Building and Environment 266 (2024) 112123

Contents lists available at ScienceDirect

Building and Environment

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

Adversarial image-to-image model to obtain highly detailed wind fields from mesoscale simulations in urban environments

Jaime Milla-Val [a], Carlos Montañés [b], Norberto Fueyo [b,*]

[a] Nabladot S.L., Salvador Allende 75, 50015 Zaragoza, Spain
[b] Instituto de Investigación en Ingeniería de Aragón (I3A), University of Zaragoza, María de Luna 3, 50018 Zaragoza, Spain

ARTICLE INFO

Keywords:
Urban wind prediction
Computational fluid dynamics
Numerical weather predictions
Image-to-image
Deep learning
Conditional generative adversarial network

ABSTRACT

We propose a conditional Generative Adversarial Network (cGAN) that can produce detailed local wind fields in urban areas, comparable in level of detail to those from Computational Fluid Dynamics (CFD) simulations, that are generated from coarser Numerical Weather Prediction (NWP) data.

In our approach, the cGAN is trained using NWP data as input and CFD as targets. Both CFD and NWP data are presented to the network as images, using an image-to-image model based on Pix2Pix to transform coarse meteorological conditions into detailed local wind fields.

The methodology is tested in a residential district in a large Spanish city, Zaragoza. The model predictions show significant agreement with the actual CFD results, while reducing the computational time from eight hours to seconds. Feature engineering of image channels effectively reduces the model error, especially in the wind direction, achieving a mean absolute error in the wind speed of 0.35 m/s and a wind direction error of 27.0°.

1. Introduction

The characterisation of wind in urban environments is of great interest in multiple disciplines. For urban city planning, Kabošová et al. [1] developed a wind prediction tool based on architectural parametric design and wind flow analysis; they have proved its reliability in a real case study. Werner et al. [2] assessed wind comfort and general wind rose statistics in built environments using regression and classification U-Nets and Computational Fluid Dynamics (CFD) simulations. The exploitation of wind energy resources in urban environments [3] may help meet the increasing energy demand as population continues to expand rapidly. Local urban energy generation helps circumvent some of the drawbacks of large-scale on-shore wind farms, such as available placements, electricity transmission losses and infrastructure, and public acceptance [4]. K.C. et al. [5] reviewed performance evaluations of small wind turbines and pointed out flaws in the validity of some international design standards for urban installations. Crucially, they also highlighted the importance of characterising urban wind conditions to facilitate a more consistent design of small wind turbines.

Toja-Silva et al. [6] explored the broad application of CFD in the analysis of urban wind patterns, particularly for urban energy purposes, including technical issues and challenges related to modelling turbulence, numerical schemes, and meshing. Also with the goal of seeking the efficient and reliable use of CFD to characterise wind patterns, Xi et al. [7] investigated the impact on accuracy of the size of the area surrounding the domain of interest (the so called "building layers") on wind flow simulations across various scales (local, micro, and building scale).

As an alternative way of economising computational time, some authors have suggested the integration of deep neural networks and machine learning with CFD [8]. Specifically for built-environment applications, Calzolari and Liu [9] reviewed the usage of artificial neural networks as surrogate models to aid, improve or replace expensive CFD analysis.

Machine learning approaches have been adopted for the evaluation and prediction of wind flow in conjunction with traditional methods such as CFD or tunnel experiments. He et al. [10] proposed a combination framework for wind evaluation in built environments through parametric design, CFD, image processing, and machine learning. In [11], the K-nearest neighbours algorithm was used to predict wind flow in urban areas, using data from wind tunnel experiments and morphological characteristics as input.

Numerical Weather Prediction (NWP) is coarser and hence computationally less expensive than CFD, and may also benefit from the application of AI techniques. Temporal fusion transformers have been applied to predict spatial temperature variations to address the urban heat island effect using NWP data [12]. Furthermore, for short-term

* Corresponding author.
E-mail address: Norberto.Fueyo@unizar.es (N. Fueyo).

https://doi.org/10.1016/j.buildenv.2024.112123
Received 13 June 2024; Received in revised form 22 September 2024; Accepted 24 September 2024
Available online 28 September 2024

wind speed forecasting in urban settings, Zheng et al. [13] introduced a convolutional long-short-term memory model that processes NWP image sequences to learn spatial–temporal correlations.

Deep learning techniques have also been applied to urban wind prediction to enhance and create new solutions that were until recently the realm of traditional, more expensive approaches. Low et al. [14] used an U-net architecture in a Convolutional Neural Network (CNN) for the fast prediction of urban wind velocity and demonstrated its usefulness for data-driven pedestrian-level wind speed prediction. The use of Graph Neural Networks (GNN) was explored in [15]; they achieved accurate and efficient prediction of urban wind flow, with an excellent management of GPU memory limitations and a reduction in computational time compared to CFD. Shao et al. [16] also used GNN in conjunction with Physics Informed Neural Networks (PINNs) for the rapid prediction of urban wind fields in unstructured meshes. Generative Adversarial Networks (GAN) and their variants are highly effective machine learning frameworks applied in many wind-power related studies. Liang et al. [17] used GAN for data augmentation for wind turbine gearbox fault detection, a field in which obtaining field data is often difficult. Similarly, Yang et al. [18] leveraged a GAN oversampling method to successfully classify gearbox faults with small datasets. Deep GANs have been used to design new aerofoils for vertical-axis wind turbines [19]. A GAN variant that uses the Wasserstein distance was employed by Zhang et al. [20] to generate stochastic scenarios of wind power production. Ye et al. [21] developed a day-ahead model for wind power generation that combines self-attention and TimeGAN for temporal forecasting. Behara and Saha [22] proposed an innovative incremental GAN model that enhances the accuracy and reliability of wind-turbine position analysis, aiding in better wind characterisation for optimising wind energy production. For single wind turbine wake prediction, Li et al. [23] developed a model integrating transformers and conditional GANs (cGANs) [24]. Zhang and Zhao [25] utilised deep convolutional cGANs to model wake formation in wind farms.

GANs have also been adopted for urban-specific applications. Kastner and Dogan [26] introduced a GAN-based surrogate model for CFD, facilitating a quick and accessible analysis of urban airflow with various building geometries; additionally, this model was incorporated into design software to assist in the early phases of architectural design. Masoumi-Verki et al. [27] introduced a framework using GANs to develop "non-intrusive" reduced-order models (*i.e.*, reduced-order models that do not require access to the CFD source code) to predict turbulent wakes in high-rise buildings, utilising bidirectional long-short-term memory neural networks and transfer learning for improved prediction accuracy with limited data.

In this work, we use a conditional Generative Adversarial Network (cGAN) to obtain high-detail urban wind fields, with a resolution similar to that of CFD. We employ NWP data as input to our cGAN, and CFD simulations in an urban layout as targets; we treat both types of data as pseudo-images. We base our model on the image-to-image Pix2Pix model proposed by Isola et al. [28]. NWP simulations provide a synoptic prediction of the meteorological state of the atmosphere in a region of study. Unlike high-resolution CFD wind-field simulations, NWP predictions account for the full range of meteorological phenomena, including not only fluid dynamics, but also radiation and precipitation. NWP simulations are, in this sense, complementary of the more focused local CFD simulations of the local wind field. For the present work, a key feature of NWP simulations is that they are much more economical than CFD ones.

Therefore, the main objective of the model proposed in this work is to obtain high-fidelity CFD data in an urban environment without incurring its elevated computational cost by using coarser and faster NWP simulations as input. Once trained, the model can transform NWP results into CFD results.

One of the advantages of the proposed methodology is the great time reduction achieved by using the proposed deep learning model compared to traditional simulations. The model can take advantage of the mesoscale simulations used daily for meteorological forecasting to achieve highly detailed wind fields. Unlike previously published work, we demonstrate the applicability of our methodology in a whole, actual, complex urban district, rather than using isolated buildings or canonical configurations of buildings as a test bed. We also predict the local flow field under real wind conditions, with the subsequent diversity in wind speed and direction. The use of a real-world urban configuration and real-world wind conditions adds to the complexity of the problem, and places an additional burden on the proposed AI method.

The structuring of the CFD and NWP data as matrices that can be understood as images, as done in this paper, broadens the applicability of AI methods already used in the field of image detection, such as the image-to-image model that we employ in this study, and to take advantages of new developments in image-to-image translation. For instance, Jiang et al. [29] applied image-to-image translation via the StyleGAN model [30] to generate anime scenes. Similarly, Dalva et al. [31] used it for editing facial images, and Sun et al. [32] to improve underwater imagery. Image-to-image translation models have also been used for processing unpaired medical physics images [33] to improve image quality, for autonomous annotation of medical images, and to reduce radiation exposure during Computed Tomography imaging. Romero et al. [34] employed an image-to-image model that used wind farm layouts and unperturbed wind fields as input to derive the perturbed wind fields produced by the wind farm, as a tool for optimising the wind farm layout. To the best of our knowledge, there are no studies that utilise image-to-image models to convert mesoscale wind fields into the highly-detailed, CFD-like ones for urban environments; the present study addresses this gap.

2. Model description

This research investigates the use of a conditional Generative Adversarial Network (cGAN) [24] that uses coarser Numerical Weather Prediction (NWP) data to achieve detailed local wind forecasts in urban areas, with a resolution comparable to those from Computational Fluid Dynamics (CFD) at a fraction of the cost.

The general idea is to obtain detailed velocity data for an urban area from coarser, and hence relatively inexpensive, meteorological data. For this, a deep learning algorithm is used; the proposed AI model is a cGAN, based on the Pix2Pix [28] image-to-image model. Mesoscale simulations, using a NWP model, are carried out over an extended period of time (the year 2018 in this study). The NWP outputs are used as boundary conditions for detailed wind field CFD simulations over the urban area studied (a district in a large Spanish city). CFD simulations are more expensive than NWP, so only a few events are simulated in detail (214 simulations for training and 80 for testing). Then the NWP results are used as input and the CFD results as targets, to train the AI model. Once the AI model has been trained, NWP simulations can be used as input to obtain highly detailed wind fields as output.

In Fig. 1 an overview of the proposed method and the training process is shown. The training process starts with the reanalysis data for the year 2018 from `ds.090.0` [35]; this is a Global Forecast System (GFS) that serves as input for the NWP simulations. The NWP simulations are performed with the Weather Research Forecast (WRF) [36] software, from which hourly simulation results are obtained for the whole year, 8760 timestamps or events. From these events, we identify the most representative ones (details provided in Section 2.1), which will be simulated with CFD; 214 events are selected for training and 80 for testing. Both NWP and CFD results are post-processed in order to arrange the data as 2D square images; in some cases, an image channel featuring engineering (see Section 2.5) is applied to improve the performance of the AI model. The image of the NWP data is the input for the AI model, which is a cGAN in this study; the NWP data are passed to the model generator and produces an image similar to

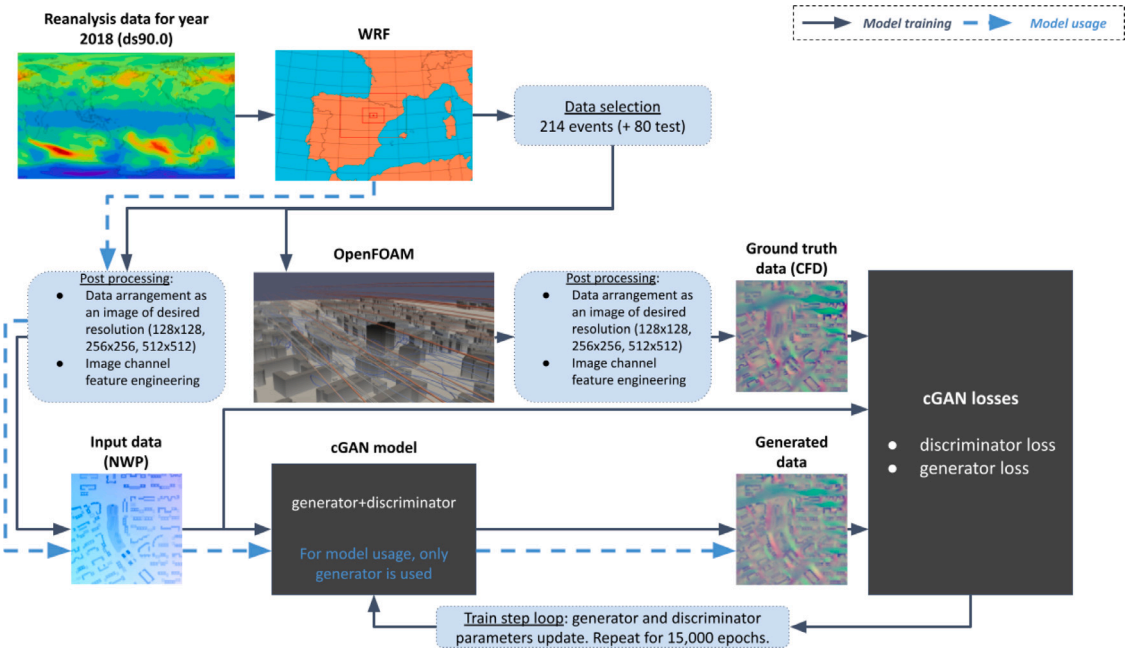


Fig. 1. Overview of the proposed method and the training process. The black arrows denote model training, and the blue dashed arrows denote model usage.

that given by the CFD. Then, the NWP image, the CFD image and the image produced by the AI model are used to calculate the cGAN losses to update the network parameters; this process is repeated for 15,000 epochs during the training process. In usage mode (blue dashed arrows in Fig. 1), the CFD simulation is not needed, so the NWP data is passed on directly to the cGAN generator to produce results similar to those of CFD.

2.1. Numerical simulations

The area of study of this work is an urban environment, specifically a district in the fifth largest Spanish city, Zaragoza. The city is located inland, in the Ebro river valley; the valley runs roughly from the Atlantic Ocean (to the North) to the Mediterranean Sea (to the East). This geographical location causes the city to experience recurrent strong wind events throughout the year. For the prediction of wind phenomena in a built environment, Zaragoza presents a challenging environment. In Fig. 2 some images of relevant buildings and the rendering of the studied area in the CFD mesh are presented. The district is located relatively close to the river bed, to the north of the city. It consists of open streets and wide avenues, with buildings that often reach 30 m high. The buildings are mainly residential; however, the area also includes shopping centres, a few educational institutions, and healthcare facilities, along with a business centre consisting of three tall buildings, the highest two standing at 80 m.

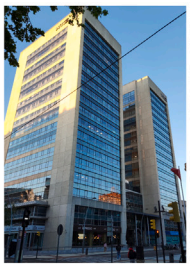
Numerical Weather Prediction (NWP) simulations are conducted for the year 2018 at the mesoscale level around the region of interest; this produces 8760 hourly results (or *events*) for the year. The NWP simulations are carried out using Weather Research Forecast v4.1 (WRF) [36], with data from `ds090.0` [35] as input, and a fourfold-nested rectangular domain centred in Zaragoza; the innermost dimensions are 81 × 54 km, with a horizontal grid resolution of 900 m. These simulations serve as boundary conditions for highly detailed wind simulations of the same area using Computational Fluid Dynamics (CFD). The CFD simulations are performed with OpenFOAM v6 [37] using the SIMPLE algorithm for the pressure–velocity coupling. The flow is modelled as isothermal, and the Reynolds-averaged Navier–Stokes equations of continuity and momentum are solved. For turbulence, the Spalart–Allmaras one-equation model [38] is used. The mesh has about 2 million cells; a rendering of the geometry as represented in the mesh can be seen in Fig. 2(d). It has a quasi-cylindrical shape to accommodate any possible wind direction, with a diameter of 3 km; in Fig. 3 a top view of the domain is shown. The simulation of wind fields in an urban environment is a well developed subject (see, for instance, [39–41]). The primary goal of this paper is not to contribute to this well established field, and therefore our CFD calculations, while fully converged and physically plausible, have not been validated. Validation of the CFD model in a real urban setting and with forecast boundary conditions linked to NWP is a very complex task; indeed, we acknowledge that certain aspects of our CFD model, including mesh resolution and turbulence model, should perhaps be refined for more accurate CFD predictions. Our goal is to develop a methodology for creating CFD-like result from NWP forecasts; our validation effort is geared to this end.

2.2. Network architecture

The essence of the methodology presented in this paper is to consider numerical simulation results as images, which can be handled by the Pix2Pix AI deep learning model [28].

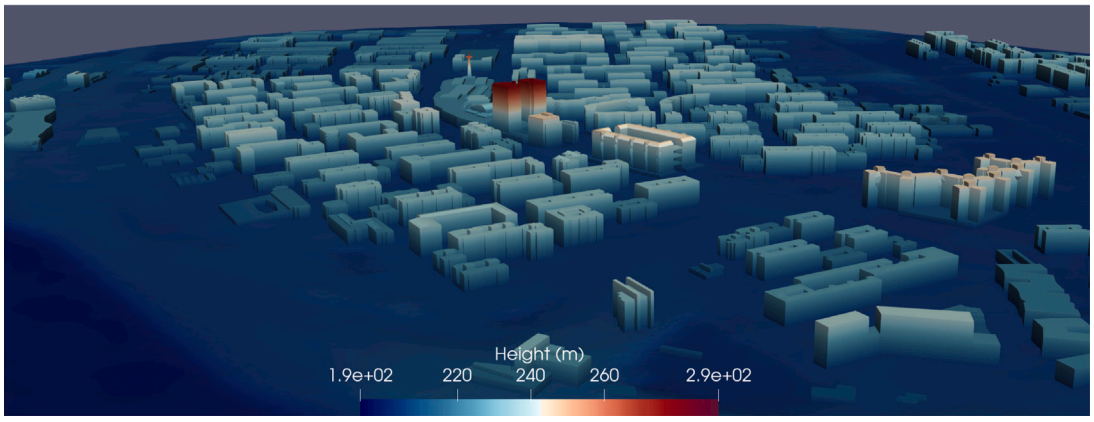
Pix2Pix is a conditional Generative Adversarial Network (cGAN) [24] designed for image-to-image translation tasks. The goal of this model is to learn a mapping from an input image to an output image. For example, it can be used to translate a day-time scene to a night-time scene or to convert a sketch into a coloured image [28]. In our case, the input images are the NWP predictions and the output images the CFD results. A generator and a discriminator are the two principal components of the Pix2Pix model. The generator's objective is to create images that appear as realistic as possible. The task of the discriminator is to discern whether a given image is real (from the training set) or artificial (created by the generator). One of the key features of Pix2Pix is the use of cGAN. In a cGAN, both the generator and the discriminator are conditioned on some additional information. In the case of Pix2Pix, this extra information is the input image; in the application presented in this paper, the conditioning information is the NWP predictions. This conditioning helps the model focus on learning the mapping from the input image to the output image, rather than on generating arbitrary images from random noise. Given the input image (NWP predictions), the model generates the corresponding output image (CFD-like results). The loss function of the Pix2Pix combines adversarial loss and L1 loss.

(a) Real view: shopping centre.



(b) Real view: World Trade Centre Zaragoza

(c) Real view: residential building.



(d) CFD mesh rendering of the studied urban area.

Fig. 2. Sample buildings in the modelled district (a) (b) (c), and district representation in the CFD mesh (d).

The adversarial loss is a cross-entropy element; the adversarial loss seeks to render the generated images indiscernible from real images. The L1 loss is the distance using the L1 norm between the target and the predicted images, *i.e.*, the pixel-wise mean absolute error; the L1 loss ensures that the generated images are structurally similar to the target images. A hyperparameter, λ, balances the adversarial and the L1 loss contributions to the loss function. This loss combination attempts to ensure that the images produced by Pix2Pix are both genuine and accurate.

The generator with a U-net-based architecture is comprised of to elements [42]: an encoder (downsampler) and a decoder (upsampler), with skip connections between them. The encoder is conformed by downsampler blocks, consisting of a convolution operation with a filter size of four and a LeakyReLU [43] activation function, and a batch normalisation layer. The decoder is made of upsampler blocks, consisting of a transposed convolution with an activation function ReLU [44], dropout (applied to the first three blocks), and also a batch normalisation layer. The discriminator architecture consists of a convolutional PatchGAN classifier (proposed in [28]). The PatchGAN classifier evaluates the image by regions instead of as a whole. By assessing the authenticity of local image patches, it improves the texture detail and stability of the generated images. Originally, Pix2Pix was designed to work with 256 × 256 images; in this work we also investigate the performance of image resolutions of 128 × 128 and 512 × 512. To accommodate these dimensions, pairs of downsampler/upsampler blocks are added or removed in the generator; the discriminator is not changed. Two ADAM [45] optimisers are used to update the network weights of the generator and the discriminator with a learning rate of $2 \cdot 10^{-4}$; these optimisers are independent of each other. This autonomy between generator and discriminator provides the model with its "adversarial" and competing characteristics. Image pixel values are normalised to the range $[-1, 1]$. Additionally, mixed precision float16 is used, which greatly increases efficiency during training and inference time; this results in lower GPU-RAM consumption and faster training steps.

2.3. Network training and testing

The proposed AI model therefore utilises NWP predictions as input and CFD calculations as targets. To train the AI model, an entire year (specifically 2018) is modelled using NWP. Out of the 8760 hourly events generated with these simulations, 214 events are chosen and simulated with CFD to create the training dataset.

Given the very high computational cost of CFD simulations compared to NWP ones, the most representative events of the year are chosen to be simulated in CFD, resulting in a total of 214 events for the training dataset. "Most representative" here should not be taken as meaning the most frequent ones. Rather, event selection, as outlined below, ensures that the set of chosen training events spans the whole space of meteorological conditions. For the selection of representative events, the hourly NWP events for a year are sorted into categories, or "bins"; an event is assigned to a bin according to the wind speed and direction at the central location in the domain. For creating the training set, events are randomly chosen from each bin in order of decreasing bin population. Once an event has been chosen from a bin, the bin and its adjacent ones are marked as ineligible for further selection. A similar approach was developed by the authors in [46], and further details can be found there.

A test dataset is assembled by randomly choosing 80 additional hourly NWP events for the same year, none of which is present in the training dataset. These test events are also simulated with CFD but are not used for model training.

The model is trained on a NVIDIA GeForce RTX 4060 Ti GPU with 16 Gb of vRAM during 15 000 epochs.

2.4. Data as images

The AI model as presented so far is a cGAN whose input (NWP simulations) and output (CFD simulations) are images. The challenge

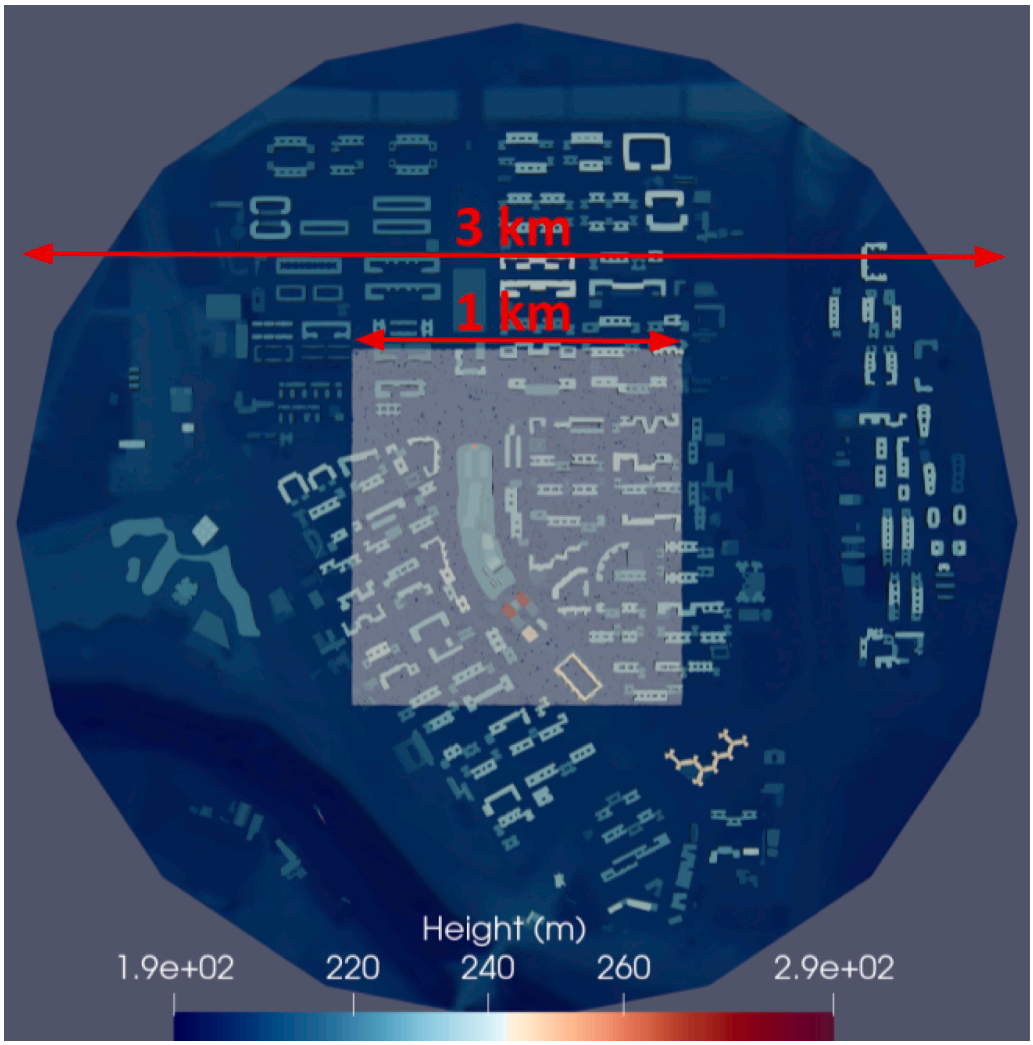


Fig. 3. Top view of the CFD mesh of the simulated district. The lighter shaded square indicates the area where wind fields will be modelled as images. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

in this application is to reconcile the output data structure of the simulations with the expected data structure of the AI model. To this end, the data from the simulations are interpolated and rearranged into a square image format.

Fig. 3 presents a top view of the CFD domain; the lighter-coloured square is the region where wind fields will be modelled as images. It is a square area, with a 1 km side, centred in the CFD mesh. The images obtained have two channels, corresponding to the West–East wind component u and the North–South wind component v. This is because the primary interest are the horizontal wind components, while the vertical component is not modelled. The images from the NWP simulations are interpolations of u and v on a surface that at each point is $\Delta = 20$ m above the ground level or the local rooftop; for the CFD images, $\Delta = 2$ m. In the CFD mesh, the cell centre closest to the ground is generally about 1 m high, and therefore the data extracted at 2 m is interpolated, as intended, from CFD results. Finer meshes close to walls can enhance the accuracy of the CFD simulation, but the current mesh arrangements do not impair the challenge set out in this work of achieving CFD-like resolution from NWP simulations.

Fig. 3 shows that the area represented in the AI model images is smaller than the complete domain used in the CFD simulation. This is for two reasons. First, we choose to move the model domain away from the boundary conditions used for the CFD simulations (the dark, approximately circular footprint in the figure); secondly, we wish to use, for parametric analysis, images of varying resolutions. To achieve manageable computational times with the existing computational resources, we generate three sets of resolutions: 128×128, 256×256, and 512×512. The use of the square shown in a light shade in Fig. 3 for generating images ensures that, in the image with the lowest resolution, a pixel represents $7.8\ \text{m} \times 7.8\ \text{m}$, that we consider as coarse as it should be given the typical street dimensions.

The final result of this image generation process is two sets (from NWP and CFD) of two-channel images; the two channels contain values of u and v, which have been interpolated from the NWP and CFD simulations. The two sets of images will later be divided into a training ensemble (with 214 pairs of images) and a testing set (with 80 pairs of images).

2.5. Image channel feature engineering

Typically, an (actual) image has three channels, one for each RGB colour. In this case, the pseudo-images we build from the simulation results have only two: the two horizontal wind components u and v. However, the cGAN model that we use does not have a hard limit on the number of channels it can handle. We can also encode in the image channels any information that may increase the accuracy of the local wind field prediction. Feature engineering attempts to improve the performance of machine learning models by using knowledge of the problem to create new features from raw data. In the case presented in this paper, for wind field prediction, the raw data are the two horizontal

Table 1
Mean Absolute Errors (MAE) in wind speed and direction for the models with resolution images of 128 × 128. Comparison for `u-v` model and `V-sin-cos` model. Mean (μ) and standard deviation (σ) for CFD wind speed and direction for the test events are presented in the last row as a reference.

Model	MAE_{speed} [m/s]	MAE_{dir} [°]
`u-v`	0.42	33.8
`V-sin-cos`	0.35	27.0
Ground truth: CFD ($\mu \pm \sigma$)	1.10 ± 1.18 m/s	$-3.2 \pm 99.2°$

wind components u and v. An alternative means of presenting this information is with the magnitude of wind speed V and the wind direction θ (as elaborated upon below, it is more practical to represent θ in terms of $\sin\theta$ and $\cos\theta$ than as a direction in degrees). The feature engineering technique applied in this study entails replacing the image channels u and v with three new channels V, $\sin\theta$, and $\cos\theta$.

During this study, sensibly larger wind direction errors are detected in areas with low u and v values. This is because the wind direction is calculated from u and v using the inverse tangent function. For low u and v values, small prediction errors in either u or v (which may even, within a small error, change their sign) result in large errors in the calculation of the wind direction.

To seek a remedy to this problem, three quantities from u and v are derived to act as (alternative) image channels. Firstly, we define the wind-speed magnitude V and the wind direction $\theta \in [-\pi, \pi]$:

$$V = \sqrt{u^2 + v^2} \tag{1}$$

$$\theta = \arctan 2\,(-u, v) \tag{2}$$

The $\arctan 2$ function calculates the inverse tangent, but it takes into account the signs of the arguments to choose the right quadrant; its behaviour is the same as the function `np.arctan2` from the Numpy Python library [47], or `tf.math.atan2` from TensorFlow [48]. The minus sign in $-u$ introduces an angular rotation so that $\theta = 0$ corresponds to a wind direction from the South to the North, $\theta = -\pi/2$ from the West to the East, $\theta = \pm\pi$ from the North to the South, and $\theta = \pi/2$ from the East to the West. There is still a discontinuity for the North to South direction, $\theta = \pm\pi$; we avoid this discontinuity by using $\sin\theta$ and $\cos\theta$. These two functions approach $\theta = \pm\pi$ smoothly. With these considerations, we build new images with three channels, instead of the previous two (u and v), for use with the cGAN model; the three channels are the wind-speed magnitude, V, the sine and the cosine of the wind direction, $\sin\theta$ and $\cos\theta$.

In this work, we analyse the performance of the cGAN model when using images with, alternatively, the two original channels (u and v) and with the three engineered channels (V, $\sin\theta$ and $\cos\theta$). In the following, we label these two alternative models as `u-v` and `V-sin-cos`. The AI model predicts the same channels as its input image. To convert V and θ back to u and v the following relations are applied:

$$u = V\cos\tilde{\theta}; \quad v = V\sin\tilde{\theta} \tag{3}$$

where $\tilde{\theta}$ is computed from the predictions of $\sin\theta$ and $\cos\theta$ given by the model, $\widehat{\sin\theta}$ and $\widehat{\cos\theta}$, as:

$$\tilde{\theta} = \arctan 2\left(\widehat{\cos\theta}, -\widehat{\sin\theta}\right) \tag{4}$$

3. Results and analysis

The models are trained on 214 events and evaluated on 80 events that are not part of the training set. The input data is obtained from NWP results, while the targets are CFD simulations, in both cases in the urban area described before and shown in Fig. 3. Data input, data output, and targets are arranged as matrices that can be interpreted as images, with varying resolutions and number of channels for the several analyses, or experiments, that will be presented in this work.

Two sets of experiments are carried out as follows. First, two models are trained using images with a resolution of 128 × 128, but with different channels, *viz* those described above as `u-v` model (which uses the velocity components as channels) and `V-sin-cos` model (which uses the velocity magnitude and the sine and cosine of the velocity direction as channels). The results are reported in Section 3.1. Then, four additional `u-v` and `V-sin-cos` models are trained with increasing resolutions of 256 × 256 and 512 × 512, to assess the influence of image resolution on the model performance. The results are reported in Section 3.2.

3.1. Analysis of feature engineering

In this section, the results for the `u-v` and `V-sin-cos` models using 128 × 128 images are presented and discussed. In Table 1, the Mean Absolute Errors (MAE) in wind speed and direction are presented. The MAE is defined as:

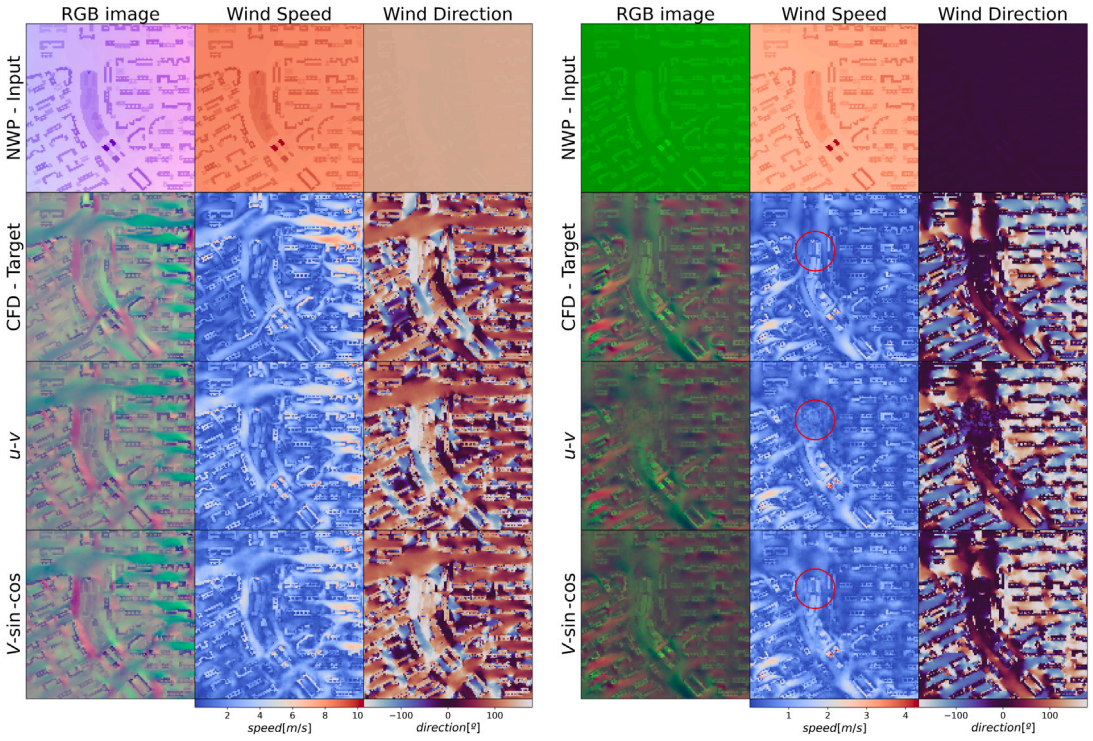
$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{5}$$

where y_i is the true value, $\hat{y}_i$ is the predicted value and n is the number of predicted points. The results indicate that `V-sin-cos` model outperforms `u-v` model, with a slight improvement of 0.07 m/s in wind speed, but a significant improvement of 6.8° in wind direction. Therefore, the benefit of using `V-sin-cos` model is an overall improvement in the prediction of the wind direction. This is a consequence of the codification in this approach of the wind direction with sine and cosine functions; this greatly improves the prediction of the wind direction in pixels where the wind components, u and v, are small. This is because, when the components are small, insignificantly small errors in their prediction can result in a large error in the wind direction when this is calculated through the components.

In Fig. 4, the outputs from the models for two events are shown; the outputs are presented alongside the model inputs (NWP) and targets (CFD) for `u-v` and `V-sin-cos` models. These two events are selected because they feature very different predominant wind directions, as shown in the upper right corner of Fig. 4(a) (northeasterly wind) and Fig. 4(b) (southerly wind). It can be seen that both models succeed in capturing the general features of the CFD results with a great level of detail. Although there are small discrepancies between target and prediction, for instance in wind speed magnitude, the models reproduce most of the wake development and eddies caused by the interaction between wind conditions and urban layout. There are no major differences between the models for the first of the events shown (Fig. 4(a)), but for the second event (Fig. 4(b)) `V-sin-cos` model clearly outperforms `u-v` model. In this second event, the forecast from `u-v` model is fuzzy in the central part of the modelled area (red circle in the central column of Fig. 4(b)), without a precise prediction of the flow or a clear appearance of the building footprint. In contrast, `V-sin-cos` model produces a sharp and well-defined output without blurring.

Fig. 5 presents the spatial distribution of MAE_{speed} and MAE_{dir} for `u-v` and `V-sin-cos` models averaged over all 80 events in the test set. For `V-sin-cos` model, smaller errors can be observed in both wind speed and wind direction. The performance of both models is markedly different in the inner courtyards of the buildings, particularly in the case of the wind direction. Although the accuracy of `u-v` model is significantly low in these areas, `V-sin-cos` model results in errors that are close to zero. These inner courtyards are sheltered from the wind by surrounding obstacles, and thus have low wind speed. According to these results, in these situations of low wind speed, `V-sin-cos` model outperforms `u-v` model due to the feature engineering carried out on the input images (see Section 2.5).

A linear regression is performed on the predicted values of `u-v` model and `V-sin-cos` model and the ground truth, the CFD data. R^2 values are presented in Table 2 for each of the wind components u and



(a) Event date and time: 2018-05-06 at 19:00.

(b) Event date and time: 2018-07-08 at 21:00.

Fig. 4. Comparison between `u-v` and `V-sin-cos` models for two events in the test dataset. *First row*: input data (NWP results); *second row*: target values (CFD results); *third row*: results from `u-v` model; *fourth row*: results from `V-sin-cos` model. *First column*: RGB representation of u and v velocity components (for `V-sin-cos` model, the values are obtained by converting from $\sin\theta$, $\cos\theta$ and V); *second column*: wind speed; *third column*: wind direction (0° blowing to the North, −90° to the East, 90° to the West and −180° and 180° to the South).

Table 2
R^2 values for u and v for `u-v` model and `V-sin-cos` model with resolution images of 128 × 128.

Model	u	v
`u-v`	0.8094	0.7529
`V-sin-cos`	0.8221	0.7597

v. R^2 is a statistical measure that indicates how similar these predicted values are to the true values. It is defined as:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{6}$$

where y_i are the observed values, $\hat{y}_i$ are the predicted values, $\bar{y}$ is the mean of the observed values, n is the number of observations. R^2 ranges from $-\infty$ to 1, where $R^2 = 1$ indicates a perfect agreement between predicted and true values, $R^2 = 0$ means that the predicted values are performing close to random data, and $R^2 < 0$ indicates that the predicted values are performing even worse than random predictions. From Table 2, R^2 values relatively close to 1 are observed, indicating a good performance of the models. Again, `V-sin-cos` model outperforms `u-v` model. The difference in R^2 between u and v, although small, is still noticeable for both models; the u component is systematically better predicted than the v one. This is likely due to the orientation of the axes in the model: the u velocity component is aligned with the prevailing wind direction, and thus it is probably easier to predict.

Fig. 6 reports for each test event the MAE values versus the mean wind speed and direction. For a fairer comparison among test events, for wind speed, we graph the relative MAE, defined as $\text{MAE}_{\text{speed}}$ divided by the mean value of the CFD predictions for wind speed in the event. `V-sin-cos` model (the model that predicts the velocity absolute value and the sine and cosine of the velocity direction) shows better performance than the alternative `u-v` model (which predicts the velocity components directly); `V-sin-cos` model results in lower MAE values and, overall, more uniform errors across the data set for the several mean wind speeds and directions in the test dataset. Further examination of the $\text{MAE}_{\text{speed}}$ plots, Figs. 6(a) and 6(c) reveals that, as the CFD-mean wind speed increases, the relative $\text{MAE}_{\text{speed}}$ decreases until it reaches an approximately constant value. Generally, stronger winds generate more consistent and directed flows through the urban fabric, which leads to wind fields having similar flow patterns. This pattern for stronger wind conditions makes them more amenable to prediction by the ML models. In the wind direction graphs, Figs. 6(b) and 6(d), the predominant characteristic wind directions of the geographical location are observed around −40 (southerly wind) and 40° (northwesterly wind). The contours in Fig. 6 are the error density estimation using Kernel Density Estimation (KDE) [49].

The hyperparameter λ balances the two elements of the generator loss in the model: the adversarial loss (the cross-entropy term) and the one that ensures that the generated images are structurally similar to the target images (the L1 error, *i.e.* the mean absolute error of the predicted images with respect to the target ones). For `u-v` model, a value of $\lambda = 100$ [28] provides a balance between the two losses during the training process. However, for `V-sin-cos` model the same value of $\lambda = 100$ is not suitable as the contributions of the adversarial loss and the L1 error to the total loss function of the model turn out not to be of a comparable magnitude. A more detailed exploration is carried out to find a value that ensures this balance during training; and finally $\lambda = 5$ is chosen. Furthermore, the contribution to the L1 error of V is multiplied by a factor of 10 because it is one order of magnitude smaller than those of $\sin\theta$ and $\cos\theta$. Overall, `V-sin-cos` model is found to perform better in the prediction of urban wind fields. However, this indicates that hyperparameter tuning may require additional attention when sophisticated feature engineering is used; for instance, the definition of the loss function may be crucially relevant

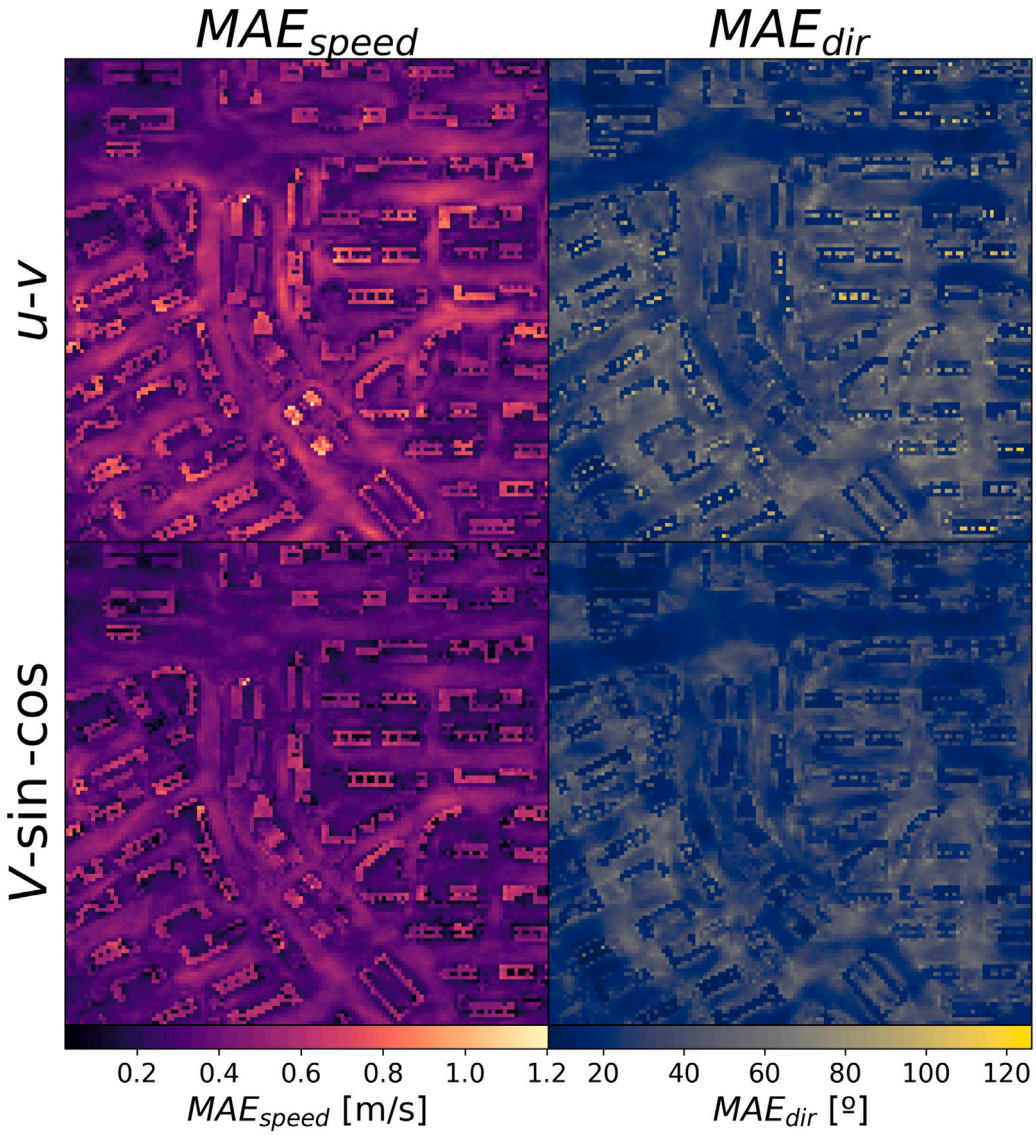


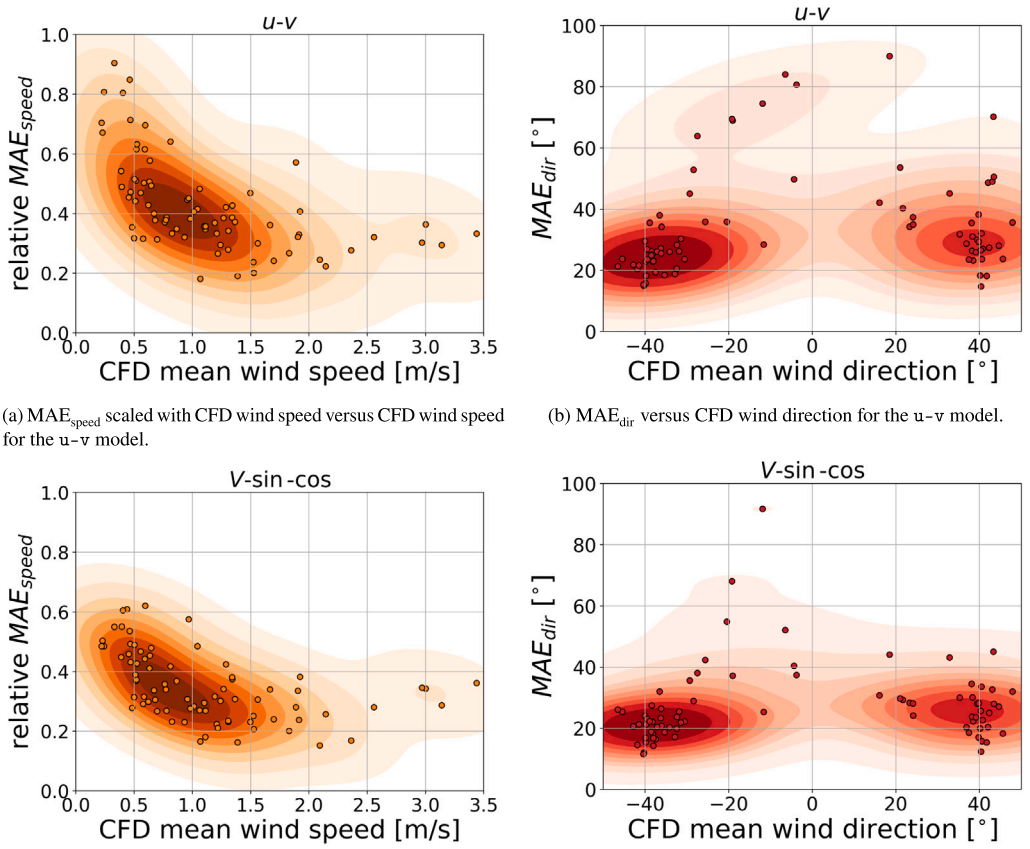
Fig. 5. Spatial MAE_{speed} and MAE_{dir} values for `u-v` model and `V-sin-cos` model averaged over all 80 events in the test set.

when the channels have different physical meanings, and therefore dissimilar values.

Next, we report the computational costs of the models. NWP and CFD simulations are run on a 6-core Intel i7-6800k CPU while AI model training and inferring are done on a NVIDIA GeForce RTX 4060 Ti GPU. The NWP mesoscale model requires 0.25 core-hours for the simulation of each hourly event, amounting to a total of 2190 core-hours (equivalent to 91 core-days) for a complete year comprising 8760 events. The CFD simulation of each event takes around 48 core-hours, thus requiring 588 core-days to simulate all 294 events in the training and test sets. For the AI models with 15,000 epochs, training takes 80 GPU-min for `u-v` model and 86 GPU-min for `V-sin-cos` model. The retrieval times of the results during the execution of the AI model are under 1 s for both AI models. Therefore, a drastic reduction in computational time is achieved, from 8 h for a traditional CFD simulation (on an 6-core machine) to just seconds using our AI model.

3.2. Influence of image resolution

Next, we investigate how the resolution of the images affects the quality of the results from the `u-v` and `V-sin-cos` models. In Fig. 7 MAE_{speed} and MAE_{dir} are presented for `u-v` and `V-sin-cos` models, trained with image resolutions of 128 × 128 (as presented in the previous section), 256 × 256 and 512 × 512. (Note that the images shown in Fig. 8 are not the same data up- or down-scaled; they are different output from three different models trained separately with different image resolutions.) We term each possible combination with the resolution size, followed by a dash and the model used; for instance, 256_`u-v` corresponds to `u-v` model trained with images with a resolution of 256 × 256. For image resolutions of 128 × 128 and 256 × 256 there is almost no difference in the errors, for either the `u-v` or the `V-sin-cos` model. However, for both models trained with 512 × 512 images, a slight reduction in errors is obtained: less than 0.02 m/s for wind speed and around 1° for wind direction. This difference behaviour with resolution increase is likely caused by the batch size selection rather than image resolution itself. The batch size refers to the number of training samples used to train a model in one iteration within an epoch. For the 128 × 128 and 256 × 256 images, a batch size of 128 is used, while for 512 × 512 it is reduced to 32 due to GPU memory limitations. Smaller batch sizes mean that more gradients are being computed to update the network weights, which is less memory demanding but more time-consuming and may result in instabilities and convergence issues during the training process.



(a) MAE_{speed} scaled with CFD wind speed versus CFD wind speed for the `u-v` model.

(b) MAE_{dir} versus CFD wind direction for the `u-v` model.

(c) MAE_{speed} scaled with CFD wind speed versus CFD wind speed for the `V-sin-cos` model.

(d) MAE_{dir} versus CFD wind direction for the `V-sin-cos` model.

Fig. 6. Relative MAE_{speed} and MAE_{dir} versus CFD wind speed and direction averaged for each of the 80 test events for 128 × 128 resolution `u-v` model and `V-sin-cos` model. Relative MAE_{speed} is MAE_{speed} divided by the mean CFD wind speed of the event.

Table 3
Training times for each model and image resolution. All times are reported in hours on a NVIDIA GeForce RTX 4060 Ti.

Resolution	`u-v` model	`V-sin-cos` model
128 × 128	1.3	1.4
256 × 256	4.9	5.8
512 × 512	20.4	23.7

In summary, there is no clear evidence that increasing the image resolution reduces the model errors. The main advantage of using high-resolution images for training is the gain in spatial detail, as seen in Fig. 8. The square domain that is represented in all images has approximately a 1 km side. A pixel in an image with 128 × 128 resolution is 7.8 × 7.8 m, with 256 × 256 resolution is 3.9 × 3.9 m, and with 512 × 512 is 1.9 × 1.9 m. A comparison of the training times for the various image resolutions is presented in Table 3. The training time increases roughly fourfold with resolution; this is to be expected as the amount of data (pixels in the image) the neural network has to process quadruples as the image resolution doubles.

4. Discussion and future research

The foregoing results have shown the ability of our image-based, cGAN model to predict local wind fields in a complex urban environment using coarser weather predictions as input. However, our cGAN models are trained on the specific urban region that will be the target of the predictions; therefore, the training datasets are domain-specific. The applicability of this approach would be greatly generalised if the model were capable of zero-shot modelling. Zero-shot modelling means that the ML model is able to predict scenarios or data that it has not been specifically trained on. In our application, this would allow the prediction of wind fields over geographical areas that the model has not seen before, that is, without needing to re-train the model with CFD results for the new areas.

Neural operators are useful in mapping the functional parametric space of partial differential equations (such as the Navier–Stokes equations that govern fluid flow) into its solutions. The Fourier neural operator does this in the Fourier space, which allows for an efficient architecture. Fourier neural operators have been shown to be capable of zero-shot modelling of turbulent flows [50]. Further research for this problem should explore the use of Fourier neural operators [50].

Different resolution images (128 × 128, 256 × 256 and 512 × 512) are explored in this work. The time and memory required for training have been shown to increase four-fold each time the resolution of the working image is doubled. In the case study, the real domain processed by the model has 1 km length; that is pixel sizes of about 8, 4, and 2 m for each of the resolutions. These are levels of detail in accordance with the requirements of standard applications such as urban wind energy generation or passerby comfort. To meet these same standards in possible real-case scenarios, where the processed domain is required to be much larger than 1 km (urban building planning of districts or monitoring an entire city), the image resolutions should increase. This may be a time inconvenience and a memory hurdle in terms of GPU hardware in order to train and allocate larger networks.

In this study, the Pix2Pix cGAN model is used to demonstrate the feasibility of the methodology; however, other models can be used

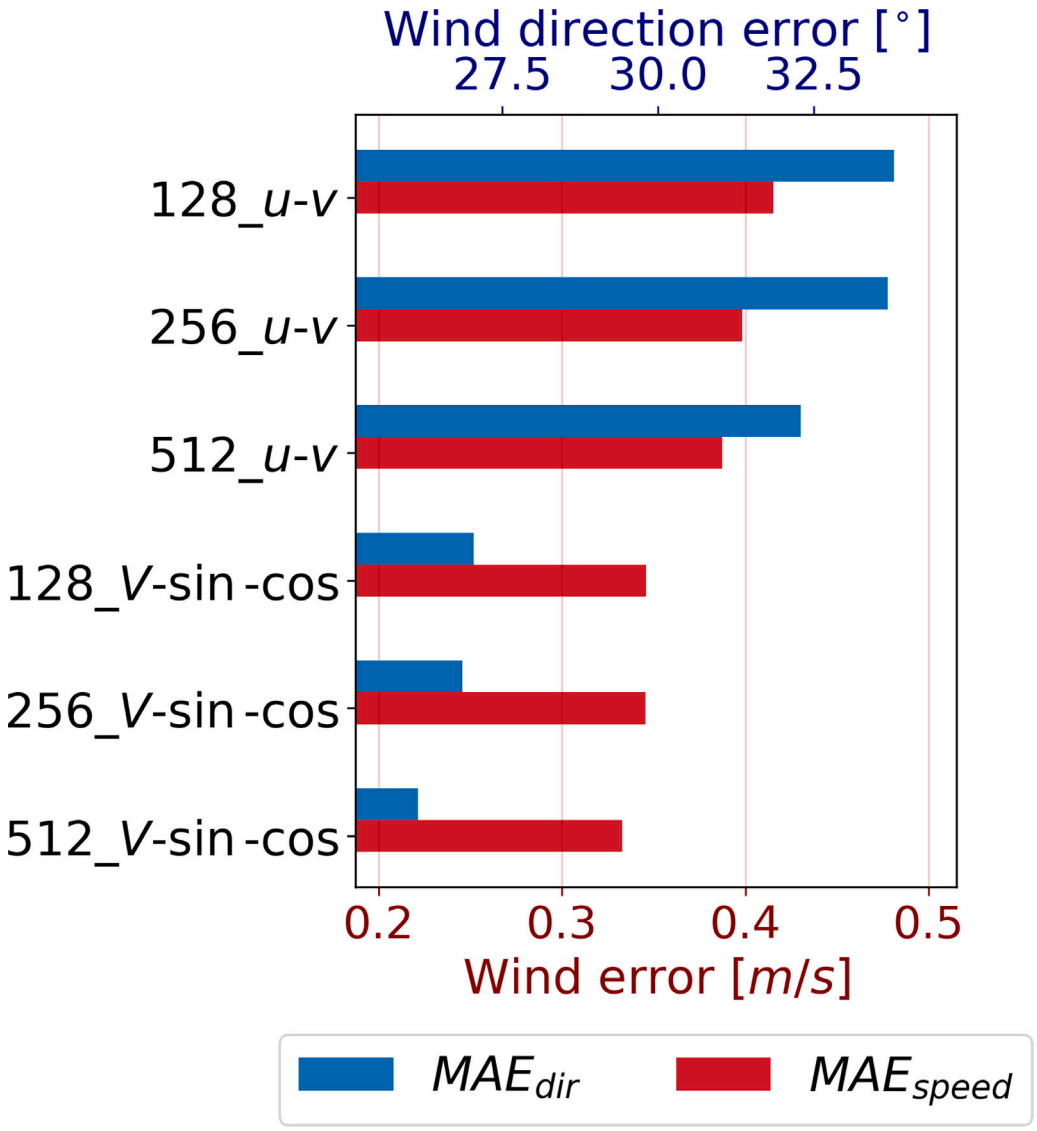


Fig. 7. MAE_{speed} and MAE_{dir} for `u-v` model and `V-sin-cos` model trained with image resolutions of 128 × 128, 256 × 256 and 512 × 512.

instead. Other state-of-the-art neural networks that have a great potential for use with this methodology are: EfficientNetV2 [51] with faster training times and a more efficient architecture; convolutional neural networks such as ConvNeXt [52]; and more recent approaches leveraging MobileNet [53] with visual transformers [54] in MobileFormer [55]. All of them can potentially be embedded in the proposed methodology; and their benefits in terms of errors, efficiency, and training times should be studied.

The aim of this paper is to explore the possibility of using AI to produce CFD-like results from NWP predictions. As indicated in Section 2.1, the true validation of the methodology is therefore by comparing AI-generated output with CFD results. Consequently, we have not validated the CFD results with, for instance, measured data. This is indeed a very complex endeavour for large, actual urban areas. Beyond the obvious CFD challenges (such as mesh resolution or turbulence modelling), an added difficulty is obtaining accurate boundary conditions. For a proper validation of our complete methodological chain, these boundary conditions should come from NWP predictions; but then these themselves are the results of numerical models, that should be separately validated; and therefore it is difficult to attribute any discrepancy in the CFD validation to any of the NWP or CFD methods, or to a combination of both. Work is nevertheless underway to validate the NWP predictions and our CFD results with wind field data in this urban setting. An intermediate validation, which provides controlled boundary conditions and therefore avoids the uncertainties associated with the NWP predictions, is to use a laboratory configuration of a complex urban setting, such as those investigated by Yoshie et al. [39]. Work in this direction is also underway, but it should be noted that any laboratory (wind-tunnel) configuration of an urban area will be at a reduced scale.

5. Conclusions

In this work, a cGAN model is used to get detailed CFD-like data from coarser NWP results for the wind field in an urban, highly built-up area. The essence of the proposed approach is to treat the CFD and NWP data as images; this allows the use of an image-to-image model, based on Pix2Pix [28], that is capable of translating coarse meteorological weather conditions (NWP) into the corresponding detailed wind field (CFD) over a built-up area.

The findings indicate that there is a significant agreement between the predicted data from the model and the actual CFD simulations, with a substantial decrease in computational time from eight hours required for a single CFD simulation to a few seconds using the AI model. Careful feature engineering of the image channels is shown to be effective

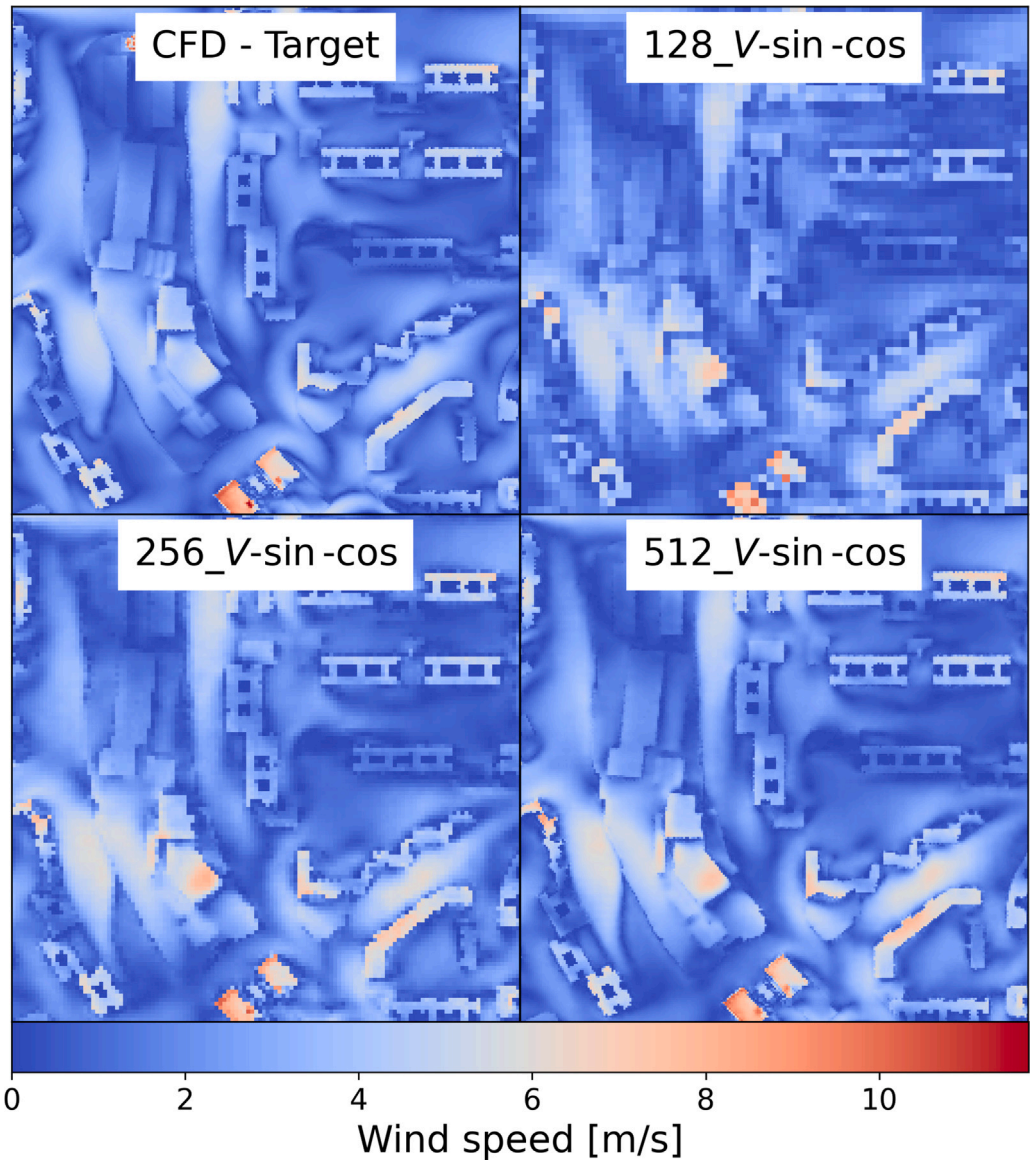


Fig. 8. Wind speed for the event 2018-03-12 at 22:00. Zoomed-in representation of the domain to show the differences in image resolution. *Upper left*: CFD data (ground truth) with a 512 × 512 resolution; *upper right*: predicted results with `V-sin-cos` model trained with images with 128 × 128 resolution; *bottom left*: predicted results with `V-sin-cos` model trained with images with 256 × 256 resolution; *bottom right*: predicted results with `V-sin-cos` model trained with images with 512 × 512 resolution;.

in reducing the error obtained by the model, especially in the wind direction. A mean absolute error in wind speed of 0.35 m/s is achieved, along with a wind direction error of 27.0°. In addition, the sensitivity of the model to the resolution of the input images is assessed; no clear evidence is found that increasing image resolution improves the model errors.

Funding

This work was partly funded by Grant DIN2019-010452 from MCIN/AEI/10.13039/501100011033, Spain; by the Departamento de Ciencia, Universidad y Sociedad del Conocimiento del Gobierno de Aragón, Spain, as Group T32_23R Tecnologías Fluidodinámicas; and by Project TED2021-131861B-I00, financed by MCIN/AEI/10.13039/-501100011033 (Spain) and by the European Union "NextGeneration-EU"/PRTR.

Software and data availability

- Input dataset for the NWP software: dataset `ds090.0` [35].
- NWP software: WRF v4.1 [36]
- CFD software: OpenFOAM v6 [37]
- AI suite Python framework: TensorFlow v2.8 [48]

CRediT authorship contribution statement

Jaime Milla-Val: Writing – review & editing, Writing – original draft, Software, Methodology, Investigation, Data curation, Conceptualization. **Carlos Montañés:** Writing – review & editing, Supervision, Methodology, Conceptualization. **Norberto Fueyo:** Writing – review & editing, Supervision, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

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

During the preparation of this work the authors used ChatGPT and Writefull occasionally for suggestions to improve the readability of some paragraphs. The authors reviewed any suggested wording and edited it as needed; they take full responsibility for the content of the publication.

References

[1] L. Kabošová, A. Chronis, T. Galanos, Fast wind prediction incorporated in urban city planning, Int. J. Archit. Comput. 20 (3) (2022) 511–527, http://dx.doi.org/10.1177/14780771221121034.

[2] J. Werner, D. Nowak, F. Hunger, T. Johnson, A. Mark, A. Gösta, F. Edelvik, Predicting wind comfort in an urban area: A comparison of a regression- with a classification-CNN for general wind rose statistics, Mach. Learn. Knowl. Extract. 6 (1) (2024) 98–125, http://dx.doi.org/10.3390/make6010006.

[3] T. Ishugah, Y. Li, R. Wang, J. Kiplagat, Advances in wind energy resource exploitation in urban environment: A review, Renew. Sustain. Energy Rev. 37 (2014) 613–626, http://dx.doi.org/10.1016/j.rser.2014.05.053.

[4] P. James, M. Sissons, J. Bradford, L. Myers, A. Bahaj, A. Anwar, S. Green, Implications of the UK field trial of building mounted horizontal axis micro-wind turbines, Energy Policy 38 (10) (2010) 6130–6144, http://dx.doi.org/10.1016/j.enpol.2010.05.070, The socio-economic transition towards a hydrogen economy - findings from European research, with regular papers.

[5] A. K.C., J. Whale, T. Urmee, Urban wind conditions and small wind turbines in the built environment: A review, Renew. Energy 131 (2019) 268–283, http://dx.doi.org/10.1016/j.renene.2018.07.050.

[6] F. Toja-Silva, T. Kono, C. Peralta, O. Lopez-Garcia, J. Chen, A review of computational fluid dynamics (CFD) simulations of the wind flow around buildings for urban wind energy exploitation, J. Wind Eng. Ind. Aerodyn. 180 (2018) 66–87, http://dx.doi.org/10.1016/j.jweia.2018.07.010.

[7] C. Xi, C. Ren, F. Haghighat, S.-J. Cao, Improving the urban wind flow prediction efficiency of target area by considering its surrounding buildings impact, Energy Build. 303 (2024) 113815, http://dx.doi.org/10.1016/j.enbuild.2023.113815.

[8] R. Vinuesa, S.L. Brunton, Enhancing computational fluid dynamics with machine learning, Nature Comput. Sci. 2 (6) (2022) 358–366, http://dx.doi.org/10.1038/s43588-022-00264-7.

[9] G. Calzolari, W. Liu, Deep learning to replace, improve, or aid CFD analysis in built environment applications: A review, Build. Environ. 206 (2021) 108315, http://dx.doi.org/10.1016/j.buildenv.2021.108315.

[10] Y. He, X.-H. Liu, H.-L. Zhang, W. Zheng, F.-Y. Zhao, M. Aurel Schnabel, Y. Mei, Hybrid framework for rapid evaluation of wind environment around buildings through parametric design, CFD simulation, image processing and machine learning, Sustainable Cities Soc. 73 (2021) 103092, http://dx.doi.org/10.1016/j.scs.2021.103092.

[11] N. BenMoshe, E. Fattal, B. Leitl, Y. Arav, Using machine learning to predict wind flow in urban areas, Atmosphere 14 (6) (2023) http://dx.doi.org/10.3390/atmos14060990.

[12] H.-C. Zhu, C. Ren, J. Wang, Z. Feng, F. Haghighat, S.-J. Cao, Fast prediction of spatial temperature distributions in urban areas with WRF and temporal fusion transformers, Sustainable Cities Soc. 103 (2024) 105249, http://dx.doi.org/10.1016/j.scs.2024.105249.

[13] L. Zheng, W. Lu, Q. Zhou, Weather image-based short-term dense wind speed forecast with a ConvLSTM-LSTM deep learning model, Build. Environ. 239 (2023) 110446, http://dx.doi.org/10.1016/j.buildenv.2023.110446.

[14] S.J. Low, Venugopalan, S.G. Raghavan, H. Gopalan, J.C. Wong, J. Yeoh, C.C. Ooi, FastFlow: AI for fast urban wind velocity prediction, 2022, http://dx.doi.org/10.48550/arXiv.2211.12035.

[15] Z. Liu, S. Zhang, X. Shao, Z. Wu, Accurate and efficient urban wind prediction at city-scale with memory-scalable graph neural network, Sustainable Cities Soc. 99 (2023) 104935, http://dx.doi.org/10.1016/j.scs.2023.104935.

[16] 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 (2023) 110056, http://dx.doi.org/10.1016/j.buildenv.2023.110056.

[17] P. Liang, C. Deng, X. Yuan, L. Zhang, A deep capsule neural network with data augmentation generative adversarial networks for single and simultaneous fault diagnosis of wind turbine gearbox, ISA Trans. 135 (2023) 462–475, http://dx.doi.org/10.1016/j.isatra.2022.10.008.

[18] S. Yang, Y. Zhou, X. Chen, C. Deng, C. Li, Fault diagnosis of wind turbines with generative adversarial network-based oversampling method, Meas. Sci. Technol. 34 (4) (2023) 044004, http://dx.doi.org/10.1088/1361-6501/acad20.

[19] G.B. Santos, A.V. Pantaleão, L.O. Salviano, Using deep generative adversarial network to explore novel airfoil designs for vertical-axis wind turbines, Energy Convers. Manage. 282 (2023) 116849, http://dx.doi.org/10.1016/j.enconman.2023.116849.

[20] X. Zhang, D. Li, X. Fu, A novel wasserstein generative adversarial network for stochastic wind power output scenario generation, IET Renew. Power Gener. (2024) http://dx.doi.org/10.1049/rpg2.12932.

[21] L. Ye, Y. Peng, Y. Li, Z. Li, A novel informer-time-series generative adversarial networks for day-ahead scenario generation of wind power, Appl. Energy 364 (2024) 123182, http://dx.doi.org/10.1016/j.apenergy.2024.123182.

[22] R.K. Behara, A.K. Saha, Analysis of wind characteristics for grid-tied wind turbine generator using incremental generative adversarial network model, IEEE Access 12 (2024) 38315–38334, http://dx.doi.org/10.1109/ACCESS.2024.3372862.

[23] H. Li, Q. Yang, T. Li, Wind turbine wake prediction modelling based on transformer-mixed conditional generative adversarial network, Energy 291 (2024) 130403, http://dx.doi.org/10.1016/j.energy.2024.130403.

[24] M. Mirza, S. Osindero, Conditional generative adversarial nets, 2014, http://dx.doi.org/10.48550/arXiv.1411.1784.

[25] J. Zhang, X. Zhao, Wind farm wake modeling based on deep convolutional conditional generative adversarial network, Energy 238 (2022) 121747, http://dx.doi.org/10.1016/j.energy.2021.121747.

[26] P. Kastner, T. Dogan, A GAN-based surrogate model for instantaneous urban wind flow prediction, Build. Environ. 242 (2023) 110384, http://dx.doi.org/10.1016/j.buildenv.2023.110384.

[27] S. Masoumi-Verki, F. Haghighat, N. Bouguila, U. Eicker, The use of GANs and transfer learning in model-order reduction of turbulent wake of an isolated high-rise building, Build. Environ. 246 (2023) 110948, http://dx.doi.org/10.1016/j.buildenv.2023.110948.

[28] P. Isola, J.-Y. Zhu, T. Zhou, A.A. Efros, Image-to-image translation with conditional adversarial networks, 2016, http://dx.doi.org/10.48550/arXiv.1611.07004.

[29] Y. Jiang, L. Jiang, S. Yang, C.C. Loy, Scenimefy: Learning to craft anime scene via semi-supervised image-to-image translation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, ICCV, 2023, pp. 7357–7367, http://dx.doi.org/10.48550/arXiv.2308.12968.

[30] T. Karras, S. Laine, T. Aila, A style-based generator architecture for generative adversarial networks, 2019, arXiv:1812.04948.

[31] Y. Dalva, H. Pehlivan, O.I. Hatipoglu, C. Moran, A. Dundar, Image-to-image translation with disentangled latent vectors for face editing, IEEE Trans. Pattern Anal. Mach. Intell. 45 (12) (2023) 14777–14788, http://dx.doi.org/10.1109/TPAMI.2023.3308102.

[32] B. Sun, Y. Mei, N. Yan, Y. Chen, UMGAN: Underwater image enhancement network for unpaired image-to-image translation, J. Mar. Sci. Eng. 11 (2) (2023) http://dx.doi.org/10.3390/jmse11020447.

[33] J. Chen, S. Chen, L. Wee, A. Dekker, I. Bermejo, Deep learning based unpaired image-to-image translation applications for medical physics: a systematic review, Phys. Med. Biol. 68 (5) (2023) 05TR01, http://dx.doi.org/10.1088/1361-6560/acba74.

[34] D.A. Romero, S. Hasanpoor, E.G.A. Antonini, C.H. Amon, Predicting wind farm wake losses with deep convolutional hierarchical encoder–decoder neural networks, APL Mach. Learn. 2 (1) (2024) 016111, http://dx.doi.org/10.1063/5.0168973.

[35] National Centers for Environmental Prediction, National Weather Service, NOAA, U.S. Department of Commerce, NCEP/NCAR global reanalysis products, 1948-continuing, 1994, URL: https://rda.ucar.edu/datasets/ds090.0/.

[36] W. Skamarock, J. Klemp, J. Dudhia, D. Gill, L. Zhiquan, J. Berner, W. Wang, J. Powers, M.G. Duda, D.M. Barker, X.-Y. Huang, A Description of the Advanced Research WRF Model Version 4, NCAR Technical Note NCAR/TN-475+STR, 2019, p. 145, http://dx.doi.org/10.5065/1dfh-6p97.

[37] H.G. Weller, G. Tabor, H. Jasak, C. Fureby, A tensorial approach to computational continuum mechanics using object-oriented techniques, Comput. Phys. 12 (6) (1998) 620, http://dx.doi.org/10.1063/1.168744.

[38] P. Spalart, S. Allmaras, A one-equation turbulence model for aerodynamic flows, AIAA 439 (1992) http://dx.doi.org/10.2514/6.1992-439.

[39] R. Yoshie, A. Mochida, Y. Tominaga, H. Kataoka, K. Harimoto, T. Nozu, T. Shirasawa, Cooperative project for CFD prediction of pedestrian wind environment in the architectural institute of Japan, J. Wind Eng. Ind. Aerodyn. 95 (9) (2007) 1551–1578, http://dx.doi.org/10.1016/j.jweia.2007.02.023.

[40] 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 (2023) 109947, http://dx.doi.org/10.1016/j.buildenv.2022.109947.

[41] T. Talwar, C. Yuan, Impact of natural urban terrain on the pedestrian wind environment in neighborhoods: A CFD study with both wind and buoyancy-driven scenarios, Build. Environ. 261 (2024) 111746, http://dx.doi.org/10.1016/j.buildenv.2024.111746.

[42] O. Ronneberger, P. Fischer, T. Brox, U-Net: Convolutional networks for biomedical image segmentation, 2015, http://dx.doi.org/10.48550/arXiv.1505.04597.

[43] A.L. Maas, A.Y. Hannun, A.Y. Ng, et al., Rectifier nonlinearities improve neural network acoustic models, in: International Conference on Machine Learning, 2013, p. 3.

[44] V. Nair, G.E. Hinton, Rectified linear units improve restricted boltzmann machines, in: Proceedings of the 27th International Conference on Machine Learning (ICML-10), 2010, pp. 807–814.

[45] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2017, arXiv:1412.6980.

[46] J. Milla-Val, C. Montañés, N. Fueyo, Economical microscale predictions of wind over complex terrain from mesoscale simulations using machine learning, Model. Earth Syst. Environ. 10 (1) (2024) 1407–1421, http://dx.doi.org/10.1007/s40808-023-01851-x.

[47] C.R. Harris, K.J. Millman, S.J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N.J. Smith, R. Kern, M. Picus, S. Hoyer, M.H. van Kerkwijk, M. Brett, A. Haldane, J.F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, T.E. Oliphant, Array programming with NumPy, Nature 585 (7825) (2020) 357–362, http://dx.doi.org/10.1038/s41586-020-2649-2.

[48] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, 2015, URL: https://www.tensorflow.org/. Software available from tensorflow.org.

[49] M. Rosenblatt, Remarks on some nonparametric estimates of a density function, Ann. Math. Stat. 27 (3) (1956) 832–837, http://dx.doi.org/10.1214/aoms/1177728190.

[50] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, A. Anandkumar, Fourier neural operator for parametric partial differential equations, 2021, arXiv:2010.08895.

[51] M. Tan, Q.V. Le, EfficientNetV2: Smaller models and faster training, 2021, arXiv:2104.00298.

[52] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, S. Xie, A ConvNet for the 2020s, 2022, arXiv:2201.03545.

[53] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, MobileNets: Efficient convolutional neural networks for mobile vision applications, 2017, arXiv:1704.04861.

[54] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16x16 words: Transformers for image recognition at scale, 2021, arXiv:2010.11929.

[55] Y. Chen, X. Dai, D. Chen, M. Liu, X. Dong, L. Yuan, Z. Liu, Mobile-former: Bridging MobileNet and transformer, 2022, arXiv:2108.05895.