is 50 ($N_e = 50$). Each ensemble member is integrated in time using the trained ConvLSTM and DCGAN models.

In ML-EnKF models, the forecasts at the lead-time steps (i.e., $\hat{\mathcal{X}}^f(t_{s+1}, \ldots, t_{s+H})$ in Algorithm 7.2) are obtained from the trained ConvLSTM and DCGAN models. For the

next-MIMO-step forecasting, the inputs are the forecasts $\hat{\mathcal{X}}^f(t_s)$ at the previous time steps $(t_{s-M+1}, \ldots, t_s)$, which are updated using the EnKF whenever an ongoing sequence of observations are available. In comparison to the true values (from physical modeling-L96), the performance of the ConvLSTM-EnKF and DCGAN-EnKF are evaluated through:

(1) *the forecast state variables* during the lead-time period using the ConvLSTM and DCGAN, where the inputs are the updated forecast results using EnKF (hereafter referred to ConvLSTM-EnKF1 and DCGAN-EnKF1);

(2) *the updated forecast results* during the lead-time period, where both outputs and inputs in the ConvLSTM and DCGAN models are updated using EnKF (hereafter referred to ConvLSTM-EnKF2 and DCGAN-EnKF2). At this step, the outputs (forecast states) from ConvLSTM-EnKF1 and DCGAN-EnKF1 are further updated using EnKF.

As shown in Fig. 7.6, the trajectories of X_2, X_8, X_{16} and X_{32} predicted using the ConvLSTM-EnKF1 and DCGAN-EnKF1 follow the trend of the true ones during the whole lead-times ($180.6 - 183.6$ MTU). It can be observed that a relatively good agreement exists between the forecasts and true values and small differences can be noticed at forecasting steps. In general, the underlying dynamic pattern is captured well by the predictive ConvLSTM-EnKF1 and DCGAN-EnKF1. During the 300-steps-ahead forecasting period, the correlation coefficients of four state variables are between 0.79 and 0.98 for the ConvLSTM and between 0.8 and 0.95 for the DCGAN respectively.

To further demonstrate the forecasting capability of the hybrid models, the error analysis is carried out by the RMSE and correlation coefficient of predicted state variables. As shown in Fig. 7.7, it can be noticed that the RMSEs of forecasting states are reduced when the inputs for the ConvLSTM and DCGAN models are updated with EnKF. For example, the RMSEs of the ConvLSTM and DCGAN models have a rising trend (with a maximum

value of 8) during the predictive period $[180.6, 183.6]$ MTU in Fig. 7.7(a), while for the ConvLSTM-EnKF1 and DCGAN-EnKF1 the values fluctuate between 0 and 4 except for the initial 30-step forecasts ($180.6 - 180.9$ MTU). The correlation coefficients of ConvLSTM-EnKF1 and DCGAN-EnKF1 are around $0.75 - 0.9$, while without EnKF, the correlation coefficients are down to 0.2. The introduction of EnKF to the ConvLSTM and DCGAN models has mitigated the cumulated forecast error problem in recursively MIMO forecasting. This can be seen in Fig. 7.7(b) that the mean RMSE of the ConvLSTM model decreases from 5.88 to 1.82 after the observations are assimilated using EnKF, while that from the DCGAN model declines from 6.35 to 2.36. The mean correlation coefficients of the ConvLSTM-EnKF1 and DCGAN-EnKF1 models are 0.92 and 0.86 respectively.

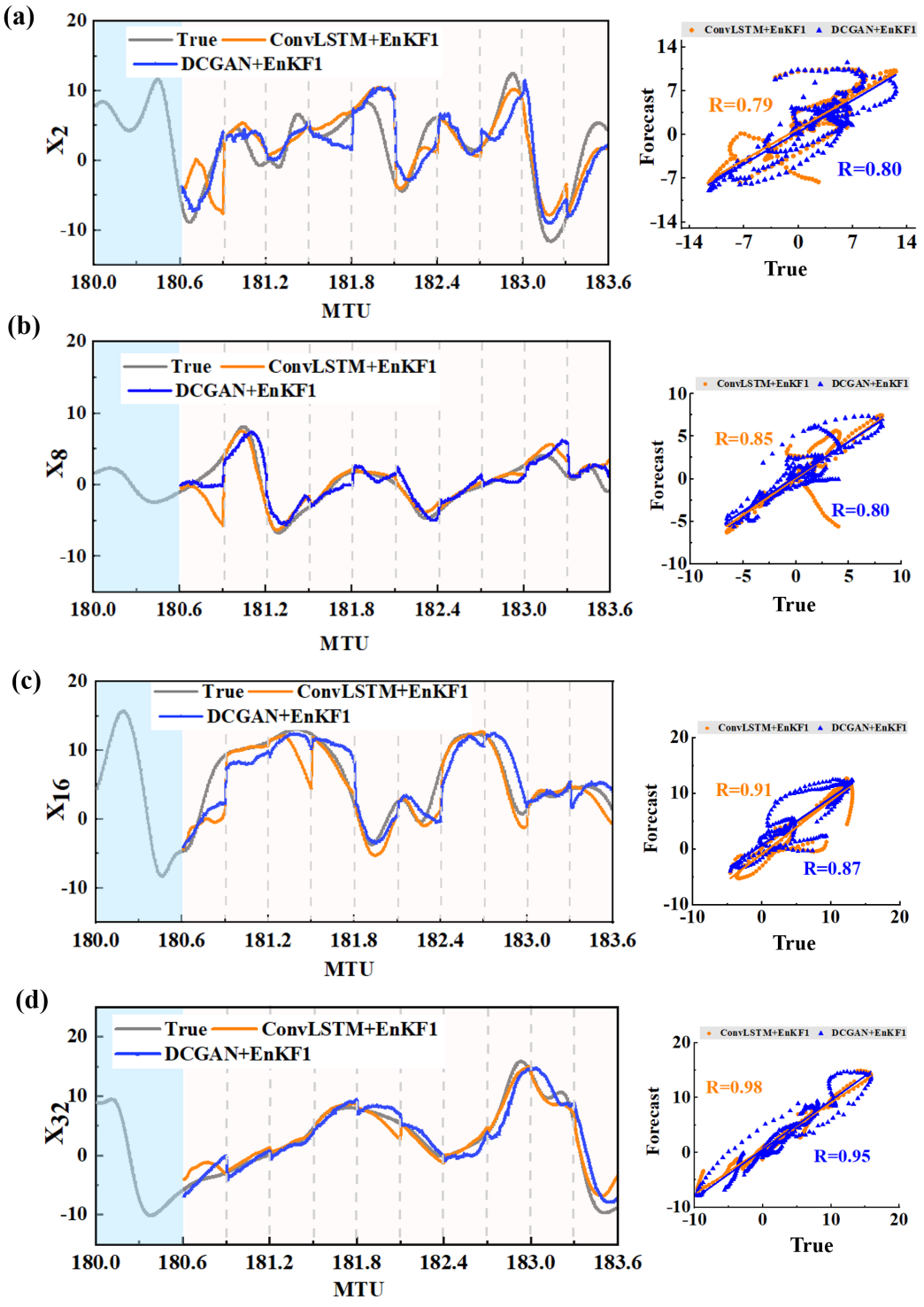


Fig. 7.6. The trajectories of X_2, X_8, X_{16} and X_{32} predicted using the ConvLSTM-EnKF1 and DCGAN-EnKF1, where the length of the known series is $M = 60$ (0.06 MTU) and the predicted series is $\Phi \times H = 300$ (3 MTU), i.e., 300-steps-ahead forecasts.

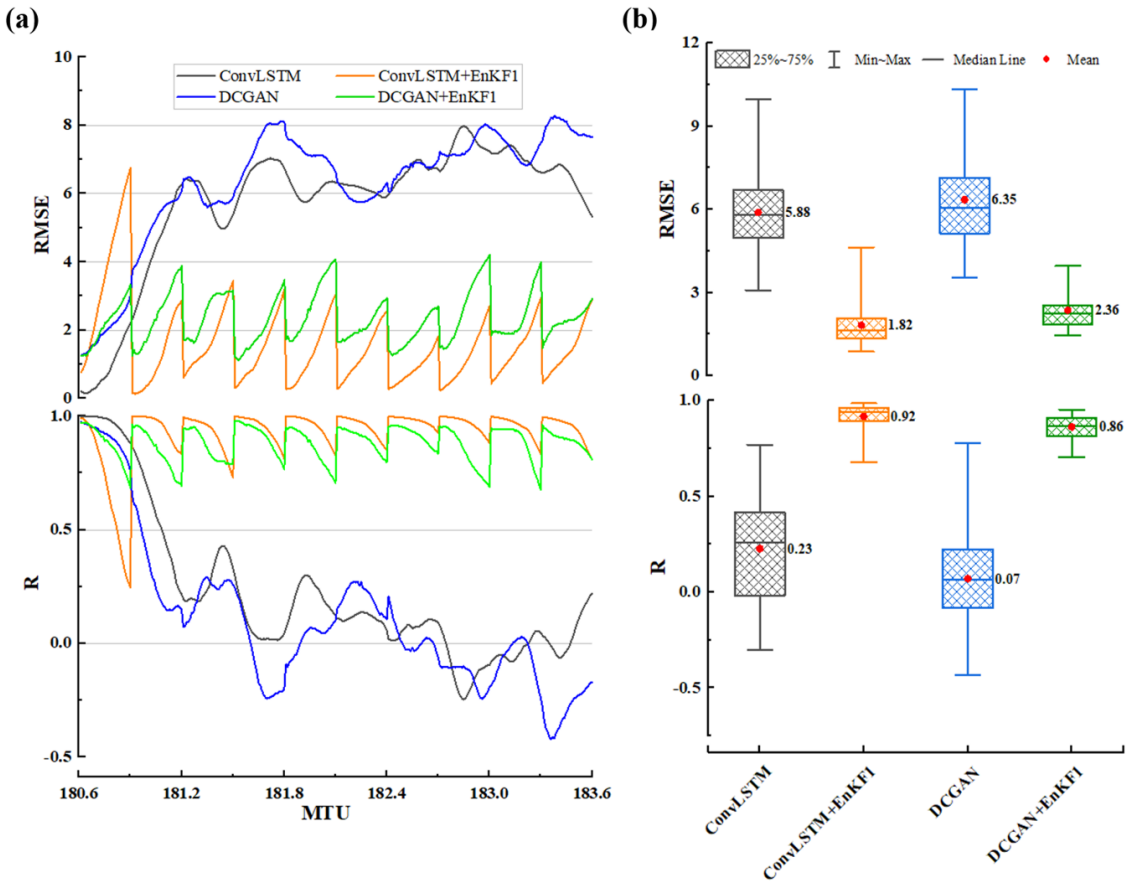


Fig. 7.7. The RMSEs and correlation coefficients of the ConvLSTM, DCGAN, ConvLSTM-EnKF1, and DCGAN-EnKF1 models during the forecasting period of $180.6-183.6$ MTU, where (a) the spatial-averaged RMSEs and (b) the temporal-averaged RMSEs. Note that EnKF are not introduced in the ConvLSTM and DCGAN models while EnKF is used for updating the inputs in ConvLSTM-EnKF1 and DCGAN-EnKF1.

The accuracy of forecast states during the lead-times can further be enhanced by updating the outputs from the ConvLSTM-EnKF1 and DCGAN-EnKF1 using EnKF. Fig. 7.8 depicts the forecasts of X_2, X_8, X_{16} and X_{32} from the ConvLSTM-EnKF2 and DCGAN-EnKF2 during the whole lead-time ($180.6 - 183.6$ MTU). Visually, very little difference between the true values and results from the ConvLSTM-EnKF2 and DCGAN-EnKF2 can be noticed. The results demonstrate that the two-hybrid ConvLSTM-EnKF and DCGAN-EnKF models have successfully captured dynamic features and provide accurate long lead-time forecasts. This can serve as a promising tool by using the hybrid ML-EnKF model to efficiently forecast complex dynamic systems for realistic applications.

A Taylor diagram in Fig. 7.9 is employed for further assessment of results between the ConvLSTM, ConvLSTM+EnKF1, ConvLSTM+EnKF2, DCGAN, DCGAN+EnKF1, DCGAN+EnKF2 and physical modelling-L96 for X_2, X_8, X_{16}, and X_{32} state variables during the forecasting period of $180.6-183.6$ MTU. It can be noticed that the results from the hybrid models (i.e., ConvLSTM+EnKF1, ConvLSTM+EnKF2, DCGAN+EnKF1, DCGAN+EnKF2) have a higher correlation coefficient than that of the ConvLSTM and DCGAN models (orange and cyan points), for example, close to 1.0 for the ConvLSTM+EnKF2 and DCGAN+EnKF2 (green and yellow points).

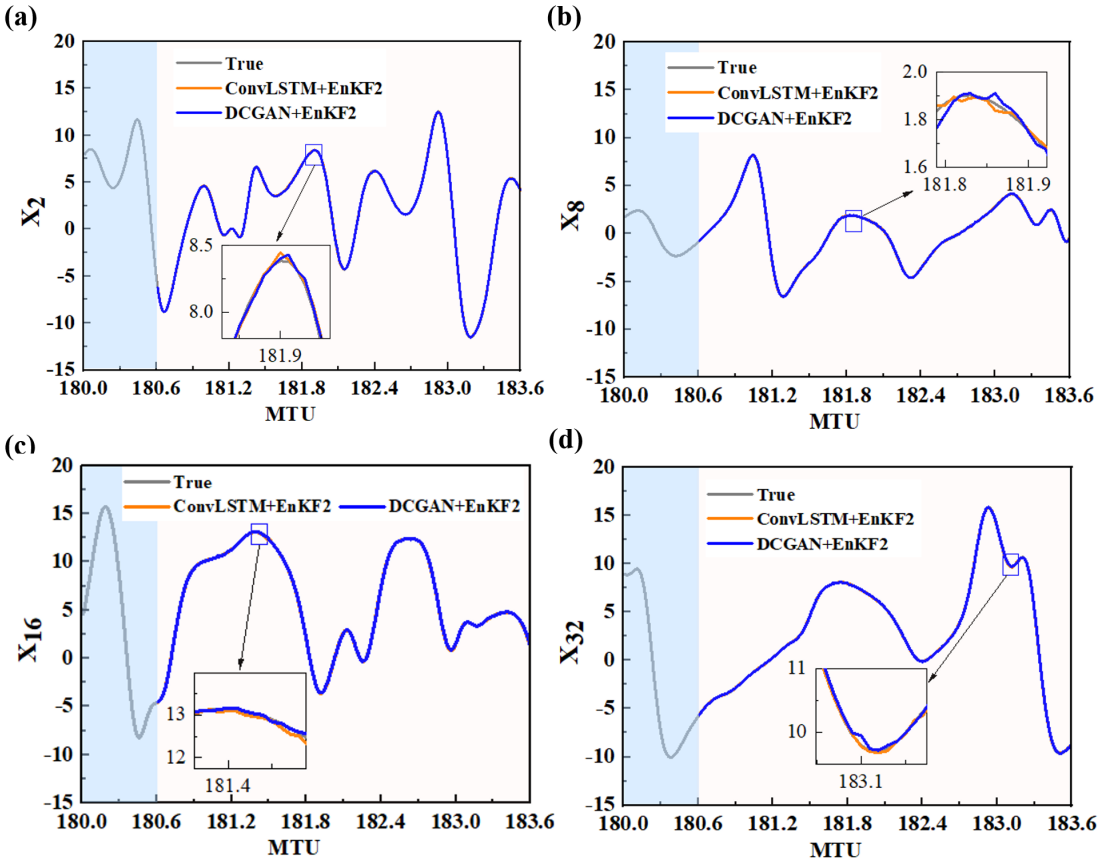


Fig. 7.8. Same as Fig. 7.6 except for using the ConvLSTM-EnKF2 and GAN-EnKF2 models, where both inputs and outputs obtained from the trained machine learning models are updated using EnKF.

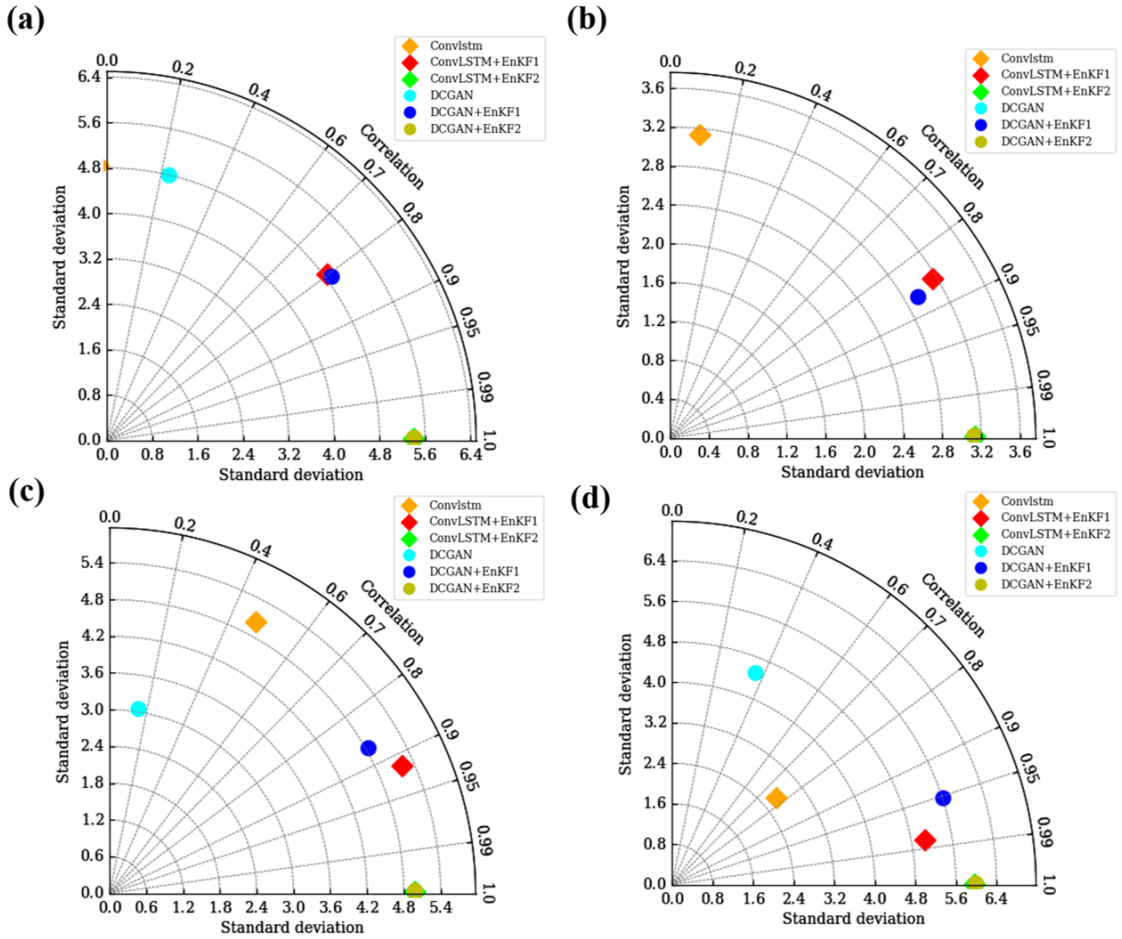


Fig. 7.9. Tylor diagrams representing correlation coefficient together with the standard deviation difference for proposed ConvLSTM, ConvLSTM+EnKF1, ConvLSTM+EnKF2, DCGAN, DCGAN+EnKF1, DCGAN+EnKF2 for (a) X_2 (b) X_8 (c) X_{16}, and (d) X_{32} state variables during the forecasting period of $180.6 - 183.6$ MTU.

To further compare the performance of the hybrid ML-EnKF models, the trajectories of full states from ConvLSTM-EnKF1, ConvLSTM-EnKF2, DCGAN-EnKF1 and DCGAN-EnKF2 models are presented in Fig. 7.10(a). It can be seen that the predictive maps from the four models attain a good agreement to that of the physical modeling-L96. The differences of forecast states between the machine learning models and physical modeling-L96 are illustrated in Fig. 7.10(b)-(e). It can be noticed that the differences are quite small over the whole domain, especially for ConvLSTM-EnKF2 and DCGAN-EnKF2 models. The bias of ConvLSTM-EnKF2 and DCGAN-EnKF2 models is hardly visible, which suggests the ConvLSTM-EnKF2 and DCGAN-EnKF2 models are capable to obtain accurate long

lead-time forecasts for complex physical dynamics.

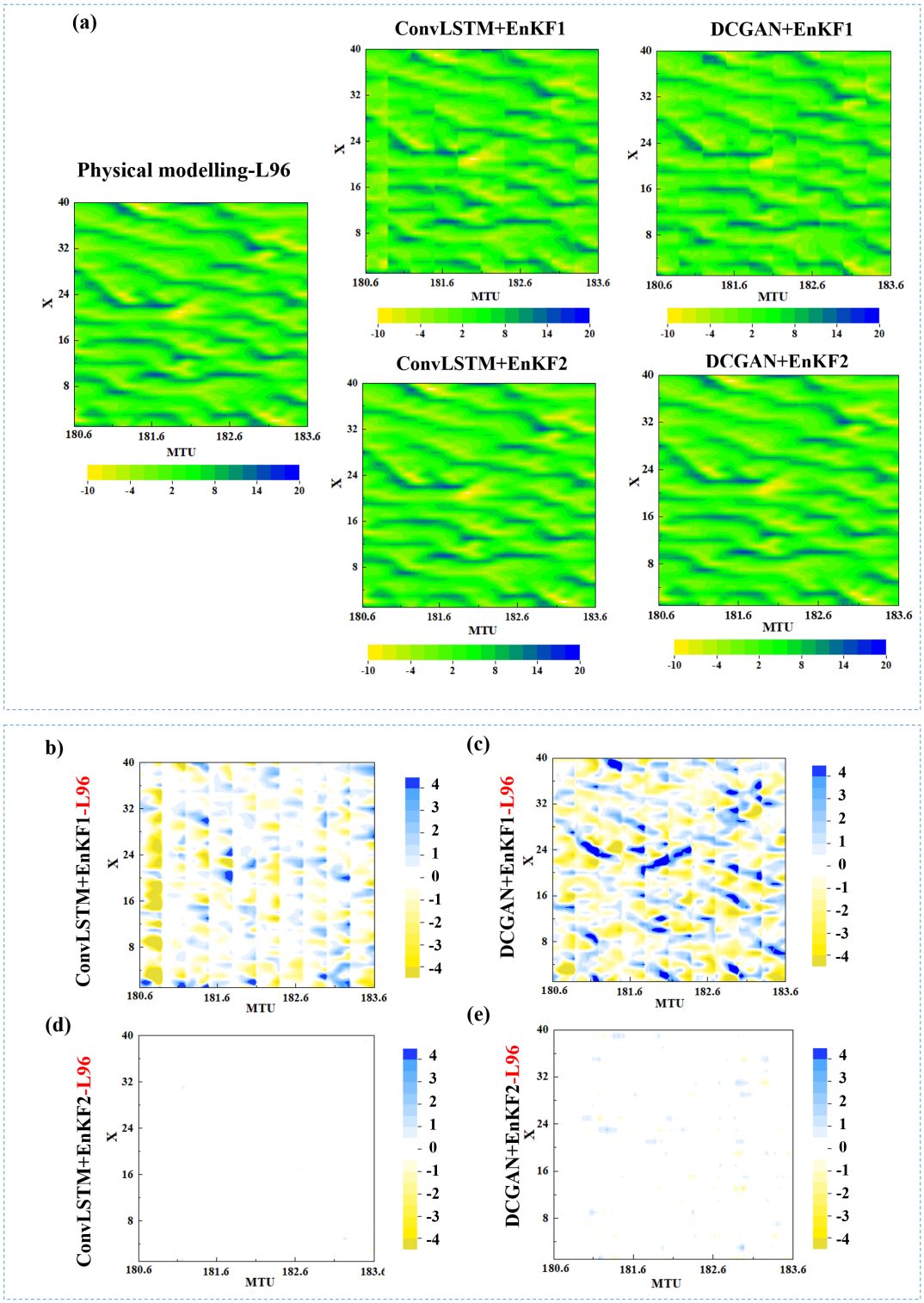


Fig. 7.10. Comparison of 300-step-ahead forecast trajectories of full states between the ConvLSTM-EnKF1, ConvLSTM-EnKF2, DCGAN-EnKF1 and DCGAN-EnKF2 models with physical modelling-L96 during the forecasting period of $180.6 - 183.6$ MTU.

7.5 Conclusions

In this work, a hybrid ML-EnKF framework has first been developed for long lead-time forecasts of nonlinear dynamic systems ($\mathcal{M}_{ML} + \mathcal{E}_{DA+ML}$ *mode*). The ConvLSTM and DCGAN have been trained for learning the complex nonlinear relationships between the past and future states, and then used as the dynamic surrogate model in long term forecasts based on an iterative MIMO algorithm. During each iteration MIMO forecast stage, the trained machine learning models are used for multiple time-step-ahead forecasts while the forecast results are updated using EnKF whenever the observations are available during the online forecast process. In comparison to traditional ENKF approaches, the trained ConvLSTM and DCGAN models are used for efficient calculation of ensembles, rather than a physical-informed model which requires a high computational cost.

The performance of the hybrid ConvLSTM-EnKF and DCGAN-EnKF models has been demonstrated by a Lorenz 96 model. To evaluate the model forecasting capability, a detailed accuracy assessment has been performed through the correlation coefficient and RMSE. Our results illustrate that the ConvLSTM model can better capture temporal dynamics in a high-dimensional nonlinear system and predict accurate dynamics than the DCGAN model. In addition, the use of EnKF in ConvLSTM and DCGAN models significantly reduces forecast errors along with lead-times.

Overall, the hybrid ML-EnKF ($\mathcal{M}_{ML} + \mathcal{E}_{DA+ML}$ *mode*) is an efficient and reliable tool for real-time forecasting of dynamic systems. Compared to physical models, the use of machine learning approaches can reduce the computational cost by several orders of magnitude, thus speeding up forecast and data assimilation processes significantly. It has a promising potential for a wide range of applications, for instance, flooding and real-time air pollution forecasting. The accuracy and efficiency of the proposed ML-DA forecasting model applied to realistic and complex applications will be the subject of our future work.

CHAPTER 8

Conclusion and Future work

8.1 Conclusions

In this thesis, I have developed a deep convolutional generative adversarial network (DCGAN) for predicting spatiotemporal flow distributions. This model has been further developed by combining a variational autoencoder (referred to as VAE-GAN) for predicting large data-driven nonlinear fluid flows in spatial and temporal dimensions. Additionally, a real-time predictive machine learning model, i.e., artificial neural network (ANN), long short term memory (LSTM), DCGAN, VAE-GAN, ConvLSTM has been developed for basin streamflow, urban flooding, and national air pollution forecasting problems respectively. Finally, a unique integrated data assimilation-based machine learning model framework has been developed by introducing an ensemble Kalman filter (EnKF) to GAN and ConvLSTM (GAN-ConvLSTM).

In Chapter 2, a DCGAN has been proposed for nonlinear spatiotemporal fluid flow prediction for the first time. In deep convolutional networks, the high-dimensional flows can be converted into the low-dimensional "latent" representations. The complex features of flow dynamics can be captured by the adversarial networks. The DCGAN is applied to prediction of spatially-temporally-coherent and high-dimensional nonlinear fluid flows in

two test cases of Hokkaido tsunami with different incoming waves along the coastal line. It is demonstrated that the results from the DCGAN are comparable with those from the original high fidelity model (Fluidity). The spatiotemporal flow features have been represented as the flow evolves, especially, the wave phases and flow peaks can be captured accurately. In addition, the results illustrate that the online CPU cost is reduced by five orders of magnitude compared to the original high fidelity model simulations.

In Chapter 3, I have further developed a hybrid deep adversarial autoencoder (VAE-GAN) by integrating generative adversarial network (GAN) and variational autoencoder (VAE) for predicting large data-driven fluid flows in spatial and temporal dimensions. By using a VAE, high-dimensional inputs are compressed into the low-dimensional representations by nonlinear functions in a convolutional encoder. Then the predictive fluid flows reconstructed in a convolutional decoder contain the dynamic fluid flow physics of high nonlinearity and chaotic nature. In addition, the low-dimensional representations are applied to the adversarial network for model training and parameter optimization, which enables a fast computation process. The capability of the hybrid VAE-GAN is illustrated by varying inputs on a flow past a cylinder test case as well as the second case of water column collapse. Results show that this hybrid VAE-GAN has successfully captured the spatiotemporal flow features with CPU speed-up of three orders of magnitude.

In Chapter 4, an artificial neural network (ANN) and a long short term memory (LSTM) have been adopted to forecast streamflow at daily and monthly scales for a long lead-time period. A recursive forecasting framework has been developed in the ANN and LSTM forecasting systems, which takes the last one-step-ahead forecast as a new input for the next-step-ahead forecast. Two models are trained and validated for streamflow forecasting using the rainfall and runoff datasets collected from the Nan River Basin and Ping River Basin, Thailand, covering the period 1974 to 2014. The main findings are highlighted here. First, with an optimal setting up of model parameters, both the ANN and LSTM model can provide accurate daily forecasting (up to 20 days ahead). Second,

in comparison to the ANN model, the LSTM model exhibits better model performance in long lead-time daily forecasting, but is less satisfactory in multi-monthly forecasting due to the lack of a large monthly training dataset. Third, the selection of the length of the time lag and number of maximum epochs used in both ANN and LSTM modelling are the key to long lead-time streamflow forecasting at daily and monthly scales. Our study shows that the LSTM method is superior to the ANN model in the daily streamflow forecasting for a long lead-time.

In Chapter 5, a real-time predictive DCGAN has been developed for flooding forecasting in Denmark. It has also been implemented under the framework of a recursive forecasting strategy. The model capability is assessed using a 100-year return period extreme flood event occurred in Greve, Denmark. Results indicate that the predictive fluid flows from the DCGAN and the high fidelity model are in a good agreement (the correlation coefficient $\geq 97\%$ and the mean absolute error $\leq 0.008m$) for a lead-900 timestep forecast. This is an important step toward real-time flow forecasting although further evaluation of the DCGAN performance is required in complex realistic cases in the future.

In Chapter 6, I further presented efficient and accurate real-time forecasting of national spatial ozone distribution by using machine learning models developed here. First, I have systematically analysed the hourly and daily spatiotemporal predictive performances using ConvLSTM and DCGAN models over the Beijing-Tianjin-Hebei region, China. Secondly, I have introduced a hybrid model VAE-GAN to provide ozone forecasts for a long lead-time beyond the training period. Results demonstrate that the machine learning-based models achieve better spatiotemporal ozone concentration prediction performance with multiple meteorological conditions. The ConvLSTM model demonstrates the practical feasibility of identifying high ozone concentration distribution and capturing spatiotemporal ozone variation patterns at a high spatial resolution (here $15km \times 15km$). In addition, VAE-GAN has achieved a reasonable accuracy in the prediction of both the spatial and temporal evolution patterns of hourly and daily ozone fields, as compared

to the Nested Air Quality Prediction Modelling System (commonly used in China), the reanalysis data and observations during the validation period.

In Chapter 7, the ensemble Kalman filter (EnKF) has been introduced to machine learning-based long term forecast models to reduce the uncertainty of long lead-time forecasts of chaotic dynamic systems. Both the deep convolutional generative adversarial network (DCGAN) and convolutional long short term memory (ConvLSTM) are used for learning the complex nonlinear relationships between the past and future states of dynamic systems. Using an iterative Multi-Input Multi-Output (MIMO) algorithm, the two-hybrid forecast models (DCGAN-EnKF and ConvLSTM-EnKF) are able to yield long lead-time forecasts of dynamic states. The performance of the hybrid models has been demonstrated by a Lorenz 96 model. Our results show that the use of EnKF in ConvLSTM and DCGAN models successfully corrects online model errors and significantly improves the real-time forecasting of dynamic systems for a long lead-time.

8.2 Future Work

While the newly developed GANs and their applications have been validated in numerous idealistic and realistic cases in this thesis, there remain some further developments and possible extensions.

8.2.1 Combining machine learning with physical models

Despite the great progress of GANs in predicting spatiotemporal physics problems in this thesis, large datasets for training purposes are not always available for scientific problems. In this thesis, GANs are capable of learning the underlying relationships from datasets, however, they are difficult to provide accurate predictions owing to extrapolation or observational biases. The physics-informed machine learning (Karniadakis et al., 2021;

Kashinath et al., 2021) is a new research topic that can integrate neural networks with mathematical models. It would also be interesting to develop a physical-informed GAN based on the physical models, which can automatically satisfy the physical invariants for better accurate and efficient prediction. For example, taking the established physical equations as a subnetwork and coupling it with machine learning (Han et al., 2018), adding physical equations and accounting for different initial/boundary conditions into the machine learning acted as additional loss functions for model prediction (Raissi et al., 2019), and constructing the residual in the loss function using automatic differentiation for the integer-order operators in the form of time- and/or space-fractional derivatives (Pang et al., 2019). The physical-informed machine learning seamlessly integrates noisy data and physical models, and implements them with neural networks or other regression networks (Cai et al., 2022). This hybrid approach also combines the strengths of neural networks for rich representation learning, physics embedding and automatic differentiation to approximate the solution of system variables (Chen et al., 2021). In addition, the network architectures of the hybrid approach could be designed specifically which can satisfy some of the physical invariants for better accuracy, faster training and improved generalization (Karniadakis et al., 2021).

8.2.2 Machine learning for problems with multi-inputs

Machine learning requires a set of input predictors for prediction. When a single predictive variable is of interest, the prediction process using machine learning may involve one-to-one or many-to-one methods depending on the predictors. For example, the influence of meteorological factors on ozone prediction and forecasting has been discussed in Chapter 6. The DCGAN model performed better with given temperature, humidity and wind as model inputs. However, it can be also noted that the performance of the VAE-GAN model becomes worse with multi-meteorological factors. For multiple variable inputs (many-to-one), the quality of generated outputs suffers from the generator

(decoder P in VAE-GAN) attempting to cover all data manifolds in the data space, leading to a poor model performance (Pandeva and Schubert, 2019). Multi-modal distribution learning would be a good choice for the GANs when dealing with many-to-one prediction problems (Sharma and Hamarneh, 2019; Chartsias et al., 2017). The multi-modal GANs can take multiple modalities of available information as input and synthesize one sequence, which use additional loss functions to constrain bias and uncertainty of predictions.

8.2.3 Realistic applications

It has a promising potential of the GANs extended for a wide range of realistic applications in geo- and environmental sciences, for instance, climate change, urban environment, renewable energy and optimal sensor location. In these applications, the spatiotemporal structure of data acts as a prevalent feature in geoscience applications, for example, remote sensing images. Therefore, it will be interesting to apply the GANs to spatiotemporal prediction for these applications.

Another possible extension of the GANs is the combination with data assimilation. As I have developed the EnKF data assimilation with the ConvLSTM and DCGAN model in Chapter 7, the hybrid forecasting ML-EnKF model of sequentially updating the forecasts with the incoming observations is not restricted to EnKFs. In the future, I may instead employ other nonlinear filtering methods such as four-dimensional variational (4D-Var) data assimilation; similarly, making accurate forecasts is not restricted to the ConvLSTM and DCGAN models but may employ other machine learning models. It would be interesting to combine machine learning and data assimilation to improve the accuracy and efficiency of forecasting in realistic and complex applications.

8.2.4 Machine learning for unresolved subgrid parametrization

Considering a general prediction problem of large and complex spatiotemporal systems (e.g., earth climate and atmosphere), parameterization physical models are usually used to present the unresolved small-scale processes, for example, $k - \epsilon$ turbulent modelling, and parameterization schemes for cloud microphysics. However, the inaccuracies persist in the parameterization and the interaction with other processes due to the complexity of the physical systems. Machine learning is a promising approach for unresolved subgrid parameterization of high-dimensional spatiotemporal systems (Rasp et al., 2018; Yuval and O'Gorman, 2020). Machine learning can be trained at different grid sizes (global, regional, national, city or street scales) without the need to change the physical closure assumption. In the future, GANs have a great potential to perform spatiotemporal prediction across a range of scales and provide insights into the scale dependence of the parameterization problem.

Bibliography

Abadi, M., 2016. Tensorflow: learning functions at scale, in: Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming, pp. 1–1.

Abolghasemi, M.A., Piggott, M.D., Spinneken, J., Viré, A., Cotter, C.J., Crammond, S., 2016. Simulating tidal turbines with multi-scale mesh optimisation techniques. Journal of Fluids and Structures 66, 69–90.

Adamowski, J., Sun, K., 2010. Development of a coupled wavelet transform and neural network method for flow forecasting of non-perennial rivers in semi-arid watersheds. Journal of Hydrology 390, 85–91.

Adnan, R.M., Liang, Z., Heddam, S., Zounemat-Kermani, M., Kisi, O., Li, B., 2019. Least square support vector machine and multivariate adaptive regression splines for streamflow prediction in mountainous basin using hydro-meteorological data as inputs. Journal of Hydrology , 124371.

Alizadeh, M.J., Kavianpour, M.R., Kisi, O., Nourani, V., 2017. A new approach for simulating and forecasting the rainfall-runoff process within the next two months. Journal of hydrology 548, 588–597.

Alvisi, S., Franchini, M., 2011. Fuzzy neural networks for water level and discharge forecasting with uncertainty. Environmental Modelling & Software 26, 523–537.

Asadi, S., Shahrabi, J., Abbaszadeh, P., Tabanmehr, S., 2013. A new hybrid artificial neural networks for rainfall–runoff process modeling. Neurocomputing 121, 470–480.

Assem, H., Ghariba, S., Makrai, G., Johnston, P., Gill, L., Pilla, F., 2017. Urban water flow and water level prediction based on deep learning, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer. pp. 317–329.

Badrzadeh, H., Sarukkalige, R., Jayawardena, A., 2013. Impact of multi-resolution analysis of artificial intelligence models inputs on multi-step ahead river flow forecasting. Journal of Hydrology 507, 75–85.

Bao, J., Li, L., Redoloza, F., 2020. Coupling ensemble smoother and deep learning with generative adversarial networks to deal with non-gaussianity in flow and transport data assimilation. Journal of Hydrology 590, 125443.

Barzegar, R., Aalami, M.T., Adamowski, J., 2020. Short-term water quality variable prediction using a hybrid CNN–LSTM deep learning model. Stochastic Environmental Research and Risk Assessment , 1–19.

Basheer, I.A., Hajmeer, M., 2000. Artificial neural networks: fundamentals, computing, design, and application. Journal of microbiological methods 43, 3–31.

Bento, P., Pombo, J., Mendes, R., Calado, M., Mariano, S., 2020. Ocean wave energy forecasting using optimised deep learning neural networks. Ocean Engineering , 108372.

Berbel Roman, S., 2014. Modelling flooding from the sea interacting with the drainage system under the influence of combined flood hazards to develop risk management strategies for the coastal region of greve, denmark. University of Nice Sophia Antipolis, Nice, France (Master's thesis) .

Bloomer, B.J., Stehr, J.W., Piety, C.A., Salawitch, R.J., Dickerson, R.R., 2009. Observed relationships of ozone air pollution with temperature and emissions. Geophysical Research Letters 36.

Bloomer, B.J., Vinnikov, K.Y., Dickerson, R.R., 2010. Changes in seasonal and diurnal cycles of ozone and temperature in the eastern US. Atmospheric Environment 44, 2543–2551.

Blum, J., Le Dimet, F.X., Navon, I.M., 2009. Data assimilation for geophysical fluids, in: Handbook of numerical analysis. Elsevier. volume 14, pp. 385–441.

Bolton, T., Zanna, L., 2019. Applications of deep learning to ocean data inference and subgrid parameterization. Journal of Advances in Modeling Earth Systems 11, 376–399.

Bontempi, G., Taieb, S.B., 2011. Conditionally dependent strategies for multiple-step-ahead prediction in local learning. International journal of forecasting 27, 689–699.

Bontempi, G., Taieb, S.B., Le Borgne, Y.A., 2012. Machine learning strategies for time series forecasting, in: European business intelligence summer school, Springer. pp. 62–77.

Boucher, M.A., Quilty, J., Adamowski, J., 2020. Data assimilation for streamflow forecasting using extreme learning machines and multilayer perceptrons. Water Resources Research 56, e2019WR026226.

Bowden, G.J., Dandy, G.C., Maier, H.R., 2005. Input determination for neural network models in water resources applications. part 1—background and methodology. Journal of Hydrology 301, 75–92.

Brajard, J., Carrassi, A., Bocquet, M., Bertino, L., 2020. Combining data assimilation and machine learning to emulate a dynamical model from sparse and noisy observations: A case study with the lorenz 96 model. Journal of Computational Science 44, 101171.

Brajard, J., Carrassi, A., Bocquet, M., Bertino, L., 2021. Combining data assimilation and machine learning to infer unresolved scale parametrization. Philosophical Transactions of the Royal Society A 379, 20200086.

Breiman, L., 2001. Random forests. Machine learning 45, 5–32.

Brunton, S.L., Noack, B., Koumoutsakos, P., 2019. Machine learning for fluid mechanics. ArXiv abs/1905.11075.

Brunton, S.L., Noack, B.R., Koumoutsakos, P., 2020. Machine learning for fluid mechanics. Annual Review of Fluid Mechanics 52, 477–508.

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., et al., 2013. API design for machine learning software: experiences from the scikit-learn project. arXiv preprint arXiv:1309.0238 .

Buizza, C., Casas, C.Q., Nadler, P., Mack, J., Marrone, S., Titus, Z., Le Cornec, C., Heylen, E., Dur, T., Ruiz, L.B., et al., 2021. Data learning: Integrating data assimilation and machine learning. Journal of Computational Science , 101525.

Burgers, G., Jan van Leeuwen, P., Evensen, G., 1998. Analysis scheme in the ensemble kalman filter. Monthly weather review 126, 1719–1724.

Cai, S., Mao, Z., Wang, Z., Yin, M., Karniadakis, G.E., 2022. Physics-informed neural networks (pinns) for fluid mechanics: A review. Acta Mechanica Sinica , 1–12.

Chang, F.J., Tsai, M.J., 2016. A nonlinear spatio-temporal lumping of radar rainfall for modeling multi-step-ahead inflow forecasts by data-driven techniques. Journal of Hydrology 535, 256–269.

Chang, L.C., Chang, F.J., Chiang, Y.M., 2004. A two-step-ahead recurrent neural network for stream-flow forecasting. Hydrological Processes 18, 81–92.

Chartsias, A., Joyce, T., Giuffrida, M.V., Tsaftaris, S.A., 2017. Multimodal mr synthesis via modality-invariant latent representation. IEEE transactions on medical imaging 37, 803–814.

Chattopadhyay, A., Nabizadeh, E., Hassanzadeh, P., 2020. Analog forecasting of extreme-causing weather patterns using deep learning. Journal of Advances in Modeling Earth Systems 12, e2019MS001958.

Chen, P., Liu, R., Aihara, K., Chen, L., 2020a. Autoreservoir computing for multistep ahead prediction based on the spatiotemporal information transformation. Nature communications 11, 1–15.

Chen, Y., Lai, Y.K., Liu, Y.J., 2018. Cartoongan: Generative adversarial networks for photo cartoonization, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 9465–9474.

Chen, Z., Li, R., Chen, D., Zhuang, Y., Gao, B., Yang, L., Li, M., 2020b. Understanding the causal influence of major meteorological factors on ground ozone concentrations across China. Journal of Cleaner Production 242, 118498.

Chen, Z., Liu, Y., Sun, H., 2021. Physics-informed learning of governing equations from scarce data. Nature communications 12, 1–13.

Cheng, M., Fang, F., Kinouchi, T., Navon, I., Pain, C., 2020a. Long lead-time daily and monthly streamflow forecasting using machine learning methods. Journal of Hydrology , 125376.

Cheng, M., Fang, F., Navon, I., Pain, C., 2021. A real-time flow forecasting with deep convolutional generative adversarial network: Application to flooding event in denmark. Physics of Fluids 33, 056602.

Cheng, M., Fang, F., Navon, I.M., Pain, C., a. Ensemble kalman filter for gan-convlstm based long lead-time forecasting. Will be submitted to Computer Methods in Applied Mechanics and Engineering .

Cheng, M., Fang, F., Navon, I.M., Zheng, J., Tang, X., Zhu, J., Pain, C., b. Spatio-temporal hourly and daily ozone forecasting in china using a hybrid machine learning

model: Autoencoder and generative adversarial networks. Journal of Advances in Modeling Earth Systems , e2021MS002806.

Cheng, M., Fang, F., Navon, I.M., Zheng, J., Zhu, J., Pain, C., c. Assessing uncertainty and heterogeneity in machine learning-based spatiotemporal ozone prediction in china. Will be submitted to Journal of Advances in Modeling Earth Systems .

Cheng, M., Fang, F., Pain, C., Navon, I., 2020b. An advanced hybrid deep adversarial autoencoder for parameterized nonlinear fluid flow modelling. Computer Methods in Applied Mechanics and Engineering 372, 113375.

Cheng, M., Fang, F., Pain, C., Navon, I., 2020c. 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.

Chenyue, X., Zelong, Y., Jianchun, W., et al., 2021. Artificial neural network-based subgrid-scale models for large-eddy simulation of turbulence. Chinese Journal of Theoretical and Applied Mechanics 53, 1–16.

Choi, Y., Choi, M., Kim, M., Ha, J.W., Kim, S., Choo, J., 2018. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8789–8797.

Clark, M.P., Rupp, D.E., Woods, R.A., Zheng, X., Ibbitt, R.P., Slater, A.G., Schmidt, J., Uddstrom, M.J., 2008. Hydrological data assimilation with the ensemble kalman filter: Use of streamflow observations to update states in a distributed hydrological model. Advances in water resources 31, 1309–1324.

Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., Bharath, A.A., 2018. Generative adversarial networks: An overview. IEEE Signal Processing Magazine 35, 53–65.

Dai, H., Abdelkefi, A., Wang, L., 2014. Modeling and nonlinear dynamics of fluid-conveying risers under hybrid excitations. International Journal of Engineering Science 81, 1–14.

DHI, 2017. Mike 3 flow model fm: Hydrodynamic module user guide. https://manuals.mikepoweredbydhi.help/2017/Coast˙and˙Sea/MIKE˙FM˙HD˙3D.pdf .

Dumoulin, V., Visin, F., 2016. A guide to convolution arithmetic for deep learning. arXiv preprint arXiv:1603.07285 .

Emery, C., Liu, Z., Russell, A.G., Odman, M.T., Yarwood, G., Kumar, N., 2017. Recommendations on statistics and benchmarks to assess photochemical model performance. Journal of the Air & Waste Management Association 67, 582–598.

Erichson, N.B., Muehlebach, M., Mahoney, M.W., 2019. Physics-informed autoencoders for lyapunov-stable fluid flow prediction. arXiv preprint arXiv:1905.10866 .

Evensen, G., 1994. Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics. Journal of Geophysical Research: Oceans 99, 10143–10162.

Eyre, J., 2016. Observation bias correction schemes in data assimilation systems: A theoretical study of some of their properties. Quarterly Journal of the Royal Meteorological Society 142, 2284–2291.

Ezimand, K., Kakroodi, A., 2019. Prediction and spatio–temporal analysis of ozone concentration in a metropolitan area. Ecological indicators 103, 589–598.

Falchetti, S., Alvarez, A., 2018. The impact of covariance localization on the performance of an ocean enkf system assimilating glider data in the ligurian sea. Journal of Marine Systems 180, 76–89.

Fan, S., Xiao, N., Dong, S., 2020. A novel model to predict significant wave height based on long short-term memory network. Ocean Engineering 205, 107298.

Fang, F., Pain, C., Navon, I.M., Xiao, D., 2017. An efficient goal-based reduced order model approach for targeted adaptive observations. International Journal for Numerical Methods in Fluids 83, 263–275.

Fang, F., Pain, C.C., Navon, I., Elsheikh, A., Du, J., Xiao, D., 2013. Non-linear petrov–galerkin methods for reduced order hyperbolic equations and discontinuous finite element methods. Journal of Computational Physics 234, 540–559.

Fang, F., Piggott, M., Pain, C., Gorman, G., Goddard, A., 2006. An adaptive mesh adjoint data assimilation method. Ocean Modelling 15, 39–55.

Fang, F., Zhang, T., Pavlidis, D., Pain, C., Buchan, A., Navon, I., 2014. Reduced order modelling of an unstructured mesh air pollution model and application in 2d/3d urban street canyons. Atmospheric Environment 96, 96–106.

Fang, W., Huang, S., Ren, K., Huang, Q., Huang, G., Cheng, G., Li, K., 2019. Examining the applicability of different sampling techniques in the development of decomposition-based streamflow forecasting models. Journal of hydrology 568, 534–550.

Farajpour, A., Farokhi, H., Ghayesh, M.H., Hussain, S., 2018. Nonlinear mechanics of nanotubes conveying fluid. International Journal of Engineering Science 133, 132–143.

Farchi, A., Bocquet, M., Laloyaux, P., Bonavita, M., Malartic, Q., 2021. A comparison of combined data assimilation and machine learning methods for offline and online model error correction. Journal of computational science 55, 101468.

Farimani, A.B., Gomes, J., Pande, V.S., 2017. Deep learning the physics of transport phenomena. arXiv preprint arXiv:1709.02432 .

Ferreira, L.B., da Cunha, F.F., 2020. Multi-step ahead forecasting of daily reference evapotranspiration using deep learning. Computers and Electronics in Agriculture 178, 105728.

Fluidity, 2015. Fluidity manual v4.1.12, applied modelling and computation group, imperial college. https: //figshare.com/articles/F luidity Manual/1387713, manual .

Freire, L., Gerken, T., Ruiz-Plancarte, J., Wei, D., Fuentes, J., et al., 2017. Turbulent mixing and removal of ozone within an amazon rainforest canopy. Journal of Geophysical Research: Atmospheres 122, 2791–2811.

Fresca, S., Dede, L., Manzoni, A., 2021. A comprehensive deep learning-based approach to reduced order modeling of nonlinear time-dependent parametrized pdes. Journal of Scientific Computing 87, 1–36.

de la Fuente, A., Meruane, V., Meruane, C., 2019. Hydrological early warning system based on a deep learning runoff model coupled with a meteorological forecast. Water 11, 1808.

Fujita, T., Stensrud, D.J., Dowell, D.C., 2007. Surface data assimilation using an ensemble kalman filter approach with initial condition and model physics uncertainties. Monthly weather review 135, 1846–1868.

Funke, S., Pain, C., Kramer, S., Piggott, M., 2011. A wetting and drying algorithm with a combined pressure/free-surface formulation for non-hydrostatic models. Advances in water resources 34, 1483–1495.

Gagne, D.J., Christensen, H.M., Subramanian, A.C., Monahan, A.H., 2020. Machine learning for stochastic parameterization: Generative adversarial networks in the lorenz'96 model. Journal of Advances in Modeling Earth Systems 12, e2019MS001896.

Gamahara, M., Hattori, Y., 2017. Searching for turbulence models by artificial neural network. Physical Review Fluids 2, 054604.

Gao, N., Xue, H., Shao, W., Zhao, S., Qin, K.K., Prabowo, A., Rahaman, M.S., Salim,

F.D., 2022. Generative adversarial networks for spatio-temporal data: A survey. ACM Transactions on Intelligent Systems and Technology (TIST) 13, 1–25.

Geneva, N., Zabaras, N., 2019. Quantifying model form uncertainty in reynolds-averaged turbulence models with bayesian deep neural networks. Journal of Computational Physics 383, 125–147.

Geuzaine, C., Remacle, J.F., 2009. Gmsh: A 3-d finite element mesh generator with built-in pre-and post-processing facilities. International journal for numerical methods in engineering 79, 1309–1331.

Gomes, J., Pavlidis, D., Salinas, P., Xie, Z., Percival, J.R., Melnikova, Y., Pain, C.C., Jackson, M.D., 2017. A force-balanced control volume finite element method for multi-phase porous media flow modelling. International Journal for Numerical Methods in Fluids 83, 431–445.

Gonzalez, F.J., Balajewicz, M., 2018a. Deep convolutional recurrent autoencoders for learning low-dimensional feature dynamics of fluid systems. arXiv preprint arXiv:1808.01346 .

Gonzalez, F.J., Balajewicz, M., 2018b. Learning low-dimensional feature dynamics using deep convolutional recurrent autoencoders. arXiv preprint arXiv:1808.01346 .

Goodfellow, I., 2016. Nips 2016 tutorial: Generative adversarial networks. arXiv preprint arXiv:1701.00160 .

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets, in: Advances in neural information processing systems, pp. 2672–2680.

Gopal, S., Woodcock, C., 1996. Remote sensing of forest change using artificial neural networks. IEEE Transactions on Geoscience and Remote Sensing 34, 398–404.

Gottwald, G.A., Reich, S., 2021. Supervised learning from noisy observations: Combining machine-learning techniques with data assimilation. Physica D: Nonlinear Phenomena 423, 132911.

Greybush, S.J., Kalnay, E., Miyoshi, T., Ide, K., Hunt, B.R., 2011. Balance and ensemble kalman filter localization techniques. Monthly Weather Review 139, 511–522.

Gu, Y., Li, K., Xu, J., Liao, H., Zhou, G., 2020. Observed dependence of surface ozone on increasing temperature in Shanghai, China. Atmospheric Environment 221, 117108.

Gulli, A., Pal, S., 2017. Deep learning with Keras. Packt Publishing Ltd.

Guo, J., Zhou, J., Qin, H., Zou, Q., Li, Q., 2011. Monthly streamflow forecasting based on improved support vector machine model. Expert Systems with Applications 38, 13073–13081.

Ham, Y.G., Kim, J.H., Luo, J.J., 2019. Deep learning for multi-year enso forecasts. Nature 573, 568–572.

Han, J., Jentzen, A., Weinan, E., 2018. Solving high-dimensional partial differential equations using deep learning. Proceedings of the National Academy of Sciences 115, 8505–8510.

Heidaryan, E., 2019. A note on model selection based on the percentage of accuracy-precision. Journal of Energy Resources Technology 141.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural computation 9, 1735–1780.

Hosseini, H., Xiao, B., Jaiswal, M., Poovendran, R., 2017. On the limitation of convolutional neural networks in recognizing negative images, in: 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE. pp. 352–358.

Howard, A.G., 2013. Some improvements on deep convolutional neural network based image classification. arXiv preprint arXiv:1312.5402 .

Hu, C., Wu, Q., Li, H., Jian, S., Li, N., Lou, Z., 2018a. Deep learning with a long short-term memory networks approach for rainfall-runoff simulation. Water 10, 1543.

Hu, R., Fang, F., Pain, C., Navon, I., 2019a. Rapid spatio-temporal flood prediction and uncertainty quantification using a deep learning method. Journal of Hydrology .

Hu, R., Fang, F., Salinas, P., Pain, C., 2018b. Unstructured mesh adaptivity for urban flooding modelling. Journal of hydrology 560, 354–363.

Hu, R., Fang, F., Salinas, P., Pain, C., Domingo, N.S., Mark, O., 2019b. Numerical simulation of floods from multiple sources using an adaptive anisotropic unstructured mesh method. Advances in Water Resources 123, 173–188.

Huang, C.L., Hsu, N.S., Wei, C.C., 2015. Coupled heuristic prediction of long lead-time accumulated total inflow of a reservoir during typhoons using deterministic recurrent and fuzzy inference-based neural network. Water 7, 6516–6550.

Humphrey, G.B., Gibbs, M.S., Dandy, G.C., Maier, H.R., 2016. A hybrid approach to monthly streamflow forecasting: integrating hydrological model outputs into a bayesian artificial neural network. Journal of Hydrology 540, 623–640.

Hussain, M., Bird, J.J., Faria, D.R., 2018. A study on cnn transfer learning for image classification, in: UK Workshop on computational Intelligence, Springer. pp. 191–202.

Ingram, J.C., Dawson, T.P., Whittaker, R.J., 2005. Mapping tropical forest structure in southeastern madagascar using remote sensing and artificial neural networks. Remote Sensing of Environment 94, 491–507.

Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: International conference on machine learning, PMLR. pp. 448–456.

Irrgang, C., Saynisch-Wagner, J., Thomas, M., 2020. Machine learning-based prediction of spatiotemporal uncertainties in global wind velocity reanalyses. Journal of Advances in Modeling Earth Systems 12, e2019MS001876.

Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A., 2017. Image-to-image translation with conditional adversarial networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1125–1134.

Jiang, Z., Tahmasebi, P., Mao, Z., 2021. Deep residual U-net convolution neural networks with autoregressive strategy for fluid flow predictions in large-scale geosystems. Advances in Water Resources 150, 103878.

Kaiser, E., Morzyński, M., Daviller, G., Kutz, J.N., Brunton, B.W., Brunton, S.L., 2018. Sparsity enabled cluster reduced-order models for control. Journal of Computational Physics 352, 388–409.

Kaiser, E., Noack, B.R., Cordier, L., Spohn, A., Segond, M., Abel, M., Daviller, G., Östh, J., Krajnović, S., Niven, R.K., 2014. Cluster-based reduced-order modelling of a mixing layer. Journal of Fluid Mechanics 754, 365–414.

Kalman, R.E., 1960. Kalman. a new approach to linear filtering and prediction problems. Journal of basic Engineering 82, 35–45.

Kalra, A., Ahmad, S., Nayak, A., 2013. Increasing streamflow forecast lead time for snowmelt-driven catchment based on large-scale climate patterns. Advances in Water Resources 53, 150–162.

Kalteh, A.M., 2013. Monthly river flow forecasting using artificial neural network and support vector regression models coupled with wavelet transform. Computers & Geosciences 54, 1–8.

Kang, X., Kokkinaki, A., Power, C., Kitanidis, P.K., Shi, X., Duan, L., Liu, T., Wu, J., 2021. Integrating deep learning-based data assimilation and hydrogeophysical data for

improved monitoring of dnapl source zones during remediation. Journal of Hydrology 601, 126655.

Kani, J.N., Elsheikh, A.H., 2017. Dr-rnn: A deep residual recurrent neural network for model reduction. arXiv preprint arXiv:1709.00939 .

Karniadakis, G.E., Kevrekidis, I.G., Lu, L., Perdikaris, P., Wang, S., Yang, L., 2021. Physics-informed machine learning. Nature Reviews Physics 3, 422–440.

Kashinath, K., Mustafa, M., Albert, A., Wu, J., Jiang, C., Esmaeilzadeh, S., Azizzadenesheli, K., Wang, R., Chattopadhyay, A., Singh, A., et al., 2021. Physics-informed machine learning: case studies for weather and climate modelling. Philosophical Transactions of the Royal Society A 379, 20200093.

Kasiviswanathan, K., He, J., Sudheer, K., Tay, J.H., 2016. Potential application of wavelet neural network ensemble to forecast streamflow for flood management. Journal of hydrology 536, 161–173.

Keller, J., Franssen, H.J.H., Nowak, W., 2021. Investigating the pilot point ensemble kalman filter for geostatistical inversion and data assimilation. Advances in Water Resources 155, 104010.

Kim, B., Azevedo, V.C., Thuerey, N., Kim, T., Gross, M., Solenthaler, B., 2019. Deep fluids: A generative network for parameterized fluid simulations, in: Computer Graphics Forum, Wiley Online Library. pp. 59–70.

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 .

Kingma, D.P., Welling, M., 2013. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 .

Kinouchi, T., Yamamoto, G., Komsai, A., Liengcharernsit, W., 2018. Quantification of seasonal precipitation over the upper chao phraya river basin in the past fifty years

based on monsoon and el niño/southern oscillation related climate indices. Water 10, 800.

Kisi, O., Cimen, M., 2011. A wavelet-support vector machine conjunction model for monthly streamflow forecasting. Journal of Hydrology 399, 132–140.

Kisi, O., Shiri, J., Nikoofar, B., 2012. Forecasting daily lake levels using artificial intelligence approaches. Computers & Geosciences 41, 169–180.

Kong, L., Tang, X., Zhu, J., Wang, Z., Li, J., Wu, H., Wu, Q., Chen, H., Zhu, L., Wang, W., et al., 2021. A 6-year-long (2013–2018) high-resolution air quality reanalysis dataset in China based on the assimilation of surface observations from CNEMC. Earth System Science Data 13, 529–570.

Kow, P.Y., Wang, Y.S., Zhou, Y., Kao, I.F., Issermann, M., Chang, L.C., Chang, F.J., 2020. Seamless integration of convolutional and back-propagation neural networks for regional multi-step-ahead pm2. 5 forecasting. Journal of Cleaner Production 261, 121285.

Kratzert, F., Klotz, D., Brenner, C., Schulz, K., Herrnegger, M., 2018. Rainfall–runoff modelling using long short-term memory (lstm) networks. Hydrol. Earth Syst. Sci 22, 6005–6022.

Kutz, J.N., 2017. Deep learning in fluid dynamics. Journal of Fluid Mechanics 814, 1–4.

Kutz, J.N., Brunton, S.L., Brunton, B.W., Proctor, J.L., 2016. Dynamic mode decomposition: data-driven modeling of complex systems. SIAM.

Kwon, S., et al., 2020. A cnn-assisted enhanced audio signal processing for speech emotion recognition. Sensors 20, 183.

Laloy, E., Hérault, R., Jacques, D., Linde, N., 2018. Training-image based geostatistical inversion using a spatial generative adversarial neural network. Water Resources Research 54, 381–406.

Le, X.H., Ho, H.V., Lee, G., Jung, S., 2019. Application of long short-term memory (lstm) neural network for flood forecasting. Water 11, 1387.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. nature 521, 436.

LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., Jackel, L., 1989. Handwritten digit recognition with a back-propagation network. Advances in neural information processing systems 2.

Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al., 2017. Photo-realistic single image super-resolution using a generative adversarial network, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4681–4690.

Lee, K., Carlberg, K., 2018. Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. arXiv preprint arXiv:1812.08373 .

Lee, S., You, D., 2018. Data-driven prediction of unsteady flow over a circular cylinder using deep learning. Bulletin of the American Physical Society .

Lee, S., You, D., 2019. Data-driven prediction of unsteady flow over a circular cylinder using deep learning. Journal of Fluid Mechanics 879, 217–254.

Legates, D.R., McCabe Jr, G.J., 1999. Evaluating the use of "goodness-of-fit" measures in hydrologic and hydroclimatic model validation. Water resources research 35, 233–241.

Legler, S., Janjić, T., 2021. Combining data assimilation and machine learning to estimate parameters of a convective-scale model. Quarterly Journal of the Royal Meteorological Society .

Leinonen, J., Guillaume, A., Yuan, T., 2019. Reconstruction of cloud vertical structure with a generative adversarial network. Geophysical Research Letters 46, 7035–7044.

Li, C., Wand, M., 2016. Precomputed real-time texture synthesis with markovian generative adversarial networks, in: European Conference on Computer Vision, Springer. pp. 702–716.

Li, Z., Liu, F., Yang, W., Peng, S., Zhou, J., 2021. A survey of convolutional neural networks: analysis, applications, and prospects. IEEE Transactions on Neural Networks and Learning Systems .

Li, Z., Meier, M.A., Hauksson, E., Zhan, Z., Andrews, J., 2018. Machine learning seismic wave discrimination: Application to earthquake early warning. Geophysical Research Letters 45, 4773–4779.

Lin, G.F., Wu, M.C., 2011. An rbf network with a two-step learning algorithm for developing a reservoir inflow forecasting model. Journal of hydrology 405, 439–450.

Ling, J., Kurzawski, A., Templeton, J., 2016. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. Journal of Fluid Mechanics 807, 155–166.

Liu, H., Liu, J., Liu, Y., Ouyang, B., Xiang, S., et al., 2020a. Analysis of wintertime O3 variability using a random forest model and high-frequency observations in Zhangjiakou—an area with background pollution level of the North China Plain. Environmental Pollution 262, 114191.

Liu, M.Y., Huang, X., Yu, J., Wang, T.C., Mallya, A., 2020b. Generative adversarial networks for image and video synthesis: Algorithms and applications. arXiv preprint arXiv:2008.02793 .

Liu, T., de Haan, K., Rivenson, Y., Wei, Z., Zeng, X., Zhang, Y., Ozcan, A., 2019. Deep learning-based super-resolution in coherent imaging systems. Scientific Reports 9, 3926.

Liu, Z., Zhou, P., Chen, G., Guo, L., 2014. Evaluating a coupled discrete wavelet transform and support vector regression for daily and monthly streamflow forecasting. Journal of hydrology 519, 2822–2831.

Lohani, A.K., Goel, N., Bhatia, K., 2014. Improving real time flood forecasting using fuzzy inference system. Journal of hydrology 509, 25–41.

Loiseau, J.C., Brunton, S.L., 2018. Constrained sparse galerkin regression. Journal of Fluid Mechanics 838, 42–67.

Loiseau, J.C., Noack, B.R., Brunton, S.L., 2018. Sparse reduced-order modelling: sensor-based dynamics to full-state estimation. Journal of Fluid Mechanics 844, 459–490.

Lu, X., Hong, J., Zhang, L., Cooper, O.R., Schultz, M.G., et al., 2018. Severe surface ozone pollution in China: a global perspective. Environmental Science & Technology Letters 5, 487–494.

Lu, X., Zhang, L., Shen, L., 2019. Meteorology and climate influences on tropospheric ozone: a review of natural sources, chemistry, and transport patterns. Current Pollution Reports 5, 238–260.

Lusch, B., Kutz, J.N., Brunton, S.L., 2018. Deep learning for universal linear embeddings of nonlinear dynamics. Nature communications 9, 4950.

Ma, M., Yao, G., Guo, J., Bai, K., 2021a. Distinct spatiotemporal variation patterns of surface ozone in China due to diverse influential factors. Journal of Environmental Management 288, 112368.

Ma, R., Ban, J., Wang, Q., Zhang, Y., Yang, Y., He, M.Z., Li, S., Shi, W., Li, T., 2021b. Random forest model based fine scale spatiotemporal O3 trends in the Beijing-Tianjin-Hebei region in China, 2010 to 2017. Environmental Pollution 276, 116635.

Ma, X., Yin, Z., 2021. Dipole Pattern of Summer Ozone Pollution in the east of China

and Its Connection with Climate Variability. Atmospheric Chemistry and Physics Discussions , 1–23.

Maier, H.R., Jain, A., Dandy, G.C., Sudheer, K.P., 2010. Methods used for the development of neural networks for the prediction of water resource variables in river systems: current status and future directions. Environmental modelling & software 25, 891–909.

Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., Frey, B., 2015. Adversarial autoencoders. arXiv preprint arXiv:1511.05644 .

Meresa, H., 2019. Modelling of river flow in ungauged catchment using remote sensing data: application of the empirical (scs-cn), artificial neural network (ann) and hydrological model (hec-hms). Modeling Earth Systems and Environment 5, 257–273.

Merghadi, A., Yunus, A.P., Dou, J., Whiteley, J., ThaiPham, B., Bui, D.T., Avtar, R., Abderrahmane, B., 2020. Machine learning methods for landslide susceptibility studies: A comparative overview of algorithm performance. Earth-Science Reviews 207, 103225.

Metz, L., Poole, B., Pfau, D., Sohl-Dickstein, J., 2016. Unrolled generative adversarial networks. arXiv preprint arXiv:1611.02163 .

MIKE21, D., MIKE3 Flow Model, F., 2009. Hydrodynamic and transport module scientific documentation. Denmark: DHl water & Environment .

Miyanawala, T.P., Jaiman, R.K., 2017. An efficient deep learning technique for the navier-stokes equations: Application to unsteady wake flow dynamics. arXiv preprint arXiv:1710.09099 .

Moghaddam, A.A., Sadaghiyani, A., 2018. A deep learning framework for turbulence modeling using data assimilation and feature extraction. arXiv preprint arXiv:1802.06106 .

Mohan, A.T., Gaitonde, D.V., 2018. A deep learning based approach to reduced order modeling for turbulent flow control using lstm neural networks. arXiv preprint arXiv:1804.09269 .

Mosser, L., Dubrule, O., Blunt, M.J., 2017. Reconstruction of three-dimensional porous media using generative adversarial neural networks. Physical Review E 96, 043309.

Mouatadid, S., Adamowski, J.F., Tiwari, M.K., Quilty, J.M., 2019. Coupling the maximum overlap discrete wavelet transform and long short-term memory networks for irrigation flow forecasting. Agricultural Water Management 219, 72–85.

Murata, T., Fukami, K., Fukagata, K., 2020. Nonlinear mode decomposition with convolutional neural networks for fluid dynamics. Journal of Fluid Mechanics 882.

Nair, A.G., Yeh, C.A., Kaiser, E., Noack, B.R., Brunton, S.L., Taira, K., 2019. Cluster-based feedback control of turbulent post-stall separated flows. Journal of Fluid Mechanics 875, 345–375.

Nash, J.E., Sutcliffe, J.V., 1970. River flow forecasting through conceptual models part i—a discussion of principles. Journal of hydrology 10, 282–290.

Noori, N., Kalin, L., 2016. Coupling swat and ann models for enhanced daily streamflow prediction. Journal of Hydrology 533, 141–151.

Nourani, V., Komasi, M., 2013. A geomorphology-based anfis model for multi-station modeling of rainfall–runoff process. Journal of Hydrology 490, 41–55.

Oishi, Y., Piggott, M.D., Maeda, T., Kramer, S.C., Collins, G.S., Tsushima, H., Furumura, T., 2013. Three-dimensional tsunami propagation simulations using an unstructured mesh finite element model. Journal of Geophysical Research: Solid Earth 118, 2998–3018.

Östh, J., Kaiser, E., Krajnović, S., Noack, B.R., 2015. Cluster-based reduced-order modelling of the flow in the wake of a high speed train. Journal of Wind Engineering and Industrial Aerodynamics 145, 327–338.

Pain, C., Umpleby, A., De Oliveira, C., Goddard, A., 2001. Tetrahedral mesh optimisation and adaptivity for steady-state and transient finite element calculations. Computer Methods in Applied Mechanics and Engineering 190, 3771–3796.

Pandeva, T., Schubert, M., 2019. Mmgan: Generative adversarial networks for multimodal distributions. arXiv preprint arXiv:1911.06663 .

Pang, G., Lu, L., Karniadakis, G.E., 2019. fpinns: Fractional physics-informed neural networks. SIAM Journal on Scientific Computing 41, A2603–A2626.

Papacharalampous, G., Tyralis, H., Koutsoyiannis, D., 2019. Comparison of stochastic and machine learning methods for multi-step ahead forecasting of hydrological processes. Stochastic Environmental Research and Risk Assessment 33, 481–514.

Park, I., Kim, K., Kim, J., Van, S., 2009. A volume-of-fluid method for incompressible free surface flows. International Journal for Numerical Methods in Fluids 61, 1331–1362.

Pascual, S., Bonafonte, A., Serra, J., 2017. Segan: Speech enhancement generative adversarial network. arXiv preprint arXiv:1703.09452 .

Pawar, S., Ahmed, S.E., San, O., Rasheed, A., Navon, I.M., 2020. Long short-term memory embedded nudging schemes for nonlinear data assimilation of geophysical flows. Physics of Fluids 32, 076606.

Pawar, S., San, O., 2021. Data assimilation empowered neural network parametrizations for subgrid processes in geophysical flows. Physical Review Fluids 6, 050501.

Pazikadin, A.R., Rifai, D., Ali, K., Malik, M.Z., Abdalla, A.N., Faraj, M.A., 2020. Solar irradiance measurement instrumentation and power solar generation forecasting based

on artificial neural networks (ann): A review of five years research trend. Science of The Total Environment 715, 136848.

Perez-Roa, R., Castro, J., Jorquera, H., Perez-Correa, J., Vesovic, V., 2006. Air-pollution modelling in an urban area: Correlating turbulent diffusion coefficients by means of an artificial neural network approach. Atmospheric Environment 40, 109–125.

Petrie, R., 2008. Localization in the ensemble kalman filter. MSc Atmosphere, Ocean and Climate University of Reading .

Peyron, M., Fillion, A., Gürol, S., Marchais, V., Gratton, S., Boudier, P., Goret, G., 2021. Latent space data assimilation by using deep learning. Quarterly Journal of the Royal Meteorological Society 147, 3759–3777.

Piggott, M., Gorman, G., Pain, C., Allison, P., Candy, A., Martin, B., Wells, M., 2008. A new computational framework for multi-scale ocean modelling based on adapting unstructured meshes. International Journal for Numerical Methods in Fluids 56, 1003–1015.

POST, 2007. Urban Flooding. Technical Report.

Prakash, O., Sudheer, K., Srinivasan, K., 2014. Improved higher lead time river flow forecasts using sequential neural network with error updating. Journal of Hydrology and Hydromechanics 62, 60–74.

Prasad, R., Deo, R.C., Li, Y., Maraseni, T., 2017. Input selection and performance optimization of ann-based streamflow forecasts in the drought-prone murray darling basin region using iis and modwt algorithm. Atmospheric Research 197, 42–63.

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

Raissi, M., Perdikaris, P., Karniadakis, G., 2019. Physics-informed neural networks: A

deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational Physics 378, 686–707.

Raissi, M., Yazdani, A., Karniadakis, G.E., 2018. Hidden fluid mechanics: A navier-stokes informed deep learning framework for assimilating flow visualization data. arXiv preprint arXiv:1808.04327 .

Rao, V., Sandu, A., 2016. A time-parallel approach to strong-constraint four-dimensional variational data assimilation. Journal of Computational Physics 313, 583–593.

Rasouli, K., Hsieh, W.W., Cannon, A.J., 2012. Daily streamflow forecasting by machine learning methods with weather and climate inputs. Journal of Hydrology 414, 284–293.

Rasp, S., Pritchard, M.S., Gentine, P., 2018. Deep learning to represent subgrid processes in climate models. Proceedings of the National Academy of Sciences 115, 9684–9689.

Rathinasamy, M., Adamowski, J., Khosa, R., 2013. Multiscale streamflow forecasting using a new bayesian model average based ensemble multi-wavelet volterra nonlinear method. Journal of Hydrology 507, 186–200.

Rawat, W., Wang, Z., 2017. Deep convolutional neural networks for image classification: A comprehensive review. Neural computation 29, 2352–2449.

Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., Lee, H., 2016. Generative adversarial text to image synthesis. arXiv preprint arXiv:1605.05396 .

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., et al., 2019. Deep learning and process understanding for data-driven earth system science. Nature 566, 195.

Sampson, C., Carrassi, A., Aydoğdu, A., Jones, C.K., 2021. Ensemble kalman filter for nonconservative moving mesh solvers with a joint physics and mesh location update. Quarterly Journal of the Royal Meteorological Society 147, 1539–1561.

Seyedashraf, O., Rezaei, A., Akhtari, A.A., 2017. Dam break flow solution using artificial neural network. Ocean Engineering 142, 125–132.

Shao, M., Zhang, Y., Zeng, L., Tang, X., Zhang, J., Zhong, L., Wang, B., 2009. Ground-level ozone in the Pearl River Delta and the roles of VOC and NOx in its production. Journal of Environmental Management 90, 512–518.

Sharma, A., Hamarneh, G., 2019. Missing mri pulse sequence synthesis using multi-modal generative adversarial network. IEEE transactions on medical imaging 39, 1170–1183.

Shi, X., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.c., 2015. Convolutional lstm network: A machine learning approach for precipitation nowcasting. Advances in neural information processing systems 28.

Shortridge, J.E., Guikema, S.D., Zaitchik, B.F., 2016. Machine learning methods for empirical streamflow simulation: a comparison of model accuracy, interpretability, and uncertainty in seasonal watersheds. Hydrology & Earth System Sciences 20.

da Silva, R.G., Ribeiro, M.H.D.M., Moreno, S.R., Mariani, V.C., dos Santos Coelho, L., 2021. A novel decomposition-ensemble learning framework for multi-step ahead wind energy forecasting. Energy 216, 119174.

Ströfer, C.M., Wu, J., Xiao, H., Paterson, E., 2018. Data-driven, physics-based feature extraction from fluid flow fields. arXiv preprint arXiv:1802.00775 .

Su, X., An, J., Zhang, Y., Zhu, P., Zhu, B., 2020. Prediction of ozone hourly concentrations by support vector machine and kernel extreme learning machine using wavelet transformation and partial least squares methods. Atmospheric Pollution Research 11, 51–60.

Sudheer, C., Maheswaran, R., Panigrahi, B.K., Mathur, S., 2014. A hybrid svm-pso

model for forecasting monthly streamflow. Neural Computing and Applications 24, 1381–1389.

Sun, A.Y., Wang, D., Xu, X., 2014. Monthly streamflow forecasting using gaussian process regression. Journal of Hydrology 511, 72–81.

Tahmasebi, P., Kamrava, S., Bai, T., Sahimi, M., 2020. Machine learning in geo-and environmental sciences: From small to large scale. Advances in Water Resources 142, 103619.

Taieb, S.B., Bontempi, G., Atiya, A.F., Sorjamaa, A., 2012. A review and comparison of strategies for multi-step ahead time series forecasting based on the nn5 forecasting competition. Expert systems with applications 39, 7067–7083.

Taieb, S.B., Sorjamaa, A., Bontempi, G., 2010. Multiple-output modeling for multi-step-ahead time series forecasting. Neurocomputing 73, 1950–1957.

Taira, K., Hemati, M.S., Brunton, S.L., Sun, Y., Duraisamy, K., Bagheri, S., Dawson, S., Yeh, C.A., 2019. Modal analysis of fluid flows: Applications and outlook. arXiv preprint arXiv:1903.05750 .

Tang, Q., Schilling, O.S., Kurtz, W., Brunner, P., Vereecken, H., Hendricks Franssen, H.J., 2018. Simulating flood-induced riverbed transience using unmanned aerial vehicles, physically based hydrological modeling, and the ensemble kalman filter. Water resources research 54, 9342–9363.

Tarwani, K.M., Edem, S., 2017. Survey on recurrent neural network in natural language processing. Int. J. Eng. Trends Technol 48, 301–304.

Taylor, R., 1990. Interpretation of the correlation coefficient: a basic review. Journal of diagnostic medical sonography 6, 35–39.

Thawakar, O., Patil, P.W., Dudhane, A., Murala, S., Kulkarni, U., 2019. Image and video super resolution using recurrent generative adversarial network, in: 2019 16th IEEE in-

ternational conference on advanced video and signal based surveillance (AVSS), IEEE. pp. 1–8.

Tieleman, T., Hinton, G., 2012. Rmsprop: Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning. COURSERA Neural Networks Mach. Learn .

Viré, A., Xiang, J., Milthaler, F., Farrell, P.E., Piggott, M.D., Latham, J.P., Pavlidis, D., Pain, C.C., 2012. Modelling of fluid–solid interactions using an adaptive mesh fluid model coupled with a combined finite–discrete element model. Ocean Dynamics 62, 1487–1501.

Wang, H.W., Li, X.B., Wang, D., Zhao, J., Peng, Z.R., et al., 2020a. Regional prediction of ground-level ozone using a hybrid sequence-to-sequence deep learning approach. Journal of Cleaner Production 253, 119841.

Wang, M., Cheung, S.W., Leung, W.T., Chung, E.T., Efendiev, Y., Wheeler, M., 2020b. Reduced-order deep learning for flow dynamics. the interplay between deep learning and model reduction. Journal of Computational Physics 401, 108939.

Wang, W.c., Chau, K.w., Qiu, L., Chen, Y.b., 2015. Improving forecasting accuracy of medium and long-term runoff using artificial neural network based on eemd decomposition. Environmental research 139, 46–54.

Wang, Z., Xiao, D., Fang, F., Govindan, R., Pain, C.C., Guo, Y., 2018. Model identification of reduced order fluid dynamics systems using deep learning. International Journal for Numerical Methods in Fluids 86, 255–268.

Wang, Z.b., Li, J.x., Liang, L.w., 2020c. Spatio-temporal evolution of ozone pollution and its influencing factors in the Beijing-Tianjin-Hebei Urban Agglomeration. Environmental Pollution 256, 113419.

Werbos, P.J., 1990. Backpropagation through time: what it does and how to do it. Proceedings of the IEEE 78, 1550–1560.

Weyn, J.A., Durran, D.R., Caruana, R., 2020. Improving data-driven global weather prediction using deep convolutional neural networks on a cubed sphere. Journal of Advances in Modeling Earth Systems 12, e2020MS002109.

Wichakul, S., Tachikawa, Y., Shiiba, M., Yorozu, K., 2013. Development of a flow routing model including inundation effect for the extreme flood in the chao phraya river basin, thailand 2011. Journal of Disaster Research 8, 415.

Wiewel, S., Becher, M., Thuerey, N., 2019. Latent space physics: Towards learning the temporal evolution of fluid flow, in: Computer Graphics Forum, Wiley Online Library. pp. 71–82.

Wikner, A., Pathak, J., Hunt, B.R., Szunyogh, I., Girvan, M., Ott, E., 2021. Using data assimilation to train a hybrid forecast system that combines machine-learning and knowledge-based components. Chaos: An Interdisciplinary Journal of Nonlinear Science 31, 053114.

Williams, R.J., Zipser, D., 1989. A learning algorithm for continually running fully recurrent neural networks. Neural computation 1, 270–280.

Wu, P., Chang, X., Yuan, W., Sun, J., Zhang, W., Arcucci, R., Guo, Y., 2021. Fast data assimilation (fda): Data assimilation by machine learning for faster optimize model state. Journal of Computational Science 51, 101323.

Xiao, D., Fang, F., Zheng, J., Pain, C., Navon, I., 2019a. Machine learning-based rapid response tools for regional air pollution modelling. Atmospheric Environment 199, 463–473.

Xiao, D., Heaney, C., Fang, F., Mottet, L., Hu, R., Bistrian, D., Aristodemou, E., Navon,

I., Pain, C., 2019b. A domain decomposition non-intrusive reduced order model for turbulent flows. Computers & Fluids 182, 15–27.

Xiao, D., et al., 2016. Non-intrusive reduced order models and their applications. Ph.D. thesis. Imperial College London.

Xie, Y., Franz, E., Chu, M., Thuerey, N., 2018. tempogan: A temporally coherent, volumetric gan for super-resolution fluid flow. ACM Transactions on Graphics (TOG) 37, 95.

Xu, L., Chen, N., Chen, Z., Zhang, C., Yu, H., 2021. Spatiotemporal forecasting in earth system science: Methods, uncertainties, predictability and future directions. Earth-Science Reviews 222, 103828.

Xu, X., Zhang, X., Fang, H., Lai, R., Zhang, Y., Huang, L., Liu, X., 2017. A real-time probabilistic channel flood-forecasting model based on the bayesian particle filter approach. Environmental Modelling & Software 88, 151–167.

Yafouz, A., Ahmed, A.N., Zaini, N., Sherif, M., Sefelnasr, A., El-Shafie, A., 2021. Hybrid deep learning model for ozone concentration prediction: comprehensive evaluation and comparison with various machine and deep learning algorithms. Engineering Applications of Computational Fluid Mechanics 15, 902–933.

Yan, R., Liao, J., Yang, J., Sun, W., Nong, M., Li, F., 2021. Multi-hour and multi-site air quality index forecasting in Beijing using CNN, LSTM, CNN-LSTM, and spatiotemporal clustering. Expert Systems with Applications 169, 114513.

Yang, C.C., Lacroix, R., Prasher, S., 1998. The use of back-propagation neural networks for the simulation and analyses of time-series data in subsurface drainage systems. Transactions of the ASAE 41, 1181.

Yang, G., Liu, Y., Li, X., 2020. Spatiotemporal distribution of ground-level ozone in China at a city level. Scientific reports 10, 1–12.

Yang, T., Asanjan, A.A., Welles, E., Gao, X., Sorooshian, S., Liu, X., 2017. Developing reservoir monthly inflow forecasts using artificial intelligence and climate phenomenon information. Water Resources Research 53, 2786–2812.

Yang, Y., Dunham, E.M., Barnier, G., Almquist, M., 2019. Tsunami wavefield reconstruction and forecasting using the ensemble kalman filter. Geophysical Research Letters 46, 853–860.

Yaseen, Z.M., Allawi, M.F., Yousif, A.A., Jaafar, O., Hamzah, F.M., El-Shafie, A., 2018. Non-tuned machine learning approach for hydrological time series forecasting. Neural Computing and Applications 30, 1479–1491.

Yaseen, Z.M., El-Shafie, A., Jaafar, O., Afan, H.A., Sayl, K.N., 2015. Artificial intelligence based models for stream-flow forecasting: 2000–2015. Journal of Hydrology 530, 829–844.

Yaseen, Z.M., Kisi, O., Demir, V., 2016. Enhancing long-term streamflow forecasting and predicting using periodicity data component: application of artificial intelligence. Water resources management 30, 4125–4151.

Yetilmezsoy, K., Ozkaya, B., Cakmakci, M., 2011. Artificial intelligence-based prediction models for environmental engineering. Neural Network World 21.

Yin, Z., Cao, B., Wang, H., 2019. Dominant patterns of summer ozone pollution in eastern China and associated atmospheric circulations. Atmospheric Chemistry and Physics 19, 13933–13943.

Young, T., Hazarika, D., Poria, S., Cambria, E., 2018. Recent trends in deep learning based natural language processing. ieee Computational intelligenCe magazine 13, 55–75.

Yuval, J., O'Gorman, P.A., 2020. Stable machine-learning parameterization of subgrid

processes for climate modeling at a range of resolutions. Nature communications 11, 1–10.

Zadeh, M.R., Amin, S., Khalili, D., Singh, V.P., 2010. Daily outflow prediction by multi layer perceptron with logistic sigmoid and tangent sigmoid activation functions. Water resources management 24, 2673–2688.

Zhan, Y., Luo, Y., Deng, X., Grieneisen, M.L., Zhang, M., Di, B., 2018. Spatiotemporal prediction of daily ambient ozone levels across China using random forest for human exposure assessment. Environmental Pollution 233, 464–473.

Zhang, D., Lin, J., Peng, Q., Wang, D., Yang, T., Sorooshian, S., Liu, X., Zhuang, J., 2018. Modeling and simulating of reservoir operation using the artificial neural network, support vector regression, deep learning algorithm. Journal of hydrology 565, 720–736.

Zhang, K., Thé, J., Xie, G., Yu, H., 2020. Multi-step ahead forecasting of regional air quality using spatial-temporal deep neural networks: a case study of huaihai economic zone. Journal of Cleaner Production 277, 123231.

Zhang, Q., Zhang, M., Chen, T., Sun, Z., Ma, Y., Yu, B., 2019. Recent advances in convolutional neural network acceleration. Neurocomputing 323, 37–51.

Zhong, Z., Sun, A.Y., Jeong, H., 2019. Predicting co2 plume migration in heterogeneous formations using conditional deep convolutional generative adversarial network. Water Resources Research 55, 5830–5851.

Zhou, Y., Chang, F.J., Chang, L.C., Kao, I.F., Wang, Y.S., 2019. Explore a deep learning multi-output neural network for regional multi-step-ahead air quality forecasts. Journal of cleaner production 209, 134–145.

Zhou, Z., De Kat, J., Buchner, B., 1999. A nonlinear 3d approach to simulate green

water dynamics on deck, in: Proceedings of the Seventh International Conference on Numerical Ship Hydrodynamics, Nantes, France, pp. 1–15.

APPENDIX A

Long lead-time daily and monthly streamflow forecasting using machine learning methods

A.1 Supplementary information related to structure of ANN and LSTM models

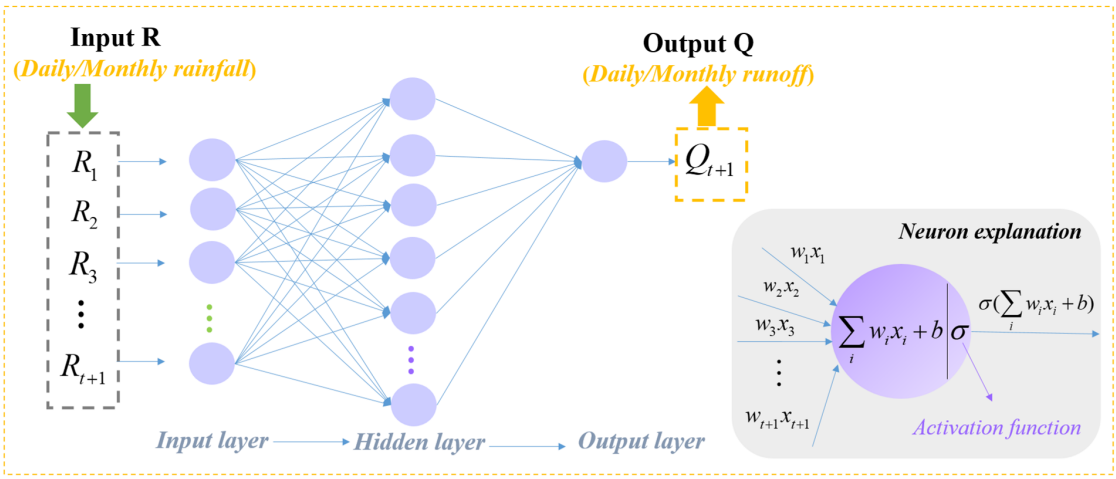


Fig. A.1. A typical three-layer feedforward artificial neural networks.

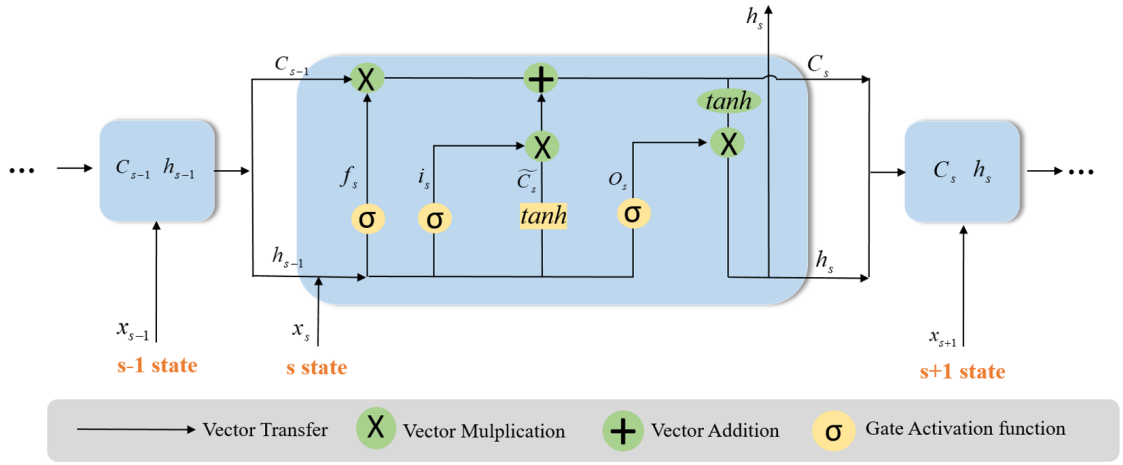


Fig. A.2. The structure of the cell connection in the Long Short-Term Memory (LSTM) neural network.

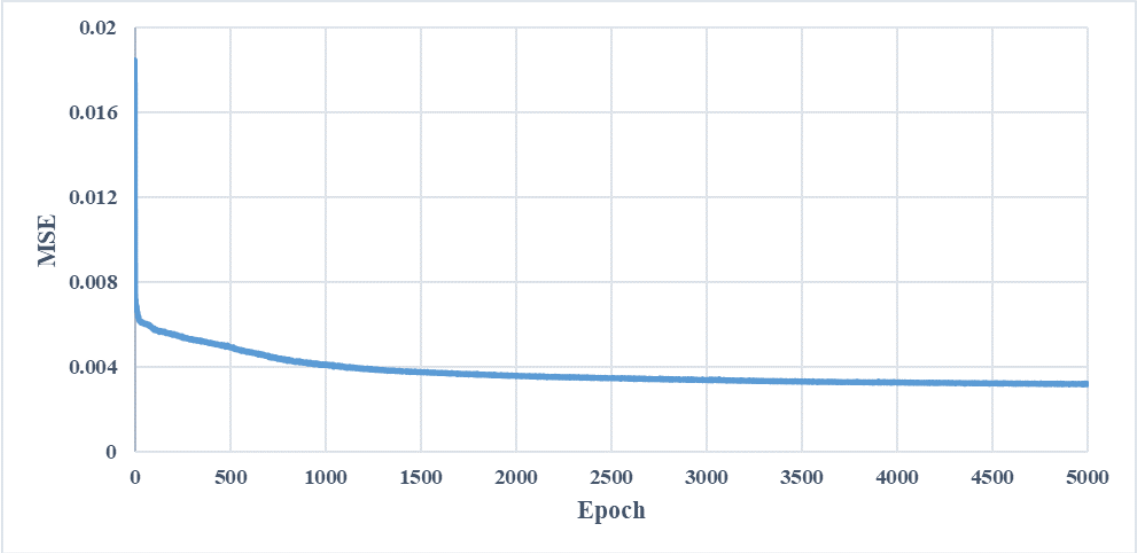


Fig. A.3. Loss variation in the ANN model training process.

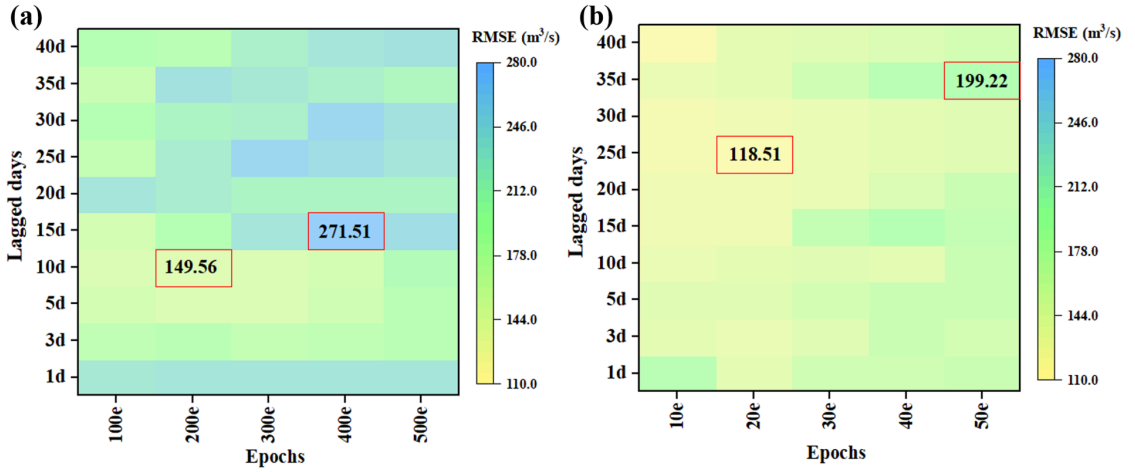


Fig. A.4. The map of the RMSE of daily forecasting streamflow varying with the number of time lag and epochs in (a) ANN modelling, and (b) LSTM modelling.

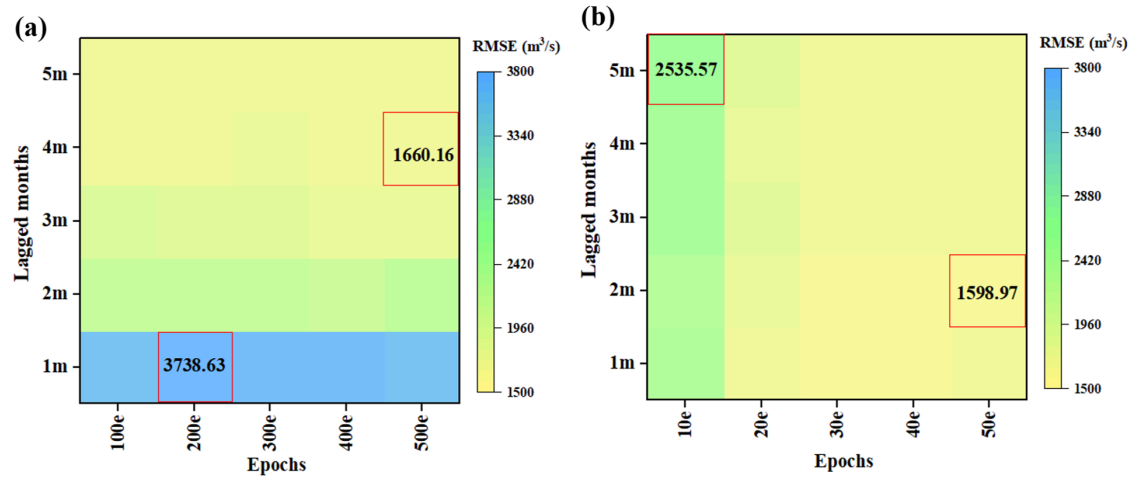


Fig. A.5. The map of the RMSE of monthly forecasting streamflow varying with the number of time lag and epochs in (a) ANN modelling, and (b) LSTM modelling.

A.2 Multi-day-ahead streamflow forecast at PRB

A.2.1 Study area

The second study site PRB, as one of major subbasins of CPRB, covers the area of 26,300 km^2. The PRB covers substantial portions of Tak and Kamphaeng Phet provinces, but only a small part of the Nakhon Sawan province. The Bhumibol Dam reservoir discharge

a significant amount of water to supply domestic and industrial use, irrigation and environmental purposes as shown in Fig. 4.2. The basin is in a subtropical monsoon region, with relatively abundant rainfall and a humid climate from May to September. The average annual precipitation over the basin is about 1200 mm. To determine the relationship between rainfall and runoff, streamflow into the Bhumibol Dam reservoir and rainfall data from gauges were collected at daily and monthly scales from 1975 to 2014.

A.2.2 Multi-day-ahead streamflow forecast

The ANN and LSTM models have been further applied to multi-day-ahead streamflow forecasting in PRB. To evaluate the forecasting performance, the comparative plots of the forecasting results obtained from the ANN and LSTM models at the lead-time = 1 day, 2 day, 3 day, 4 day, and 5 day are shown in Fig. A.6. It can be observed a relatively good agreement between the observed and forecasting streamflow especially in low streamflow with lead-time = 1 day and 2 day from 27 February 2014 to 9 March 2014. With increasing the lead-time, the arrival time of forecasting streamflow peaks becomes slightly delayed, which is caused by the accumulation of forecast errors. Comparing with high streamflow simulation from 28 April 2013 to 19 May 2013, the streamflow peaks of the ANN model are generally higher than observed streamflow, while the streamflow peaks of the LSTM model match better.

Four indicators, RMSE, NSE, CC, and MAE, are chosen to evaluate the multi-day-ahead forecast performance during the validation period (from 3 April 2012 to 25 July 2014). The corresponding evaluation of forecasting streamflow results during the lead-times of 1-20 days is illustrated in Fig. A.7. It can be seen that the LSTM model generally outperforms the ANN model with the smaller RMSE and MAE (in Fig. A.7 (a) and (d)), and higher NSE and CC especially at the longer lead-times (in Fig. A.7 (b) and (c)). It is also visible that the predicted accuracy of the ANN and LSTM models decays as the forecasting horizon extends further in time (i.e. with the increasing lead-time length). As the

lead-time increases from 1 to 20 days, the RMSE and MAE in ANN modelling increase approximately by 52% and 40% respectively, while in LSTM modelling, the RMSE and MAE increase approximately by 78% and 45% respectively. The NSE ranges between 0.2 and 0.3 for the ANN model, while between 0.1 and 0.52 for the LSTM model from lead-time = 1 day to lead-time = 6 day. The CC decreases from 0.71 to 0.37 for the ANN model and from 0.8 and 0.43 for the LSTM model as the lead-time increases. The results suggest that the forecasting performance for the LSTM model is better than that of the ANN model for long lead-time daily forecasting.

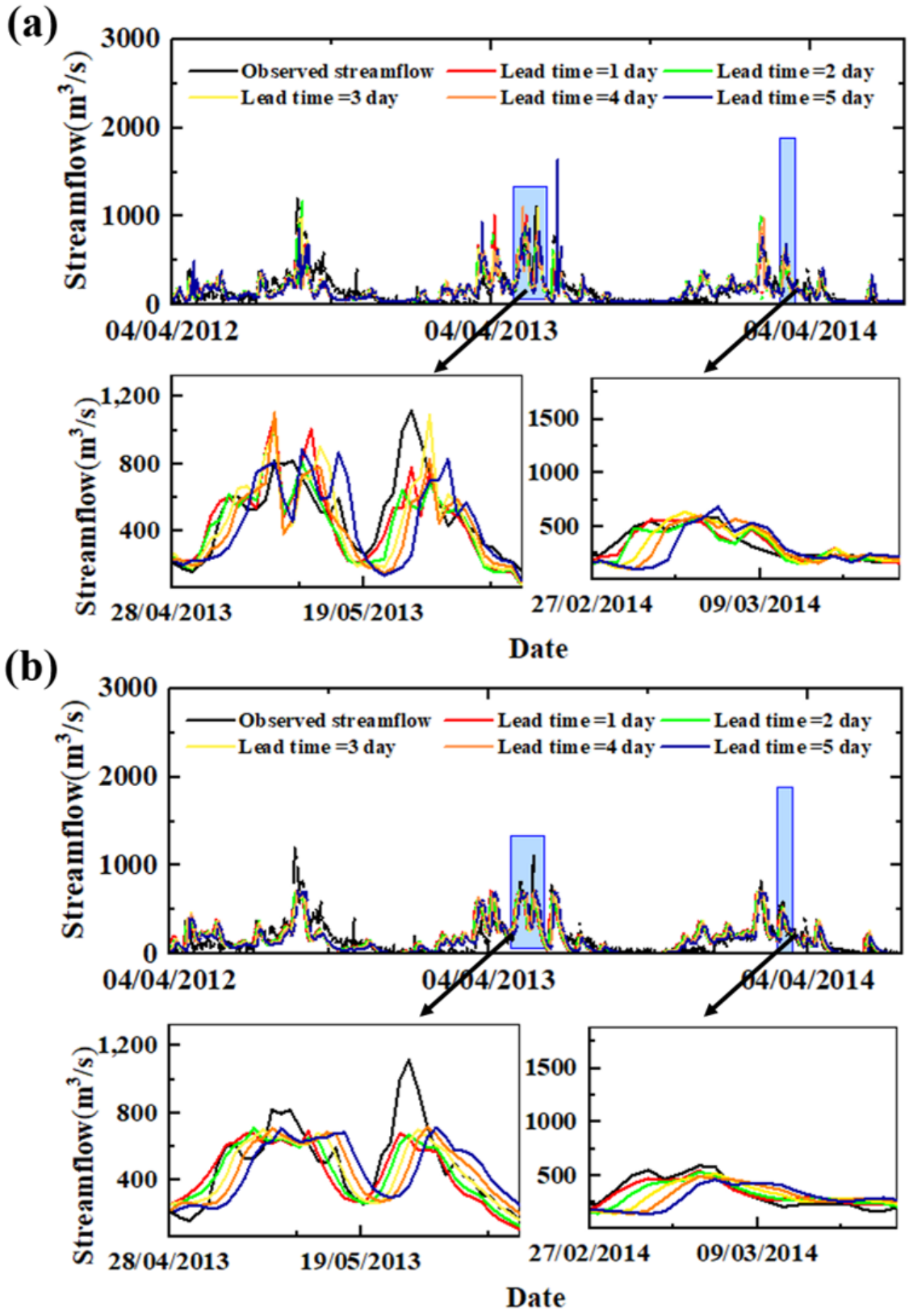


Fig. A.6. Forecast performance at the lead-times 1, 2, 3, 4 and 5 days: (a) with the ANN model and (b) with the LSTM model.

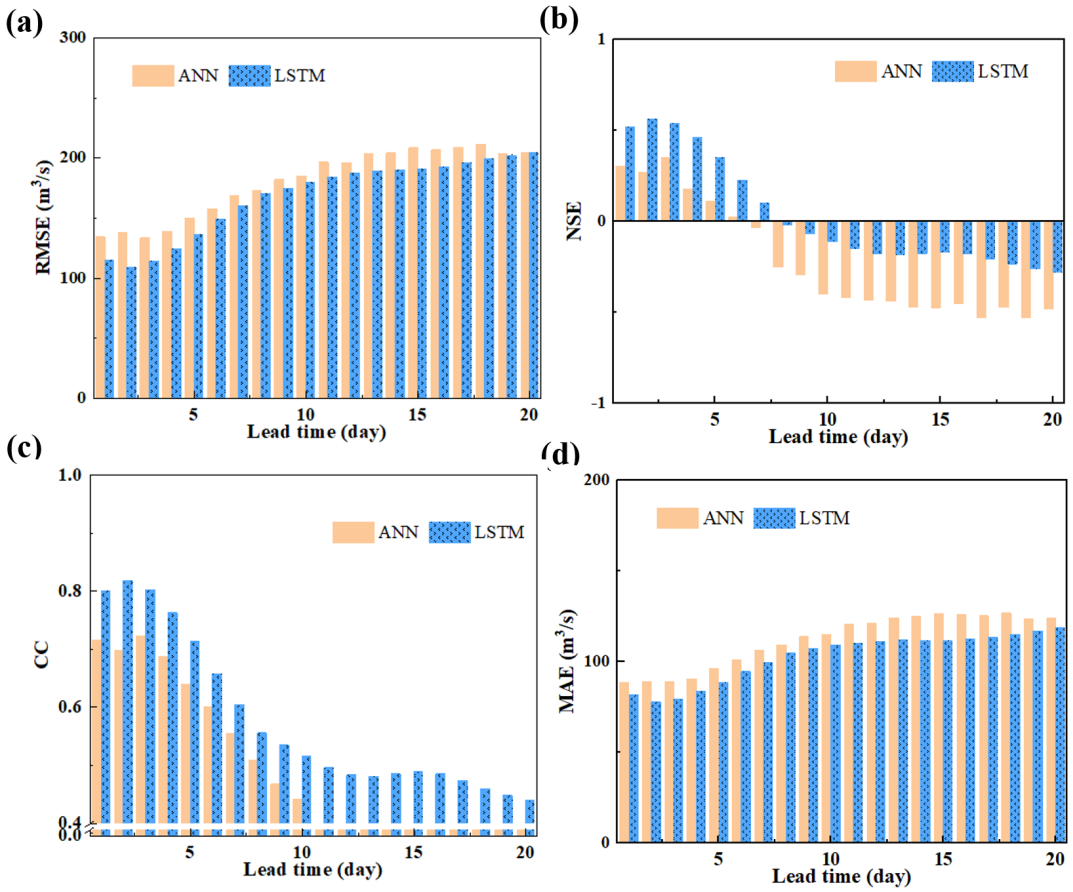


Fig. A.7. Forecast performance with the ANN and LSTM models over the validation period of year 2012 to 2014 at the lead-times of 1-20 days: (a) RMSE, (b) NSE, (c) CC and (d) MAE.

APPENDIX B

Spatio-temporal hourly and daily ozone forecasting in China using a hybrid machine learning model: Autoencoder and Generative Adversarial Networks

B.1 Model comparison

In the section, the performance of our VAE-GAN model is evaluated against a VAE-GAN-LSTM model, which introduces the convolutional long short-term memory (ConvLSTM) layer (Shi et al., 2015) into the encoder and decoder of the VAE-GAN. The comparative plots of hourly ozone forecasting results obtained from the VAE-GAN and VAE-GAN-LSTM models during the predictive period of June in 2018 are shown in Fig. B.1. It can be observed in Fig. B.1(a) that a relatively good agreement between the observed and forecasting hourly ozone concentration is achieved, while the arrival time of forecasting ozone peaks is slightly delayed. We can see that during the validation period of days 22-26 in June 2018, the VAE-GAN-LSTM model can better predict the ozone peaks than

the VAE-GAN model, where the reanalysis data is considered as the reference. However, the VAE-GAN-LSTM model fails to capture the low ozone concentration at night, thus having higher the RMSEs than that of the VAE-GAN model in Fig. B.1(b). This situation also reflects on the spatial distribution of ozone forecasts as shown in Fig. B.2. Generally speaking, the use of LSTM in the VAE-GAN can well capture the high ozone concentration although the RMSE of results from the VAE-GAN-LSTM is slightly higher than that of the VAE-GAN. It is thus suggested that the VAE-GAN-LSTM model has the potential for hourly high ozone forecasting in China.

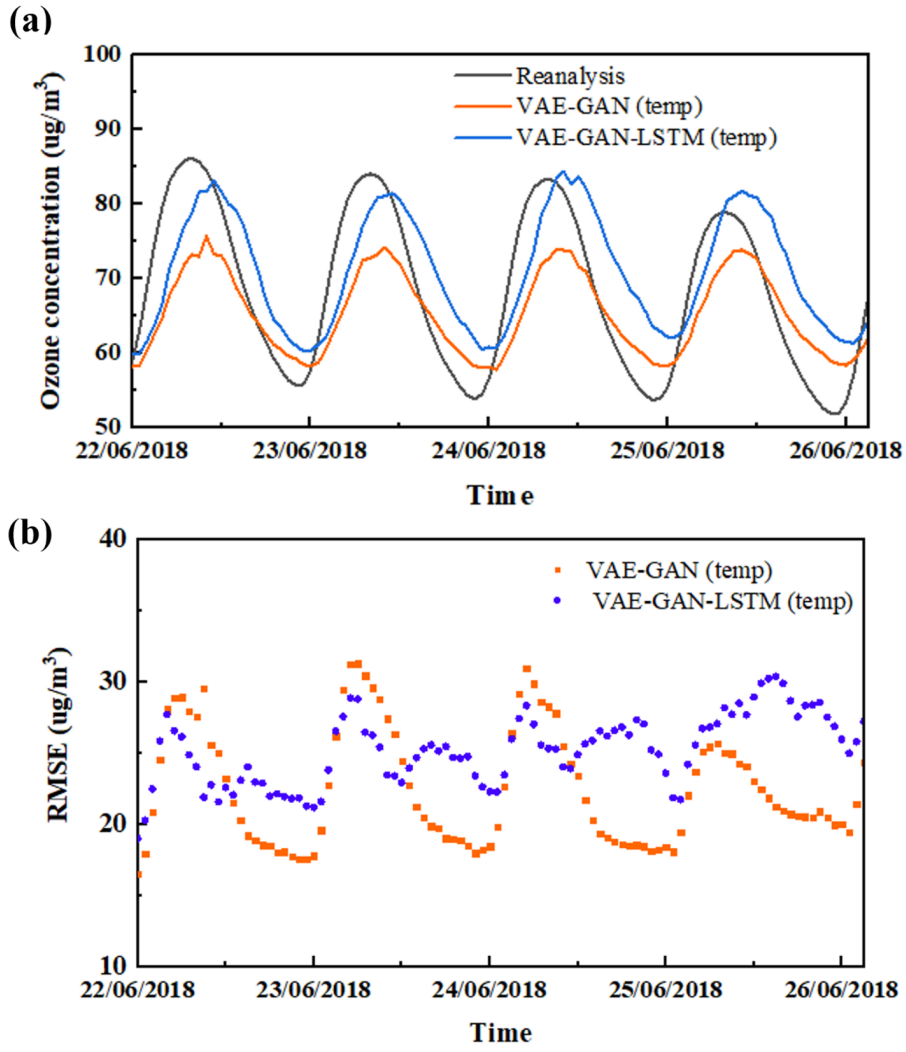


Fig. B.1. Spatial-averaged ozone concentration variation during predictive period of June in 2018. The VAE-GAN and VAE-GAN-LSTM models take both the temperature and last-step predictive ozone as the model inputs. The VAE-GAN-LSTM model adopts the convolutional long short-term memory (ConvLSTM) layer into the encoder and decoder of the VAE-GAN.

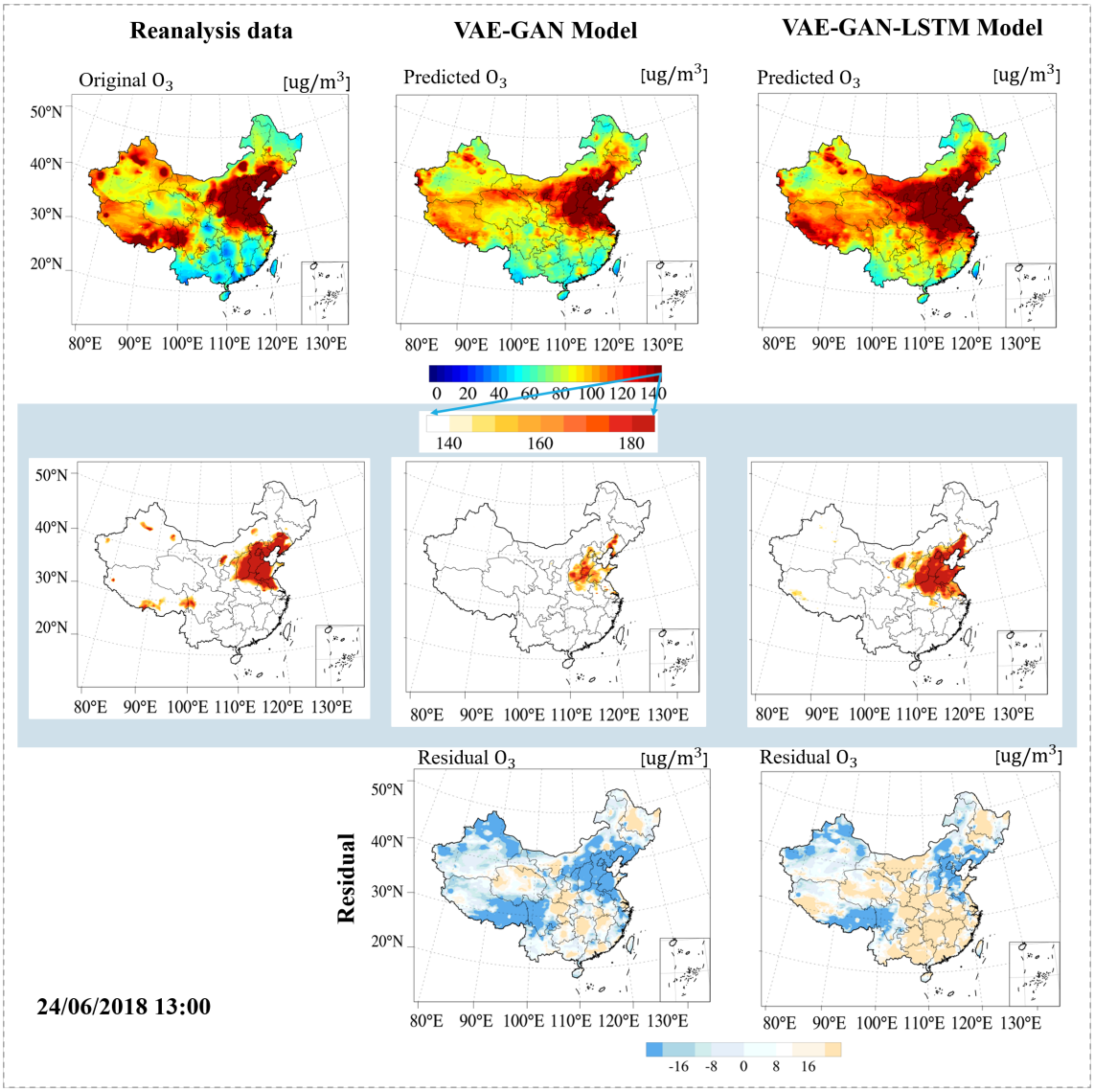


Fig. B.2. Comparison of the spatial distribution of ozone fields obtained from the VAE-GAN model and the VAE-GAN-LSTM at the lead-time of $60h$ in June of 2018.

B.2 Ozone forecasting using NO_2

NO_2 is the primary element involved in the process of forming ozone. When sunlight interacts with NO_2, the gas molecules break down and release monoatomic oxygen (O) atoms which then react to diatomic oxygen (O_2) molecules and form the trioxygen molecule known as ozone (Ezimand and Kakroodi, 2019). Here the impacts of the precursors (here, NO_2) on ozone concentrations have been explored using the proposed VAE-GAN model.

B.2.1 Hourly ozone forecasting using NO_2 from reanalysis datasets

In real-time ozone forecasting in realistic scenarios, both NO_2 and ozone are the unknown variables (outputs) to be predicted/solved using either the air quality model or the AI-based model (here VAE-GAN). Thus it is not suitable to consider NO_2 as the model input in real-time ozone forecasting. Even so, it is worth exploring the relationship between NO_2 and ozone using the VAE-GAN model based on history ozone and NO_2 datasets, where the NOx concentration is the model input while the ozone concentration is the model output.

the hourly NO_2 and ozone datasets are collected from the reanalysis CAQRA datasets. We used the hourly datasets in 2017 for training, and 2018 for validation. The spatial performance of the VAE-GAN model for hourly ozone forecasting in June and December of 2018 is shown in Fig. B.3. It can be seen that the VAE-GAN model captures a large part of the spatial distribution of ozone concentration at lead-times $t = 30, 60h$ in December. The last row of Fig. B.3 shows the residual maps between ozone maps predicted by the VAE-GAN and reanalysis data. In June, the predicted ozone concentration is slightly overestimated over the south of China at different lead-times, while slightly underestimated over the NCP region.

The RMSE and correlation coefficient of predictive results from the VAE-GAN are shown in Fig. B.4. It can be noticed that the RMSE values in June and December in 2018 are generally below $17ug/m^3$ during the whole forecasting period. The average RMSE values are $14.56ug/m^3$ and $8.17ug/m^3$ for June and December respectively. The corresponding correlation coefficient values are shown in Fig. B.4(b). We note that in particular, models which accurately capture the regime behaviour will also show good correlation statistics when averaged over a long lead-time. Most of the correlation coefficients are around or beyond 0.8 during the forecasting period. It is worth noting that the magnitude of the correlation coefficient in December is close to 0.9. This means the VAE-GAN performs

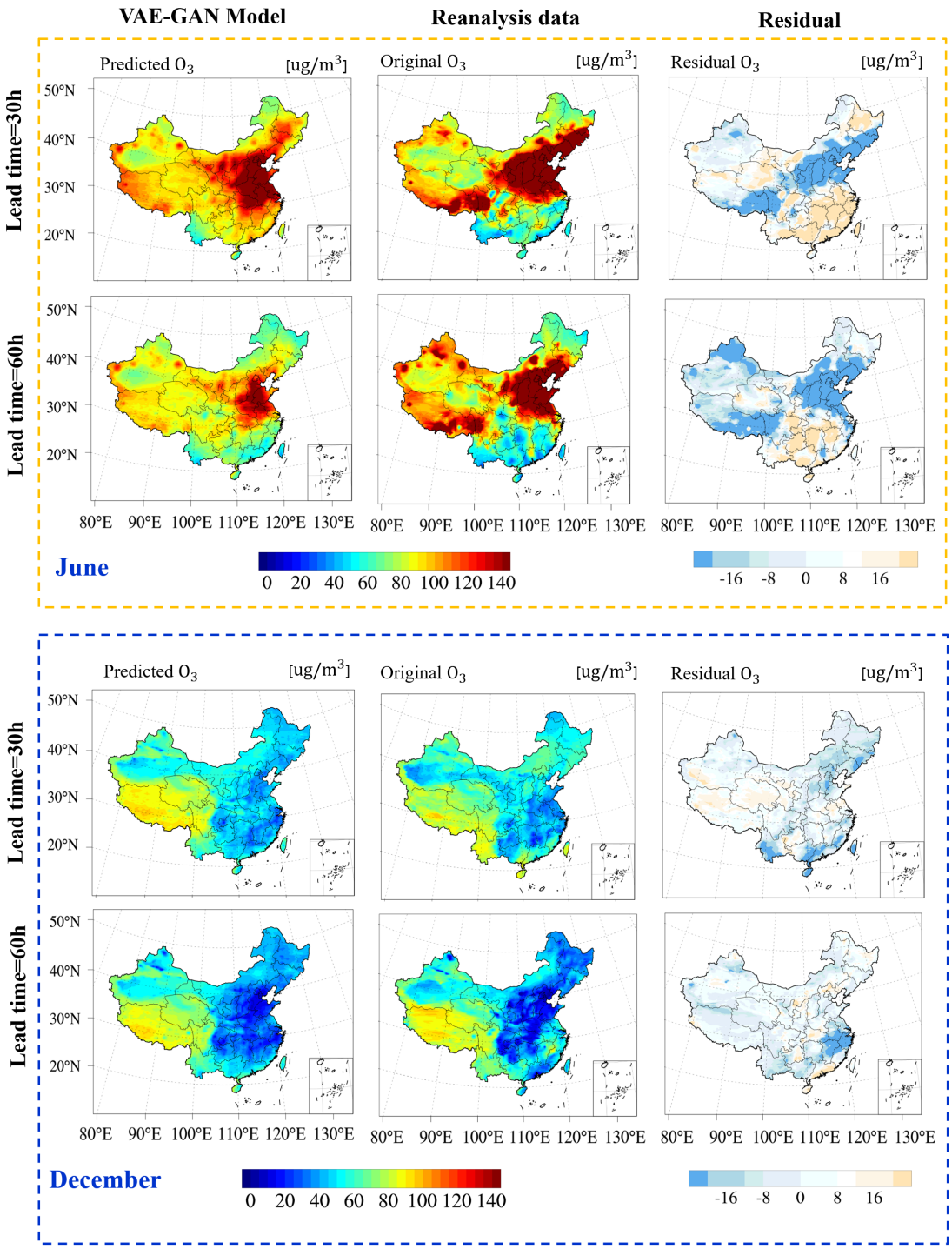


Fig. B.3. Comparison of the spatial distribution of ozone fields obtained from the VAE-GAN model (left) and the reanalysis data (middle) at the lead-times of $30h$ and $60h$ in June and December in 2018 respectively.

better in hourly ozone forecasting in December.

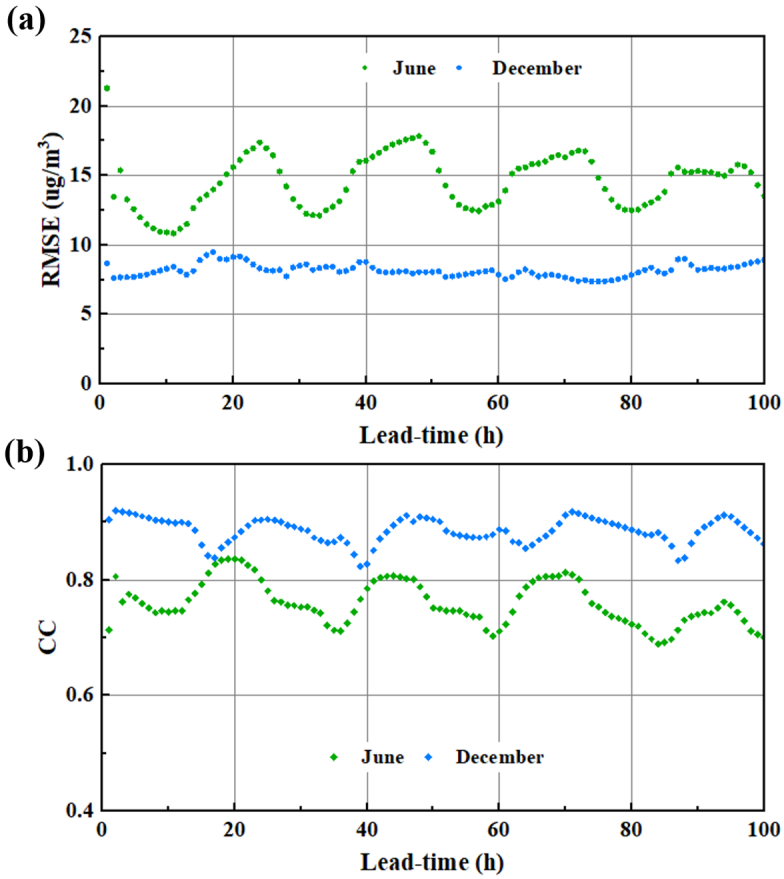


Fig. B.4. The RMSE and correlation coefficients of ozone concentration obtained from the VAE-GAN and the reanalysis data in two months in 2018.

B.2.2 Impact of NO_2 emissions on ozone forecasting

In realistic scenarios, the local NOx emission from massive anthropogenic emissions, industries, or transportation is another factor impacting ozone forecasting. In this work, the NOx and VOC emissions were considered as inputs, but removed during the selection of driving variables. There are two reasons for this. First, the available NOx emission data (source) are averaged monthly or daily. The daily variation of NOx emission remains the same during the same month. The NOx emissions are constant at a daily-temporal scale. Thus, the hourly-temporal variation of ozone concentrations will not be affected by a constant NOx emission during the training process. However, the emission of NOx

varies over space, e.g. a high emission in densely populated cities. The impact of the spatial variation of NOx emission is implicitly considered during the training process. The reason for this is that the training datasets (the reanalysis datasets) combine the physical modelling results, where NOx and VOC emissions have been considered in NAQPMS model (Kong et al., 2021) and observations. We speculate the predictive results can reflect the contribution of other driven factors (including NOx and VOC) on ozone concentration. To validate this hypothesis, we have undertaken one modelling experiment, where the average temperature ($27°C \pm 3°C$) in June is used as inputs everywhere over China. Can we expect the VAE-GAN models are able to predict the spatial distribution of ozone concentrations over China? The right column in Fig. B.5 shows the ozone concentration variation where the input temperature is around 300K ($27°C \pm 3°C$). We can see that the predictive ozone varies in space with the same temperature as model inputs. It is noted that the ozone variation over the regions can match well with the NO_2 emission (right column in Fig. B.5) in June, a high ozone concentration in the densely populated cities, mainly in the east of China. Therefore, the predictive ozone in space has indicated the patterns of ozone formation with emission intensities associated with industrial and agricultural activities.

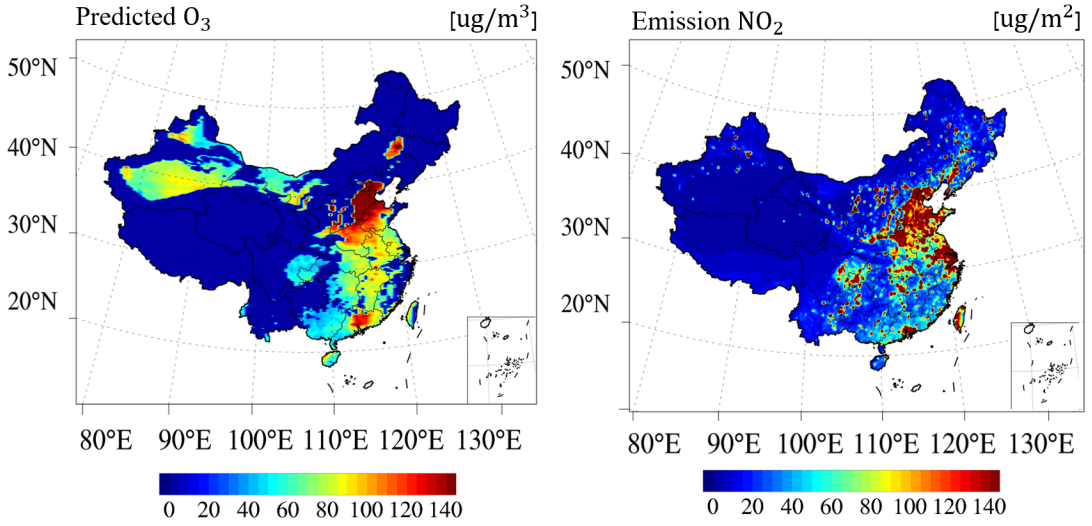


Fig. B.5. The predictive ozone (left) and emission of NO_2 (right) distribution of China in June of 2018.

Second, in ozone forecasting, the importance of NOx and NMVOC emissions is lower than expected although the ozone formation is associated with NOx emissions. Zhan et al. (2018) pointed out that this may be due to regional transport or titration effects. NOx and NMVOC may be transported from urban to rural areas via advection, causing the location of ozone formation to occur some distance away from the emission source. NOx titration tends to counteract the importance of NOx emissions to ozone levels. In the work of Ma et al. (2021a), they also found that ozone precursors such as NO_2 and satellite-based formaldehyde (HCHO) observations were found to play relatively weak roles in modulating ozone variations, though these factors were crucial to the ozone formation. In addition, Zhan et al. (2018) found that the inclusion of NOx and NMVOC into the machine learning model can introduce a noise which tends to be higher than the gained information, which is consistent with what occurred in our study.

Finally, we would like to further discuss the importance of the choice of driving variables in ozone forecasting. For machine learning-based ozone forecasting at stations, the model performance may be improved with increasing inputs (Su et al., 2020; Yafouz et al., 2021). In contrast, for spatio-temporal ozone forecasting, with the increase of the number of driving factors, the number of modelling parameters to be optimially estimated will significantly increase, especially on a large spatial scale. It would thus compromise all the drivers during the training process, thus weakening the relationship between the most important driver (e.g. temperature) and ozone concentrations. It is therefore suggested that only key-driven variables should be chosen for model prediction. This is why the NOx emission is removed from modelling inputs due to its importance.

APPENDIX C

Software and error analysis

C.1 Fluidity

Fluidity is an open source and multi-phase CFD code and capable of solving numerically the Navier-Stokes and accompanying field equations on unstructured finite element meshes in one, two and three dimensions (Pain et al., 2001; Piggott et al., 2008). The main feature of Fluidity is its ability to adapt the mesh dynamically in time and space of interest. The code is parallelizable which makes it attractive for larger-scale computationally challenging applications, such as tidal turbines (Abolghasemi et al., 2016), 3D tsunami propagation (Oishi et al., 2013), fluid-solid interactions (Viré et al., 2012), and multiphase porous media flow (Gomes et al., 2017).

More details on governing equations, numerical discretization and adaptive remeshing in Fluidity can refer to Fluidity manual (Fluidity, 2015).

C.2 MIKE 3 FM

MIKE 3 FM (MIKE21 and MIKE3 Flow Model, 2009), is a software tool that simulates flows, cohesive sediments, water quality, and ecology in lakes, rivers, coastal areas and seas in three dimensions. In flooding modelling, the hydrodynamic module calculates the flow fields based on a variety of forcing and boundary conditions. The governing equations are the shallow water equations, which invoke a hydrostatic pressure assumption. The time integration of the shallow water equations and the transport (advection-dispersion) equations is performed using a semi-implicit scheme, where the horizontal terms are treated explicitly and the vertical terms are treated implicitly. The Courant–Friedrichs–Lewy (CFL) number is set less than 1. A variable time step interval is used in the calculation of both the shallow water equations and the transport equations determined so that the CFL number is less than a critical CFL number in all computational nodes.

For more details of MIKE 3 FM, see DHI (2017).

C.3 Error analysis

The statistical metrics for error analysis of the developed models in this thesis are summarized below:

Table C.1: List of the statistical metrics used in model evaluation.

Metrics	Equation	Notes	Used in Chapters
Correlation Coeffcient (R)	$R = \frac{\sum_{i=1}^{n}(\phi_i-\overline{\phi})(\phi_{d,i}-\overline{\phi}_d)}{\sqrt{\sum_{i=1}^{n}(\phi_i-\overline{\phi})^2}\sqrt{\sum_{i=1}^{n}(\phi_{d,i}-\overline{\phi}_d)^2}}$	range: (-1,1), close to 1 is better	Chapters 2-7
Root Mean Squared Error (RMSE)	$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\phi_i-\phi_{d,i})^2}{n}}$	range: $(0, \infty)$, smaller is better	Chapters 2-7
Nash–Sutcliffe Effciency Coeffcient (NSE)	$NSE = 1 - \left[\frac{\sum_{i=1}^{n}(\phi_{d,i}-\phi_i)^2}{\sum_{i=1}^{n}(\phi_{d,i}-\overline{\phi}_d)^2}\right]$	range: (0,1), close to 1 is better	Chapters 4 and 5
Mean Absolute Error (MAE)	$MAE = \frac{1}{n}\sum_{i=1}^{n}\left\|\phi_i - \phi_{d,i}\right\|$	range: (0, ∞), smaller is better	Chapter 4
Mean Magnitude of Relative Error (MMRE)	$MMRE = \frac{1}{n}\sum_{i=1}^{n}\left\|\frac{\phi_i-\phi_{d,i}}{\phi_{d,i}}\right\|$	range: (0, ∞), (0,1) is acceptable	Chapter 5
Nash-Sutcliffe efficiency ratio (NSER)	$NSER = \left[\frac{\sum_{i=1}^{n}(\phi_{d,i}-\phi_i)^2}{\sum_{i=1}^{n}(\phi_{d,i}-\overline{\phi}_d)^2}\right]$	range: (0, ∞), (0,1) is acceptable	Chapter 5
Percentage of Accuracy-Precision (PAP)	$PAP = 100\left(1 - \frac{\sqrt{2}}{2}\sqrt{\left(\frac{1}{n}\sum_{i=1}^{n}\left\|\frac{\phi_i-\phi_{d,i}}{\phi_{d,i}}\right\|\right)^2 + \left[\frac{\sum_{i=1}^{n}(\phi_i-\phi_{d,i})^2}{\sum_{i=1}^{n}(\phi_{d,i}-\overline{\phi_{d,i}})^2}\right]^2}\right)$	range: (0,100), close to 100 is better	Chapter 5

Note: ϕ and ϕ_d represent the predicted and observed datasets.