Proceedings of the 2025 Winter Simulation Conference
E. Azar, A. Djanatliev, A. Harper, C. Kogler, V. Ramamohan, A. Anagnostou, and S. J. E. Taylor, eds

RELIABILITY ASSESSMENT OF CONVOLUTIONAL AUTOENCODER-BASED WIND MODELING FOR AUTONOMOUS DRONE TRAINING

Jiahao Wu[1], Hengxu You[1], and Jing Du[1]

[1]Dept. of Civil and Coastal Eng., University of Florida, Gainesville, USA

ABSTRACT

Computational Fluid Dynamics (CFD) is a widely used method for wind modeling in autonomous drone training, yet its computational expense limits real-time applications. This study explores the reliability of a convolutional autoencoder-based approach as a replacement for CFD in wind field generation. A convolutional autoencoder (CAE)is trained on CFD-generated wind data to learn its spatial and dynamic distribution patterns, enabling the generation of realistic wind fields with significantly lower computational costs. The generated wind fields are then used to train reinforcement learning (RL)-based drones, with their policies evaluated in real CFD environments. Results demonstrate that the CAE accurately replicates CFD wind patterns, supports stable drone navigation, and facilitates seamless transferability of RL-trained policies from autoencoder-generated wind environments to CFD-based environments. These findings highlight the convolutional autoencoder's potential as a computationally efficient and reliable alternative to traditional CFD in autonomous drone simulations.

1 INTRODUCTION

Autonomous drones have gained widespread adoption across various applications, including search and rescue, environmental monitoring, infrastructure inspection, and logistics (Capolupo et al. 2015; Jacobsen et al. 2023; Mayer et al. 2019). Their ability to operate in complex and dynamic environments makes them particularly useful for urban navigation, where obstacles such as tall buildings and unpredictable wind conditions pose significant challenges. One critical aspect of drone operation in urban settings is their ability to adapt to aerodynamic disturbances, including wind currents that vary in intensity and direction due to interactions with the built environment (Blocken et al. 2008). This problem is particularly pronounced in dense urban areas, where local wind variations can lead to unexpected turbulence and instability, affecting flight safety and control accuracy (Biao et al. 2019).

To address these challenges, RL-based approaches have been increasingly explored for training autonomous drones. RL enables drones to learn complex navigation strategies by interacting with simulated environments and optimizing their control policies based on predefined objectives such as stability, energy efficiency, and obstacle avoidance (Kaelbling et al. 1996). However, a key limitation in training RL-based drones in a simulated windy environment is accurately modeling the aerodynamic forces they will encounter in real-world conditions. Acquiring actual wind field data is often challenging due to its limited availability and the complexity of real-world measurements. Consequently, traditional approaches rely on CFD to simulate wind environments, offering high-fidelity aerodynamic data but at a significant computational cost (Calzolari and Liu 2021). The high computational cost of CFD-based training makes it impractical for large-scale simulations or real-time adaptation, limiting the scalability and efficiency of RL-based drone training.

To overcome this issue, data-driven approaches have emerged as a viable alternative to CFD. Deep learning models, particularly convolutional autoencoders, have demonstrated strong capabilities in learning complex fluid dynamics and compressing high-dimensional aerodynamic data into a latent

979-8-3315-8726-0/25/$31.00 ©2025 IEEE

representation (Ribeiro et al. 2020). Autoencoders are a class of neural networks designed to encode input data into a compressed feature space and subsequently reconstruct the original data from this representation, preserving essential spatial and temporal structures (Ng 2011). In wind modeling, convolutional autoencoders can learn the underlying patterns of wind distribution and generate synthetic wind fields with significantly lower computational requirements than traditional CFD solvers (Milla-Val et al. 2023). This approach enables faster wind field inference while maintaining sufficient accuracy for drone navigation tasks.

A major concern in replacing CFD with autoencoder-generated wind data is the reliability and generalization of the learned representations. If a CAE can accurately reconstruct wind distributions and generate realistic wind fields, it should also enable RL-trained drones to generalize their learned policies to real-world wind conditions. Thus, an essential aspect of evaluating autoencoder-based wind modeling is assessing the transferability of RL policies trained in autoencoder-generated environments to real CFD-generated environments. Ensuring that policies remain robust across different wind models is critical for validating the effectiveness of deep learning-based wind simulation as a reliable alternative to CFD.

In this study, we introduced an innovative approach to assess the feasibility of autoencoder-based wind modeling as a reliable substitute for CFD in drone training. The research objectives focus on three key aspects: (1) Validate whether the CAE can generate wind distributions that closely resemble CFD-derived wind fields using quantitative metrics, (2) Investigate how reinforcement learning agents trained in autoencoder-generated wind environments perform in comparison to those trained with CFD-based wind fields and (3) Examine whether RL policies trained on autoencoder-generated wind fields can be successfully deployed in CFD-based environments without significant performance degradation. To achieve these goals, we trained a CAE on CFD wind field data and leveraged its outputs to create a simulated training environment for RL-based drone navigation. Once trained, the drone policies are tested in a real CFD-generated wind environment to determine the viability of the autoencoder-based model as an alternative to CFD. The evaluation considers three core aspects: computational efficiency, fidelity of the wind field reconstruction, and drone performance in navigation tasks.

2 RELATED WORK

Autonomous drones are increasingly deployed in environments where wind disturbances significantly affect their stability and navigational performance. The ability to simulate wind conditions accurately is crucial for training reinforcement learning-based drone control systems, enabling them to adapt to real-world aerodynamic effects. Various approaches have been explored to model wind interactions with drones, ranging from high-fidelity CFD simulations to data-driven learning-based methods that aim to reduce computational costs while maintaining accuracy.

Traditional CFD-based simulations have been widely used to analyze and replicate wind patterns influencing drone performance. Paz et al. (2021) applied CFD techniques to model the airflow generated by quadcopter propellers, evaluating how aerodynamic interactions impact flight stability when operating near obstacles. Similarly, Qu et al. (2023) utilized CFD-generated local wind data to improve reinforcement learning-based multi-drone coordination strategies for disaster response. Gianfelice et al. (2022) used CFD to analyze the effects of urban building geometry on wind fields in downtown Toronto. They integrated CFD data with historical meteorological records to generate real-time, historical, forecast, and statistical wind visualizations for enhancing drone flight safety and efficiency. Besides, Large Eddy Simulation (LES) is another CFD technique used to model turbulent airflow by directly resolving large-scale turbulent structures while approximating smaller-scale turbulence through subgrid-scale modeling. LES provides a more accurate representation of unsteady, three-dimensional wind patterns than traditional Reynolds-Averaged Navier-Stokes (RANS) models, making it particularly useful for simulating complex aerodynamic interactions in urban environments. Giersch et al. (2022) developed a LES model to capture turbulent airflow in urban settings, demonstrating its effectiveness in predicting drone behavior under complex wind conditions. Additionally, Jeong et al. (2021) integrated the Weather

Research and Forecasting (WRF) model with LES to simulate wind patterns influenced by urban terrain, providing high-resolution wind fields for drone training. Despite their accuracy, CFD and LES methods impose a significant computational burden, limiting their applicability in reinforcement learning tasks that require extensive simulation data. The increasing demand for efficient and scalable wind data simulation has led to adopting deep learning techniques, which offer the potential to generate synthetic aerodynamic data with significantly reduced computational costs. This shift has prompted research into machine learning models capable of learning and generating realistic wind distributions for drone training environments.

Among these techniques, autoencoders have emerged as a highly effective tool for compressing and reconstructing high-dimensional aerodynamic data. Autoencoders learn compact representations of wind fields and generate reconstructions that closely resemble high-fidelity CFD outputs while requiring a fraction of the computational cost. Milla-Val et al. (2023) demonstrated that supervised learning techniques could produce wind predictions similar to CFD-based results, balancing accuracy and computational efficiency. Ma et al. (2021) further explored the potential of physics-driven convolutional neural networks (CNNs) to predict complex, high-resolution flow fields, achieving first-order accuracy while significantly reducing the computational cost of training multiple CFD cases. Zhang et al. (2021) applied a CAE to generate 3D realizations of time-averaged velocity in wind turbine wakes at the SWiFT facility, training the model on LES data. The trained model demonstrated strong predictive capabilities for unseen flow conditions while being significantly more computationally efficient than LES, making it a promising tool for wind turbine flow field and power production analysis.

This study builds upon these advancements by training a CAE on high-resolution CFD wind fields and then using the CAE to generate wind data for RL-based drone training. We evaluate the reliability of autoencoder-generated wind fields as a direct replacement for CFD by comparing the performance of RL policies trained on CAE outputs to those trained in traditional CFD environments, assessing transferability, computational efficiency, and robustness. Section 3.1 describes our CFD data generation procedure, Section 3.2 details the CAE architecture, and Section 3.3 validates the CAE's reconstructions against CFD using MAE and SSIM metrics. In Section 4, we present the experimental setup, including the RL pipeline that consumes CAE-generated wind data, and in Section 5, we report quantitative comparisons between CAE and CFD reconstructions as well as the resulting drone navigation performance under both wind models.

3 METHODOLOGY

3.1 Computational Fluid Dynamics Data Generation

In this study, we use OpenFOAM to generate high-fidelity CFD data, which serves as the training dataset for the convolutional autoencoder (Jasak 2009). The wind field is simulated by solving the incompressible transient two-dimensional Navier-Stokes equations governing mass and momentum conservation. The governing equations are given as follows:

$$\nabla \cdot u = 0 \tag{1}$$

$$\rho \left(\frac{\partial u}{\partial t} + u \cdot \nabla u\right) = -\nabla p + \nabla \cdot \tau + f \tag{2}$$

where u represents the velocity field with components in the x and y directions, ρ is the fluid density, p is the pressure field, τ is the stress tensor, and f accounts for external body forces such as gravity.

To simulate steady-state flow conditions, the time-dependent term is neglected, and the momentum equation is rewritten for the velocity components as:

$$u_x \left(\frac{\partial u_x}{\partial x}\right) + u_y \left(\frac{\partial u_x}{\partial y}\right) = -\left(\frac{1}{\rho}\right)\left(\frac{\partial p}{\partial x}\right) + \nu \nabla^2 u_x + g_x \tag{3}$$

$$u_x\left(\frac{\partial u_y}{\partial x}\right)+\,u_y\left(\frac{\partial u_y}{\partial y}\right)=\,-\left(\frac{1}{\rho}\right)\left(\frac{\partial p}{\partial y}\right)+\,\nu\,\nabla^2 u_y+\,g_y \tag{4}$$

where g represents gravitational acceleration and ν denotes the dynamic viscosity of the fluid. The terms on the left side describe convective transport, while the right side accounts for pressure gradients and diffusive transport effects.

The numerical solution of these equations is performed using OpenFOAM, an open-source CFD solver widely used for aerodynamic simulations. Specifically, we employ the simpleFoam solver, which implements the Semi-Implicit Method for Pressure-Linked Equations (SIMPLE) algorithm. This algorithm iteratively refines the velocity and pressure fields by solving discretized equations in two steps. First, an explicit velocity field solution is computed based on the current pressure field. Second, an implicit pressure correction step refines the pressure values using information from the previous iteration.

3.2 Convolutional Autoencoder Data Generation

A convolutional autoencoder (CAE) is a type of neural network architecture that combines the feature extraction capabilities of convolutional neural networks (CNNs) with the dimensionality reduction ability of autoencoders (AEs). This architecture leverages convolutional layers to exploit the local, translationally invariant patterns in wind fields, ensuring that key aerodynamic features are encoded into a compact latent space. By constraining the bottleneck, we force the network to focus on the most salient structures rather than noise, balancing representational power with computational efficiency. While deeper encoders, skip connections, or multi-scale filters could yield finer reconstructions, especially in highly turbulent regions, these variants often increase sensitivity to small-scale fluctuations or incur higher inference costs. Our design thus prioritizes a lean yet expressive CAE, with future work able to explore these alternative configurations for targeted fidelity improvements.

As shown in Figure 1, the CAE consists of two main components: an encoder, which processes data into a lower-dimensional latent space, and a decoder, which reconstructs the original data from this representation. By learning to capture key aerodynamic patterns, the CAE can generate synthetic wind field data that closely approximates high-fidelity CFD simulations while significantly reducing computational costs.

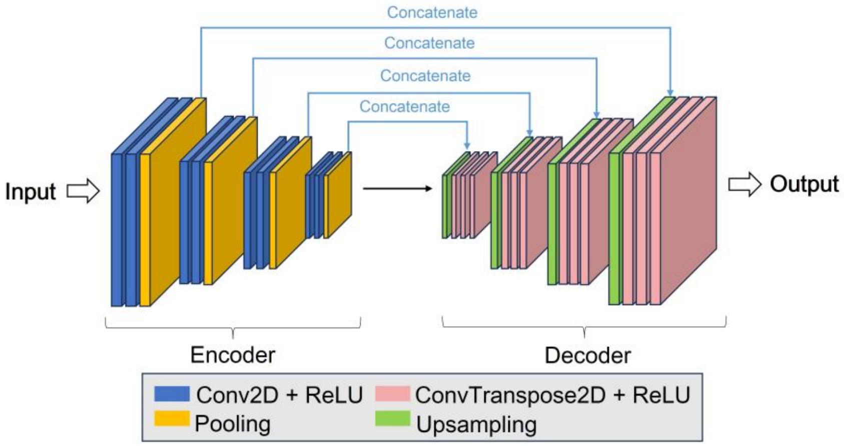


Figure 1: The structure of the convolutional autoencoder.

We develop a comprehensive methodology to generate and simulate the aerodynamic characteristics of wind flow around tall buildings. While CFD has traditionally been used for this purpose, it is computationally expensive and impractical for Deep Reinforcement Learning (DRL) applications, which require running thousands of simulations under dynamically changing conditions. One of the major challenges in traditional CFD data generation lies in handling the Navier-Stokes equations. As mentioned, these equations describe fluid motion based on the conservation of momentum, defining how velocity changes over time under external forces. In CFD applications, boundary conditions and obstacle placements significantly affect simulation results. Therefore, the CAE model must incorporate geometric and boundary condition data to ensure accurate input representation. We introduce a parameterization technique to represent these conditions numerically, defining five distinct regions within the computational domain. As shown in Figure 2, the flow region has a length of 200 and a width of 200. In the flow region, 0 represents obstacles, 1 denotes free-flow areas, 2 corresponds to the upper and lower no-slip wall conditions, 3 signifies the constant velocity inlet, and 4 indicates the zero-gradient velocity outlet. By encoding these conditions directly, the CAE model can process the geometric characteristics of the simulated environment as part of its input, ensuring that aerodynamic features are accurately learned and reproduced in the synthetic CFD data.

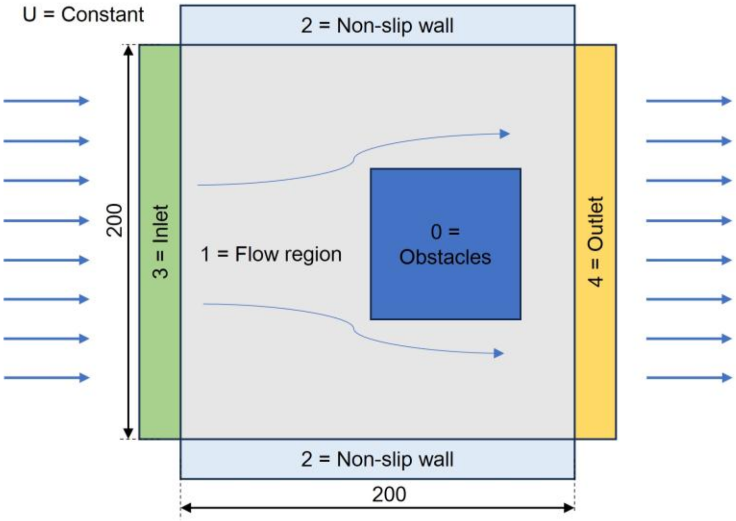


Figure 2: The flow region and boundary conditions for wind simulation.

To further enhance spatial representation, we incorporate the Signed Distance Function (SDF) to describe geometric relationships within the flow domain. The Signed Distance Function (SDF) is a mathematical tool widely used in computer graphics, computational geometry, and physics simulations to measure the distance between a point and a surface. It provides additional spatial context by encoding the distance and direction to the nearest obstacle, assigning a negative sign to points inside an object and a positive sign to points outside.

In our case, for a point x in the computational domain X and a geometric shape S, the first signed distance function $SDF_1(x)$ is defined as:

$$SDF_1(x) = \begin{cases} -d(x, \partial S) & if\ x \in S \\ d(x, \partial S) & else \end{cases} \tag{5}$$

where ∂S represents the boundary of a single obstacle and $d(x, \partial S)$ calculates the minimum spatial distance from the point x to the obstacle boundary. When multiple obstacles exist within the flow domain, this function will determine the minimum distance to any obstacle boundary in the environment. This representation enables the CAE model to interpret spatial relationships between fluid flow and obstacles.

The second signed distance function $SDF_2(x)$ is defined as:

$$SDF_2(x) = \begin{cases} -L(x,\theta) & if\ x \in S \\ L(x,\theta) & else \end{cases} \tag{6}$$

where θ denotes the center of the obstacle and $L(x,\theta)$ represents the horizontal distance from a given point to the obstacle's center. Similarly, the second SDF can be adapted to consider multiple obstacle centers. When multiple obstacles are in the flow domain, this function will be modified to reference the nearest obstacle center. This additional distance metric enhances the network's ability to distinguish variations in flow regions.

3.3 Validation of Autoencoder-Generated Wind Fields

To validate the autoencoder-generated wind fields, we use two complementary metrics. First, we report the Mean Absolute Error (MAE) to quantify pointwise deviations in velocity magnitude, an important factor because accurate velocity predictions directly influence RL policy performance. Second, we include the Structural Similarity Index (SSIM) to evaluate whether key spatial flow structures (e.g., vortices, wakes) are preserved, since MAE alone cannot capture these higher-order features.

The MAE quantifies the average absolute difference between the CAE-predicted wind field and the CFD reference data. A lower MAE value indicates higher accuracy in reconstructing the aerodynamic properties. The MAE is calculated as follows:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|X_i - Y_i| \tag{7}$$

where N represents the total number of data points, X_i denotes the CAE-generated wind field values, and Y_i refers to the corresponding CFD-generated values.

To further assess the quality of the reconstructed wind fields, we employ the SSIM, which evaluates the perceptual similarity between the predicted and ground-truth fields. Unlike MAE, SSIM considers structural information, luminance, and contrast to assess the generated wind distributions comprehensively. The SSIM is computed as:

$$SSIM(X,Y) = \frac{(2\mu_X\mu_Y + C_1)(2\sigma_{XY} + C_2)}{(\mu_X^2 + \mu_Y^2 + C_1)(\sigma_X^2 + \sigma_Y^2 + C_2)} \tag{8}$$

where μ_X and μ_Y are the means of the CAE-generated and CFD-generated wind fields, respectively; σ_X^2 and σ_Y^2 are their variances and σ_{XY} is the covariance between them. The constants C_1 and C_2 ensure numerical stability.

4 EXPERIMENT

This study evaluates the reliability of using a CAE for wind modeling by comparing its results with high-fidelity CFD simulations. The objective is to determine whether CAE-generated wind fields can be a computationally efficient alternative to traditional CFD methods used for autonomous drone simulation while maintaining aerodynamic accuracy. The experiment involves designing a simulated environment, training reinforcement learning-based drones under different wind conditions, and validating the accuracy, generalization, and transferability of CAE-generated wind fields.

4.1 Simulation Environment and Drone Parameters

To evaluate the impact of CAE-generated wind fields on drone navigation, we developed a simulation environment in Unity 3D, leveraging its scalability and interactive capabilities to model aerodynamic effects and integrate RL training seamlessly. The simulated environment represents an urban district populated with several randomly placed tall buildings, with widths and lengths ranging from 130 to 260 feet and heights varying between 200 and 400 feet. Wind conditions within the urban landscape are modeled without physical boundaries, allowing natural airflow to develop freely throughout the simulation.

For realistic drone behavior, we configured the simulation to reflect the specifications of a DJI Mavic 2 Pro, a widely used commercial UAV. The drone is modeled with a weight of 2 lbs., a maximum speed of 30 mph, and a maximum altitude of 200 feet, ensuring that it adheres to real-world flight constraints and responds accurately to wind disturbances. The primary objective of the drone is to navigate through the urban environment while optimizing flight efficiency and stability under various wind conditions.

To achieve adaptive navigation, we implemented a DRL algorithm based on Proximal Policy Optimization (PPO) to train the drone's control policy (Wu et al. 2024). The model follows a multi-objective learning framework, balancing shortest-path efficiency, flight stability, and time optimization. To enhance adaptability, Long Short-Term Memory (LSTM) networks are integrated within the RL framework, allowing the agent to retain temporal dependencies and adjust its control actions dynamically in response to changing wind conditions (Hochreiter and Schmidhuber 1997). The agent receives state information, including position, velocity, and wind field data, which it processes to generate optimal navigation decisions.

During training, wind conditions are randomized in each episode to introduce variability, with wind speeds ranging from 3 to 45 mph and wind directions varying from 0° to 360°. This setup ensures that the RL agent generalizes across diverse aerodynamic conditions rather than memorizing specific wind patterns, enabling robust navigation strategies in unpredictable environments.

4.2 Computational Fluid Dynamics Data Generation

We generated high-resolution CFD wind fields using OpenFOAM to provide ground-truth aerodynamic data for CAE training. The simulation domain replicates urban wind interactions with obstacles, applying boundary conditions that define inflow, outflow, and surface interactions. Table 1 illustrates the CFD settings, including boundary conditions, material properties, and solver details.

Table 1: Computational fluid dynamics details.

Aspect/Parameter	**Details/Choice**
Material in Flow	Air
Air Density	1.196 kg/m^3
Wind Speed (Inlet)	5 m/s
Air Pressure (Outlet)	0.015 kPa
OpenFOAM Solver	simpleFoam
Meshing Generation	snappyHexMes
Grid Convergence Study	Yes
Turbulence Model	k-ε turbulence model
Boundary Conditions	No-slip wall, velocity inlet, pressure outlet
Simulation Time Step	0.001 s
Number of Iterations	100,000
Computational Resources Used	Intel i9-14900K, 32GB RAM, RTX 4090

The wind simulations solve the incompressible Navier-Stokes equations, modeling air with a 1.196 kg/m³ density. The inlet wind speeds were set at 5 m/s, while the outlet pressure conditions were maintained at 0.015 kPa. These CFD-generated wind fields serve as training data for the CAE model.

To train and validate the CAE, we generated a diverse set of steady-state CFD simulations by randomly placing 1-10 buildings in each urban scene and varying wind direction among four angles (0°, 90°, 180°, 270°). Each steady-state run produces one full 2D wind-velocity field. In total, approximately 1,000 simulations were performed, which we split into 800 training and 200 validation samples. The CAE was trained in the 800 training fields (batch size = 16, learning rate = 1 × 10^{-4}, Adam optimizer) for 200 epochs using MAE as the loss function, terminating once both training and validation losses converged.

4.3 Convolutional Autoencoder Architecture

The convolutional autoencoder is trained to learn aerodynamic patterns from CFD-generated data and predict wind distributions based on input geometry. Table 2 presents the details of each layer in this architecture.

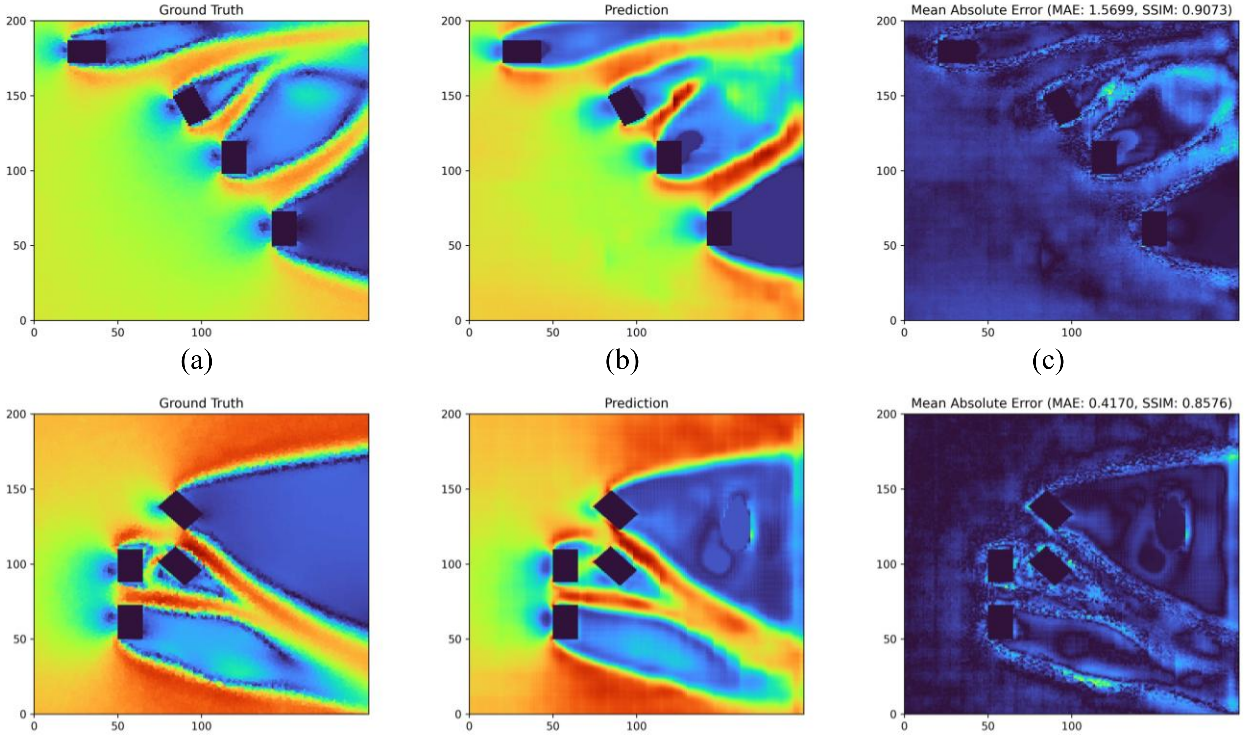
Table 2: Convolutional autoencoder architecture details.

No	Layer	Operation	Activation Function	Input Dimensions	Output Dimensions
1	Input	Geometry Encoding		3	8
2	Conv2D	Feature Extraction	ReLU	8	8
3	Conv2D+ MaxPool	Feature Extraction	ReLU	8	16
4	Conv2D	Feature Extraction	ReLU	16	16
5	Conv2D+ MaxPool	Feature Extraction	ReLU	16	32
6	Conv2D	Feature Extraction	ReLU	32	32
7	Conv2D+ MaxPool	Feature Extraction	ReLU	32	32
8	Copy and Concatenate	Feature Merging		32	32+32
9	ConvTranspose2D	Upsampling	ReLU	32+32	32
10	ConvTranspose2D	Upsampling	ReLU	32	32
11	ConvTranspose2D	Upsampling	ReLU	32	32
12	ConvTranspose2D	Upsampling	ReLU	32	16+16
13	ConvTranspose2D	Upsampling	ReLU	16+16	16
14	ConvTranspose2D	Upsampling	ReLU	16	8+8
15	ConvTranspose2D	Upsampling	ReLU	8+8	8
16	ConvTranspose2D	Final Output	ReLU	8	1

5 RESULTS

To evaluate the accuracy of the CAE in generating wind fields, we compare its predictions against the ground truth CFD results. Figures 3(a) and 3(d) display the ground truth wind fields for two distinct test cases, while Figures 3(b) and 3(e) show the corresponding CAE-generated predictions, and Figures 3(c) and 3(f) present the absolute-error maps. Figures 3(a)-3(c) show a complex wake downstream of multiple obstacles. Although the CAE captures the large-scale velocity gradients in Figure 3(b), it produces noticeable discrepancies in highly turbulent patches, resulting in a relatively high MAE of 1.5699 (SSIM 0.9073). The error distribution in Figure 3(c) is concentrated around sharp shear regions and small vortices. In contrast, Figures 3(d)-3(f) correspond to a milder turbulence scenario: the CAE prediction in

Figure 3(e) closely matches the ground truth in Figure 3(d), yielding a much lower MAE of 0.4170 (SSIM 0.8576). The error map in Figure 3(f) shows only minor, localized differences in fine-scale eddies. Together, these results demonstrate that while the CAE reliably reconstructs primary aerodynamic structures, its accuracy improves substantially under moderate turbulence and diminishes in highly turbulent regions.


Ground Truth

Prediction

Mean Absolute Error (MAE: 1.5699, SSIM: 0.9073)

Ground Truth

Prediction

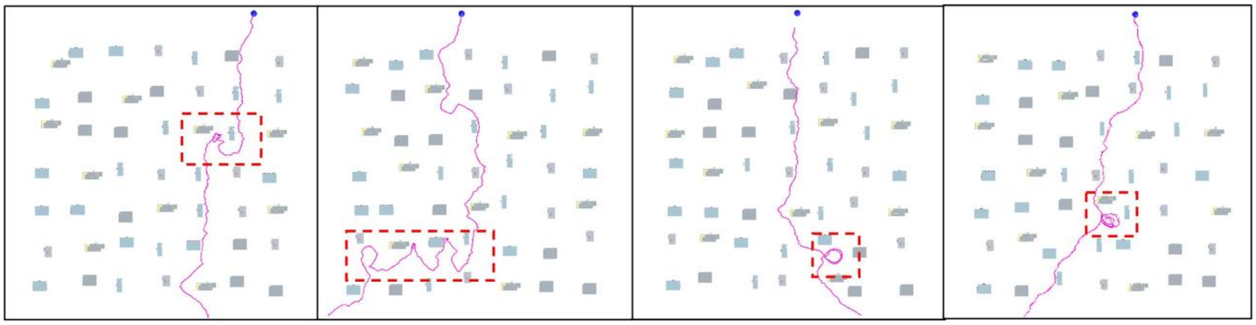
Mean Absolute Error (MAE: 0.4170, SSIM: 0.8576)

(a) (b) (c)

(d) (e) (f)

Figure 3: Comparison of ground truth CFD wind fields and CAE predictions.

The CAE model shows strong potential for accelerating aerodynamic simulations while maintaining reasonable accuracy. The errors observed suggest that further refinements, such as incorporating physics-informed loss functions or additional training data, may enhance predictive performance in complex flow scenarios.



(a) (b) (c) (d)

Figure 4: Drone trajectories under strong wind influence.

Figure 4 illustrates the trajectory of a drone navigating an urban environment under varying wind conditions generated by the CAE model. The target location is marked by a blue sphere, while the purple lines represent the drone's trajectory as it moves through the environment. The presence of strong wind zones significantly affects the drone's path, leading to detours, which are highlighted using red dashed rectangles. The subfigures show that the drone attempts to reach its target while adapting to the aerodynamic disturbances caused by localized strong wind regions. In these areas, the wind forces create instability, pushing the drone off its optimal path and forcing it to perform corrective maneuvers. The severity of these deviations varies across different wind scenarios, as seen in the differences between the trajectories in each subfigure. The observed detours demonstrate how the wind-aware navigation policy adjusts the flight path to avoid unstable zones and minimize drift, prioritizing safety over the shortest possible route. These results validate the importance of incorporating wind field modeling in the training process to enhance the drone's adaptability in real-world conditions.

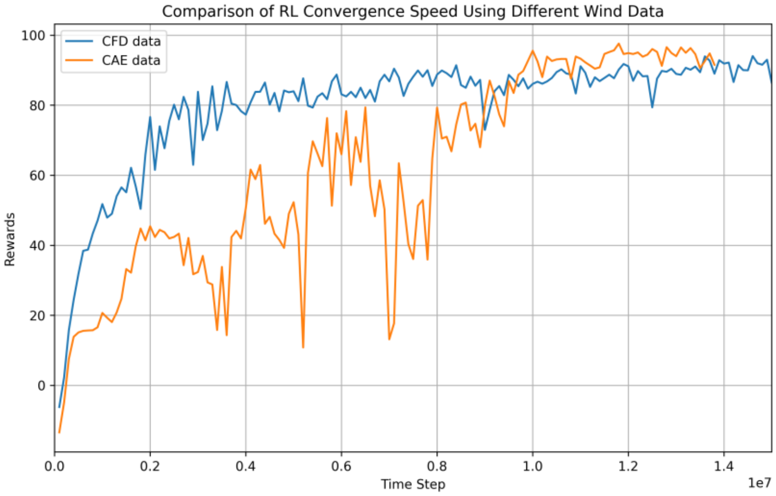


Figure 5: RL convergence comparison between CFD and CAE-generated wind data.

Figure 5 presents a comparison of RL convergence speed when training with CFD-generated wind data (blue curve) and CAE-generated wind data (orange curve). The y-axis represents the smoothed average reward over the time steps, while the x-axis denotes cumulative environment time steps (up to 10^7 steps), aggregated across multiple episodes. Each episode samples a new wind speed and direction and terminates when the drone reaches the target or exceeds 2,000 steps. Because episode lengths vary, the total number of episodes in 10^7 steps is roughly 5,000-6,000. As shown, the agent trained on CFD data initially achieves higher rewards, owing to more precise aerodynamic feedback. The CAE-based agent starts with lower rewards because minor reconstruction errors in the CAE fields slow early learning. However, by around 8×10^6 steps, the CAE curve catches up, and by 10^7 steps, both agents converge to comparable reward levels (around 90). This demonstrates that, despite a longer adaptation phase, CAE-generated wind fields can ultimately train an RL policy that performs on par with the CFD-based baseline.

6 CONCLUSION

This study explores the feasibility of using a CAE to replace CFD for wind modeling in reinforcement learning-based autonomous drone training. By training a CAE on high-fidelity CFD-generated wind data, we successfully demonstrated that CAE-generated wind fields can approximate CFD results while significantly reducing computational costs. The generated wind distributions were validated using quantitative metrics such as MAE and SSIM, confirming the CAE's ability to reconstruct crucial aerodynamic features, including velocity gradients and wake structures around obstacles. The experimental results indicate that drones trained with CAE-generated wind fields can achieve comparable navigation performance to those trained with CFD data. Initially, RL agents trained using CFD data exhibit faster convergence due to more precise aerodynamic information. However, after extended training, the CAE-based agents reach a similar level of performance, demonstrating the potential of CAE-generated wind fields as a computationally efficient alternative to CFD. Furthermore, our policy transferability tests show that RL policies trained on CAE-generated wind fields can generalize well when deployed in real CFD environments, ensuring the robustness and reliability of the learned control strategies.

Despite these promising results, several limitations remain. While the CAE reconstructs wind fields well under low to moderate turbulence, its performance degrades in highly turbulent regions, over-smoothing eddies and increasing reconstruction error, and it struggles with complex urban wake interactions; its performance in multi-building scenarios with varying building densities also remains untested. Notably, future work will explore multi-scale convolutional branches (3×3, 5×5, 7×7 kernels), lightweight attention modules, and turbulence-aware losses to better capture small-scale features. Accelerating generalization will involve expanding the training set to include high-intensity turbulence and dynamic obstacles. Additionally, future studies should assess the CAE under varying atmospheric pressures, precipitation, and dynamic obstacles to validate its robustness in diverse environments. Rigorous, long-term real-world field tests are also recommended to confirm applicability.

In summary, this study demonstrates that CAE-based wind modeling is a viable, efficient, and scalable alternative to traditional CFD simulations for training autonomous drones. The ability to generate reliable aerodynamic data with minimal computational costs makes this approach highly valuable for large-scale reinforcement learning applications, ultimately advancing the field of data-driven aerodynamic modeling and autonomous flight navigation.

ACKNOWLEDGEMENTS

This material is supported by the Air Force Office of Scientific Research (AFOSR) under grant FA9550-22-1-0492. Any opinions, findings, conclusions, or recommendations expressed in this article are those of the authors and do not reflect the views of the AFOSR.

REFERENCES

Biao, L., J. Cunyan, W. Lu, C. Weihua, and L. Jing. 2019. "A Parametric Study of the Effect of Building Layout on Wind Flow over an Urban Area". *Building and Environment* 160:106160.

Blocken, B., T. Stathopoulos, and J. Carmeliet. 2008. "Wind Environmental Conditions in Passages between Two Long Narrow Perpendicular Buildings". *Journal of Aerospace Engineering* 21(4):280-287.

Calzolari, G., and W. Liu. 2021. "Deep Learning to Replace, Improve, or Aid Cfd Analysis in Built Environment Applications: A Review". *Building and Environment* 206:108315.

Capolupo, A., S. Pindozzi, C. Okello, N. Fiorentino, and L. Boccia. 2015. "Photogrammetry for Environmental Monitoring: The Use of Drones and Hydrological Models for Detection of Soil Contaminated by Copper". *Science of the Total Environment* 514:298-306.

Gianfelice, M., H. Aboshosha, and T. Ghazal. 2022. "Real-Time Wind Predictions for Safe Drone Flights in Toronto". *Results in Engineering* 15:100534.

Giersch, S., O. El Guernaoui, S. Raasch, M. Sauer, and M. Palomar. 2022. "Atmospheric Flow Simulation Strategies to Assess Turbulent Wind Conditions for Safe Drone Operations in Urban Environments". *Journal of Wind Engineering and Industrial Aerodynamics* 229:105136.

Hochreiter, S., and J. Schmidhuber. 1997. "Long Short-Term Memory". *Neural computation* 9(8):1735-1780.

Jacobsen, R. H., L. Matlekovic, L. Shi, N. Malle, N. Ayoub, K. Hageman, S. Hansen, F. F. Nyboe, and E. Ebeid. 2023. "Design of an Autonomous Cooperative Drone Swarm for Inspections of Safety Critical Infrastructure". *Applied Sciences* 13(3):1256.

Jasak, H. 2009. "Openfoam: Open Source Cfd in Research and Industry". *International journal of naval architecture and ocean engineering* 1(2):89-94.

Jeong, S., K. You, and D. Seok. 2021. "Hazardous Flight Region Prediction for a Small Uav Operated in an Urban Area Using a Deep Neural Network". *Aerospace Science and Technology* 118:107060.

Kaelbling, L. P., M. L. Littman, and A. W. Moore. 1996. "Reinforcement Learning: A Survey". *Journal of artificial intelligence research* 4:237-285.

Ma, H., Y. Zhang, N. Thuerey, X. Hu, and O. J. Haidn. 2021. "Physics-Driven Learning of the Steady Navier-Stokes Equations Using Deep Convolutional Neural Networks". *arXiv preprint arXiv:2106.09301.*

Mayer, S., L. Lischke, and P. Woźniak. Year. "Drones for Search and Rescue". *1st International Workshop on Human-Drone Interaction*, May 5th, Glasgow, United Kingdom.

Milla-Val, J., C. Montañés, and N. Fueyo. 2023. "Economical Microscale Predictions of Wind over Complex Terrain from Mesoscale Simulations Using Machine Learning". *Modeling Earth Systems and Environment*:1-15.

Ng, A. 2011. "Sparse Autoencoder". *CS294A Lecture notes* 72(2011):1-19.

Paz, C., E. Suárez, C. Gil, and J. Vence. 2021. "Assessment of the Methodology for the Cfd Simulation of the Flight of a Quadcopter Uav". *Journal of Wind Engineering and Industrial Aerodynamics* 218:104776.

Qu, C., F. B. Sorbelli, R. Singh, P. Calyam, and S. K. Das. 2023. "Environmentally-Aware and Energy-Efficient Multi-Drone Coordination and Networking for Disaster Response". *IEEE Transactions on Network and Service Management* 20(2):1093-1109.

Ribeiro, M. D., A. Rehman, S. Ahmed, and A. Dengel. 2020. "Deepcfd: Efficient Steady-State Laminar Flow Approximation with Deep Convolutional Neural Networks". *arXiv preprint arXiv:2004.08826.*

Wu, J., Y. Ye, and J. Du. 2024. "Multi-Objective Reinforcement Learning for Autonomous Drone Navigation in Urban Areas with Wind Zones". *Automation in construction* 158:105253.

Zhang, Z., C. Santoni, T. Herges, F. Sotiropoulos, and A. Khosronejad. 2021. "Time-Averaged Wind Turbine Wake Flow Field Prediction Using Autoencoder Convolutional Neural Networks". *Energies* 15(1):41.

AUTHOR BIOGRAPHIES

JIAHAO WU is a Ph.D. candidate at the Informatics, Cobots, and Intelligent Construction Lab in the Department of Civil and Coastal Engineering at the University of Florida in Gainesville, Florida, USA. His research interests include drone simulation, human-robot collaboration, and reinforcement learning for autonomous robots. His email address is jiahaowu@ufl.edu.

HENGXU YOU is a Ph.D. candidate at the Informatics, Cobots, and Intelligent Construction Lab in the Department of Civil and Coastal Engineering at the University of Florida in Gainesville, Florida, USA. His research interests include the 3D point cloud object detection and environmental understanding, human-instructed reinforcement learning for robot arms, and embodied AI. His email address is you.h@ufl.edu.

JING DU is a professor in the Department of Civil Engineering, the Department of Mechanical and Aerospace Engineering, and the Department of Industrial and System Engineering at the Herbert Wertheim College of Engineering, University of Florida. His primary area of research is human-robot collaboration for complex industrial operations. Dr. Du is the elected Secretary of the American Society of Civil Engineers (ASCE) Visualization, Information Modeling and Simulation (VIMS) committee and serves on the editorial board of three journals. His email address is eric.du@essie.ufl.edu, and his website is https://faculty.eng.ufl.edu/ericdu/.