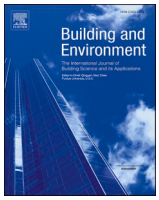
Building and Environment 284 (2025) 113495



Contents lists available at ScienceDirect

Building and Environment

journal homepage: www.elsevier.com/locate/buildenv



Deep learning for urban wind prediction: An MLP-Mixer approach with 3D encoding

Adam Clarke [a,*], Knut Erik Teigen Giljarhus [b,c], Luca Oggiano [c], Alistair Saddington [a], Karthik Depuru-Mohan [a]

[a] Centre for Defence Engineering, Cranfield University, Defence academy of the UK, Shrivenham, United Kingdom
[b] Department of Mechanical and Structural Engineering and Materials Science, University of Stavanger, Stavanger, Norway
[c] Nablaflow AS, Stavanger, Norway

ARTICLE INFO

Keywords:
Pedestrian-level wind
Wind comfort
Urban wind
Deep learning
Neural networks
Lift-up
Skyway

ABSTRACT

Pedestrian-level wind environments are strongly influenced by urban morphology, with large and tall buildings playing a significant role. City authorities increasingly mandate assessments of pedestrian wind conditions before approving new construction. Computational fluid dynamics (CFD) models can provide a detailed understanding of the aerodynamic environment; however, in early design stages, urban morphologies are subject to change, requiring multiple simulations, adding substantial financial and time burdens to projects.

To address this challenge, we develop a deep learning approach for the rapid inference of pedestrian-level wind conditions using a multi-layer perceptron (MLP)-mixer architecture. By embedding 3D structural details into the training data, our model can infer wind conditions around complex structures such as lift-up designs and skyways while maintaining inference times on the order of fractions of a second. This extends the capabilities of deep learning models that typically reduce the problem to a 2D image-to-image translation task, omitting crucial structural details.

We conduct an extensive evaluation of our model and compare its performance to the widely adopted UNet architecture, demonstrating that the MLP-mixer outperforms UNet across all evaluation metrics. Notably, the MLP-Mixer achieves a mean squared error approximately 2.6 times lower, a peak signal-to-noise ratio 3.7 dB higher and the highest recorded structural similarity index of 0.991. These results indicate improved agreement with the reference CFD data. We anticipate that the MLP-mixer model will serve as a valuable tool in early-stage urban design workflows, enabling faster and more efficient wind assessments.

1. Introduction

Urban outdoor spaces are essential to city life, supporting social, cultural, and economic activities. With 68 % of the global population expected to live in cities by 2050, one in eight in megacities, urban infrastructure faces increasing pressure. Since 1990, large cities have grown at an average rate of 1.8 % annually, particularly in lower-middle-income countries [1]. To accommodate this growth, cities must either increase density or expand, though expansion often harms agricultural land and wildlife. Compact living offers a sustainable alternative, driving the rise of skyscrapers [2], which are expected to increase eightfold by 2050 and surpass today's tallest structures by 50 % [3]. Beyond optimising land use, skyscrapers enhance a city's global image, boost economic activity, and provide proximity to work, infrastructure, and cultural amenities [2,4].

The concentration of tall buildings significantly affects pedestrian-level wind conditions. Large structures deflect high-speed winds downward, creating vortices, accelerating flows, and forming low-pressure zones on their leeward sides [5]. In dense urban areas, closely spaced buildings generate wind corridors that intensify turbulence, noise, and pollutant transport [6]. Poorly managed wind environments can elevate mean wind speeds by 50–75 %, posing safety risks and discouraging outdoor activity due to discomfort [7,8]. These challenges are further compounded by the increasing frequency of extreme weather events, which intensify urban wind effects and heighten risks to infrastructure and public safety. Conversely, well-designed wind environments enhance urban life by fostering social interactions, economic activity, and cultural engagement. Proper ventilation disperses

* Corresponding author.
E-mail address: adam.p.clarke@cranfield.ac.uk (A. Clarke).

https://doi.org/10.1016/j.buildenv.2025.113495
Received 19 April 2025; Received in revised form 14 July 2025; Accepted 27 July 2025
Available online 5 August 2025
0360-1323/© 2025 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

pollutants and supports natural cooling [9–11]. Achieving these benefits requires balancing wind flow management to ensure safety, comfort, and sustainability.

Lift-up building design, where the ground floor of a building is elevated on columns or shear walls to create open pedestrian spaces [12], is recognised for improving urban wind comfort [13,14]. This design enhances ventilation by increasing wind velocities at street level, particularly in dense urban environments where airflow is typically obstructed [12,13,15]. By allowing wind to pass through the open ground floor, lift-up designs reduce wind blockage and can create localised cooling zones that encourage outdoor activity [16]. However, their effectiveness is influenced by surrounding buildings, which can either enhance or diminish wind amplification [14]. While non-lift-up buildings may partially reduce the wind-enhancing effects, lift-up structures can still improve mean wind velocity by altering wake regions and mitigating the canyon effect caused by closely spaced buildings [13].

In addition to lift-up designs, skyway bridges, which connect buildings at elevated levels, also play a crucial role in shaping pedestrian wind comfort. Poorly designed skyway bridges can create a channelling effect, amplifying wind speeds and causing discomfort or even safety concerns for pedestrians [17]. Conversely, a well-configured skyway bridge can help moderate wind speeds, enhancing overall comfort at both street level and on the bridge itself [18].

Wind-related challenges also affect energy consumption, construction costs, and urban economies. Strong winds increase heating demands by stripping heat from buildings, while weak winds raise cooling loads [19]. Mitigating wind-induced damage necessitates costly structural reinforcements and aerodynamic designs, while poor wind conditions can deter investment, tourism, and long-term urban viability.

Pedestrian-level wind assessments, which are increasingly becoming mandated by local authorities, evaluate outdoor safety and comfort during the design phase of new buildings [20–22]. These assessments incorporate meteorological data, building aerodynamics, and human comfort metrics to generate maps highlighting areas of concern [23,24]. Comfort standards typically set limits on the probability of wind exceeding threshold velocities, categorising conditions into comfort bands [25–28].

Aerodynamic data is obtained through wind tunnel testing or computational fluid dynamics (CFD) simulations. While wind tunnels remain a standard method, they are costly, time-intensive, and require specialised expertise. Scale model construction, experimental setup, and data collection become increasingly complex for intricate geometries or large urban areas. Maintaining dynamic similarity with real-world conditions presents further challenges, while boundary layer effects and limited spatial resolution of measurement instruments can impact accuracy, particularly for tall structures or dense urban environments [29].

CFD offers a versatile alternative by solving the governing equations of fluid motion to simulate turbulent flows across diverse scenarios, providing detailed insights into flow characteristics. Its flexibility allows integration into optimisation frameworks, enabling iterative refinement of design choices, particularly for pedestrian-level wind assessments. However, CFD comes with challenges. The choice of turbulence modelling significantly impacts accuracy and computational cost [30]. High-fidelity methods like Direct Numerical Simulation (DNS) and Large Eddy Simulation (LES) capture fine-scale turbulence but are computationally expensive, making them impractical for large urban environments. The Reynolds-Averaged Navier-Stokes (RANS) approach, widely used in industry, is more efficient but struggles in regions with complex turbulence, such as wake zones behind buildings [24,31]. Nevertheless, for pedestrian-level wind assessments, where low wind speeds rarely pose critical challenges, RANS is often a practical and effective choice.

Urban wind conditions vary across temporal scales, necessitating assessments with multiple wind directions, usually between 8 and 36. While increasing inlet directions improves the accuracy of pedestrian wind comfort maps [32], it further raises computational and financial costs.

Once aerodynamic data is obtained, it is combined with statistical weather data from nearby stations, adjusted for site-specific morphology, and used to generate comfort maps based on assessment criteria [33]. During early design stages, lower-resolution qualitative insights often suffice, allowing for faster iterations. The widespread use of RANS methods reflects the industry's preference for balancing speed and accuracy, particularly when simulating multiple design variations.

Machine learning (ML) techniques have recently emerged as a promising alternative to traditional computational methods. These data-driven approaches, especially deep learning networks, offer the potential to predict wind flows with significantly reduced computational costs and improved speeds. Unlike physics-based simulations, ML models learn relationships between input data, typically geometric features or initial conditions, and outputs such as time-averaged wind flow fields.

Several studies have explored the application of machine learning in urban wind modelling [34–41]. Among these, generative adversarial networks (GANs) and convolutional neural networks (CNNs) have demonstrated notable promise. CNNs, well-established in computer vision, have proven effective for tasks that involve spatially coherent inputs and outputs, such as predicting steady-state wind velocity fields from rasterized representations of urban geometry. Due to their architecture CNNs have inherent inductive biases toward local spatial structure, which allows them to efficiently learn low-level features such as edges, corners, and near-wall effects with relatively modest amounts of data. However, it can also hinder generalisation beyond the training dataset. Key features of wind behaviour - such as the proximity to boundaries, obstacles, and spatial positioning are critical for accurate predictions. Techniques like incorporating signed distance functions (SDFs) [38] or spatial coordinates [42] during training help mitigate this limitation, enabling the model to understand its spatial context. However, CNNs' ability to capture long-range dependencies, such as upstream-to-downstream interactions, may remain limited unless mechanisms like global pooling or sufficiently large receptive fields are incorporated. Without these enhancements, CNNs might produce similar downstream flow predictions for vastly different upstream conditions, leading to inaccuracies. Clemente et al. [43] introduced a configurable CNN framework that allows users to balance accuracy and inference speed. By trading some accuracy, users could reduce the time per forward pass by up to sevenfold, providing flexibility for iterative design optimisation.

GANs, on the other hand, introduce a learnable loss function through their adversarial training framework [44]. This enables the generator to produce outputs that are perceptually sharper or more consistent with real data distributions, which can be especially useful for recovering fine-scale flow structures. In many cases, additional conditions, such as geometry or boundary constraints, are incorporated during training leading to conditional GANs (cGANs) that allow for more targeted and physically consistent generation. Kastner and Dogan [45] applied a GAN-based surrogate model to develop real-time urban wind prediction achieving high structural similarity to ground truth simulations, However, they noted that performance degraded in regions with strong velocity gradients, such as near building surfaces, though accuracy improved with larger training datasets.

Hoiness et al. [42] compared various cGAN and CNN networks trained on datasets with diverse urban morphologies. They found that while equivalent performance among the networks was observed, the additional complexity added through adversarial optimisation can make training unstable and convergence difficult, limiting practicality. Each class of networks in the study exhibited the highest errors in downstream regions, where flow conditions are typically more challenging to model accurately.

Recent advancements include the use of Fourier neural operator (FNO) networks and graph-based neural networks (GNN). FNOs model physical systems governed by partial differential equations by using

spectral transforms to learn mappings in the frequency domain, unlike other models which operate in the spatial domain. Qin et al. [46] demonstrated that FNOs could make accurate velocity predictions for full 3D urban microclimates over 60 s, achieving speeds up to 50 times faster than traditional numerical solvers. However, FNOs are still relatively slow when compared to previously mentioned models, which can operate thousands of times faster [34,47,48] although it is important to note that FNOs are typically used for transient problems, whereas many CNN and GAN-based approaches typically focus on predicting steady-state flow fields. Additionally, FNO models require initial conditions two timesteps ahead, potentially relying on short-term CFD solutions which may not be readily available in early-stage or zero-shot prediction scenarios. This constraint can limit their utility when applied to novel urban geometries without precomputed simulation data.

GNNs are a versatile architecture which have been used broadly across engineering applications owing to their ability to learn from unstructured data. A key strength of GNNs is in their ability to model spatially complex domains without requiring uniform structured grids. This makes them particularly well-suited to CFD application involving intricate urban geometries. In such cases unstructured CFD mesh data can represented as graphs with nodes and edges representing physical quantities and spatial relationships [47,49]. The networks employ an encoder-decoder framework with message passing to transform the initial graph into a representation of the resulting flow field. In this framework, the encoder maps the raw input graph into a latent space, capturing spatial relationships and flow-related features. The message passing process then iteratively propagates information between connected nodes, allowing the network to learn complex dependencies within the data. Finally, the decoder reconstructs the flow field from the learned graph representation, predicting velocity, pressure, or other relevant physical quantities. However, while GNNs excel at handling unstructured data, their high memory requirements pose a challenge, limiting their practicality for urban wind engineering applications which require large-scale or high-resolution meshes limiting their practicality. That said, ongoing research is actively exploring methods to reduce their memory footprint, enhancing their feasibility for large-scale simulations through methods such as graph sparsification or hierarchical message passing [49].

While each of these models presents unique advantages, they also carry inherent limitations—such as difficulty modelling global spatial interactions (CNNs), training instability (GANs), computational cost (FNOs), or memory inefficiency (GNNs). To address these challenges, we propose the use of MLP-Mixers.

The multi-layer perceptron (MLP)-Mixer is a supervised learning model composed entirely of linear transformations, nonlinear activations, and data transpositions. This model architecture has demonstrated state-of-the-art performance in various computer vision tasks [50], including image denoising [51], classification [52], segmentation [53] and image generation [54]. Unlike CNNs, which leverage local spatial relationships through receptive fields, MLP-Mixers primarily learn global interactions across an image. Consequently, they have a lower inherent bias toward spatial structure but excel in capturing broader contextual information that may be missed by locally focused architectures. Such global modelling capability is especially relevant to urban fluid dynamics, where wind patterns are governed not only by immediate surroundings but also by the influence of distant structures such as wakes, channelling effects, and shadow zones. Where CNNs require deep stacking or large receptive fields to learn such long-range dependencies, MLP-Mixers can capture them directly through their token-mixing layers.

Furthermore, MLP-Mixers are computationally efficient and highly parallelisable, with a parameter count that scales linearly with input dimensions. This characteristic leads to fast inference times and smaller model sizes, reducing storage requirements and enhancing deployment efficiency which is particularly valuable for scenarios requiring rapid simulation feedback such as iterative design or real-time wind assessment.

Mansour et al. [51] introduced the image-to-image MLP-Mixer, an adaptation that enables the model to generate or modify images through additional spatial processing. They demonstrated that their model could achieved state-of-the-art performance in image reconstruction performance, provided the training set and network size were sufficiently large. Additionally, they showed that their MLP-mixer architecture could match the performance of a UNet while using fewer parameters resulting in more compact models. However, a known drawback of MLP-Mixers is their limited ability to exploit local spatial correlations, due to the absence of strong spatial inductive biases. Clarke, et al. [55,56] addressed this limitation by introducing an additional local mixing step, which enhanced the MLP-Mixer's capacity to capture fine-grained spatial features. Their refined models were successfully applied to pedestrian-level wind assessment, demonstrating the potential of MLP-Mixers as data-driven surrogate models in early-stage design. While MLP-Mixers may underperform in data-scarce regimes compared to CNNs, their ability to model global interactions makes them a compelling alternative for tasks where spatial dependencies span the entire domain.

This study builds on our earlier work [56], which introduced a compact 2.5D data representation that encodes both the highest and lowest building elevations within each grid cell. This approach enables the model to account for important architectural features, such as lift-up buildings, podiums, and skyways, that are often missed by traditional 2D height maps. While some recent deep learning methods attempt to incorporate full 3D geometries through voxel grids, meshes, or point cloud representations, these approaches are typically computationally expensive, memory-intensive, and difficult to scale to large urban areas or fine-grained resolutions.

In contrast, our 2.5D encoding offers a practical middle ground, capturing key geometric structures with minimal additional complexity introducing only a constant-factor increase in input dimensionality (from one to two height values per cell), as opposed to the orders-of-magnitude increase in memory and computation required by voxel grids or point clouds. In this paper, we substantially extend that initial framework through a more rigorous and comprehensive evaluation. We benchmark the geometry-aware MLP-Mixer against the widely used UNet [57], and demonstrate that our model achieves comparable or superior predictive accuracy with significantly fewer parameters. Additionally, we perform a comprehensive ablation study to isolate the contributions of the model components, and we explore the latent feature representations within intermediate layers to better understand how the model internalises spatial and structural information. These analyses provide new insights into the model's behaviour and reinforce its suitability for efficient, scalable urban wind prediction.

2. Model development

We begin by expressing the problem of predicting wind velocities at pedestrian-level as an image-to-image translation task. Given some geometry, $x \in X$ we train a model $f(x,\ \theta)$ to predict the components of velocity at each point in space on a plane normal to the z-axis 2 m from the ground, where θ are the trainable parameters of the model. Each geometry $x \in X$ has a corresponding precomputed velocity field $y \in Y$ extracted from a RANS solution which we call the ground truth. To train the model we define a loss function $\mathscr{L}(f(x,\ \theta),\ Y)$ which we minimise with respect to θ using back propagation.

Deep learning models rely on extensive datasets to effectively capture and learn the underlying patterns and relationships between the input and output, with their accuracy often improving as the quantity and quality of data increases. To generate a sufficiently large dataset for this study, we developed an automated pipeline for parametric geometry generation, CFD simulation, and preprocessing of flow fields, which serve as training data for the model.

Table 1
Ranges of possible values for the variables in the parametric design space.

Dimension	Values (meters)
$h_{1,2}$	[10, 20, 40, 60, 80, 100]
$w_{1,2}$	[10, 20, 40, 60, 80]
$d_{1,2}$	[10, 20, 40, 60, 80]
$y_{1,2}$	[0, 10, 20]
$z_{1,2}$	[0, 3]

2.1. Geometry generation

A total of 490 unique geometries were created for this study. Each geometry consisted of two parallel cuboid buildings separated by a 20-meter gap, with a connecting bridge (skyway) structure spanning between them. The skyway had a depth of 3 m in the y-direction and a height of 3 m It was positioned 6 m above ground level, with its leading edge aligned with the front face of the rearmost building. One, both or neither of the buildings could incorporate a lift-up design. The geometries were generated using a Latin hypercube sampling technique, ensuring a well-distributed representation across the entire design space. The cuboid simplification was chosen to balance realism with computational efficiency, capturing essential aerodynamic features while keeping the CFD simulations manageable. The parametric design space is detailed in Table 1, with a schematic diagram provided in Fig. 1 and visualisation of the design space coverage in Fig. 2.

2.2. CFD model

To generate the ground truth data, a series of steady-state RANS simulations were conducted using OpenFOAM version 2206. For each unique geometry, eight independent simulations were performed to capture variations in flow behaviour under different inlet directions in increments of 45 degrees resulting in 3920 cases. Details of the computational setup and validation are provided in appendix A.

2.3. Data processing

The building geometries were converted into structured 2D height maps using the PyVista Python package [58], which enabled consistent spatial sampling of the 3D domain. Each sample was represented as a 256 × 256 pixel, two-channel image. The first channel encoded the height from the ground to the top surface of any structure at each (x, y) location, while the second channel encoded the height from the ground to the bottom surface enabling representation of elevated buildings or skyways. To standardise the data, both channels were normalised to a [0, 1] range by dividing by a reference maximum height of 100 m. Each pixel corresponded to approximately 0.75 m × 0.75 m in the physical domain, ensuring consistent spatial resolution across the dataset. Further details on the geometry encoding can be found in Section 3.2.

Similarly, the corresponding CFD solutions were converted into 256 × 256-pixel images, each consisting of three channels corresponding to the velocity components in the x, y, and z directions. The velocity field was sampled on the XY plane at a height of 2 m above the ground. This plane was extracted from the pre-computed flow field and interpolated to provide velocity values at each spatial point.

To standardise the orientation of the data samples, each geometry and velocity field pair was rotated so that the inflow direction was aligned with the top edge of the image. To further maintain consistency between images rotated by multiples of 90° and 45°, a corner mask was applied. This mask mitigated the disparity in information loss at the corners, thereby reducing potential confusion for the model during training. The masked corner regions correspond to the uniform far-field flow which does not contain any features of interest; Therefore, they do not contain meaningful information relevant to model training. A depiction of the processed input and output data can be seen in Fig. 3.

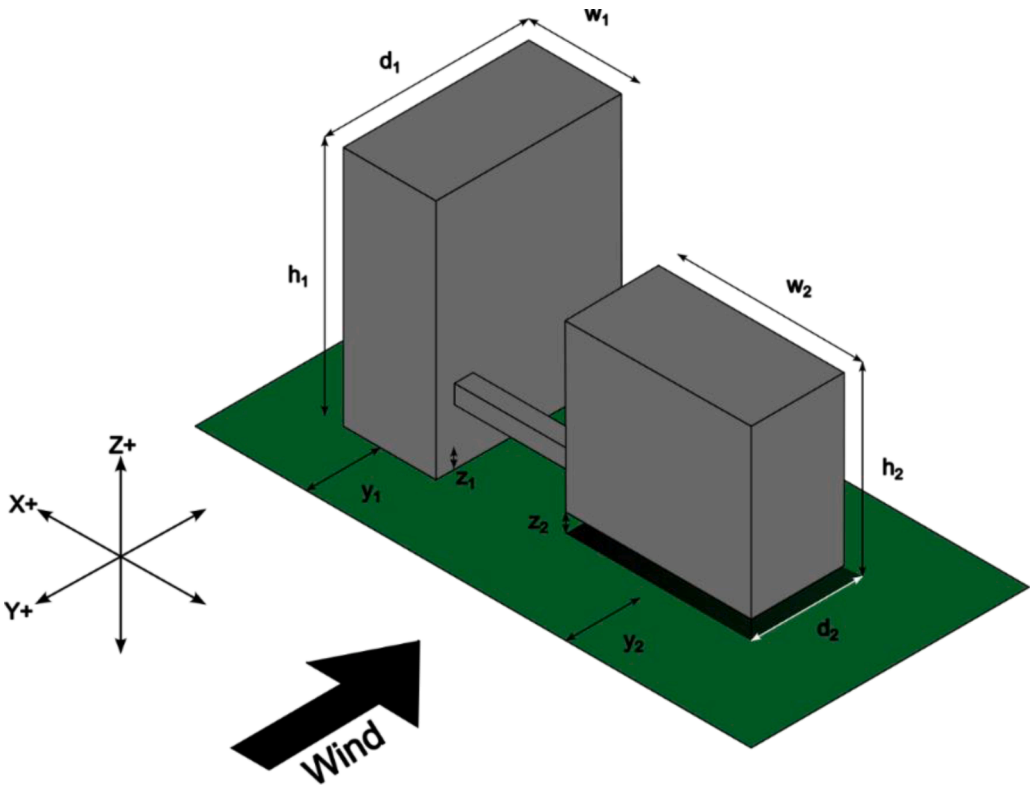


Fig. 1. A schematic diagram illustrating the training geometries.

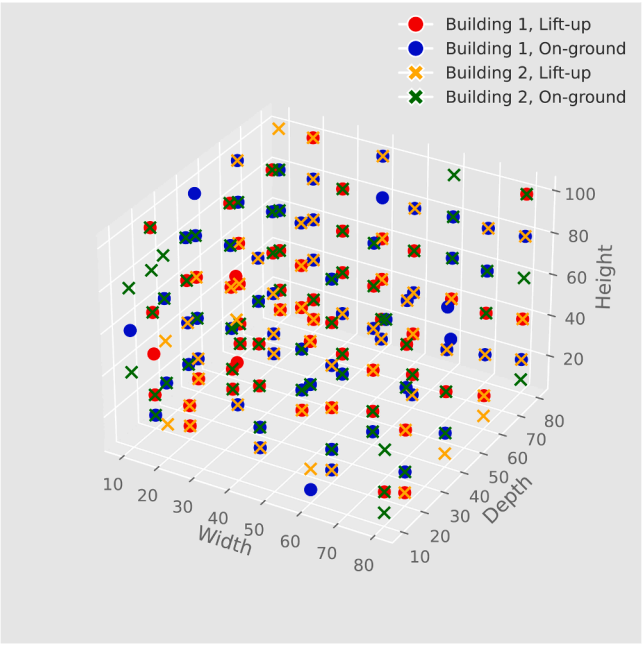


Fig. 2. Visualisation of the parametrically generated design space. A Latin hypercube sampling method was used to ensure a broad and uniform coverage across the width, depth, and height dimensions.

2.4. Model architecture

Our model builds upon the image-to-image MLP-mixer architecture introduced by Mansour, Lin and Heckel [42], incorporating an additional local mixing step enhancing its ability to accurately predict wind flow by facilitating the exchange of localised information. The network is structured into three sequential stages, each transforming the input geometry into the predicted flow field: patch embedding, mixing layers and patch reconstruction. Each mixing layer encompasses spatial mixing, channel mixing and localised mixing. A diagrammatic representation of the model architecture is show in Fig. 3.

2.4.1. Patch embedding

The first stage, the patch embedder, partitions the input image into discrete, non-overlapping patches of size $P \times P \times 2$. Each patch is then transformed by a shared trainable linear layer into a feature vector of an arbitrary embedding dimension C. These feature vectors are arranged to preserve their spatial relationships within the image, resulting in a latent tensor of dimensions $(H/P) \times (W/P) \times C$, where H and W denote the height and width of the original input image, respectively.

2.4.2. Mixing layers

The latent tensor is subsequently processed through a series of mixing layers, which iteratively mix and distribute information across the spatial and channel dimensions. Each mixing layer performs three key operations:

Channel-wise Mixing: This operation mixes information across the C channels within each feature vector, enabling interaction among the dimensions of the embedding.

Spatial Mixing: Information is redistributed across the spatial dimensions (H/P and W/P) to allow communication between patches.

Local Mixing: To enhance spatial interactions, we introduce a local mixing step. In this step each feature vector is influenced by neighbouring vectors within a predefined local region. This enhancement facilitates localised context sharing, capturing relationships between nearby patches that are important for the accurate prediction of flow fields. The local mixing step extends the original MLP-mixer architecture by incorporating spatially aware interactions without altering the overall tensor dimensions.

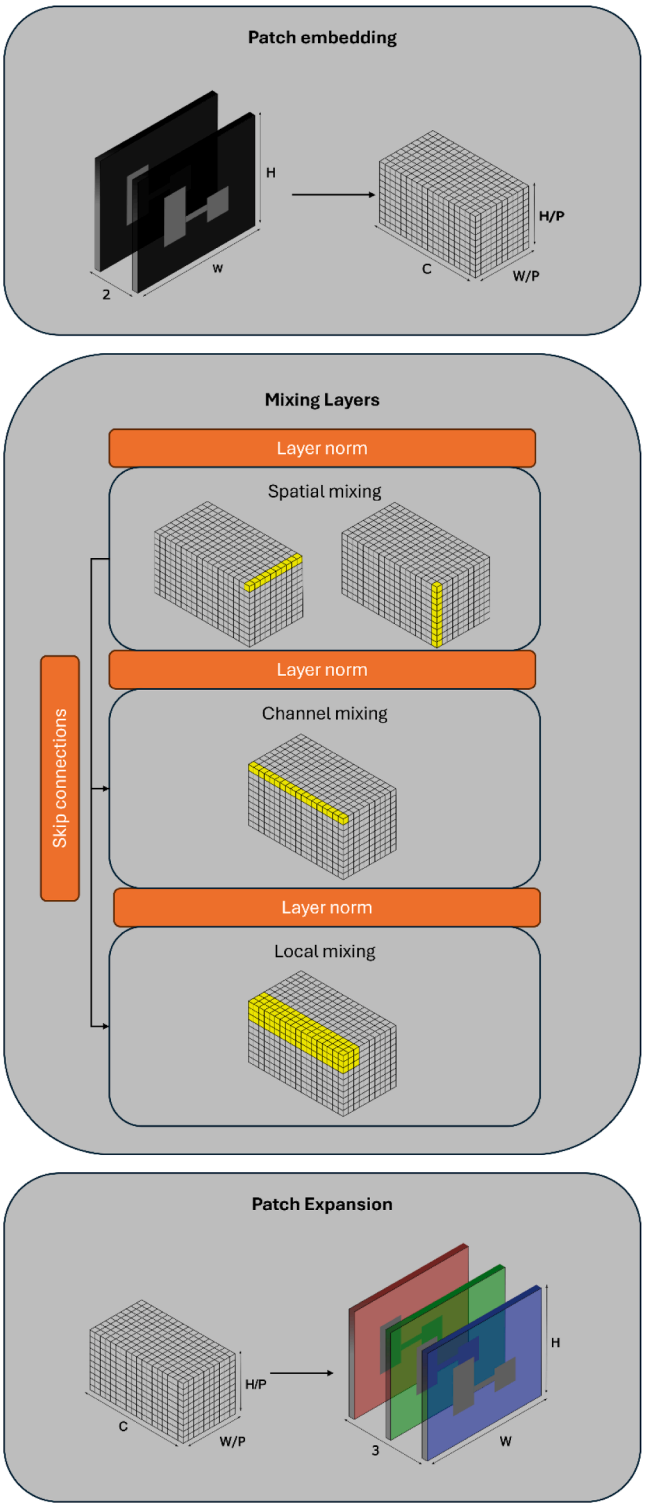


Fig. 3. Schematic representation of the MLP-mixer architecture, including a breakdown of the mixing step adapted from [51].

The implementation of these mixing operations relies on shared trainable layers. For channel-wise and spatial mixing, two shared MLP layers are applied to each input dimension, separated by a Gaussian Error Linear Units function (GeLU) non-linear activation. Local mixing is implemented using shared two-dimensional convolutional layers, which linearly transform each $N \times N \times C$ region of the latent tensor where N is the size of the neighbourhood. Similarly, these layers are separated with

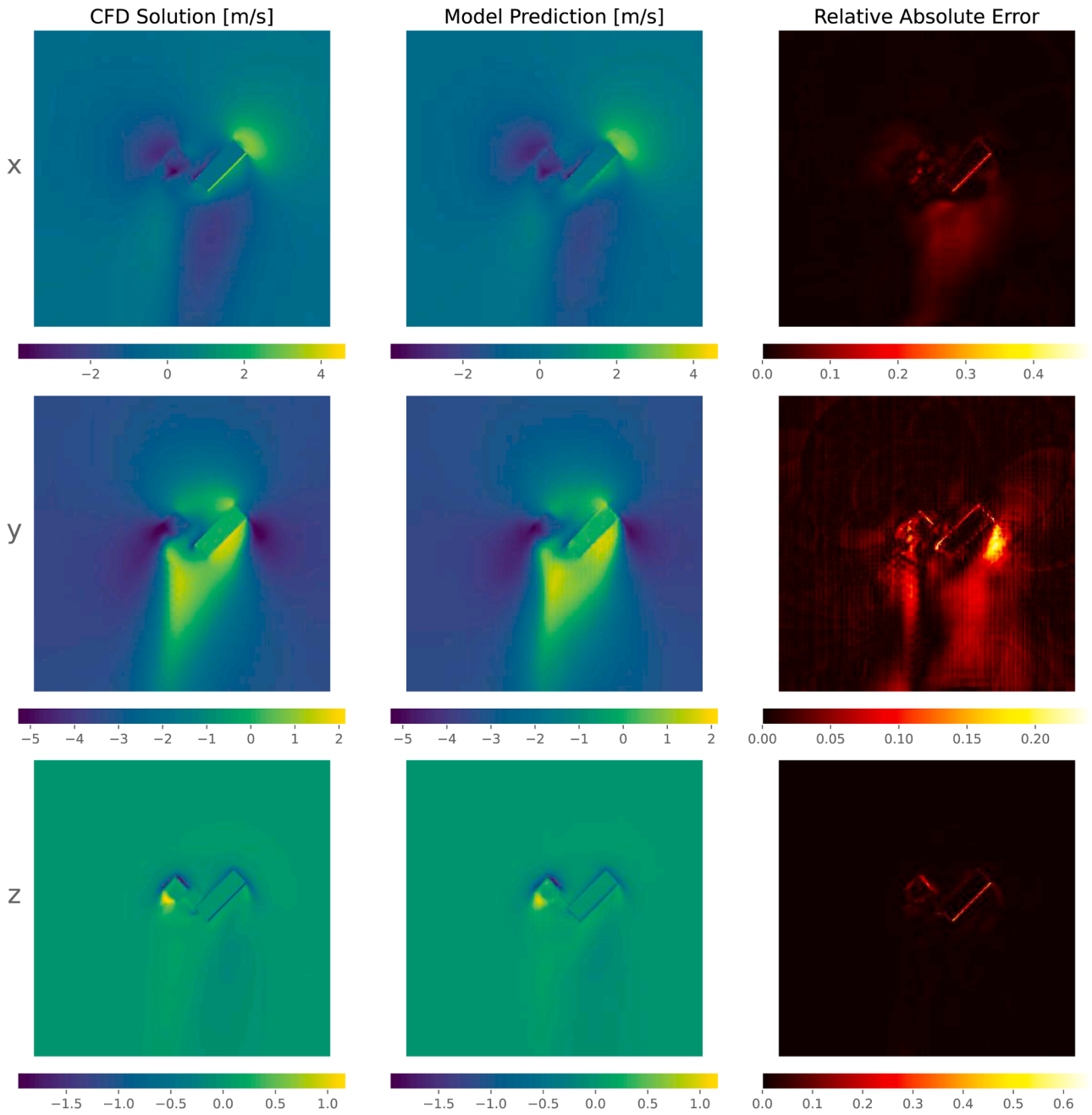


Fig. 4. Comparison of the model prediction (centre) with the post-processed CFD solution (left), showing wind flow across XY planes at a pedestrian-level height of 2 m. The right column illustrates the relative error between the model and the CFD reference data.

GeLU non-linearities to enhance the model's representational capacity. During each mixing operation, the first mixing layer expands the respective latent dimension by a factor of f prior to the non-linearity being applied, the second then reduces it back to its original size maintaining the dimensions of the input. Both N and f are hyper-parameters of the model. Skip connections are incorporated at each mixing step to improve model trainability and preserve information as it passes through the network.

2.4.3. Patch reconstruction

In the final stage, each feature vector in the latent tensor is expanded to a size of $(P \times P \times C)$ using a shared linear transformation. This produces a tensor of size $H/PxW/PxCP^2$ which is then reshaped to match the spatial dimensions of the original image. Finally, a 1×1 convolution is applied to reduce the embedding dimension C to the desired number of output channels. For our application, the output tensor has three channels, corresponding to the x, y, and z velocity components of the predicted flow.

2.5. Model training and optimisation

Of the 3920 data samples generated, 70 % were allocated for training, while two-thirds of the remaining samples used for validation. The final portion was reserved for testing to assess the model's performance on unseen data. This data split provides sufficient training samples while retaining enough validation data to optimise model parameters and prevent overfitting. Model training was conducted on a single NVIDIA A100 GPU, running for 20 epochs and requiring approximately three hours to complete.

The model was optimised using the Adam optimiser [59], paired

Table 2
Search space for hyperparameter optimisation.

Hyperparameter	Values
Model depth (Layers)	6, 10, 16, 20
Local mixing region	0, 3, 5, 7, 9, 11
Hidden layer expansion factor	4, 8
Embedding patch size	2, 4, 8, 16
Latent embedding size	128, 256

with a learning rate scheduler that reduced the learning rate at regular intervals. Specifically, the learning rate was decreased every five epochs with a decay factor (γ) of 0.3. The batch size was set to 1 to accommodate the computational complexity of the model and the data.

The loss function used during training was Mean Squared Error (MSE) defined in 2, which was chosen to minimise the difference between the predicted and actual flow field values. Key hyperparameters were determined using the open-source Optuna optimisation framework [60], resulting in an optimal initial learning rate of 0.0001, 16 sequential mixing layers, an embedding dimension of 256, a patch size of 4, an MLP hidden layer expansion factor of 8, and a neighbourhood size of 3. This configuration was chosen to balance model performance and computational efficiency.

$$MSE(y,\ \hat{y}) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{1}$$

Where y_i and $\hat{y}_i$ denote the processed CFD solution and the model prediction for the i^{th}sample and N gives the total number of samples in the evaluation.

3. Model evaluation

We evaluate the model's performance in the context of early-stage design usability. First, we present an example of the model's predictions compared to the ground truth, qualitatively assessing whether the relevant flow features are captured. We also highlight instances where the model fails to accurately represent the underlying physics. We then examine the impact of model structure, training data preparation, and data quantity on its ability to learn the desired mappings effectively. Next, we conduct an ablation study to assess the contribution of individual model components. We then analyse latent representations by inspecting outputs from intermediate layers, providing insight into how the model learns and what it prioritises. Finally, to further support our evaluation, we compare our model's performance to a UNet trained on the same dataset.

To assess the performance of the proposed model, we employ four key evaluation metrics: Mean Squared Error (MSE), Mean Relative Error (MRE), Structural Similarity Index (SSIM) [61], and Peak Signal-to-Noise Ratio (PSNR) [62], equations (1) – (4) respectively. MSE quantifies the average squared difference between predicted and ground truth values, providing a direct measure of numerical accuracy. However, as MSE does not account for relative differences across varying magnitudes, MRE is also used to normalise errors, offering a more interpretable measure of deviation in a human context. While these metrics evaluate numerical accuracy, SSIM assesses perceptual similarity by considering luminance, contrast, and structural information, aligning more closely with human visual perception than purely numerical metrics. This is particularly valuable during the exploratory phases of early-stage urban design, where qualitative assessments play a key role in guiding decisions and complimenting quantitative analyses. SSIM is a dimensionless metric ranging from -1 to 1, with higher values indicating greater similarity between the predicted and reference images. PSNR, given in decibels, measures the ratio between the maximum possible pixel intensity and the noise introduced during prediction with higher values indicating a better-quality reconstruction, making it useful for detecting blurring and loss of detail. Together, these metrics provide a comprehensive evaluation of the models' ability to generate both numerically precise and visually coherent flow predictions.

$$MRE(y,\ \hat{y}) = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{2}$$

$$SSIM(y,\ \hat{y}) = \frac{\left(2\mu_y\mu_{\hat{y}} + C_1\right)\left(2\sigma_{y\hat{y}} + C_2\right)}{\left(\mu_y^2 + \mu_{\hat{y}}^2 + C_1\right)\left(\sigma_y^2 + \sigma_{\hat{y}}^2 + C_2\right)} \tag{3}$$

$$PSNR(y,\ \hat{y}) = 10\log_{10}\left(\frac{Max(y)^2}{MSE(y,\ \hat{y})}\right) \tag{4}$$

Where y_i and $\hat{y}_i$ denote the processed CFD solution and the model prediction for the i^{th}sample, respectively, and N represents the total number of samples in the evaluation. The sample means of the CFD solutions and model predictions are given by μ_y, $\mu_{\hat{y}}$, respectively. The corresponding sample variances are σ_y^2, $\sigma_{\hat{y}}^2$ and the sample covariance is denoted by $\sigma_{y\hat{y}}$. The constants $C_1 = (k_1 L)^2$ and $C_2 = (k_2 L)^2$ are used for stability in the case of a weak denominator, where L is the dynamic range of the pixel-values and $k_1 = 0.01$, $k_2 = 0.03$.

3.1. Qualitative evaluation

Fig. 4 compares a sample model prediction alongside the post-processed CFD simulation result. The building on the right measures 20 m x 60 m with a hight of 40 m and is positioned directly the ground. The building on the left has a footprint of 20 m x 20 m and a height of 80 m. It features a lift-up design, with its base elevated 3 m above the ground, allowing wind to flow beneath the structure. The incoming wind approaches from the top of the image. Starting with the x-direction velocity component, the model correctly predicts the lateral deflection of the wind at the leading edge of both buildings, where the flow accelerates around the corners. Additionally, we see elevated lateral velocity at the left rear edge of the rightmost building as the wind is directed away from the structure. In the wake region of both buildings, the model accurately captures the flow reversal within the recirculation zone.

For the y-direction velocity component, the model effectively captures the shape of the recirculation region and the reversal in flow direction on the leading edges of the buildings, where a standing vortex typically forms.

Finally, in the z-direction velocity component, the model clearly captures the downdraft on the windward faces of both buildings, as well as the up draft from the wind rising after passing through the lift-up area beneath the lefthand building.

Certain features in the processed CFD solutions exhibit anomalous behaviour, such as a narrow strip of artificially elevated velocity along the right edge of the rightmost building in the x-direction velocity field. This artifact can be attributed to the mesh being highly refined near the building surfaces, many small cells with varying velocity values exist in these regions. When the slice is rendered, each mesh cell's value is mapped directly to the image without blending or averaging over pixels. Consequently, small-scale features such as thin regions with unexpected velocity values can appear as sharp lines or outlines in the image. These artifacts are not physical but arise from the combination of mesh refinement near boundaries and the direct visualisation of cell data without spatial smoothing. Such artifacts can be effectively mitigated through masking or smoothing techniques applied prior to data usage in training or further analysis.

3.2. Influence of model architecture and training approach

To find the optimal model we conducted a hyperparameter search using the Optuna search algorithm. The defined search space given in

Table 3
Summary of experimental results for different model configurations and training conditions. For each experiment the resulting train and test losses (MSE) are reported.

Layers	Parameters (Millions)	Train loss (MSE)	Test loss (MSE)
6	10.5	0.000127	0.0001289
10	16.8	0.000112	0.000115
16	26.3	0.000102	0.000109
20	32.6	0.000138	0.000620

Dataset size	Train loss (MSE)	Test loss (MSE)
30 %	0.000273	0.000342
50 %	0.000163	0.000201
70 %	0.000102	0.000109
100 %	0.000052	0.000076

Local mix filter size	Parameters (Millions)	Train loss (MSE)	Test loss (MSE)
0	5.0	0.000165	0.000200
3	16.8	0.000112	0.000115
5	37.8	0.000122	0.000133
7	69.3	0.000130	0.000140
11	163.6	0.000167	0.000180

the Table 2. We find the optimal configuration employs 16 layers with a local mixing region of 3 × 3, an expansion factor of 8. The optimal embedding layer divides the input into 4 × 4 patches and transforms them into 1 × 1 × 256 latent vectors. Among the key hyperparameters we further examine the effect that model depth, training set size and the size of the local mixing region has on the accuracy of the model. In these analyses, all other hyperparameters are held constant at their optimal values as determined by the Optuna search, except where noted. Specifically, for the local mixing region analysis, we used a shallower model with 10 layers due to hardware memory constraints. This deviation is noted to ensure clarity and reproducibility.

The results presented in Table 3 show that while increasing model depth initially improves performance, beyond a certain point, test loss begins to rise significantly. This suggests two possible explanations: either the model is overfitting, memorising training examples rather than generalising effectively to unseen data, or it has become too large relative to the available training data, limiting its ability to learn robust patterns from a finite number of examples.

To further investigate the role of data availability in model performance, we conducted additional experiments by training models on 30 %, 50 %, 70 %, and 100 % of the combined test and validation datasets. The results indicate that increasing the training set size leads to substantial performance gains - specifically, doubling the dataset size approximately halves the training loss and reduces test loss by a factor of three. This suggests that the model remains data-limited, meaning that further increases in dataset size could lead to continued improvements in generalisation. The interplay between model depth and data availability highlights the importance of balancing model complexity with sufficient training examples to avoid overfitting while maximising the predictive capability of the model given the time and financial burden of producing sufficient CFD solutions.

Experiments with different local mixing filter sizes reveal that while training loss remains consistently low across all configurations, a filter size of 3 produces the best overall performance on the test set, this result is compounded by the reduced parameter count for the smaller filter size making the model quicker to train and use at inference time, considerations that are especially important for rapid iteration and design optimisation studies. We further examine the performance of the model employing a 3 × 3 filter on two additional evaluation metrics SSIM and PSNR which show a marginal increase in the SSIM score and a gain of up to 5 % for PSNR. This suggests that a 3 × 3 receptive field provides an optimal balance, allowing the model to capture meaningful spatial relationships without introducing excess noise or over-smoothing. Omitting the local mixing step or using smaller filter sizes may limit the model's ability to extract sufficient spatial context, while larger filters dilute relevant information, introducing noise from regions further away that have less influence on the target area inhibiting the model's ability to generalise. The optimal filter size may correspond to a characteristic length scale of the input geometry and resulting flow patterns. Further research is needed to investigate this relationship and determine whether similar trends hold across different geometries and flow conditions.

We experimented with two methods for embedding 3D building details into the training data, as illustrated in Fig. 5. In both approaches, we define an arbitrary datum at 100 m above the ground, using it to normalise all values in the input tensor. In the first method, the input tensor consists of two channels: the first encodes the distance from the ground to the top of the building, while the second records the distance from the ground to the bottom. For locations without a structure, we assign a maximum normalised value of 1. In the second method, the second channel instead represents the distance from the datum (100 m) to the bottom of the building, while the first channel remains unchanged. Again, locations without buildings are assigned a maximum value of 1. Our experiments show that models trained using the first method consistently outperform those using the second with the first method yielding a reduction of approximately 60 % for both the train and test losses. This is likely because the first method creates a clearer distinction between locations with and without structures, reducing ambiguity and improving the model's ability to learn meaningful features of the geometry.

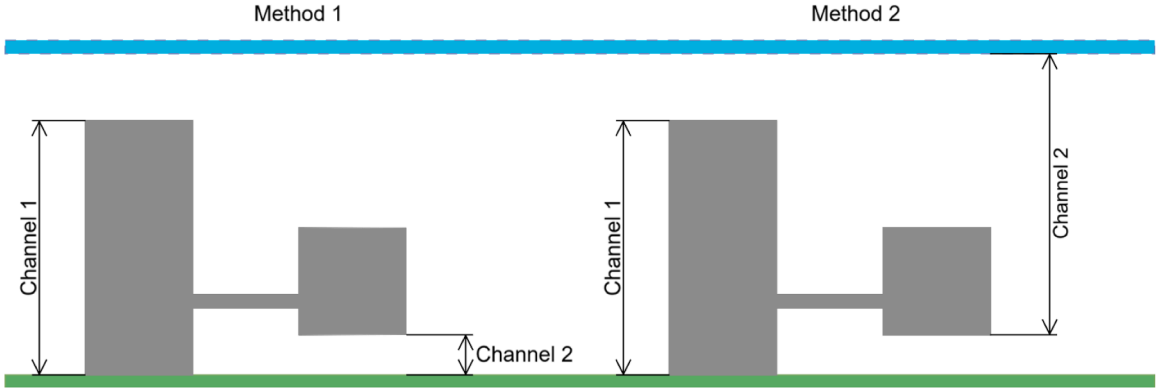


Fig. 5. Illustration of the two methods used to embed 3D building details into the training data. Method 1 encodes the height from the ground to the building top in the first channel and the height from the ground to the bottom in the second. Method 2 replaces the second channel with the height from the datum to the building bottom.

Table 4
Results of the ablation study, showing the impact of removing individual model components on training time, model complexity (number of parameters), and performance (training and test loss).

Model	Parameters (Millions)	Training time	Training loss (MSE)	Test loss (MSE)
MLP mixer	26.3	3 H	5.19×10^{-5}	7.58×10^{-5}
No channel mixing	26.3	4 H	6.42×10^{-5}	9.56×10^{-5}
No spatial mixing	26.3	2 H	1.02×10^{-4}	1.35×10^{-4}
No local mixing	26.3	6 H	1.35×10^{-4}	1.66×10^{-4}

3.3. Ablation study: assessing component contributions

The structure of the model significantly affects its ability to learn the mapping from the training data to the desired output. To better understand the role of individual components, we systematically remove them and evaluate their impact on training efficiency and predictive performance. To ensure that any change in performance is due to the removal of a specific model component rather than a reduction in the number of free parameters, we adjust the expansion of hidden layers in the remaining components to maintain parity across all experiments. We assess these modifications based on training time and both training and test loss, as detailed in Table 4.

Our results indicate that removing the local mixing component doubles the training time, as the remaining mixing layers, which utilise multi-layer perceptrons, expand to compensate and maintain the overall parameter count. This expansion leads to large matrix multiplications, increasing computational demands. Furthermore, the removal of the local mixing function results in the greatest increase in both training and test loss, suggesting it plays a crucial role in the model's performance by capturing and reinforcing the models understanding of localised fine-grained interactions such as boundary effects.

Similarly, removing the spatial mixing component severely degrades accuracy. Spatial mixing facilitates the transmission of information across spatial dimensions, and while its removal marginally reduces training time, the significant drop in model accuracy outweighs this benefit inhibiting the model's ability to propagate large-scale information about global flow structures.

Meanwhile, eliminating the channel mixing function, which enables information sharing across input channels, slightly worsens model performance. This, combined with an increase in training time, makes it an unsatisfactory modification.

These findings suggest that each component of the proposed model

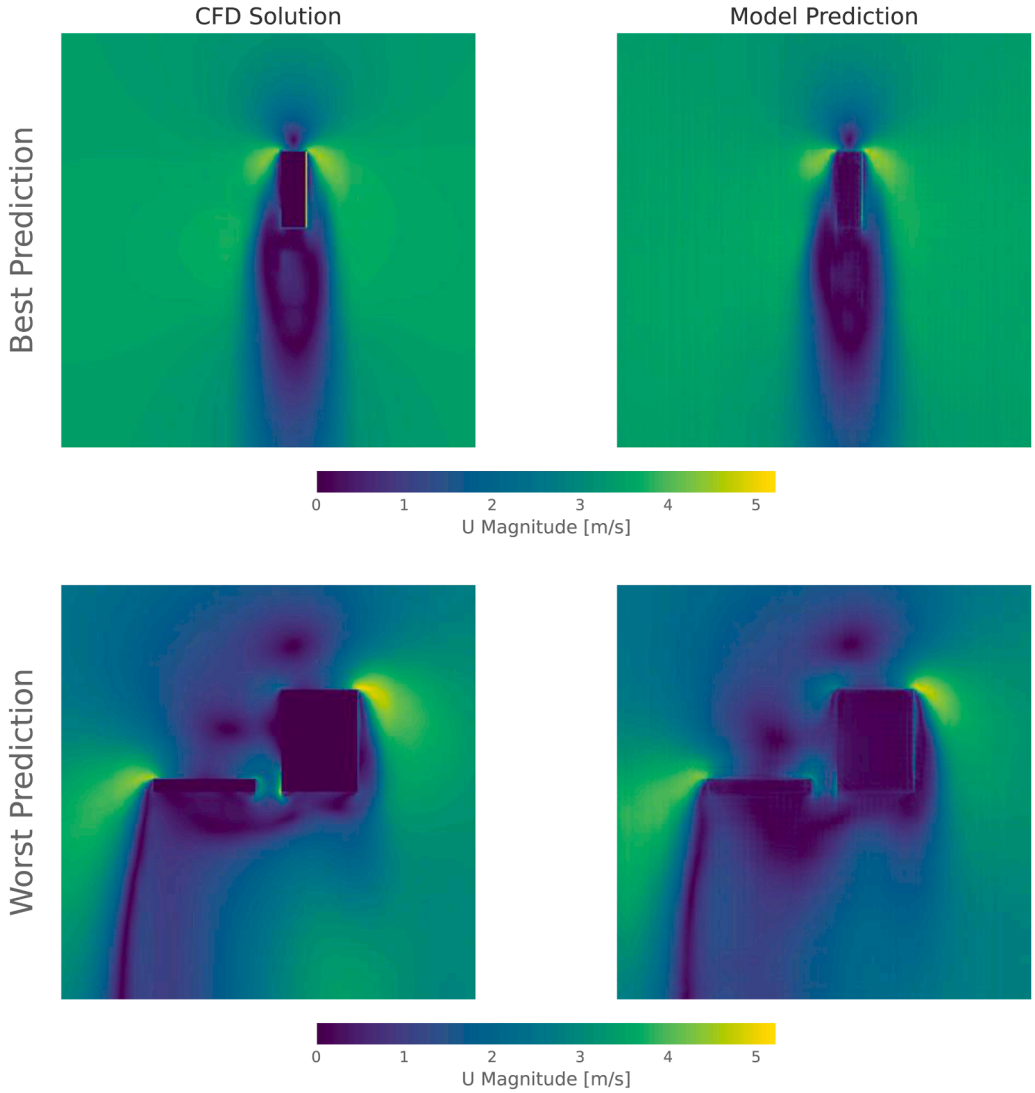


Fig. 6. Velocity magnitude fields at $z = 2$ m for the best and worst predictions in the test set. Each row compares the CFD solution (left) with the model prediction (right). Top row: Best prediction Bottom row: Worst prediction.

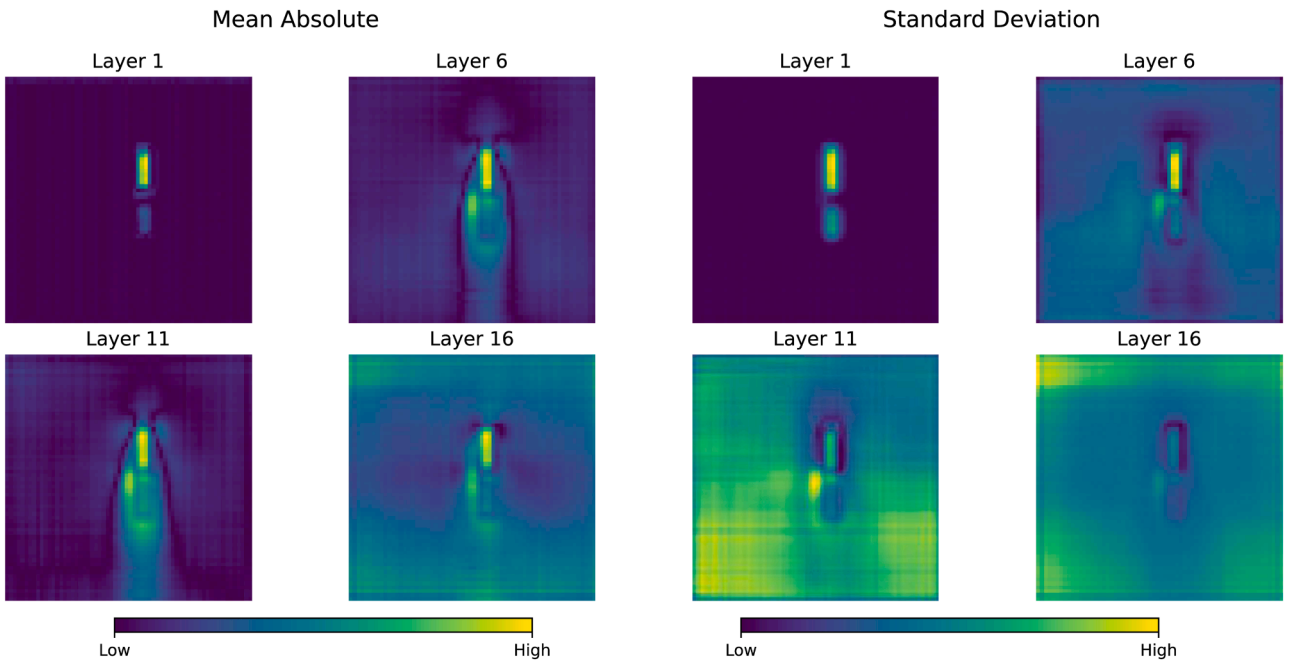


Fig. 7. Visualisation of latent activations in different layers of the model for the best prediction. The left column shows the mean absolute activations at each spatial location, while the right column displays the standard deviation of activations across the channel dimension. As depth increases, the activation patterns become more diffuse, suggesting broader feature abstraction at deeper layers.

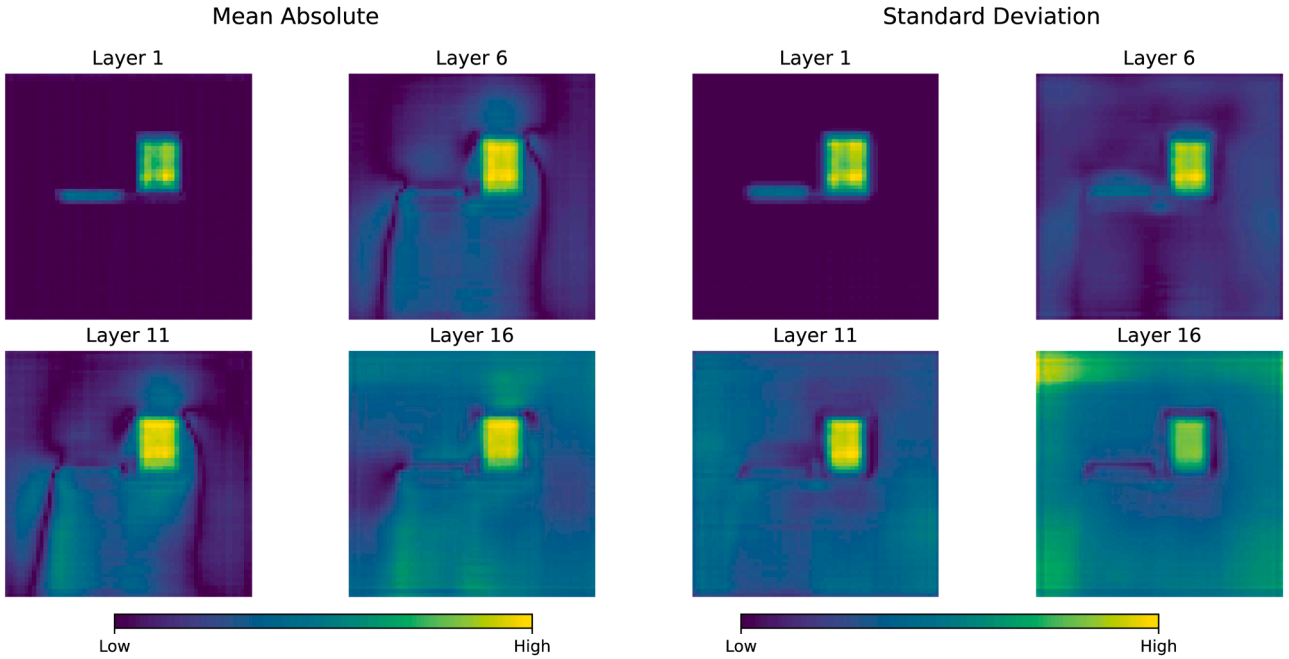


Fig. 8. Visualisation of latent activations in different layers of the model for the worst prediction. The left column shows the mean absolute activations at each spatial location, while the right column displays the standard deviation of activations across the channel dimension. As depth increases, the activation patterns become more diffuse, suggesting broader feature abstraction at deeper layers.

contributes to its ability to learn the spatially dependent wind patterns efficiently, with local and global spatial mixing playing particularly vital roles in maintaining both accuracy and computational efficiency.

3.4. Latent representation analysis

As data flows through the model, it is progressively transformed by the individual layers. During this transformation, the channel dimension is significantly expanded, enabling the model to encode key information about both the structure of the flow and the geometry, while also generating abstract representations of these features. By inspecting the internal workings of the model, we can gain insights into how it learns and identify where and how it extracts meaningful features from the geometry to build the resulting flow patterns.

With 256 channels, direct visualisation is impractical. To address this, we compute the mean absolute value across the channel dimension

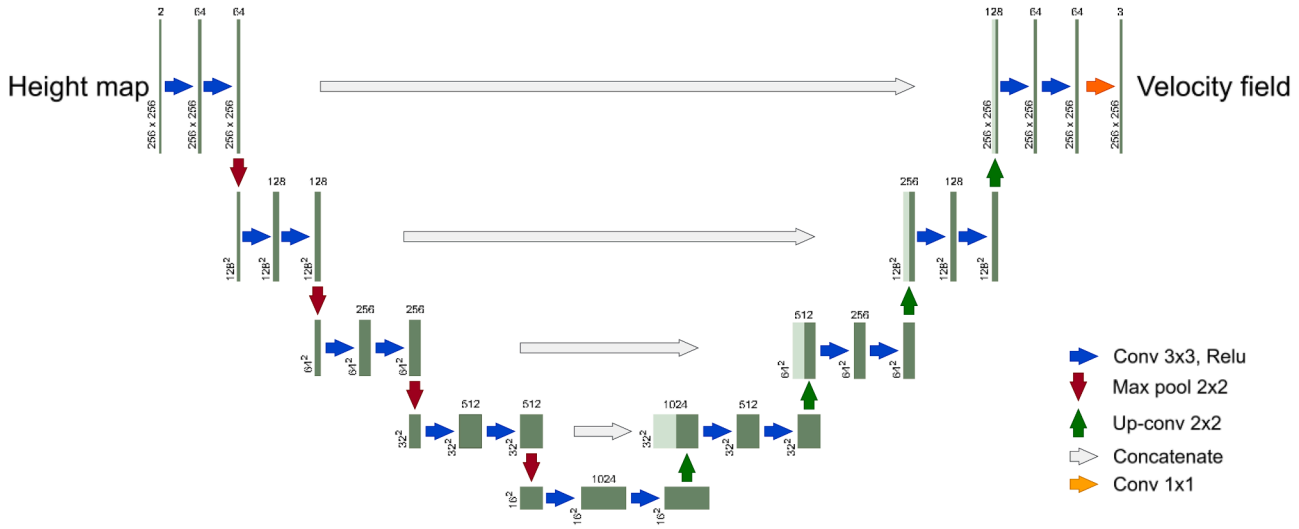


Fig. 9. Schematic representation of the UNet models adapted from [63].

Table 5
Training parameters for the UNet instances and loss achieved during training.

Model	Learning rate	Scheduler step	Epochs	Train loss (MSE)
UNet-a	0.0001	5	20	0.000152
UNet-b	0.001	5	20	0.000146
UNet-c	0.001	20	80	0.000083
UNet-d	0.001	40	160	0.000078

at each spatial point, pooling the activations for clearer interpretation. This provides a holistic view of the learned representations and allows us to identify activation hotspots. Although this method may overlook channel-dependent features, which could encode information such as edge detection or gradient identification, it provides sufficient information to reveal the model's hierarchical development of flow patterns. Additionally, we calculate the standard deviation of the activations as a measure of the spread, offering further insight into the variability of learned features across the network.

Fig. 6 shows the velocity fields for the model's best and worst predictions from the test set, each compared against the ground truth CFD providing a physical interpretation of model performance. For the best prediction (top row), the windward building is 20 m wide, 60 m deep, and 60 m high, situated directly on the ground. The building directly behind it is 20 m wide, 40 m deep, and 20 m high, featuring a lifted-up design. The model accurately captures the key flow features, including the wake structure and recirculation zones, with good agreement in both

Table 6
Quantitative comparison of model performance across multiple evaluation metrics. The table reports the mean and standard deviation for Mean Squared Error (MSE), Mean Relative Error (MRE), Structural Similarity Index Measure (SSIM), and Peak Signal-to-Noise Ratio (PSNR) for each model.

Model	MSE		MRE		SSIM		PSNR	
	Mean	Std dev	Mean	Std dev	Mean	Std dev	Mean	Std dev
MLP Mixer	0.000076	0.000066	0.0097	0.0035	0.991	0.0032	42.35	2.73
UNet-a	0.000196	0.000213	0.0157	0.0062	0.988	0.0047	38.63	3.54
UNet-b	0.000204	0.000223	0.0164	0.0065	0.988	0.0049	38.44	3.55
UNet-c	0.000204	0.000223	0.0157	0.0063	0.989	0.0049	38.57	3.81
UNet-d	0.000223	0.000248	0.0164	0.0067	0.989	0.0050	38.57	3.81

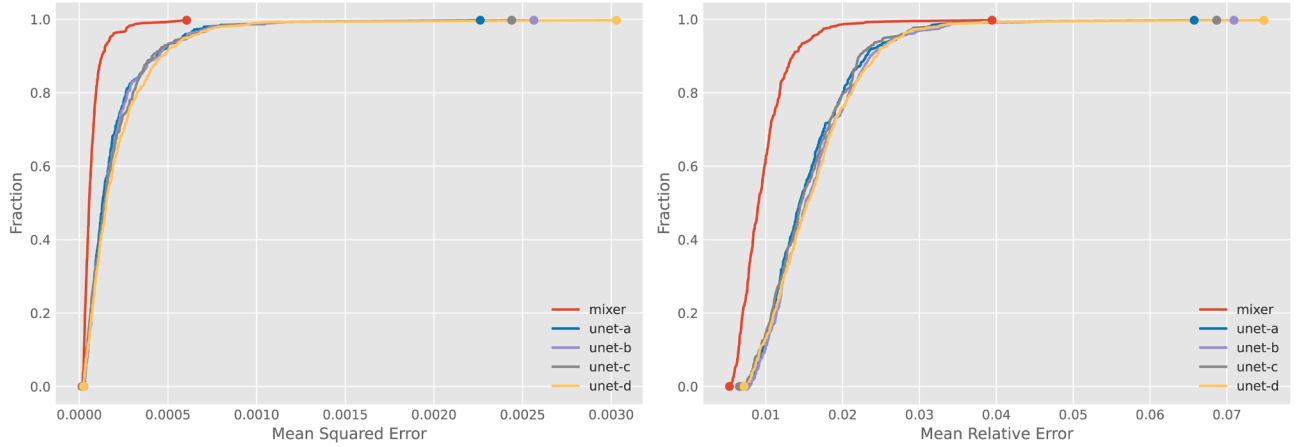


Fig. 10. Cumulative distribution functions of the MSE and MRE for the mixer and UNet variants.

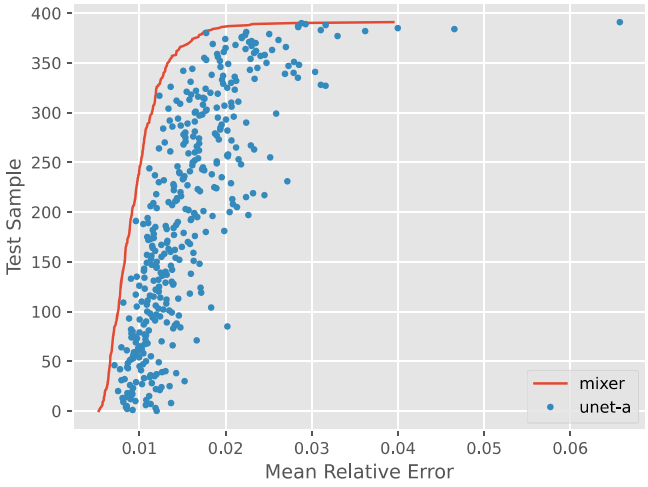


Fig. 11. Case-by-case mean relative error plot for the mixer model and the corresponding error resulting from predictions made my UNet-a.

shape and velocity gradients. In contrast, the worst prediction (bottom row) involves a more complex configuration: the left-hand building is 80 m wide, 10 m deep, and 100 m high, while the right-hand building is 60 m wide, 80 m deep, and 60 m high; both buildings rest directly on the ground. Here, the model struggles to replicate the intricate shear layers and wake interactions evident in the CFD solution. Notable discrepancies arise in the predicted recirculation zones and the intensity of velocity gradients around building corners and in the wake region.

Fig. 7 and Fig. 8 present the corresponding latent activations for these two cases. The best prediction (Fig. 7), the activations highlight clear spatial patterns that align with key flow features, suggesting a well-structured latent encoding. In the worst prediction (Fig. 8), the activations are more diffuse and less localised, reflecting the model's difficulty in capturing the complex interactions present in the flow field.

In both instances, the first layer appears to focus on identifying the geometry of the buildings. Across both cases, the high standard deviation among the activations corresponding to the buildings is maintained throughout the network, which could suggest that the model is learning to capture diverse contextual effects of the buildings on the resulting flow. As we move deeper into the model, clear boundaries begin to emerge in both the windward and leeward directions relative to the building. These regions correspond to areas of the flow with the largest gradients, which may indicate that the model is identifying key features associated with the building's influence on the flow pattern. By the 11th layer, the model shows heightened activation in the leeward flow regions, coupled with an increased standard deviation broadly across the image, suggesting a more refined understanding of how the building shapes the flow in these areas and the resulting effects for the wider flow.

In the final layer, the model shifts its focus to the entire flow field with a broader attention span and again, with elevated standard deviations. This indicates that the model is learning to apply more global features across the domain. The variability in these activations may reflect sensitivity to different aspects of the inputs from preceding layers, such as the location and shape of the features identified. Alternatively, this increased variability might suggest that the final layer is learning a more generalised response such as smoothing to mitigate the appearance of discontinuities in the predicted flow.

3.5. *Comparative study: performance against UNet*

To assess the relative performance of our proposed model and to better understand the strengths and weaknesses of our approach, we conduct a comparative study with a UNet architecture. The UNet model takes as input a $256 \times 256 \times 2$ tensor and produces a $256 \times 256 \times 3$ output. The network follows the classic UNet architecture, consisting of an encoder-decoder structure with skip connections. The encoder is composed of four convolutional blocks, each containing two sequential 3×3 convolutional layers with rectified linear unit activations, followed by a 2×2 max-pooling operation for downsampling. The bottleneck layer, located at the deepest part of the network, employs additional convolutional layers to extract high-level features. The decoder consists of four upsampling blocks, each using a transposed convolutional layer followed by concatenation with the corresponding encoder features and subsequent convolutional layers to refine the reconstruction. The final output is obtained via a 1×1 convolutional layer that maps the feature maps to the desired three output channels. A schematic representation of the model is shown in Fig. 9.

To ensure a fair comparison between our proposed model and the UNet architecture, we trained four instances of the same UNet model - denoted UNets, UNet-a, UNet-b, UNet-c and UNet-d. These variants share identical network architecture; the differences lie solely in their training configurations. As detailed in Table 5, the training regimes differ in learning rate, scheduler step size, and total training epochs, allowing us to explore the effect of these hyperparameters on model performance and identify the most competitive UNet configuration for comparison with our proposed model. Each model was trained using an Adam optimiser.

Table 6 summarises the performance of the mixer and UNet models across all evaluation metrics on the withheld test set. The MLP mixer model consistently outperforms all UNet variants achieving lower MSE and MRE values indicating more precise predictions. As well, the mixer scores higher on SSIM and PSNR, reflecting better overall image quality

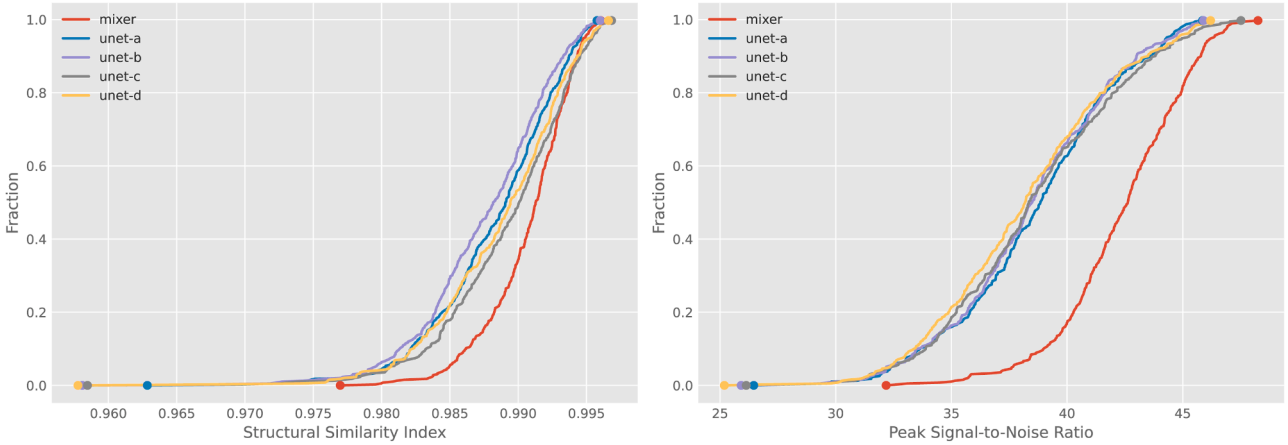


Fig. 12. Cumulative distribution functions of the SSIM and PSNR for the mixer and UNet variants.

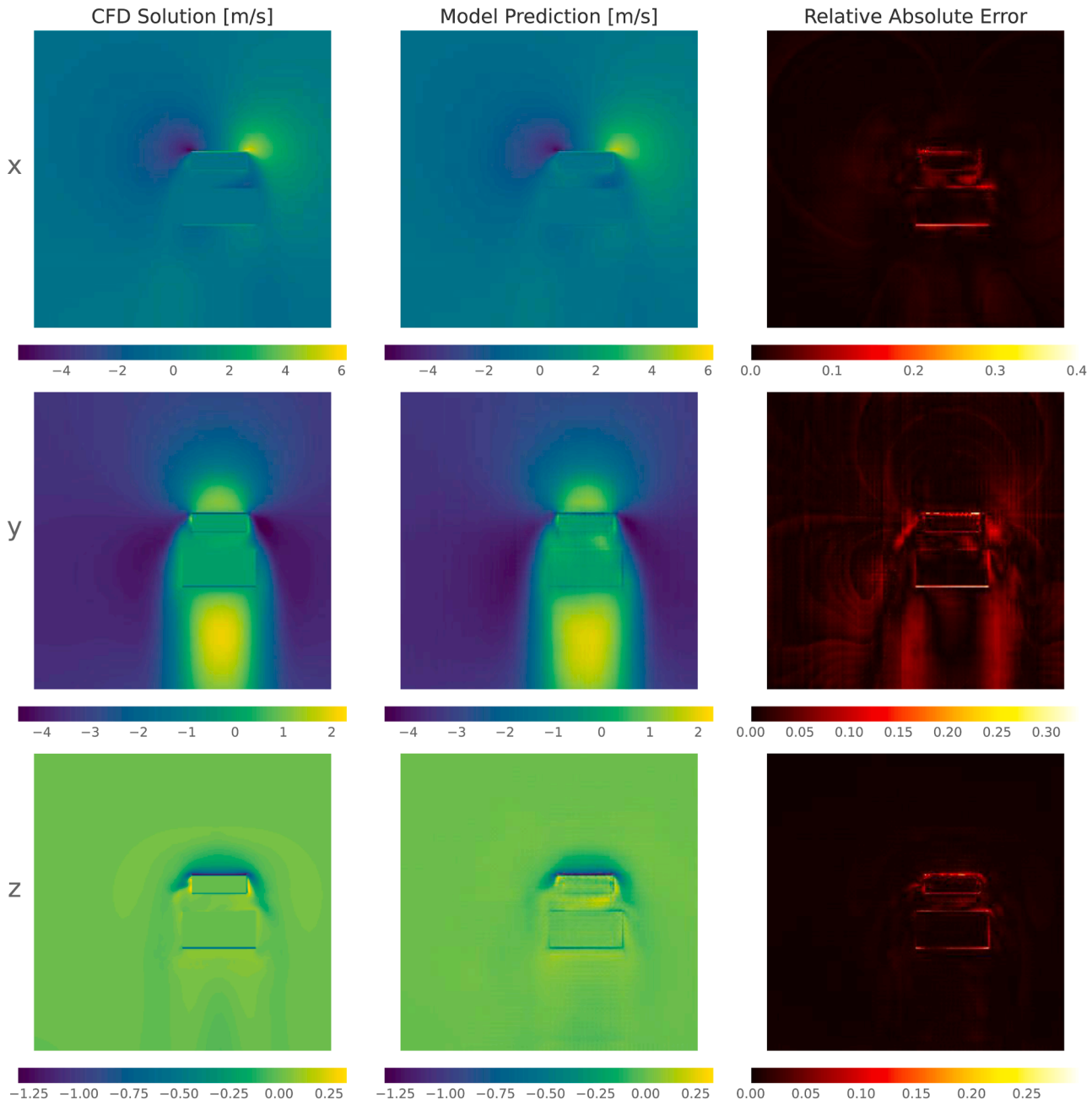


Fig. 13. A sample prediction produced by the mixer model, showing wind flow across XY planes at a pedestrian-level height of 2 m. The right column presents the relative error between the mixer model prediction (centre) and the CFD reference data (left).

and similarity. Furthermore, the mixer exhibits the lowest standard deviation across all metrics, demonstrating a more stable and reliable performance when making predictions on unseen geometries.

Fig. 10 presents the cumulative distribution functions (CDF) for MSE test loss and MRE, comparing the performance of the proposed mixer model and the UNet variants. While MSE and MRE serve as proxies for one another, MRE provides a more interpretable measure of error in a human context. The distributions highlight the mixer's superior ability to generalise to the unseen data, consistently maintaining a lower MSE and MRE across test samples. This is further supported by Fig. 11 which compares the MRE of matched samples for the mixer and UNet-a, the best performing UNet variant. Additionally, these plots reveal that UNet-d performs worst among the UNet models despite achieving the lowest training loss. This suggests that the extended training period may have led to overfitting, leading to poorer performance during test inference.

For both SSIM and PSNR the mixer generally outperforms the UNet models as depicted in Fig. 12, reinforcing its ability to generate both visually and numerically accurate results. The close grouping of the UNet models throughout most of the CDF plots indicates that training variations have minimal impact on the predictive performance, except in a limited number of cases.

Figs. 13–16 present sample predictions from the Mixer and UNet-a models, alongside the corresponding ground truth and error maps. While both models successfully capture the overall structure of the CFD solution, localised discrepancies are evident in regions with sharp velocity gradients, particularly in wake regions and around building edges.

In each sample, localised errors tend to be higher in the UNet model, suggesting greater difficulty in resolving fine-scale flow structures. Additionally, artifacts appear more frequently in the UNet's predictions, especially in the downstream regions, where flow complexity increases.

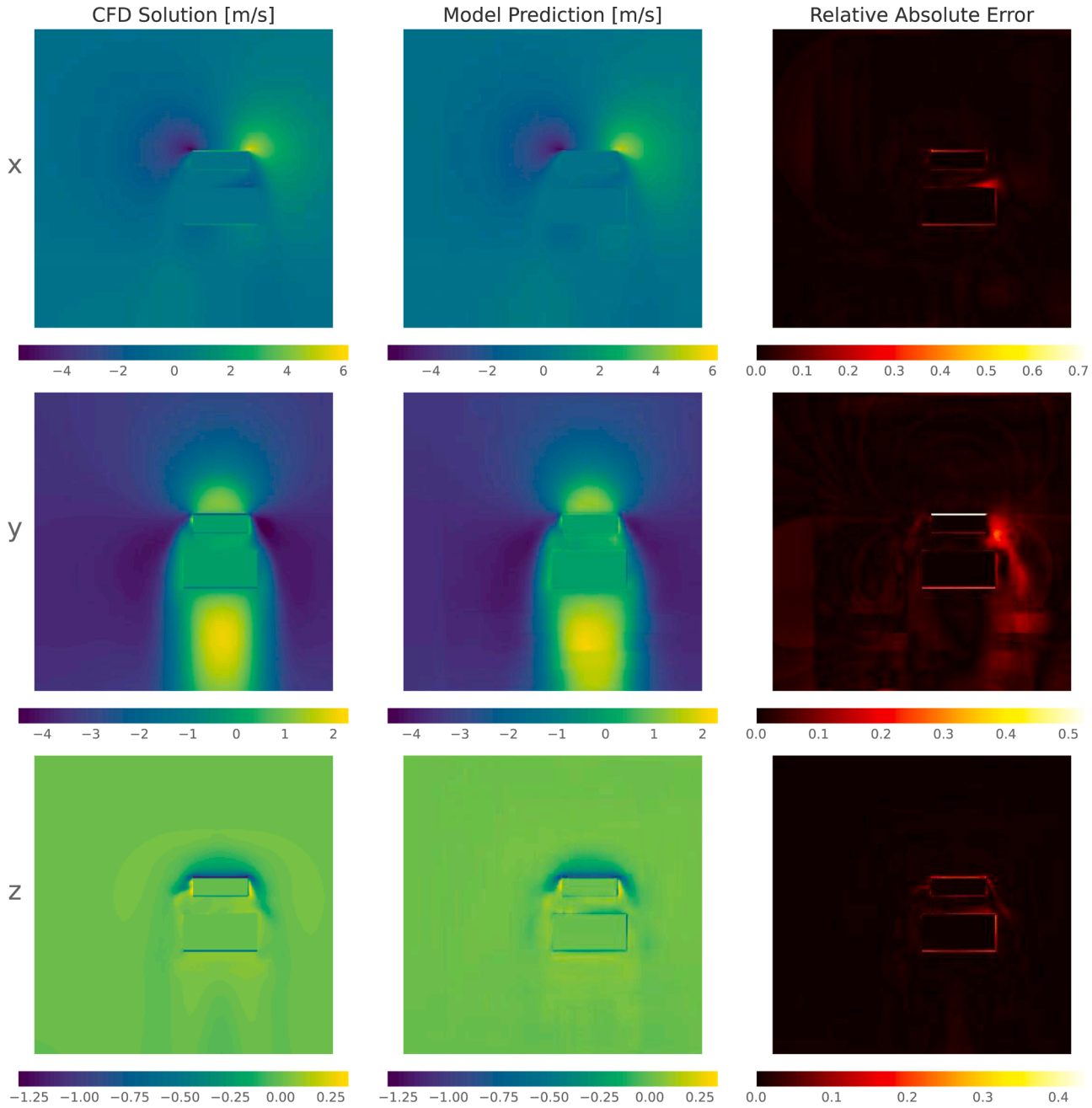


Fig. 14. A sample prediction produced by UNet-a, showing wind flow across XY planes at a pedestrian-level height of 2 m. The right column presents the relative error between the mixer model prediction (centre) and the CFD reference data (left).

This effect is particularly pronounced in Fig. 12, where the UNet model exhibits a catastrophic failure, entirely misrepresenting the resulting flow, which could be due to the limited receptive field and inductive bias towards local features leading to errors accumulating downstream where the flow interaction depends heavily on the upstream conditions. These observations highlight the importance of global and local information propagation in the mixer model's ability to generalise.

Fig. 17 shows the training loss curves for each model. In the early stages, loss decreases rapidly for all models, with the UNet variants exhibiting particularly fast convergence. However, about halfway through training, the UNet models begin to plateau, while the mixer model continues to improve, ultimately achieving the lowest test loss. The negative gradient observed toward the end of the mixer's loss curve suggests that further training might continue to reduce loss, albeit with an increased risk of overfitting. UNet-c and UNet-d show similar training trajectories across most of the curve indicating that the extended training duration of UNet-d has little effect on improving the model performance. Overall, the mixer exhibits a greater capacity to learn from the provided training data making. Its efficiency makes it a suitable candidate where data availability may be a limiting factor due to resource constraints such as early-stage design workflows, where rapid and reliable predictions are needed without the luxury of an extensive dataset.

When considering these models for an early-stage design tool, trade-offs among model size, training time, and inference time become crucial. Training times for the UNet models range from 1.5 to 9 h, depending on specific training parameters, whereas the mixer model required 3 h to train on our hardware. Although the mixer model has a reduced parameter count (26.3 M) compared to the UNet models (31 M), its inference time is slower—0.022 s per image versus 0.004 s for the UNet.

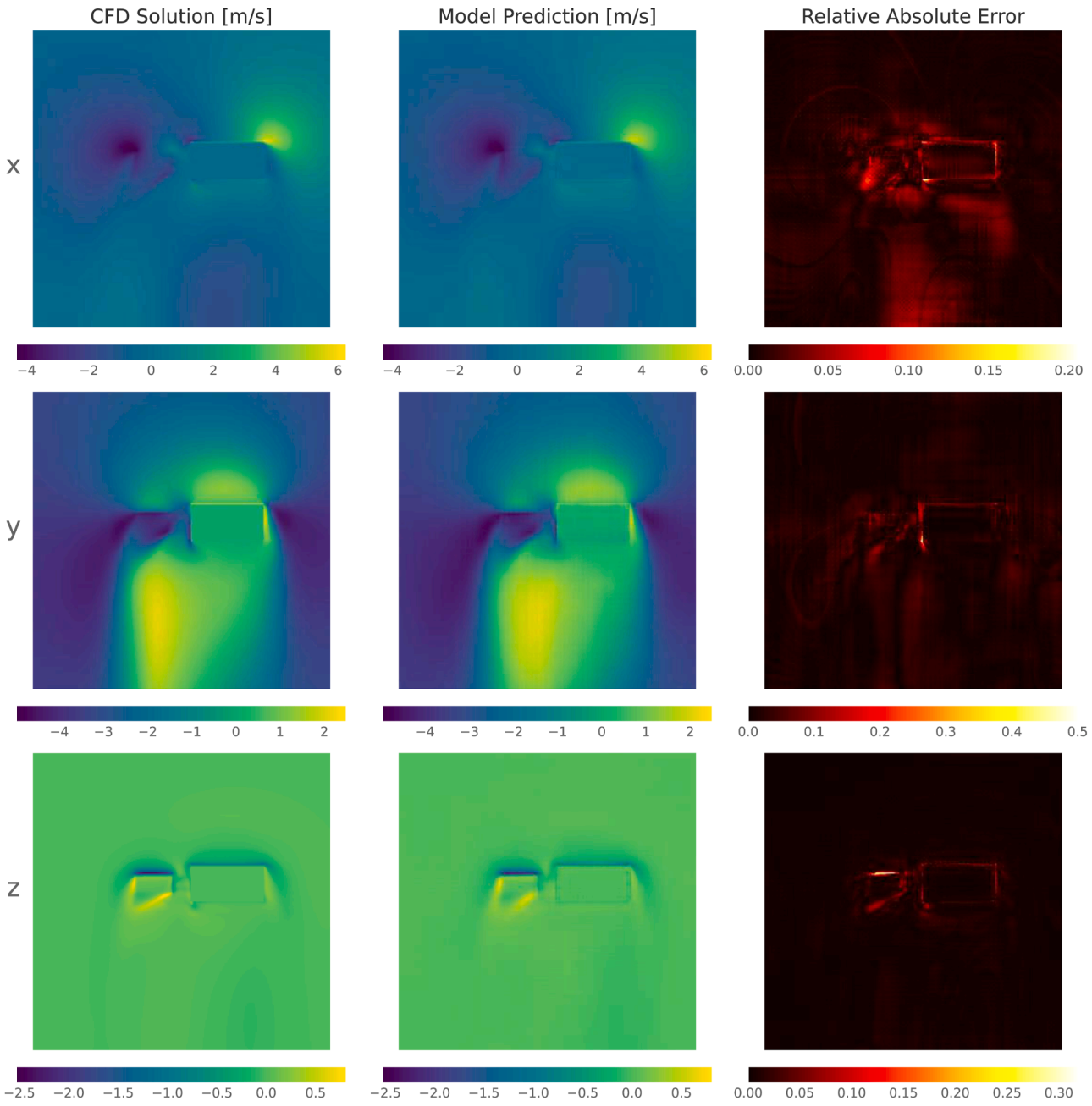


Fig. 15. A sample prediction produced by the mixer model, showing wind flow across XY planes at a pedestrian-level height of 2 m. The right column presents the relative error between the mixer model prediction (centre) and the CFD reference data (left).

To further contextualise this comparison, we evaluated the CFD solver's performance across a range of CPU counts (1, 2, 4, 8, 16, and 32). As shown in Fig. 18, compute time decreased approximately linearly with increasing CPU count, indicating efficient parallelisation. Despite this, the mixer model achieved a 3 to 4 order-of-magnitude speed advantage when executed on a CPU, and a 4 to 5 order-of-magnitude advantage when GPU-accelerated. These results highlight the transformative potential of learned surrogate models in dramatically reducing computation time while maintaining acceptable accuracy, thereby enabling fast and iterative early-stage design workflows.

4. Conclusions

This study detailed the development and evaluation of an MLP-mixer-based model for the application of predicting pedestrian-level wind conditions in urban environments. We introduced a method for encoding the highest and lowest points of structures into the training data. This Enhanced the model's ability to infer wind conditions around architectural features such as lift-up designs and skyways while maintaining a compact model size and fast inference speeds.

Our evaluation shows that the MLP-mixer model achieves high accuracy and structural similarity, with improved performance across multiple evaluation metrics, including MSE, MRE, SSIM, and PSNR when compared with a UNet architecture. Overall, the MLP-mixer exhibits greater stability and generalisation, particularly in regions with sharp velocity gradients, such as wake regions and around building edges.

Furthermore, show that each component of the MLP-mixer architecture is integral to producing rapid, accurate urban wind predictions, progressively refining the learned representations through the

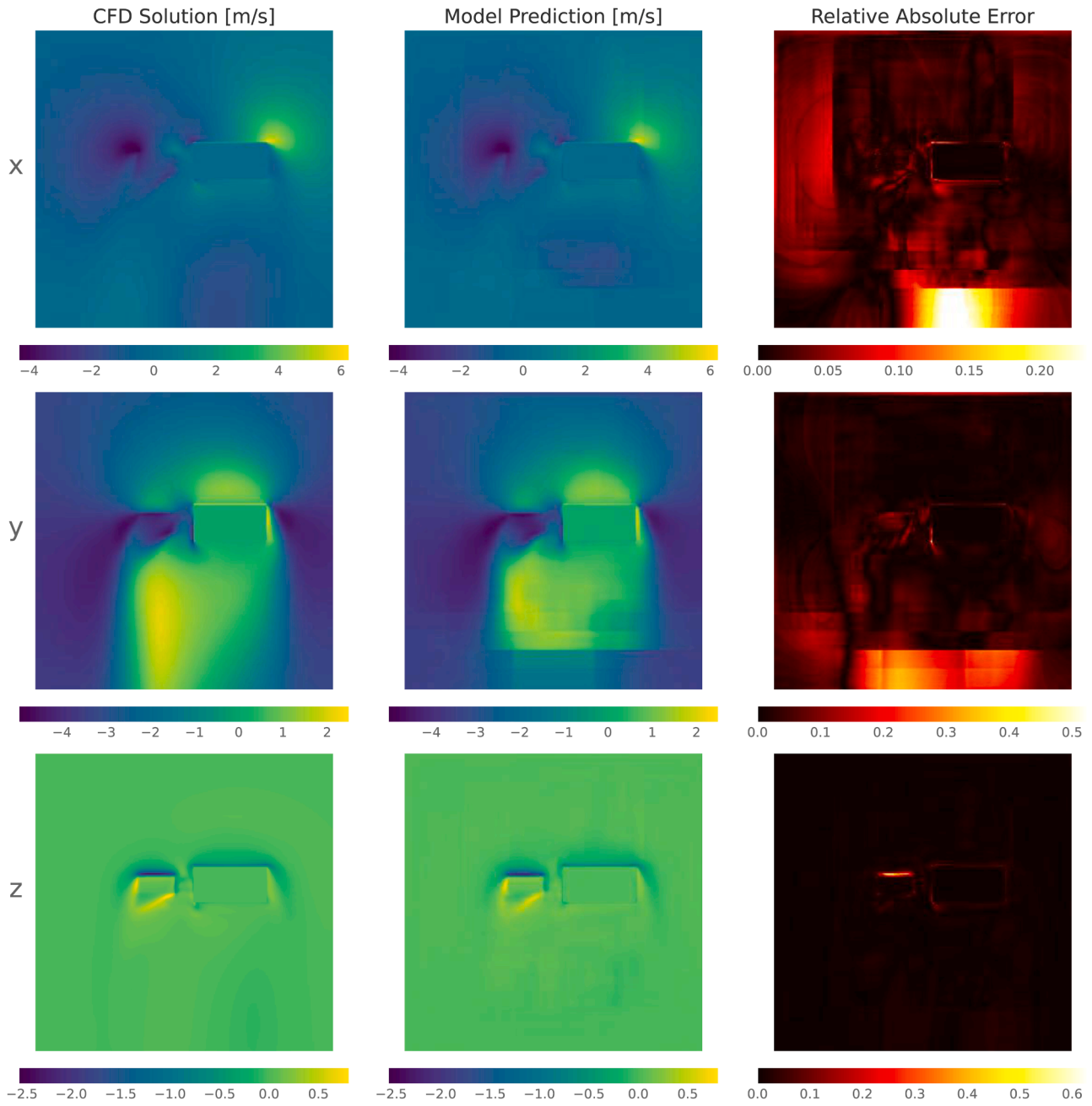


Fig. 16. A sample prediction produced by UNet-a, showing wind flow across XY planes at a pedestrian-level height of 2 m. The right column presents the relative error between the mixer model prediction (centre) and the CFD reference data (left).

network's hierarchical layers.

Several key insights emerge from the results:

- **Model Complexity and Data Availability**: Increasing model depth improves performance up to a certain point, beyond which test loss rises significantly, suggesting overfitting or excessive complexity relative to the available training data. Expanding the training dataset results in substantial performance gains, indicating that the model is data-limited rather than capacity-limited and could benefit from additional CFD-generated samples.
- **Local Mixing and Spatial Awareness**: The local mixing component plays a critical role in capturing fine-scale structures, with a neighbourhood size of 3 producing the best results. Removing this component significantly degrades performance and increases computational demands, highlighting its importance. Similarly, spatial mixing is essential for propagating global flow information, and its removal leads to severe accuracy losses.
- **Building Representation and Feature Learning**: The choice of height encoding has a direct impact on model performance, with a dual-channel representation distinguishing the top and bottom of structures proving most effective. This approach reduces ambiguity and enhances the model's ability to extract meaningful geometric features, improving predictive accuracy.
- **Latent Space Interpretation**: The mixer's early layers focus on identifying geometric boundaries, while deeper layers shift towards understanding large-scale flow structures. This hierarchical learning process contributes to the model's ability to generalise across different urban configurations.
- **Training Efficiency and Generalisation**: While the UNet variants converge quickly during early training, they plateau earlier than the

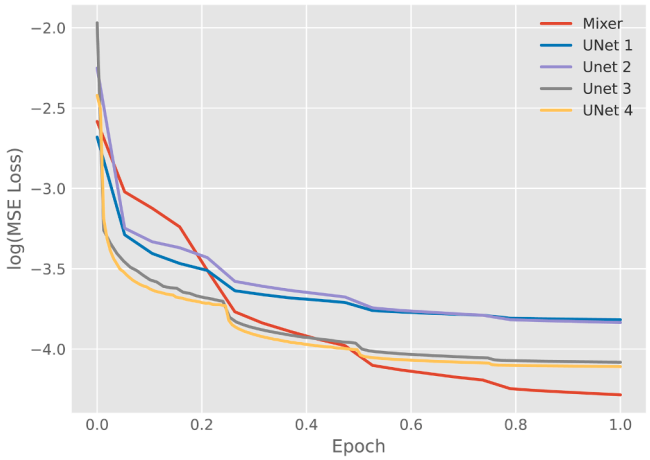


Fig. 17. Training loss curves for all models, with the x-axis representing the normalised epoch number and the y-axis showing the log of the training loss for finer detail. The normalisation allows for a direct comparison between models trained for different numbers of epochs.

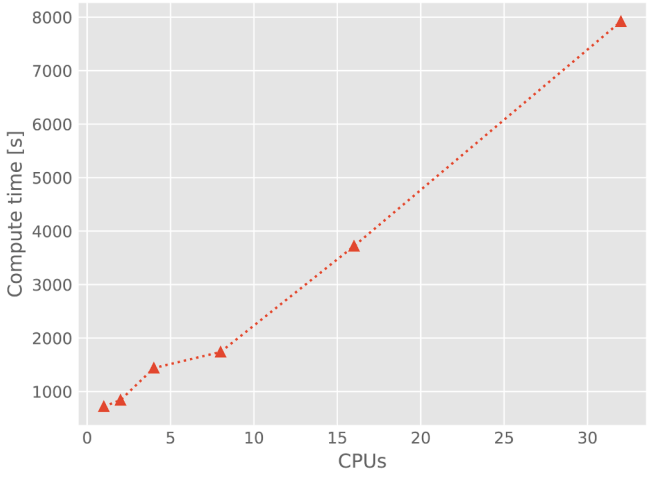


Fig. 18. Compute time (in seconds) for the CFD solver as a function of CPU count. The results demonstrate a near-linear scaling, with compute time increasing proportionally to the number of CPUs used.

Mixer, which continues improving throughout training and achieves the lowest test loss. This suggests that the mixer is capable of sustained learning and avoids premature convergence.

- **Trade-offs in Practical Deployment**: Although the mixer has fewer parameters than the UNet, its inference time is slower. However, at 0.022 s per image, it remains 3 to 5 orders of magnitude faster than CFD simulations depending on hardware configuration, making it a highly viable option for early-stage urban wind assessment where rapid feedback is essential.

The findings reinforce the potential of MLP-mixer architectures for urban wind flow prediction, particularly in applications where both accuracy and inference speed are required. The reduced inductive bias of the MLP-mixer allows for greater adaptability to varying urban configurations, making it a promising alternative to CNN-based approaches. However, the study has several limitations that should be addressed in future work.

First, the training data was restricted to simplified urban layouts consisting of only two buildings connected by a skyway. While this setup enables controlled experimentation, it does not capture the geometric complexity of real-world urban environments. For the model to the applicable to more realistic scenarios, it must be evaluated on block-level or city-scale morphologies, which exhibit a broader range of flow patterns and spatial interactions.

Second, the simulations were conducted under a neutral atmospheric stratification and with fixed inflow boundary conditions. The model was trained on data that assumes a consistent inflow profile across all samples, meaning it has effectively learned a mapping conditioned on a single flow configuration. Consequently, the model's generalisability to varying unsteady boundary conditions remains untested. Extending the model to handle such variability would require redesign of the model architecture to incorporate inflow boundary condition parameters explicitly and a substantial expansion of the training dataset to include a diverse set of inflow scenarios.

Funding sources

This research was supported by funding provided by Cranfield University and Nablaflow AS (https://nablaflow.io/).

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author(s) used ChatGPT in order to format and improve readability of the text. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the published article.

CRediT authorship contribution statement

Adam Clarke: Writing – original draft, Visualization, Validation, Software, Project administration, Methodology, Investigation, Formal analysis, Data curation. **Knut Erik Teigen Giljarhus:** Writing – review & editing, Supervision, Software, Investigation, Funding acquisition, Conceptualization. **Luca Oggiano:** Writing – review & editing, Funding acquisition, Conceptualization. **Alistair Saddington:** Writing – review & editing, Supervision. **Karthik Depuru-Mohan:** Writing – review & editing, Supervision, Project administration, Funding acquisition, Conceptualization.

Declaration of competing interest

Adam Clarke reports financial support was provided by Nablaflow AS. Adam Clarke reports financial support was provided by Whitworth Society. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

Mr Clarke is pleased to acknowledge the contribution of the IMechE Whitworth Senior Scholarship Award in supporting this research.

A. Appendix

The computational domain was defined as a cylinder with a diameter of 1600 m and a height of 300 m, allowing for adequate flow development. A no-slip boundary condition was applied to all building surfaces. For the lateral walls of the cylindrical domain, an inlet-outlet condition is used to dynamically switch between inlet or outlet conditions based on the flux. The inflow was characterised by a logarithmic velocity profile with a

reference wind speed of 5 m/s at a height of 10 m, and a zero pressure gradient. A roughness length of 0.03 m was specified at the terrain surface to model aerodynamic roughness typical of suburban environments. For the outlet conditions, a fixed pressure value was used along with zero gradient for the remaining fields.

Mesh generation was performed using snappyHexMesh. Local mesh refinement was used to increase spatial resolution in critical regions. Refinement level 4 was applied within 15 m of the skyway, while refinement level 3 was applied within 15 m of all building surfaces. This approach ensured adequate resolution of near-ground flow features. Depending on the refinement region, approximately 10–20 vertical cells were present below the 2 m pedestrian height threshold. Under lift-up style buildings sitting 3 m above the ground, this increased to approximately 16–32 cells, providing enhanced detail in the substructure flow region. An intermediate refinement level of 2 was enforced in the annular region spanning 15–20 m from the buildings. The final computational grid comprised approximately 3 million cells, with an example shown in Fig. 19.

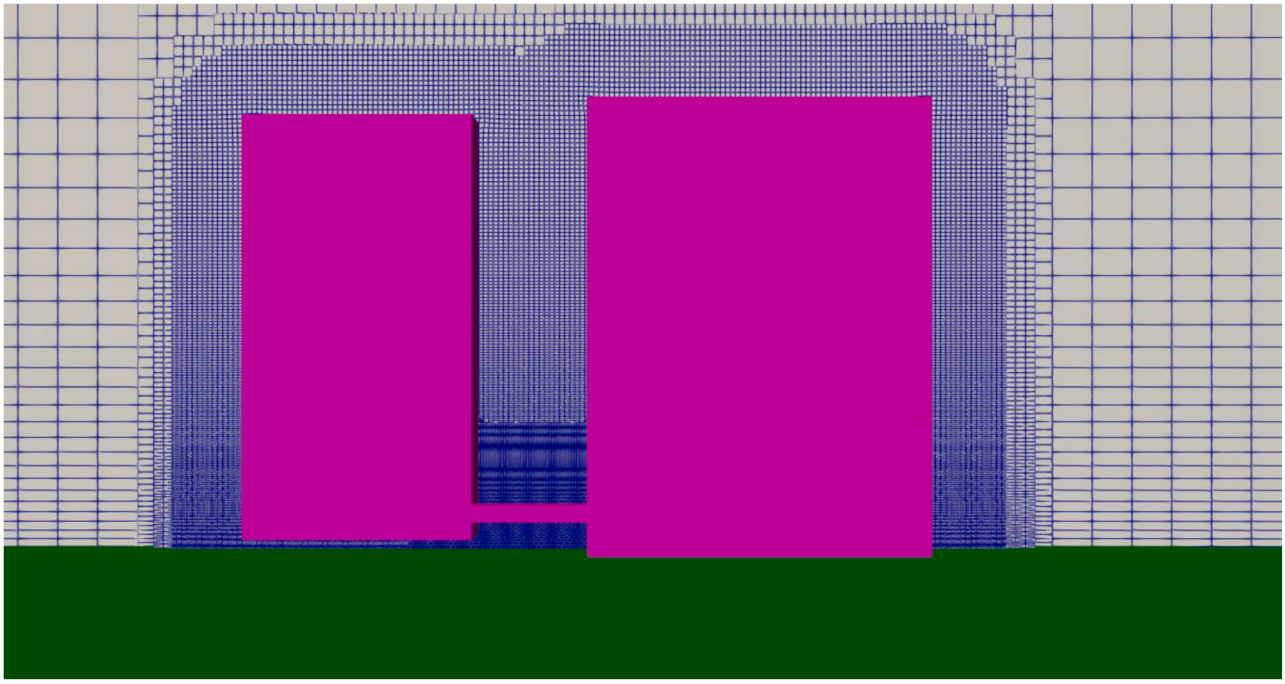


Fig. 19. Example computational mesh used for CFD simulation.

The resolution of the computational mesh can significantly influence both the accuracy and computational cost of CFD simulations. To assess grid independence, simulations were conducted using coarse and fine meshes, created by halving and doubling the number of cells in the baseline (medium) grid. The setup for this study replicates the configuration used in the wind tunnel experiments by Stathopoulos and Storms [64], which also serves as the basis for validating the CFD model. The domain consists of buildings with side lengths of 20 m and a gap width of 6 m, modelled at a scale of 1:400. Wind velocity was sampled along the centreline of the gap at a height of 2 m above ground level. The wind speed ratio - defined as the local wind speed divided by the inflow velocity at the same height - was used to compare the mesh resolutions. As shown in Fig. 20, the results from the medium and fine meshes are in close agreement, confirming that the medium-resolution mesh provides sufficient accuracy while reducing computational cost. The CFD results also show good agreement with wind tunnel measurements, supporting the validity of the simulation setup in capturing the essential flow characteristics.

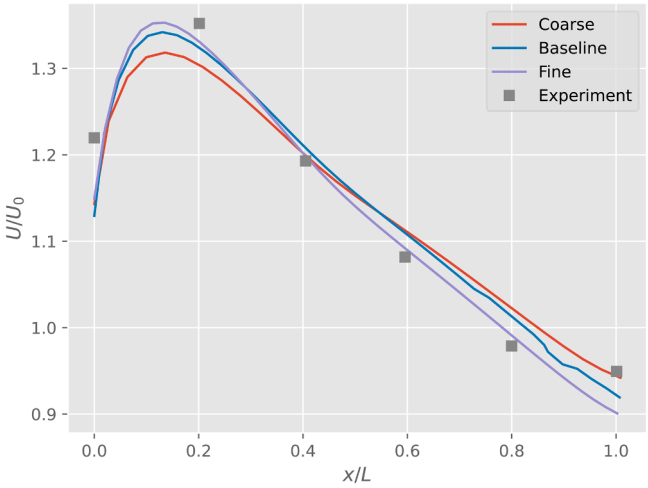


Fig. 20. Mesh independence study showing the normalized velocity profile U/U_0 along the streamwise direction x/L for coarse, baseline, and fine computational meshes. Simulation results are compared with experimental wind tunnel data (grey squares) recorded by Stathopoulos and Storms [64]. Good agreement is observed,

with diminishing differences between the baseline and fine meshes, indicating mesh convergence.

The RANS equations were solved using the SIMPLE algorithm. Turbulence effects were modelled using the standard k–ε closure model. Convective terms, including velocity were discretised using a first-order upwind scheme to ensure numerical stability, while diffusive terms employed a second-order accurate scheme. Residuals were observed to be on the order of 1×10^{-5} for velocity and pressure, and 1×10^{-6} for turbulent kinetic energy and dissipation rate, indicating a satisfactory solution quality for the purpose of dataset generation.

All simulations were performed on a high-performance computing system equipped with AMD EPYC 7543 processors with a clock speed of 2.8 GHz. Training samples were generated in parallel batches to efficiently utilise available computational resources. The total execution time of each simulation was approximately 2 h.

Data availability

Data supporting this study cannot be made available due to *Nabla-flow AS retaining ownership of the data.*

References

[1] United Nations, 'World urbanization prospects the 2018 revision', 2018.

[2] M.M. Ali, K. Al-Kodmany, Tall buildings and urban Habitat of the 21st century: a global perspective, Buildings 2 (4) (Sep. 2012) 384–423, https://doi.org/10.3390/buildings2040384.

[3] J. Auerbach, P. Wan, Forecasting the Urban skyline with extreme value theory, arXiv (2018), https://doi.org/10.48550/arXiv.1808.01514. Oct. 29.

[4] K. Al-Kodmany, The logic of vertical density: tall buildings in the 21st century City, Int. J. High-Rise Build. 1 (2) (Jun. 2012) 131–148, https://doi.org/10.21022/IJHRB.2012.1.2.131.

[5] B. Blocken, J. Carmeliet, Pedestrian wind environment around buildings: literature review and practical examples, J. Therm. Envel. Build. Sci. 28 (2) (Oct. 2004) 107–159, https://doi.org/10.1177/1097196304044396.

[6] ASHRAE, 'ASHRAE Handbook - Fundamentals', American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc., 2021.

[7] Y. Wu, Y. Fan, N. Zeng, W. Li, M. Wang, H. Liu, Clustering tall building typologies and analyzing their composite impacts on wind environments in infill development projects, Int. J. Env. Sci. Technol (Sep. 2024), https://doi.org/10.1007/s13762-024-06067-9.

[8] W. Afe, 'Wind effects due to groups of buildings', in *Royal Society Symposium Architectural Aerodynamics*, London, 1970, pp. 26–27.

[9] S. Zhang, et al., Multi-scale urban natural ventilation climate guidance: a case study in the Shijiazhuang Metropolitan area, Atmos. (Basel) 15 (6) (Jun. 2024), https://doi.org/10.3390/atmos15060676. Art. no. 6.

[10] H. Huang, R. Ooka, H. Chen, S. Kato, T. Takahashi, T. Watanabe, CFD analysis on traffic-induced air pollutant dispersion under non-isothermal condition in a complex urban area in winter, J. Wind Eng. Ind. Aerodyn. 96 (10) (Oct. 2008) 1774–1788, https://doi.org/10.1016/j.jweia.2008.02.010.

[11] V. Cheng, E. Ng, C. Chan, B. Givoni, Outdoor thermal comfort study in a sub-tropical climate: a longitudinal study based in Hong Kong, Int. J. Biometeorol. 56 (1) (Jan. 2012) 43–56, https://doi.org/10.1007/s00484-010-0396-z.

[12] C. Nevers, A. Kubilay, J. Carmeliet, D. Derome, CFD simulation of the wind flow under lift-up buildings using a porous approach, Build Environ. 263 (Sep. 2024) 111867, https://doi.org/10.1016/j.buildenv.2024.111867.

[13] X. Zhang, Y. Gao, Q. Tao, Y. Min, J. Fan, Improving the pedestrian-level wind comfort by lift-up factors of panel residence complex: field-measurement and CFD simulation, Build Environ. 229 (Feb. 2023) 109947, https://doi.org/10.1016/j.buildenv.2022.109947.

[14] J. Liu, X. Zhang, J. Niu, K.T. Tse, Pedestrian-level wind and gust around buildings with a "lift-up" design: assessment of influence from surrounding buildings by adopting LES, Build. Simul. 12 (6) (Dec. 2019) 1107–1118, https://doi.org/10.1007/s12273-019-0541-5.

[15] Y. Du, C.M. Mak, Y. Li, Application of a multi-variable optimization method to determine lift-up design for optimum wind comfort, Build Environ. 131 (Mar. 2018) 242–254, https://doi.org/10.1016/j.buildenv.2018.01.012.

[16] Y. Du, C.M. Mak, J. Liu, Q. Xia, J. Niu, K.C.S. Kwok, Effects of lift-up design on pedestrian level wind comfort in different building configurations under three wind directions, Build Environ. 117 (May 2017) 84–99, https://doi.org/10.1016/j.buildenv.2017.03.001.

[17] K. Teigen and T.-O. Hagbo, 'Simulation-based data-driven wind engineering—Analyzing the influence of building proximity and skyways on pedestrian comfort', 2024, pp. 241–253. doi: 10.1007/978-3-031-49791-9_17.

[18] W.-Y. (陳婉儀) Chen, Y.-L. (蘇益立) Su, Y.-H. (阮于軒) Juan, Wind characteristics around a skyway bridge of high-rise buildings, Phys. Fluids 36 (7) (Jul. 2024) 075134, https://doi.org/10.1063/5.0216665.

[19] M. Santamouris, et al., On the impact of urban climate on the energy consumption of buildings, Sol. Energy 70 (3) (Jan. 2001) 201–216, https://doi.org/10.1016/S0038-092X(00)00095-5.

[20] R.M. Aynsley, Politics of pedestrian level urban wind control, Build Environ. 24 (4) (Jan. 1989) 291–295, https://doi.org/10.1016/0360-1323(89)90022-X.

[21] City of London, 'Wind Microclimate Guidlines For Developments in the City of London', Aug. 2019. Accessed: Dec. 19, 2022. [Online]. Available: https://www.cityoflondon.gov.uk/assets/Services-Environment/wind-microclimate-guidelines.pdf.

[22] Leeds City Council, 'Draft Wind and Microclimate Toolkit'. 2021. Accessed: Jan. 15, 2024. [Online]. Available: https://www.leeds.gov.uk/docs/Draft%20wind%20and%20microclimate%20toolkit.pdf.

[23] Royal Institute of British Architects, 'RIBA Plan of Work 2020 Overview', 2020. Accessed: Jul. 15, 2023. [Online]. Available: https://www.architecture.com/knowledge-and-resources/resources-landing-page/riba-plan-of-work.

[24] B. Blocken, T. Stathopoulos, J.P.A.J. Van Beeck, Pedestrian-level wind conditions around buildings: review of wind-tunnel and CFD techniques and their accuracy for wind comfort assessment, Build Environ. 100 (May 2016) 50–81, https://doi.org/10.1016/j.buildenv.2016.02.004.

[25] I. Isyumov, A.G. Davenport, The ground level wind environment in built-up areas, in: Proceedings of the Fourth International Conference on Wind Effects on Buildings and Structures: Heathrow 1975, Cambridge University Press, 1977.

[26] T.V. Lawson, The widn content of the built environment, J. Wind Eng. Ind. Aerodyn. 3 (2) (Jan. 1978) 93–105, https://doi.org/10.1016/0167-6105(78)90002-8.

[27] W.H. Melbourne, Criteria for environmental wind conditions, J. Wind Eng. Ind. Aerodyn. 3 (2) (Jan. 1978) 241–249, https://doi.org/10.1016/0167-6105(78)90013-2.

[28] E. Willemsen, J.A. Wisse, Design for wind comfort in The Netherlands: procedures, criteria and open research issues, J. Wind Eng. Ind. Aerodyn. 95 (9) (Oct. 2007) 1541–1550, https://doi.org/10.1016/j.jweia.2007.02.006.

[29] H. Mittal, A. Sharma, A. Gairola, A review on the study of urban wind at the pedestrian level around buildings, J. Build. Eng. 18 (Jul. 2018) 154–163, https://doi.org/10.1016/j.jobe.2018.03.006.

[30] Stefano Cammeli, francesco Dorigatti, Daniel Hackett, Alex Ping To, Bernado Vazquez, A Position Paper On Experimental and Computational Methods in Wind Engineering, UK Wind Engineering Society, 2022. Accessed: Jul. 14, 2023. [Online]. Available: https://www.windengineering.org.uk/uploadedresources/1654687206-a_position_paper_on_experimental_and_computational_methods_in_wind_engineering_-_uk_wind_engineering_society.pdf.

[31] B. Blocken, LES over RANS in building simulation for outdoor and indoor applications: a foregone conclusion? Build. Simul. 11 (5) (Oct. 2018) 821–870, https://doi.org/10.1007/s12273-018-0459-3.

[32] T.-O. Hågbo, K.E.T. Giljarhus, Pedestrian wind comfort assessment using computational fluid dynamics simulations with varying number of wind directions, Front. Built Environ. 8 (Jul. 2022) 858067, https://doi.org/10.3389/fbuil.2022.858067.

[33] P. Moonen, T. Defraeye, V. Dorer, B. Blocken, J. Carmeliet, Urban physics: effect of the micro-climate on comfort, health and energy demand, Front. Archit. Res. 1 (3) (Sep. 2012) 197–228, https://doi.org/10.1016/j.foar.2012.05.002.

[34] S. Mokhtar, A. Sojka, C.C. Davila, Conditional generative adversarial networks for pedestrian wind flow approximation, Soc. Model. Simul. Int. (SCS) SimAUD 2020 (2020) 8.

[35] B. Kim, D.-E. Lee, K.R.S. Preethaa, G. Hu, Y. Natarajan, K.C.S. Kwok, Predicting wind flow around buildings using deep learning, J. Wind Eng. Ind. Aerodyn. 219 (Dec. 2021) 104820, https://doi.org/10.1016/j.jweia.2021.104820.

[36] A.U. Weerasuriya, X. Zhang, B. Lu, K.T. Tse, C.H. Liu, A Gaussian process-based emulator for modeling pedestrian-level wind field, Build Environ. 188 (Jan. 2021) 107500, https://doi.org/10.1016/j.buildenv.2020.107500.

[37] Y. Wu, Q. Zhan, S.J. Quan, Improving local pedestrian-level wind environment based on probabilistic assessment using gaussian process regression, Build Environ. 205 (Nov. 2021) 108172, https://doi.org/10.1016/j.buildenv.2021.108172.

[38] H. Wang, W. Ma, J. Niu, R. You, Evaluating a deep learning-based surrogate model in predicting wind distribution for urban microclimates, Build Environ. (Dec. 2024) 112426, https://doi.org/10.1016/j.buildenv.2024.112426.

[39] S.J. Low, et al., FastFlow: AI for fast urban wind velocity prediction, arXiv (Nov. 22, 2022), https://doi.org/10.48550/arXiv.2211.12035.

[40] N. Benmoshe, E. Fattal, B. Leitl, Y. Arav, Using machine learning to predict wind flow in urban areas, Atmos. (Basel) 14 (Jun. 2023) 990, https://doi.org/10.3390/atmos14060990.

[41] Y. He, et al., Hybrid framework for rapid evaluation of wind environment around buildings through parametric design, CFD simulation, image processing and machine learning, Sustain. Cities Soc. 73 (Oct. 2021) 103092, https://doi.org/10.1016/j.scs.2021.103092.

[42] H. Hoeiness, K. Gjerde, L. Oggiano, K.E.T. Giljarhus, M. Ruocco, Positional encoding augmented GAN for the assessment of wind flow for pedestrian comfort in urban areas, arXiv (Jan. 04, 2022), https://doi.org/10.48550/arXiv.2112.08447.

[43] A.V. Clemente, K.E.T. Giljarhus, L. Oggiano, M. Ruocco, Rapid pedestrian-level wind field prediction for early-stage design using Pareto-optimized convolutional

neural networks, Comput.-Aided Civ. Infrastruct. Eng. 39 (2024) 2826–2839, https://doi.org/10.1111/mice.13221.
[44] I.J. Goodfellow, et al., Generative Adversarial Networks, arXiv (Jun. 10, 2014), https://doi.org/10.48550/arXiv.1406.2661.
[45] P. Kastner, T. Dogan, A GAN-based surrogate model for instantaneous urban wind flow prediction, Build Environ. 242 (Aug. 2023) 110384, https://doi.org/10.1016/j.buildenv.2023.110384.
[46] S. Qin, et al., Modeling multivariable high-resolution 3D urban microclimate using localized fourier neural operator, arXiv (Nov. 18, 2024), https://doi.org/10.48550/arXiv.2411.11348.
[47] X. Shao, Z. Liu, S. Zhang, Z. Zhao, C. Hu, PIGNN-CFD: a physics-informed graph neural network for rapid predicting urban wind field defined on unstructured mesh, Build Environ. 232 (Mar. 2023) 110056, https://doi.org/10.1016/j.buildenv.2023.110056.
[48] G. Calzolari, W. Liu, Deep learning to replace, improve, or aid CFD analysis in built environment applications: a review, Build Environ. 206 (Dec. 2021) 108315, https://doi.org/10.1016/j.buildenv.2021.108315.
[49] Z. Liu, S. Zhang, X. Shao, Z. Wu, Accurate and efficient urban wind prediction at city-scale with memory-scalable graph neural network, Sustain. Cities Soc. 99 (Dec. 2023) 104935, https://doi.org/10.1016/j.scs.2023.104935.
[50] I. Tolstikhin, et al., MLP-Mixer: an all-MLP architecture for vision, arXiv (Jun. 11, 2021), https://doi.org/10.48550/arXiv.2105.01601.
[51] Y. Mansour, K. Lin, and R. Heckel, 'Image-to-Image MLP-mixer For Image Reconstruction', Feb. 2023, Accessed: Apr. 07, 2023. [Online]. Available: https://openreview.net/forum?id=wsuQ2h6KZXQ.
[52] J. Yang, X. Yu, P. Qiu, D2-MLP: dynamic decomposed MLP mixer for medical image segmentation, arXiv (Sep. 13, 2024), https://doi.org/10.48550/arXiv.2409.08905.
[53] H. Zhang, Z. Dong, B. Li, S. He, Multi-scale MLP-mixer for image classification, Knowl Based Syst 258 (Dec. 2022) 109792, https://doi.org/10.1016/j.knosys.2022.109792.
[54] G. Cazenavette, M.L. De Guevara, MixerGAN: an MLP-based architecture for unpaired image-to-image translation, arXiv (Aug. 19, 2021), https://doi.org/10.48550/arXiv.2105.14110.
[55] A. Clarke, K.E.T. Giljarhus, L. Oggiano, A. Saddington, K. Depuru-Mohan, MLP-Mixer-based deep learning network for pedestrian level wind assessment, in: presented at the Climate Informatics 2024, Apr. 2024.
[56] A. Clarke, K.E.T. Giljarhus, L. Oggiano, A. Saddington, K. Depuru-Mohan, Bridging the gap: embedding 3D details into fast deep-learning model for pedestrian-level wind prediction, presented at the, in: 14th UK Conference on Wind Engineering, University of Southampton, 2024, pp. 56–59, https://doi.org/10.5258/WES/P0017.
[57] X. Feng, et al., ConvWin-UNet: uNet-like hierarchical vision transformer combined with convolution for medical image segmentation, MBE 20 (1) (2023), https://doi.org/10.3934/mbe.2023007. Art. no. mbe-20-01-007.
[58] Bane Sullivan, Alexander Kaszynski, PyVista: 3D plotting and mesh analysis through a streamlined interface for the Visualization Toolkit (VTK), J. Open Source Softw. 4 (37) (May 2019) 1450, https://doi.org/10.21105/joss.01450.
[59] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, arXiv (Jan. 29, 2017). Accessed: Dec. 13, 2022. [Online]. Available: http://arxiv.org/abs/1412.6980.
[60] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: a next-generation hyperparameter optimization framework, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Association for Computing Machinery, New York, NY, USA, Jul. 2019, pp. 2623–2631, https://doi.org/10.1145/3292500.3330701. KDD '19.
[61] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Trans. Image Process. 13 (4) (Apr. 2004) 600–612, https://doi.org/10.1109/TIP.2003.819861.
[62] A.C. Bovik, The Essential Guide to Image Processing, Academic Press, 2009.
[63] O. Ronneberger, P. Fischer, T. Brox, U-Net: convolutional networks for biomedical image segmentation, arXiv (May 18, 2015), https://doi.org/10.48550/arXiv.1505.04597.
[64] T. Stathopoulos, R. Storms, Wind environmental conditions in passages between buildings, J. Wind Eng. Ind. Aerodyn. 24 (1) (Aug. 1986) 19–31, https://doi.org/10.1016/0167-6105(86)90070-X.



https://dspace.org/

Deep learning for urban wind prediction: an MLP-Mixer approach with 3D encoding

Clarke, Adam; Giljarhus, Knut Erik Teigen; Oggiano, Luca; Saddington, Alistair J.; Depuru-Mohan, Karthik

2025-10-01
Elsevier
https://doi.org/10.1016/j.buildenv.2025.113495
Clarke A, Giljarhus KET, Oggiano L, et al., (2025) Deep learning for urban wind prediction: An MLP-Mixer approach with 3D encoding. Building and Environment, Volume 284, October 2025, Article number 113495

Downloaded from a DSpace repository. https://dspace.org/