Imperial College London

Department of Earth Science and Engineering

MSc in Environmental Data Science and Machine Learning

Independent Research Project
Final Report

Modelling pollution in the urban environment using neural networks

by

Hanson Shen

Email: hs1623@imperial.ac.uk
GitHub username: edsml-hs1623
Repository: https://github.com/ese-msc-2023/irp-hs1623

Supervisors:

Dr Claire E. Heaney
Dr Christopher C. Pain

August 2024

Abstract

Air pollution is a critical public health concern, particularly in urban areas where vehicle emissions significantly contribute to deteriorating air quality. This study introduces a novel approach to modeling urban air pollution by integrating Convolutional Variational Autoencoders (VAEs) with computational fluid dynamics (CFD) using AI4PDEs, an advanced in-house CFD code that leverages neural networks to solve discretized systems of equations. The framework incorporates dynamic data assimilation (DA) to continuously refine the model by comparing CFD simulations with real-world observations from monitoring systems. This process, conducted in the latent space, significantly reduces computational costs while maintaining high data accuracy. The proposed method addresses the limitations of traditional CFD and statistical models by providing a more efficient and adaptive solution for urban air quality monitoring. The study demonstrates that the integration of DA enhances prediction accuracy, as shown by the reduction in data mismatches between simulations and observations. The results underscore the model's potential to inform policy decisions by providing more reliable assessments of pollution exposure in complex urban environments. This approach offers a robust framework for improving urban air quality management through enhanced predictive capabilities.

1 Introduction

Air pollution in urban environments, primarily from vehicle emissions, poses severe health risks to humans and ecosystmes [11, 19]. However, accurate modelling of pollution dispersion in cities is challenging due to the intricate nature of airflows and the diverse sources of pollutant [9, 5]. Commercially, addressing this challenge is critical for improving public health outcomes and supporting urban planning decisions. Academically, there is a growing need to develop more efficient and adaptable models that can capture the dynamics of urban air pollution with greater accuracy. This research addresses two significant issues in the current approaches to urban air quality modeling: the limitations of traditional modeling methods and the underexplored potential of integrating data assimilation (DA) in the latent dimension [1, 2] to overcome computational constraints [4].

The first problem concerns existing methods for modeling air pollution, such as traditional computational fluid dynamics (CFD) models. While these are known for their precision, they are often computationally intensive and time-consuming, requiring substantial computing resources to simulate urban airflows and pollution dispersion in detail [6, 21]. In contrast, statistical models, while less resource-intensive, often lack the precision needed for detailed urban analysis and struggle with real-time adaptability, limiting their utility in dynamic urban environments. Therefore, building on recent advancements in neural networks [6, 16, 17], this research integrates a Convolutional Variational Autoencoder (VAE) with AI4PDEs [6], a neural network-based CFD solver, to address these issues [12]. The AI4PDEs framework offers a novel approach by solving discretized equations using neural networks, significantly reducing the computational overhead without sacrificing accuracy.

The second problem addressed by this research is the underuitilisaton of data assimilation techniques in the context of air pollution modelling. Although previous studies have demonstrated the potential of neural networks in CFD [6, 16, 17], the assimilation with dynamic data in the reduced space, or the latent dimension, remains underexplored [1, 2, 20]. This research tackled this gap by incorporating observational traffic flow through data assimilation to refine the convolutional VAE model [3, 7, 15], enabling it to dynamically adjust to realistic environmental observations and providing more accurate and reliable predictions for urban air quality monitoring [10, 8, 13])

The objectives of this research were to develop a reliable model for predicting pollution dispersion, validate the model against existing air quality measurements, and conduct sensitivity analyses [14, 13, 18] to refine the model by understanding the influence of various parameters. The work completed involved a novel and comprehensive system that includes preprocessing historical and traffic data, segmenting urban domains into manageable subdomains, applying advanced neural network techniques to simulate air pollution dynamics effectively, and assimilating observational data to refine the result.

This study goes beyond the state-of-the-art by enhancing the efficiency of modeling methods and integrating dynamic data assimilation in the latent dimension, thereby advancing the capabilities of urban air quality monitoring. The original contributions of this work include the development of a novel VAE-based model integrated with AI4PDEs, and the innovative use of DA in the latent space, providing a robust framework for predicting and managing urban air pollution. This research offers significant advancements in understanding and addressing urban air quality issues, ultimately contributing to improved living conditions and reduced health risks associated with urban pollution.

2 Methods

This study employs an integrated approach to model air pollution dispersion across complex urban landscapes, utilising AI-driven computational fluid dynamics (CFD) model and real-time data assimilation. The overall workflow, outlined in Fig. 1, begins with preprocessing historical environmental data, such as pollution concentrations and wind velocities. The urban domain is divided into smaller subdomains, which are augmented by overlapping and rotation to create a robust training dataset. These subdomains serve as inputs to a Convolutional Variational Autoencoder (VAE), which compresses this data into a latent space, capturing essential features, and then reconstructs it to generate initial conditions. These generated initial conditions are then used to create a full-scale initial condition by concatenating the subdomains. This condition is fed into the AI4PDEs solver, which simulates the dispersion of pollutants across the urban landscape. The framework iteratively refines its predictions through forward simulation, data assimilation, and backpropagation, resulting in accurate modeling of pollution dispersion in complex urban settings.

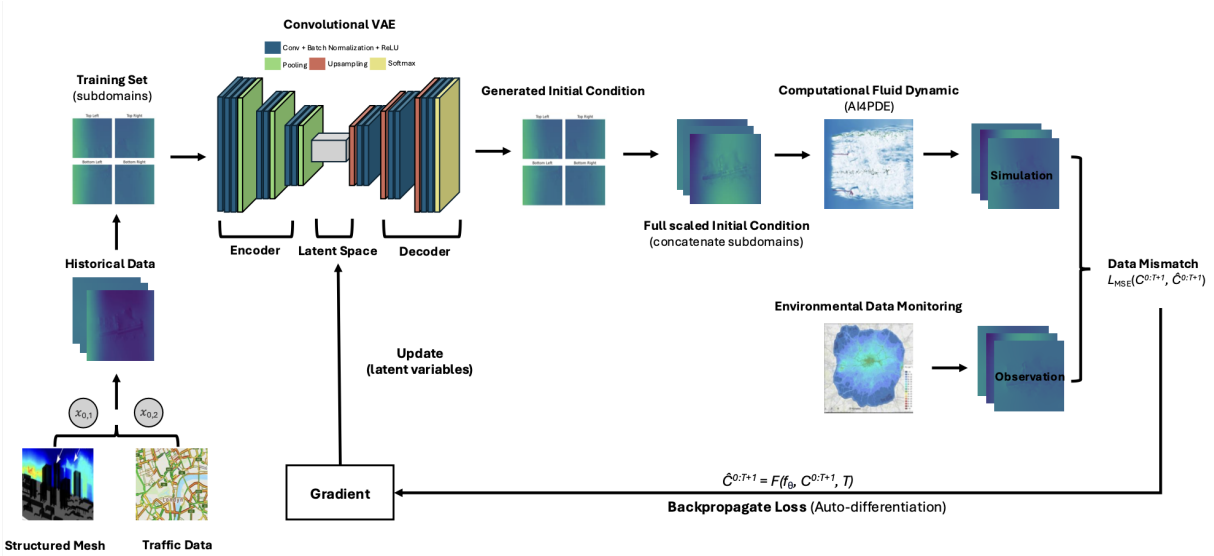

Figure 1: Overview of the workflow.

Algorithm 1 Integrated Approach for Air Pollution Dispersion Modeling

Input: Historical environmental data (pollution concentrations, wind velocities), traffic flow data, structured mesh (512x512 nodes)
Output: Accurate predictions of pollution dispersion in complex urban environments
Step 1: Preprocessing
Assemble traffic data and structured mesh to form the input $\boldsymbol{x}_0 = [\boldsymbol{x}_{0,1}, \boldsymbol{x}_{0,2}]$
Set the initial and boundary conditions to replicate the experimental conditions
Run a preliminary CFD simulation to generate pollution concentration fields $\boldsymbol{c}_d$ and velocity fields $\boldsymbol{v}_d = (v_{x,d}, v_{y,d}, v_{z,d})$ as historical dataset $\boldsymbol{d}$ = $\{\boldsymbol{c}_d, \boldsymbol{v}_d\}$
Normalize the data $\boldsymbol{d}$ to the range $[0, 1]$
Optionally divide the urban domain into smaller subdomains (4 in this study)
Apply augmentation techniques, including overlapping and rotations, to enhance the training dataset for the CVAE model
Step 2: Convolutional Variational Autoencoder (CVAE) Training
Feed the preprocessed CFD data $\boldsymbol{d}$ = $\{\boldsymbol{c}_d, \boldsymbol{v}_d\}$ into the CVAE model
Encode the data into a latent space $\boldsymbol{z}$:

$$\boldsymbol{z} = f_{\text{encoder}}(\boldsymbol{d})$$

Decode the latent space to reconstruct initial conditions:

$$\hat{\boldsymbol{d}} = f_{\text{decoder}}(\boldsymbol{z})$$

Step 3: Computational Fluid Dynamic (CFD) Forward Simulation
Set initial and boundary conditions based on reconstructed data from the CVAE model
Run the CFD simulation on 2 GPUs for $T = 30$ minutes, producing $N = 3{,}000$ timesteps
Extract pollution concentration $\boldsymbol{c}$ and velocity fields $\boldsymbol{v} = (v_x, v_y, v_z)$ as model output $\boldsymbol{O}_{sim}$ = $\{\boldsymbol{c}, \boldsymbol{v}\}$
Step 4: Data Assimilation (DA)
Compare simulated pollution concentrations $\boldsymbol{O}_{sim}$ with real-world observations $\boldsymbol{I}_{obs}$
Calculate the mismatch functional $\mathcal{M}$ using Mean Squared Error (MSE):

$$\mathcal{M} = \frac{1}{N} \sum_{i=1}^{N} \left(\boldsymbol{O}_{sim}^{i} - \boldsymbol{I}_{obs}^{i} \right)^2$$

Step 5: Iterative Refinement
Repeat the forward simulation, data assimilation, and backpropagation process
Continue iterations until convergence is achieved (10 cycles in this study)
Final Output: Accurate predictions of air pollution dispersion across the urban environment

2.1 Setup of a Real Test Case

The presented framework is applied to observational traffic and pollution data collected from Open Weather Ltd., located on Du Cane Road, Borough of Chelsea and Kensington, London, UK. The data is represented in a structured mesh format composed by 512x512 equally spaced nodes, accurately captures the detailed layout of buildings, infrastructure, as shown in Figure 2. This model assumes that pollution sources on the roads are constant and dependent on traffic flow for complexity reasons. Five sensors located in different positions, derived form coupling project (Optimising sensor location using neural networks applied to air pollution: https://github.com/ese-msc-2023/irp-xz3323), were used to record the data. During the whole period of the experiment, the predominant wind came from all three directions.

To replicate the field study experiment, a numerical simulation has been performed using the computational fluid dynamics (CFD) through an advanced in-house CFD code, AI4PDEs (https://github.com/bc1chen/AI4PDE/tree/main). The same simulation has been used in previous paper [6] and only the main details of the CFD setup are re-called and modified here. The initial and boundary conditions are set to replicate the experimental conditions. The simulation was run in parallel on 2 GPU and 40 minutes were simulated rendering 10,000 timesteps. In this research, the working variable of interest is the pollution concentration and velocity field in three directions. However, it is worth noting that after the timestep 2,500 the concentration reaches a dynamic steady state everywhere in the field.

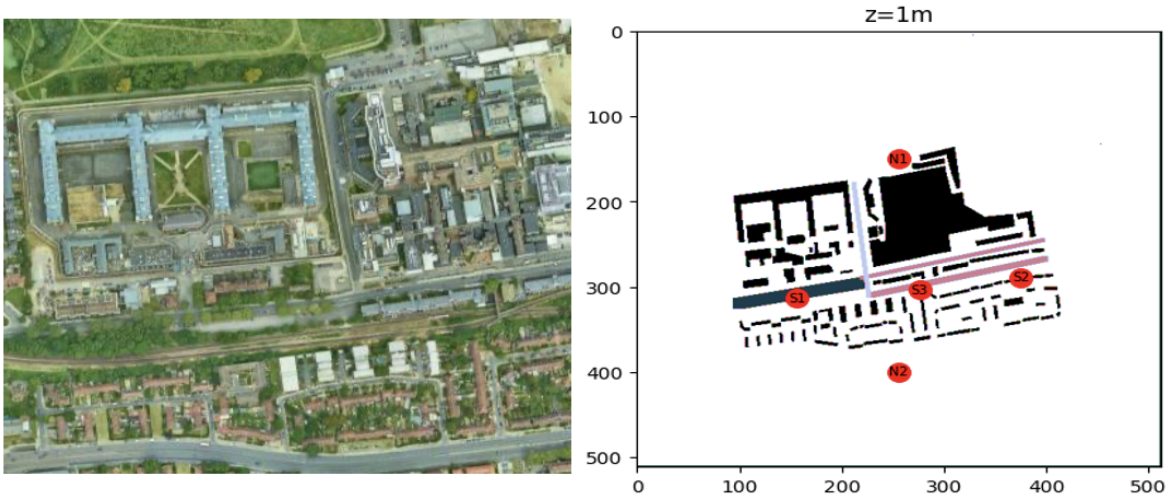

Figure 2: Test case located at Du Cane Road, Borough of Chealsea and Kensington, London, UK. Black rectangles and red dots denote the buildings and sensor locations, respectively. Constant pollution sources are showed with different colours depending on the values.

2.2 Data Preprocessing

The data generated by the preliminary CFD simulations are stored on a structured mesh in order to apply the model for the whole system, as illustrated in Fig. 3. This structured mesh is divided into four smaller and more manageable subdomains, each subjected to various augmentations to enhance the robustness of the training dataset. This process includes further breaking it down into smaller grids that overlaps the adjacent ones by 50%, preserving the sequence of data and continuity of information. Additionally, some grids are rotated to ensure that the model is exposed to a wider variety of scenarios, improving its ability to generalise across different conditions and handle unseen data effectively.

The observational pollution data for data assimilation sourced from Open Weather Ltd. is the

historical daily data of depth from 11/08/2024 to 17/08/2024 in South Kensington area, with a resolution of 500 meters. For the test case, only the nodes located within the field were selected to work with, thus excluding the rest of the domain. As a final step for both dataset, all the data are normalised between 0 and 1.

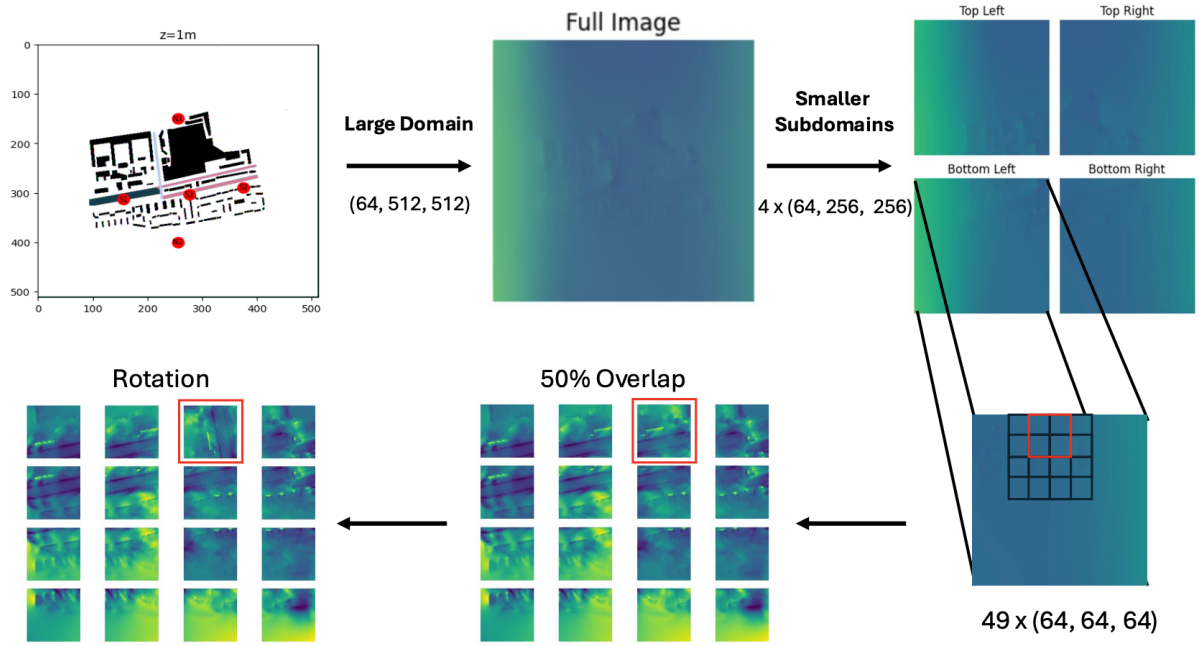

Figure 3: The steps involved in preparing priors for CVAE, including dividing the structured mesh into four subdomains, breaking them into overlapping grids, and applying rotations to enhance the dataset's robustness and improve the model's generalisation across various scenarios.

2.3 Convolutional Variational Autoencoder

After preprocessing, the historical data is used as priors and fed into the CVAE to handle the high dimensionality of the input data while capturing the essential features necessary for accurate predictions. In this study, dimensionality reduction is achieved via a deep learning approach using a Convolutional Variational Autoencoder (CVAE). This choice leverages the advantages of convolutional operations, which takes into account the spatial information and is specially well-suited when working with images or grid data.

The CVAE model comprises two main components: an encoder and a decoder, as illustrated in Fig. 4. The encoder compresses the input data through a series of convolutional layers, which apply filters to detect spatial hierarchies in the data. Each convolutional layer is followed by batch normalization and Rectified Linear Unit (ReLU) activations, which help to stabilize and speed up the training process by normalizing the inputs to each layer and ensuring non-linear transformation of the data. Pooling layers are strategically placed to reduce the spatial dimensions of the data, which not only lessens the computational load but also helps in focusing the model on the most critical features. The essential features of the data are then captured in a latent space, a lower-dimensional representation that preserves the most critical information. The decoder then reconstructs these features from the latent space, generating realistic initial conditions for the subsequent CFD simulations.

The training of the CVAE is conducted in an unsupervised manner, where the network learns to encode and decode the input data without needing labeled outputs. The optimisation process is

handled using the Adam optimiser, configured with a learning rate of 1e-3 and beta values (0.9, 0.999) to ensure convergence. The loss function guiding the learning process is a combination of the reconstruction error, measured by Mean Squared Error (MSE), and a Kullback-Leibler (KL) divergence term that enforces the desired distribution in the latent space. This carefully selected learning rate, combined with the MSE loss criterion, ensures that the model trains efficiently over 1000 epochs without converging too quickly to a suboptimal solution. The choice of these hyperparameters reflects a balance between convergence speed and the accuracy of the final model.

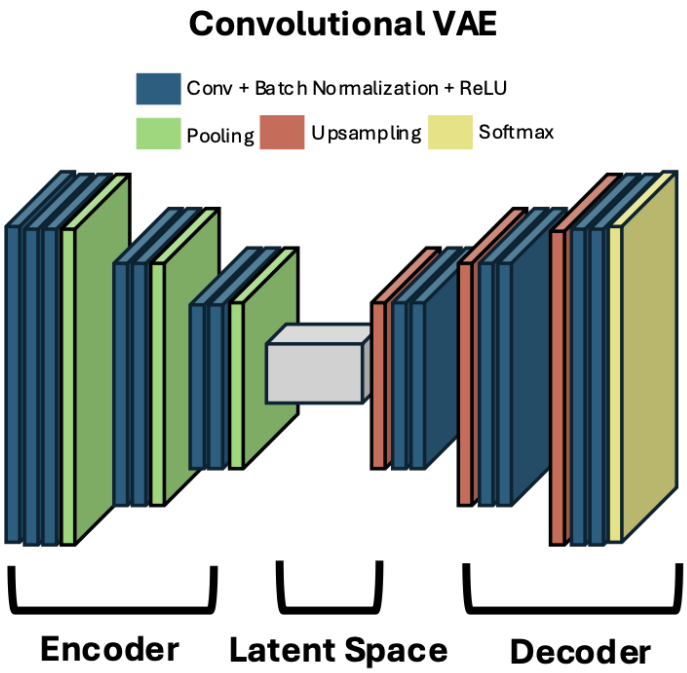

Figure 4: Architecture of the convolution Variational Autoencoder, highlighting its encoder and decoder and various layers. The encoder compresses high-dimensional input data into a lower dimensional latent space, while the decoder reconstructs the data to generate realistic initial conditions for CFD simulations.

2.4 Computational Fluid Dynamic Forward Simulation

The final step before simulation involves generating the initial conditions for the full urban domain. The generated subdomain initial conditions from the CVAE are concatenated to form a comprehensive initial condition that represents the entire urban area. This full-scale initial condition is then used as input for the AI4PDEs solver.

The code is written in Python and is available at the following link: https://github.com/bc1chen/AI4PDE/tree/mai At the core of this workflow is the fusion of a Convolutional Variational Autoencoder (VAE) with AI4PDEs, a specialized neural network-based CFD solver that uses pre-configured weights derived from different discretisation methods, eliminating the need for extensive training. As we can deduce from the name, this autoencoder uses the convolutional operation. This solver operates on structured mesh grids, as demonstrated in Fig. 5, representing urban environments with detailed building and infrastructure layouts. Pollution sources, such as vehicle emissions, are defined within these grids, serving as the basis for simulating pollutant dispersion under various conditions. Using the AI4PDEs code, we will conduct detailed simulations of pollution movement within the urban environments, providing insights into the concentration under various conditions.

The same simulation has been used in previous paper [6] and only the main details of the CFD setup are re-called and modified here. The initial and boundary conditions are set to replicate the experimental conditions. The simulation was executed in parallel across 2 GPUs, covering a 30-minute period and producing 3,000 timesteps. As determined in prior analyses in section 2.1, the concentration field achieves a dynamic steady state after timestep 2,500.

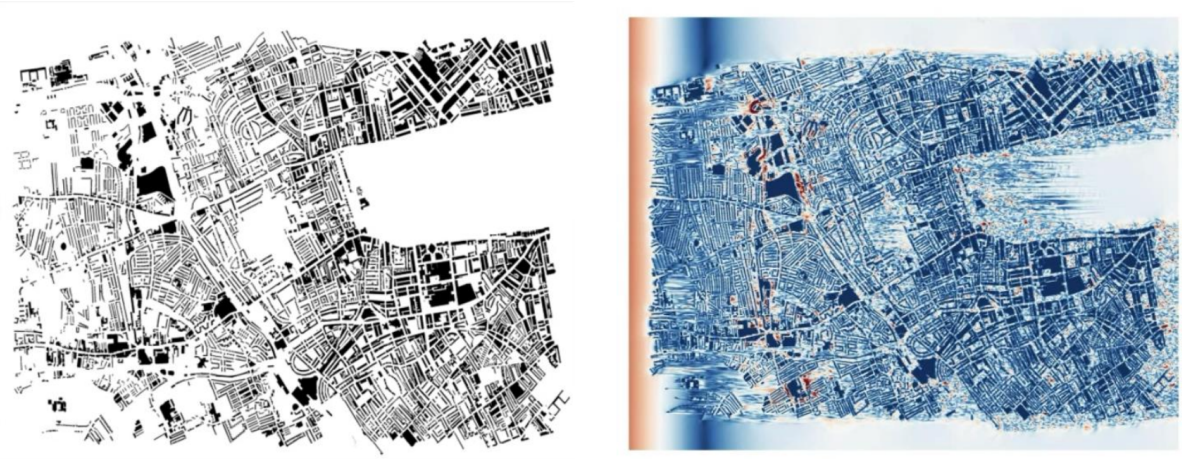

Figure 5: Left: Structured mesh of the area of interest (South Kensington area, London, UK). Right: Effect after computational fluid dynamics simulation.

2.5 Data Assimilation

To enhance the accuracy of these predictions, the workflow integrates a data assimilation (DA) process, as illustrated in Fig. 6. This process involves comparing the simulated pollution concentrations with real-world observations, which in this study are sourced from Open Weather Ltd. Any discrepancies or mismatches between the simulations and actual data are identified, prompting an update to the parameters of the CVAE.

The DA process is essential for maintaining the alignment of the model's predictions with actual environmental conditions. This alignment is achieved through backpropagation, a neural network technique where the computed loss, the Mean Squared Error (MSE) in this model, is propagated back through the network. The loss calculation takes into account both the reconstruction error in the CVAE and the deviations between the simulated and observed data. During backpropagation, the weights and biases in the decoder of the CVAE are updated, refining the latent space representation and improving the accuracy of generated initial conditions for future simulations.

This iterative cycle of forward simulation, data assimilation, and backpropagation continues until the model converges to a satisfactory level of accuracy. The continuous updating of the parameters in response to observational data allows the model to remain adaptable, ensuring that it can provide reliable and accurate predictions of air pollution dispersion in complex urban environments. For this study, a total of 10 iterative cycles were conducted to attain robust results, as detailed in the following section.

2.6 Environment

The simulation framework for this study is developed and executed on a Linux-based platform, ensuring compatibility with a wide range of open-source tools and libraries. The core of the model is built using Python, a versatile and widely-used programming language in scientific

Algorithm 2 Sensitivity Analysis and Data Assimilation

Input: Trained CVAE model, AI4Urban model, number of iterations N_{sens}, latent representations $\boldsymbol{Z}$, observational data $\boldsymbol{I}_{obs}$, original data $\boldsymbol{I}_{org}$, additional parameters
Output: Updated CVAE model and reconstruction progress
Step 1: Initialise Decoder
Set the decoder parameters of the CVAE model to be trainable.
Step 2: Sensitivity Analysis Loop
for each iteration $n = 1$ to N_{sens} (5 in this study) **do**
    a) Reset Gradients
    Reset gradients for the optimizer.
    b) Reconstruct Initial Conditions with Decoder
    Decode the latent representations $\boldsymbol{Z}$ to obtain reconstructed data $\hat{\boldsymbol{d}}$:

$$\hat{\boldsymbol{d}} = f_{\text{decoder}}(\boldsymbol{z})$$

    Form the full-sized reconstructed data as initial conditions$\hat{\boldsymbol{d}}_{\text{full}}$ from $\hat{\boldsymbol{d}}$ for CFD simulations.
    d) Computational Fluid Dynamic (CFD) Forward Simulation
    for time step $t = 1$ to N_{time} (3000 timesteps in this study) **do**
        Execute the CFD simulation using the AI4Urban model:

$$\boldsymbol{v}_{x,t+1}, \boldsymbol{v}_{y,t+1}, \boldsymbol{v}_{z,t+1}, \boldsymbol{p}_{t+1} = \text{AI4Urban}(\boldsymbol{v}_{x,t}, \boldsymbol{v}_{y,t}, \boldsymbol{v}_{z,t}, \boldsymbol{c}_t, ...)$$

        where $\boldsymbol{v}_{x,t}$, $\boldsymbol{v}_{y,t}$, $\boldsymbol{v}_{z,t}$, $\boldsymbol{c}_t$ are the state variables at time step t.
    end for
    d) Compare Sensor Error
    For each sensor s, compute relative error between model output $\boldsymbol{O}_{sim}$ and observational data $\boldsymbol{I}_{obs}$:

$$e_r = \frac{\boldsymbol{O}_{sim} - \boldsymbol{I}_{obs}}{\boldsymbol{I}_{obs}}$$

    where $\boldsymbol{O}_{sim}$ is the pollution concentration fields $\boldsymbol{c}_t$ and the velocity field $\boldsymbol{v}$ = ($\boldsymbol{v}_{x,t}$, $\boldsymbol{v}_{y,t}$, $\boldsymbol{v}_{z,t}$)
    e) Compute Loss
    Compute the Mean Squared Error (MSE) $\mathcal{M}$ loss between model output $\boldsymbol{O}_{sim}$ and observational data $\boldsymbol{I}_{obs}$:

$$\mathcal{M} = \frac{1}{N} \sum_{i=1}^{N} \left(\boldsymbol{O}_{sim}^{i} - \boldsymbol{I}_{obs}^{i}\right)^2$$

    f) Backpropagation and Model Update
    Perform backpropagation to compute gradients.
    Clip gradients to prevent explosion.
    Update model parameters using the optimizer.
    g) Store Progress and Result
    Record the current reconstruction progress, save the updated model CVAE* and data.
end for
Step 3: Return Results
Updated model state CVAE* and result for future iterations.

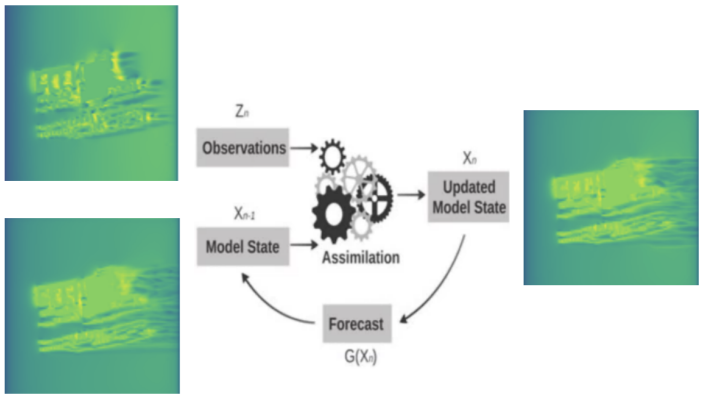

Figure 6: Illustration of the data assimilation process used to enhance the accuracy of pollution predictions. The observational data is assimilated with the simulation, highlighting how data mismatch are used to update the parameters of the Conditional Variational Autoencoder (CVAE).

computing. The neural network components, particularly the Convolutional Variational Autoencoder (CVAE) and other deep learning models, are implemented using TensorFlow and PyTorch, two leading libraries known for their robustness and flexibility in handling complex neural network architectures.

2.6.1 Computational Resources

To handle the intensive computational requirements of the model, especially given the large-scale simulations involved in urban air pollution modeling, the framework leverages the High-Performance Computing (HPC) facilities available at Imperial College. The GPU resources provided by this HPC infrastructure are critical for accelerating the training of deep learning models and the execution of forward simulations within the AI4PDEs framework. Specifically, the parallel processing capabilities of GPUs significantly reduce the time required for each Data Assimilation (DA) cycle, enabling the completion of multiple iterations within a feasible timeframe.

Additionally, the simulations require substantial memory resources, with RAM usage exceeding 900GB during peak operations. This high memory demand is crucial for handling the large datasets and complex computations involved in modeling urban airflows and pollutant dispersion

2.6.2 Solver Configuration

The AI4PDEs solver (https://github.com/bc1chen/AI4PDE/tree/main), which integrates machine learning with traditional Computational Fluid Dynamics (CFD), is configured with solver tolerances carefully tuned to match those used in conventional CFD models. This ensures that the results produced by the AI-driven approach are consistent and comparable with those obtained from standard CFD methods. The consistency of solver tolerances also plays a crucial role in maintaining the stability of simulations over extended periods, which is particularly important when dealing with the chaotic nature of urban airflow and pollutant dispersion. The same simulation has been used in previous paper [6] and only the main details of the CFD setup are re-called and modified in this study.

3 Results

This study investigates the effectiveness of integrating data assimilation (DA) into an AI-driven computational fluid dynamics (CFD) model to enhance the accuracy of urban air pollution dispersion predictions. The methods employed, including the use of a Convolutional Variational Autoencoder (CVAE) for dimensionality reduction and an iterative DA process, enable the model to refine its predictions continuously by aligning simulation results with real-world observations. The results are evaluated by comparing velocity fields, pollution concentration fields, and data mismatch reduction across multiple sensors and additional analysis points.

3.1 Data Assimilation: Velocity Field and Pollution Concentration Field

The iterative refinement process, depicted in Figure 7, highlights significant improvements in the model's predictions after 15 DA cycles, each involving 3000 time steps of forward CFD simulation. Initially, noticeable discrepancies exist between the simulated velocity fields and pollution concentration fields when compared to the observational data. These discrepancies are expected in complex urban environments due to the influence of factors such as varying building geometries, street canyons, and traffic conditions, all of which contribute to the complexity of airflow and pollutant dispersion.However, the updated fields after DA closely match the observational data, demonstrating the model's ability to correct initial inaccuracies and improve predictive accuracy.

The figure shows multiple views comparing the simulation, observational data, and updated predictions for velocity fields in the x, y, and z directions, as well as the pollution concentration field. The top rows in each panel depict the raw simulation data before DA, where we observe significant deviations from the real-world observations, especially in regions of complex flow interactions or pollutant accumulation. Following 15 iterations of DA, the updated fields (shown in the bottom rows of each panel) exhibit a marked improvement in accuracy. The visual comparison between the simulated fields and the observational data reveals that the updated predictions closely align with the observed patterns. This convergence towards observational data underscores the model's ability to adjust and refine its predictions, effectively reducing the initial inaccuracies. As the DA process progresses through each iteration, the model systematically reduces the discrepancies between the simulated and observed fields, effectively capturing the essential dynamics of urban airflows and pollutant behaviors with increasing precision.

Overall, the convergence of the velocity fields and pollution concentration field towards the observed data, as seen in the updated predictions, highlights the robustness of the DA framework. It ensures that the model not only learns from its initial errors but also continually improves, delivering more reliable and accurate predictions for urban air quality management.

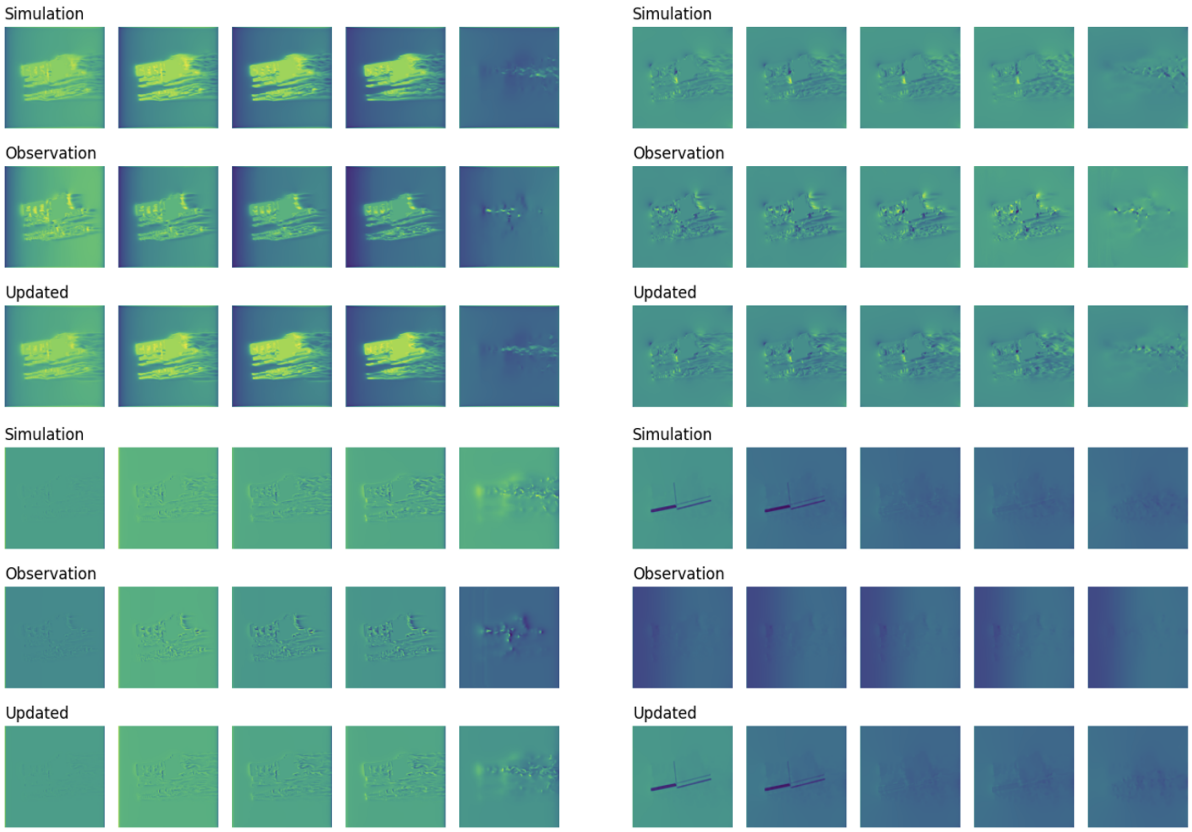

Figure 7: Result after 15 loops of data assimilation, with each iteration involving 3000 time steps of CFD simulation. Top Left: Velocity field in the x-direction. Top Right: Velocity field in the y-direction. Bottom Left: Velocity field in the z-direction. Bottom Right: Pollution concentration field.

3.2 Detailed Analysis: Velocity Field in x-direction at 1-Meter Height

Figure 8 provides a more granular view of the data assimilation results, focusing specifically on the velocity field in the x-direction at a height of 1 meter above ground level. This detailed visualization compares the simulated, observed, and updated velocity fields after 15 DA cycles, offering insights into the spatial accuracy of the model's predictions.

In the initial simulation (left panel), significant discrepancies are evident when compared to the observational data (middle panel), particularly in regions with complex building interactions that disrupt airflow. These discrepancies highlight the challenges of accurately modeling urban airflows without real-world data integration.

The updated velocity field (right panel) demonstrates a remarkable alignment with the observational data. The data assimilation process effectively refines the model's predictions, reducing the errors observed in the initial simulation. The refined model captures the nuanced airflow patterns caused by building layouts and street canyons, resulting in a more accurate representation of the urban environment.

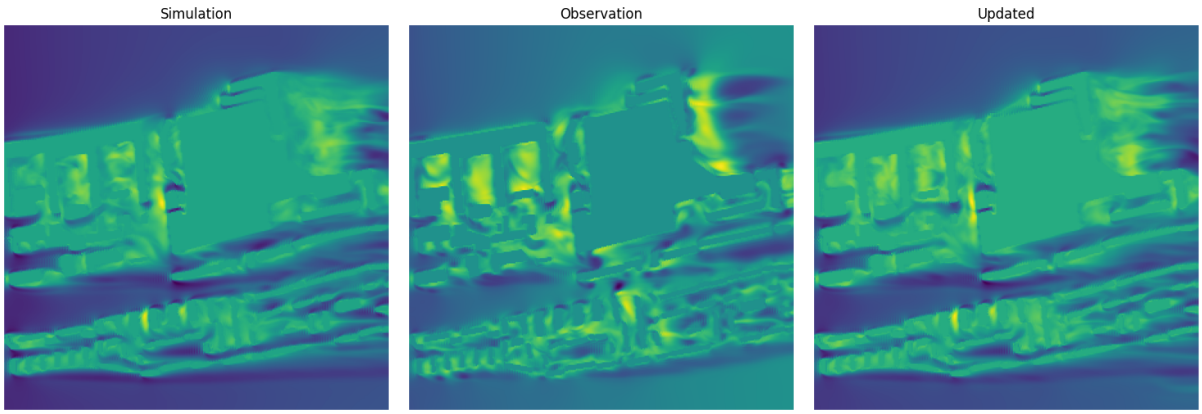

Figure 8: Data assimilation results (velocity field in the x-direction at a 1-meter height level) in more detail in terms of the spatial resolution.

3.3 Data Mismatch and Sensor Performance

Figure 9 presents the reduction in relative error across five sensors and additional analysis points across multiple channels (x, y, z directions and pollution) as the DA process progresses. Initially, the data mismatch is approximately 0.035, but it decreases substantially by 8th iteration, approaching towards a convergence. This trend demonstrates the model's ability to minimise discrepancies between simulated and observed data effectively.

Initially, the data mismatch is evident, with relative errors starting at around 70% to 50% for some channels and sensors. However, as the DA iterations proceed, a significant reduction in error is observed. By the 10th iteration, the relative errors across all sensors and nodes show marked improvements and the trend of convergence is evident, demonstrating the model's effectiveness in aligning the simulated data with real-world observations.

- Sensor 1 (coordinates: 156, 313): This sensor shows a steady decrease in relative error across all channels, with channel 1 starting at approximately 70% and dropping to just under 20% after 14 DA iterations. Channel 2 and the pollution channel also follow similar patterns, though with slightly higher residual errors.
- Sensor 2 (coordinates: 276, 304): Notably, Sensor 2 exhibits the most significant improvement, particularly in channel 2, where the relative error decreases from approximately 60% to below 10% after 14 iterations, reflecting an error reduction of nearly 85%. Channel 3 and the pollution channel also show considerable error reductions, reinforcing the effectiveness of DA at this location.
- Sensor 3 (coordinates: 375, 289): While Sensor 3 initially exhibits higher errors across all channels, it shows a consistent decline in relative error as the DA iterations progress. By the 14th iteration, the error in channel 1 reduces from around 70% to just above 20%, indicating a significant improvement in predictive accuracy.
- Sensor 4 (coordinates: 255, 150): At this point, the relative error decreases steadily across all channels. The most substantial reduction is observed in the pollution channel, where the error drops from around 60% to 15% by the 14th iteration, a reduction of approximately 75%.
- Sensor 5 (coordinates: 255, 400): Similar to the ones above, sensor 5 shows a significant decrease in relative error, particularly in channel 2, where the error falls from 60% to below 20%, representing a reduction of about 67%.

Overall, the significant reduction in relative error across all sensors and nodes confirms the effectiveness of the DA process in improving the predictive accuracy of the AI-driven CFD model. This enhanced accuracy is crucial for reliable urban air pollution dispersion predictions, making the model a valuable tool for environmental management in urban settings.

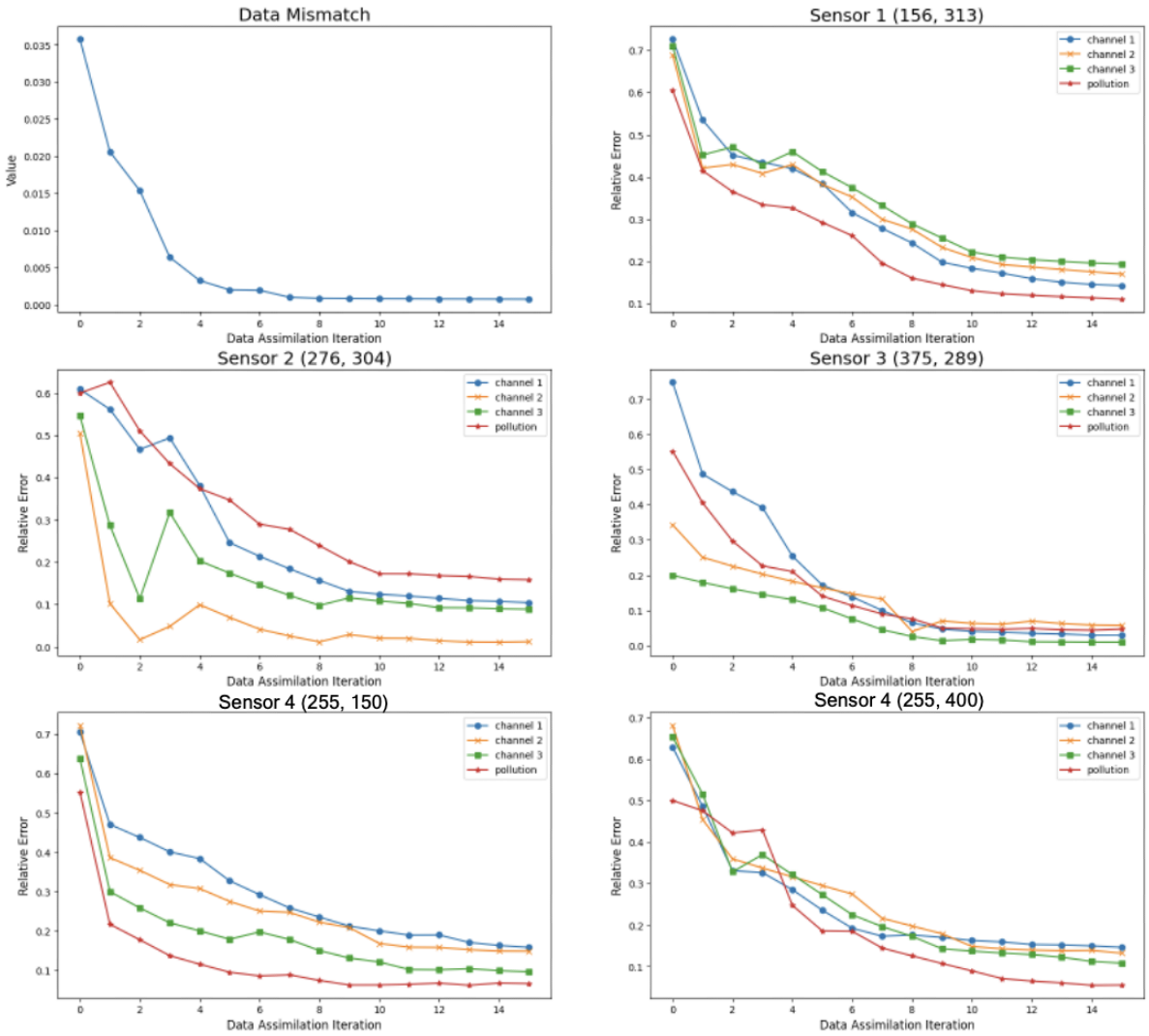

Figure 9: Data Mismatch and relative error over iterations showing the improvement of data assimilation. Channels 1, 2, 3 correspond to wind velocity in x, y, z-directions, respectively

4 Discussion and Conclusions

This study demonstrates the substantial impact of integrating data assimilation (DA) into an AI-driven computational fluid dynamics (CFD) model for predicting urban air pollution dispersion. The iterative process of forward simulation, DA, and backpropagation improved the model's predictive accuracy, with a notable reduction in relative errors across various sensors and channels. Specifically, the data mismatch, initially around 0.035, decreased significantly across the 15 DA cycles, with relative errors dropping by as much as 85% for certain sensors. This quantifiable improvement underscores the model's effectiveness in real-time air quality monitoring and decision-making, particularly in complex urban environments where traditional models struggle.

The integration of DA with the Convolutional Variational Autoencoder (CVAE) for dimensionality reduction proved to be a robust method for enhancing prediction accuracy. By the 10th DA iteration, some sensors showed a reduction in relative error from 70% to under 20%, demonstrating the model's ability to correct initial inaccuracies and align its predictions closely with observed data. For instance, Sensor 2 exhibited the most significant improvement, with the error in channel 2 reducing from approximately 60% to below 10%. These quantifiable results highlight the model's capacity to adapt and refine its predictions over multiple iterations, effectively capturing the dynamic nature of urban airflows and pollutant dispersion.

However, a significant challenge encountered during the implementation was the intensive computational resources required, particularly in terms of RAM. This issue was most pronounced during the CFD simulation, which involved thousands of time steps, and the subsequent DA process. The need to calculate gradients during backpropagation, coupled with the complexity of the neural network, placed a heavy demand on memory, leading to RAM requirements exceeding 900GB. To manage this, we employed a "moving window" strategy, where data and the updated model were stored after each iteration and reloaded for subsequent loops.

This approach allowed us to mitigate the RAM limitations by reducing the amount of data held in memory at any given time. However, this also introduced additional complexity, as it required manual intervention to store and retrieve data during the simulation process. While this solution successfully mitigated the RAM limitation, it also highlighted a trade-off between computational efficiency and manual workload. The process of manually managing data storage and retrieval, though necessary, was cumbersome and could benefit from further automation in future work.

Looking ahead, one of the most exciting future directions is to scale this model to even larger geographical areas, such as entire cities or countries. Applying the integrated DA and AI-driven CFD framework on a larger scale could provide more comprehensive insights into air pollution patterns, offering significant benefits for large-scale urban planning, environmental policy-making, and public health initiatives. However, scaling up will inevitably exacerbate the challenges related to computational resources, necessitating advancements in both algorithmic efficiency and hardware capabilities.

In conclusion, this study illustrates the powerful potential of integrating data assimilation into AI-driven CFD models to enhance urban air quality predictions. The iterative refinement process ensures that the model remains adaptable and capable of accurately simulating complex pollution dispersion scenarios in real-time. However, the substantial computational demands, particularly in terms of RAM usage during gradient calculations and neural network operations, underscore the need for optimizing resource usage and automating data management processes. Future work should focus on overcoming these challenges and extending the model's applicability to larger scales, thereby maximizing its utility for urban planners, environmental agencies, and public health officials.

References

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

[2] Rossella Arcucci, César Quilodrán Casas, Dunhui Xiao, Laetitia Mottet, Fangxin Fang, Pin Wu, Christopher Pain, and Yi-Ke Guo. A domain decomposition reduced order model with data assimilation (dd-roda). *Parallel Computing: Technology Trends*, 36, 2020.

[3] Rossella Arcucci, Laetitia Mottet, Christopher Pain, and Yi-Ke Guo. Optimal reduced space for variational data assimilation. *Journal of Computational Physics*, 379(2), 2019.

[4] J. Chan, Z. Wang, A. Modave, J.-F. Remacle, and T. Warburton. GPU-Accelerated Discontinuous Galerkin Methods on Hybrid Meshes. *Journal of Computational Physics*, 318:142–168, 2016.

[5] Chen, Heaney, Gomes, Matar, and Pain. Solving the Discretised Multiphase Flow Equations with Interface Capturing on Structured Grids Using Machine Learning Libraries. *arXiv preprint arXiv:2401.06755*, 2024.

[6] Boyang Chen, Claire E. Heaney, and Christopher C. Pain. Using ai libraries for incompressible computational fluid dynamics. *arXiv preprint arXiv:2402.17913*, 2024.

[7] D. N. Daescu. On the Sensitivity Equations of Four-Dimensional Variational (4D-Var) Data Assimilation. *Monthly Weather Review*, 136(8):3050–3065, 2008.

[8] D. N. Daescu and G. R. Carmichael. An Adjoint Sensitivity Method for the Adaptive Location of the Observations in Air Quality Modeling. *Journal of the Atmospheric Sciences*, 60(2):434–450, 2003.

[9] G. D'Amato, G. Liccardi, M. D'Amato, and M. Cazzola. The role of outdoor air pollution and climatic changes on the rising trends in respiratory allergy. *Respiratory Medicine*, 95(7):606–611, July 2001. Available online 25 May 2002.

[10] Fang, Pain, Navon, and Xiao. An efficient goal-based reduced order model approach for targeted adaptive observations. *International Journal for Numerical Methods in Fluids*, 83:263–275, 2017.

[11] Annunziata Faustini, Massimo Stafoggia, Paola Colais, Giovanna Berti, Luigi Bisanti, Ennio Cadum, Achille Cernigliaro, Sandra Mallone, Corrado Scarnato, Francesco Forastiere, et al. Air pollution and multiple acute respiratory outcomes. *European Respiratory Journal*, 42:304–313, 2013.

[12] P. Gousseau, B. Blocken, T. Stathopoulos, and G.J.F. van Heijst. Cfd simulation of near-field pollutant dispersion on a high-resolution grid: A case study by les and rans for a building group in downtown montreal. *Atmospheric Environment*, 45(2), 2011.

[13] Geer A. J. Learning earth system models from observations: machine learning or data assimilation? *Philosophical Transactions of the Royal Society A*, 379, 2021.

[14] A. C. Lorenc, R. S. Bell, and B. Macpherson. The Meteorological Office analysis correction data assimilation scheme. *Quarterly Journal of the Royal Meteorological Society*, 117(497):59–89, Jan 1991.

[15] Julian Mack, Rossella Arcucci, Miguel Molina-Solana, and Yi-Ke Guo. Attention-based convolutional autoencoders for 3d-variational data assimilation. *Computer Methods in Applied Mechanics and Engineering*, 372(2), 2020.

[16] Toby R. F. Phillips, Claire E. Heaney, Boyang Chen, Andrew G. Buchan, and Christopher C. Pain. Solving the discretised neutron diffusion equations using neural networks. *International Journal for Numerical Methods in Engineering*, 2023.

[17] Toby R. F. Phillips, Claire E. Heaney, Boyang Chen, and Christopher C. Pain. Solving the discretised boltzmann transport equations using neural networks: Applications in neutron transport. *arXiv:2301.09991*, 2023.

[18] A. Sandu, D. N. Daescu, and G. R. Carmichael. Direct and Adjoint Sensitivity Analysis of Chemical Kinetic Systems with KPP: Part II — Numerical Validation and Applications. *Atmospheric Environment*, 37(36):5097–5114, 2003.

[19] Joel Schwartz. Air Pollution and Hospital Admissions for Respiratory Disease. *Epidemiology*, 7(1):20–28, January 1996.

[20] Maike Sonnewald, Redouane Lguensat, Daniel C. Jones, Peter D. Dueben, Julien Brajard, and V. Balaji. Bridging observations, theory and numerical simulation of the ocean using machine learning. *Environmental Research Letters*, 16(7), 2021.

[21] Nils Thuerey, Konstantin Weißenow, Lukas Prantl, and Xiangyu Hu. Deep learning methods for reynolds-averaged navier–stokes simulations of airfoil flows. *AIAA Journal*, 68(1), 2019.