A Deep Learning Approach for Pedestrian Wind Comfort Prediction for Generative Design Processes

Linde van Beers¹, Mike Jäger², Pia Riedel², Veronika Heidegger¹, Henry Sallandt²
¹Arup, Amsterdam, the Netherlands
²Arup, Berlin, Germany

Abstract
This paper proposes a novel approach for integrating wind analysis in generative design processes for high rise buildings. In this context, identifying potential problem areas around the development in a very short timeframe, using indicative information, is more important than highly accurate analysis results complying with codes and guidelines. This level of data reliability and speed can possibly be achieved using machine learning techniques. A deep learning approach is proposed using a conditional generative adversarial network (cGAN) to train a surrogate model that can be integrated in a widely used parametric design software (Grasshopper 3D). This allows for wind-comfort assessment to be made available to designers and integrated seamlessly in generative design processes.

The focus of the paper is to compare 3 different models which were trained with different training datasets as the selection of the training dataset showed significant impact on the performance of the model.

Highlights
- Pedestrian wind comfort prediction using surrogate models
- Comparison of the impact of different city representations on the quality of the prediction
- Possibility to integrate surrogate models in early stage generative design processes

Introduction
Wind comfort assessments with traditional computational fluid dynamics (CFD) methods are computationally heavy and time consuming, requiring expert knowledge. The high (computational) cost means that wind analysis is often postponed to later design stages, while ideally many different building configurations should be tested early in the planning and design process, when high-level changes can still be made. In order to integrate wind comfort in generative design processes, where hundreds of options are to be evaluated on several performance indicators, the return time of the wind analysis must be shortened drastically.

There are multiple alternative numerical approaches with the goal of reducing the return time of CFD analyses, such as Fast Fluid Dynamics (FFD), Lattice-Boltzmann-Method or Discrete Velocity Method. All of them vary in terms of accuracy, computational cost, and the trade-off between both. FFD was further explored in the context of the application in building simulations. Originally applied by Stam (2003) to realistically animate fluid-like effects for games FFD runs in real time for reasonable grid sizes. Waibel et al. (2017), who applied this method to predict air flow around buildings in early design stage, conclude that it is limited in capturing turbulent flows especially in wake regions due the lack of a turbulence model and a problem of numerical diffusion. Zuo (2009) states FFD to be 50 times faster than traditional CFD.

The most promising approach for further speed up these analyses is building a surrogate model via machine learning (ML) techniques. Previous studies on this topic have been conducted with varying levels of success.

Related Work
Mokhtar et al. (2020) show the feasibility of a ML surrogate in the context of wind prediction by using a conditional generative adversarial network (cGAN) architecture called pix2pix. They investigate the performance for training with both a simpler and a more complex dataset. The latter contains urban grids and morphologies within a boundary of 550 m. The authors test different resolutions and color maps of the input images. The performance is assessed by calculating the mean absolute error. They state that ‘the developed model is capable of generating pedestrian wind flow approximations for a wide range of urban configurations at an accuracy of about 0.3 m/s’. Furthermore, the work holds an extensive summary of limitations and challenges that the authors faced, e.g. the decreasing performance of the prediction for morphologies that deviate further from the trained dataset as well as the geometry encoding being limited to continuously solid buildings.

Mokhtar et al. (2021) extent their work and tackle the latter limitation by implementing an encoding for complex geometry representations, for subtractive local features (to capture e.g. tunnels or curved balconies), for the topography as well as for additive local features such as trees and shading structures. As a result it is stated that they ‘achieve an average of 0.2 absolute wind factor error, with minor differences across test data characteristics. The range is within 1 m/s which is a reasonable deviation for cities with wind speeds below 5 m/s, and reflects the merit of [their] approach’. The authors did not tackle decrease of performance associated with a distribution.
shift within the test data. Moreover, they noted some artifacts and grainy noise in the results. This is connected to limits in resolution and the training set size due to computational costs of increased amount and finer resolved CFD analyses. Furthermore, the framework is not able to encode and therefore correctly predict curved surfaces.

Huang et al. (2022) present a workflow for auto parameter tune to support the parametric model design process focusing on the performance of pedestrian wind comfort level, solar radiation, and thermal comfort criteria (UTCI). They build a surrogate model based on a cGAN to accelerate the prediction of mentioned comfort parameter. They state that their workflow, which was trained with a large amount of simulation results of typical European urban block morphologies, achieves real-time optimization with a speed up of 120-240 times compared to numerical simulation method. In terms of accuracy, they conclude to be close to the upper limit for data-driven methods as further increase of their dataset and hyperparameter tuning will not lead to significant changes.

The research presented in this paper builds upon the work of Mokhtar et al. as it is based on a similar workflow with the aim to generate wind flow images that are close to CFD simulation results to speed up and inform the design process. In addition to the cGAN architecture used in Mokhtar et al. (2019) the present study investigates an adapted model architecture using only the generator part consisting of an U-Net architecture as proposed by Ronneberger et al. (2015) but without the discriminator part of the cGAN. Furthermore, it compares the performance of three different training datasets that vary in complexity of city block morphologies, city density as well as the building area, that was simulated with CFD. The surrogate ML model is integrated into a prototype tool in a parametric design software (Grasshopper 3D). It is capable of providing a prediction of the wind flow at pedestrian height within an analysis area of 256 m by 256 m for a given building geometry within 2 s. The performance is evaluated on a hold-out validation dataset as well as for a case study.

**Methods**

A deep learning approach is proposed using an image-to-image translation approach (U-Net architecture) for training a surrogate model to predict wind flows in a city environment. The aim is to integrate the trained surrogate model in a widely used parametric design software (Grasshopper 3D), so that it can be used in early-stage generative design processes.

On a high level the proposed workflow consists of 3 steps:

1. Data generation
2. Model training
3. Using surrogate model for wind flow prediction

The following sections address the above points in more detail.

**Data generation**

**Representation of the city geometry**

The 3D city is represented as a 2D single-channel greyscale image encoding the heights of buildings in their respective location. A total area of 256 m x 256 m is used and each square meter in the urban landscape is represented by 1 pixel. The value of each pixel corresponds to the height of the building. A value of 255 (white) means no building is there whereas a value of 0 (black) means a 2m high building. The greyer the pixel value (max. 243) the higher the building. Any buildings under 2 m high are ignored, while the maximum height is set at 200 m, above which buildings will be represented as being 200 m high. These values were chosen in line with typical projects completed in Europe over the past few years.

![Figure 1: Example of representation of building height picture for training the machine learning model](image1)

**Representation of the wind environment**

The wind environment is likewise represented in an image with the same characteristics as the city geometry representation, but rather than representing the building height, each pixel’s value represents the speedup factor of the wind at the centre of the corresponding area. The speedup factor $S$ is defined as:

$$S = \frac{u_x}{u_0}$$

where $u_x$ is the predicted speed at the location and $u_0$ is the input windspeed equal to the reference velocity at a reference height of 10 m. At locations where buildings are the speedup factor is set to pixel value of 255 (white). The maximum speedup factor is 1.75 with a pixel value of 243.

![Figure 2: Example of representation of wind environment with speedup factors for training the machine learning model](image2)
**Synthetic city generation**

The datasets for training the machine learning models consist of parametrically generated city landscapes on which wind simulations are performed with computational fluid dynamics (CFD) to yield corresponding wind environments. Three different training datasets that vary in complexity of city block morphologies (e.g. maximum building height), average density as well as the building area, that was simulated with CFD are used for training and for assessment in this research. Training dataset 1 contains small clusters of around 4 buildings, sitting in the center of the prediction area. For training dataset 2, a larger city of 1 km squared was created and simulated, from which samples of 256 m x 256 m where then taken. The same principle was applied to training dataset 3, except that the city size is increased to 1.75 km squared. The advantage of the latter two approaches is that more image data can be extracted using fewer CFD simulations and therefore less computational time, because overlapping images can be taken from each simulation result. However, this approach is based on the assumption that the larger city topology does not slow down the wind significantly and that the wind profile is kept roughly constant also within city terrain.

**Table 1- Overview of the 3 training datasets**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Training dataset 1</th>
<th>Training dataset 2</th>
<th>Training dataset 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area</td>
<td>256 m x 256 m</td>
<td>1 km x 1 km</td>
<td>1.75 km x 1.75 km</td>
</tr>
<tr>
<td>Max height</td>
<td>50 m</td>
<td>200 m</td>
<td>200m</td>
</tr>
<tr>
<td>Samples per simulation</td>
<td>1</td>
<td>36</td>
<td>64</td>
</tr>
<tr>
<td>Training samples</td>
<td>176</td>
<td>655</td>
<td>995</td>
</tr>
<tr>
<td>Validation samples</td>
<td>20</td>
<td>280</td>
<td>427</td>
</tr>
<tr>
<td>Average density</td>
<td>0.231</td>
<td>0.355</td>
<td>0.442</td>
</tr>
</tbody>
</table>

Training dataset 1 contains buildings with courtyards and towers with or without plinths, where all towers are rectangular with sharp corners. Training dataset 2 introduces in addition towers with rounded corners and small buildings of 3 to 10 m in each dimension. Training dataset 3 has more flexibility in the placement of towers and enforces some courtyard buildings to not be fully closed as well as varying their depth. Examples of all three datasets are visualised in Figure 3.

**Figure 3: Examples of the chosen training data sets for representation city landscape for the different models**

**CFD wind flow simulation**

The created geometry is placed in a numerical wind tunnel and simulated using a steady state incompressible RANS solver of OpenFOAM called simpleFoam with the turbulence model implementation kOmegaSST.

**Table 2: Overview over the boundary conditions used for the wind tunnel**

<table>
<thead>
<tr>
<th>Location</th>
<th>Velocity</th>
<th>Pressure</th>
<th>Turbulence fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inlet</td>
<td>Atmospheric boundary layer</td>
<td>$\partial_n p = 0$</td>
<td>Atmospheric boundary layer</td>
</tr>
<tr>
<td>Sky</td>
<td>fixed shear stress</td>
<td>$\partial_n p = 0$</td>
<td>$\partial_n \varphi = 0$</td>
</tr>
<tr>
<td>Sides</td>
<td>$U \cdot n = 0$</td>
<td>$\partial_n p = 0$</td>
<td>$\partial_n \varphi = 0$</td>
</tr>
<tr>
<td>Ground + Buildings</td>
<td>$</td>
<td>U</td>
<td>= 0$</td>
</tr>
<tr>
<td>Outlet</td>
<td>$\partial_n U = 0$</td>
<td>$p = p_{total}$</td>
<td>$\partial_n \varphi = 0$</td>
</tr>
</tbody>
</table>

At the inlet of the domain a logarithmic wind profile is set with $z_0 = 0.3m$ as characteristic roughness length, $U_{ref} = 10ms^{-1}$ and $Z_{ref} = 50m$. OpenFOAM’s implementation of the atmospheric boundary profile is used for the velocity and turbulence fields. The sides of the wind tunnel are set to be slip walls, the ground and buildings are flat no slip walls. At the sky a fixed shear stress is set to ensure the conservation of the wind profile at the sky. The outlet uses zero gradient boundary conditions in all fields except for the pressure where the total pressure is set to be a constant value. From the 3D solution a plane at 2m height is extracted which is the basis for the ground truth images.
Model training

Model architecture

In order to make a prediction of the wind environment based on an image representing building geometry, a surrogate model is trained. The machine learning model in question is based on the U-Net architecture as proposed by Ronneberger et al. (2015).

The loss function used was that of mean pixel-wise absolute error (MAE). This loss function does not ensure a natural-looking gradual wind pattern as we are used to getting from CFD. To improve the visual quality of the output, but not necessarily the overall accuracy, the model can be extended to a conditional generative adversarial network as used by Mokhtar et al. (2020). In this case a discriminator is introduced that is trained on determining whether a wind environment image is ‘real’, i.e. made through CFD, or ‘fake’, i.e. a prediction from the surrogate model. Next to MAE, the U-Net loss function is then additionally defined by how ‘real’ the discriminator deems the prediction to be.

Model training

The U-Net architecture was trained with each of the three training datasets described in Table 1. In the following sections model 1 refers to the surrogate model trained with training dataset 1, model 2 to a model trained with dataset 2 and model 3 consequently to a model trained with dataset 3. Model 2 is the only model that was additionally evaluated using the adapted model architecture including the discriminator part to obtain a full GAN architecture. The splits used to divide the dataset into training set and the hold-out validation are shown in Table 1. Both are drawn from the same distribution. The training for each model was performed for 100 epochs using the Adam optimizer with a learning rate of 0.0002.

Method for judging model performance

To assess the performance of the machine learning model, the prediction and CFD images are compared by putting the calculated and predicted speedup factors into different buckets.

For each image the amount of pixels within each bucket is accumulated and divided by the total amount of pixels for which a wind speed was predicted. This results in a percentage occurring of each bucket for one image.

The difference between the size of these buckets for the predicted and calculated results shows if the machine learning model over- or underpredicts the respective speedup factors.

Knowing that the accuracy of the machine learning model decreases near the edges of the image only pixels within the middle section, around the actual area of interest, are evaluated. This is based on the assumption that the building of interest is always located in the center of the image.
Results

Results of synthetic city generation

To check whether the created geometry is in fact comparable to real city geometries, the building height distribution of the generated city is evaluated and compared to a collection of places in Europe (selected use cases for this research). Figure 5 shows a comparison of the building height distributions of the different training datasets and the example use cases.

Figure 5: Comparison of the building height distributions of the different training datasets and the example use cases (in Europe).

In comparison to the example use cases dataset 1 overrepresents buildings between 5 – 10 m height. The building height distribution for dataset 3 aligns with the example cases very well. However, this is just one parameter how training data can be prepared in a reasonable way.

Training results

During training, the performance of the model was judged by its performance on a hold-out validation set drawn from the same distribution as the training data. Especially the MAE and RMSE with regard to the speedup factor was considered. Table 3 shows the performance of the different models trained with the U-Net architecture. It also shows the performance of the model trained with the second dataset, but with the addition of a discriminator, i.e. the cGAN architecture, for comparison.

Figure 6 shows the errors throughout training for the U-Net and cGAN versions of model 2. We see that while significant learning occurs for the training set, this improvement does not generalise to unseen cases. In fact no real improvement occurred in the results on the validation set after approximately 40-50 epochs. This behaviour consistently occurred throughout the different models.

Table 3 - Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) on validation set after 100 epochs of training on for different models and datasets, errors calculated on speedup factor.

<table>
<thead>
<tr>
<th>Loss</th>
<th>MAE</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model 1 (U-net)</td>
<td>0.070</td>
<td>0.125</td>
</tr>
<tr>
<td>Model 2 (U-net)</td>
<td>0.076</td>
<td>0.120</td>
</tr>
<tr>
<td>Model 2 (cGAN)</td>
<td>0.077</td>
<td>0.122</td>
</tr>
<tr>
<td>Model 3 (U-net)</td>
<td>0.086</td>
<td>0.145</td>
</tr>
</tbody>
</table>

Results on case studies

The results for prediction on real use cases show that models 1-3 can in general identify areas with high wind speeds at pedestrian level. The models learn that introducing a high building result in higher wind intensity near that building. This applies to all used datasets shown in Figure 3. Figure 7 exemplarily illustrates the described outcome for dataset 2.
Figure 7: Results with dataset 2 showing that the machine learning model learned that a high building means higher intensity in velocity.

In this example, the predictions show lower speedup factors than the CFD simulation. It can be concluded that the prediction overrepresents lower wind speeds.

One reason for this is that the training data of dataset 2 was trained with relatively too low speedup factors. Figure 8 shows the speedup factor distribution for the training images for all 3 different training datasets.

Dataset 2 shows more values in the lower speeds (0 – 0.4375) in comparison to the example cases. Therefore, this model could be biased to predict lower speedup factors.

Evaluation of model performance

To assess the model performance the described methodology in section Method for judging model performance is used. Figure 9 shows the results for model 1-3. Each group of 3 bars relates to one model trained with the respective training dataset. Each single bar represents one CFD case of the example cases. The first example case (first bar of a group of three) contains a 60m high tower, the second example case a 90m high tower and the third example case includes no tower.

A positive value (indicated with green bars) represents an overprediction of the trained model in the respective speedup factor bucket. Overprediction means that in the predicted image more pixels show speedup factors in that bucket in comparison to the CFD result. A negative value (indicated with red bars) shows an underprediction of the trained model in comparison with the CFD results. Therefore, less pixels show speedup factors in that bucket in comparison to the CFD result. A paler bar represents cases where the trained model did not predict any values in that speedup factor bucket, whereas the CFD simulation did.

In general, the trained models 2 and 3 show a clear tendency of overprediction for low speedup factors (< 0.25). Trained model 1 shows a balanced distribution of over- and underprediction for all speedup factor buckets.

Another indicator that the trained models 2 and 3 overpredict low speedups is that the predictions do not show any speedup factors greater than 0.5 for model 3 and greater than 0.75 for model 2 although the CFD shows this (paler red bars in figure 9). In comparison trained model 1 even shows values for speedups greater than 1 for one example case. There are no bars for the 3rd example case within the speedup factors of 0.75 – 1 and 1 – 1.25 for trained model 2 and 3. This means that the CFD and the prediction does not show any values in that bucket. Trained model 1 does show speedups between 0.75 – 1 as a green bar is shown in the figure for that bucket.

The absolute average deviation between prediction and CFD for all speedup factor buckets of trained model 1 is 7.3%, trained model 2 is 11.6% and trained model 3 is 23.2%.
Results on generative design use case

The trained model can provide quick predictions (around 20 seconds), considerably lowering the cost of a pedestrian wind comfort assessment. This enables its integration in early-stage design evaluation.

Additionally, the trained model was successfully integrated in parametric design workflows through Grasshopper 3D, making it available and easy-to-use for designers.

Metrics are derived from the model prediction to indicate the effect of the current building configuration on the pedestrian wind comfort. Initial tests show that the model outputs are promising, although the accuracy of the prediction needs further investigation at this stage.

Discussion

Discussion of results

Trained model 1 performs best based on the results shown in this paper. This can be partially explained through the different sizes of the domain of the different data sets. In the training dataset for model 1 one block of buildings is simulated whereas the domain size was gradually increased with training dataset 2 and 3. This increase was done to generate more samples per simulation (see table 1) with a tiling approach. This approach has been tested for applicability. However, this is especially influential when the general density is increased to create a more realistic city. Within training dataset 3 there are lot of small wind speeds in it. Therefore, this trained model shows a tendency to overpredict small speedups.

The results give an insight what type of data is most suitable for this machine learning model. A model trained with a simple dataset (dataset 1) with no buildings at the edges of the images is sufficient to predict reasonable wind flows. For high level wind flow predictions, that are used to locate areas of potential wind accelerations in an early design stage, it is not necessary to train a model with complex data such as having surroundings near the area of interest.

Further experimentation with different training data and applying a transfer learning component to the workflow could allow for even more accurate predictions.

Conclusion

The research has shown that there are several opportunities, but also challenges with the proposed approach.

Opportunities

- Possibility to train a machine learning model with 2d representations of city geometry and calculated wind speeds (using CFD)
- Possibility to generate a synthetic city training dataset that is representative for real case studies in Europe.
- Using a surrogate model for predicting wind flow in the early design stage is promising. The proposed approach provides quick results, which fits in with the requirements of early stage generative design.
- Possibility to integrate a surrogate model into common parametric design software (Grasshopper 3D).

Challenges

- The quality of the prediction highly depends on the training dataset.
- The accuracy of the produced results requires further investigation.
- The cost of running CFD simulation to generate a large enough dataset remains substantial.

Future work

Based on above challenges and opportunities, a few suggestions for follow up work can be formulated:

- Representative dataset: The results of this research have highlighted the impact of the training data set on the quality of the results. It is therefore suggested to carry out further extensive testing to determine what a representative dataset looks like and how much training data is needed to generate results with an acceptable accuracy.
- Effect of surroundings: Since a building is never isolated in a city environment, the wind flows at its base will always be partially dictated by the shape of its surrounding buildings. In the current approach, only a small portion of surrounding buildings is included in the model training. A suggestion for further research would be to increase the portion of surroundings included in the model training.
- Results on different heights: The current approach only allows to predict wind speeds at ground level. However, when designing a building also the wind comfort levels on balconies and terraces are an important factor. It is therefore suggested to extend the model training to allow for predicting wind flows at different heights.
- Generative design workflow: The research has shown the possibility to integrate a surrogate model in commonly used generative design software (Grasshopper 3D). To better judge the usability of the proposed workflow, it is advised to carry out further user testing on actual case studies.

Acknowledgement

To be added.
References


