13th International Symposium on Turbulence and Shear Flow Phenomena (TSFP13)
Montreal, Canada, June 25–28, 2024

PREDICTING TURBULENCE STRUCTURE IN STREET-CANYON FLOWS USING MACHINE LEARNING

Tomek Jaroslawski
Center for Turbulence Research
Stanford University
tomek@stanford.edu

Aakash Patil
Center for Turbulence Research
Stanford University
aakash.patil@stanford.edu

Beverley J. McKeon
Center for Turbulence Research
Stanford University
bjmckeon@stanford.edu

ABSTRACT

This contribution explores the use of machine learning to forecast spatio-temporal information in street canyons with varying canyon geometry and upstream roughness. We utilize an experimental particle image velocimetry dataset and adopt an approach that incorporates a convolutional encoder-decoder transformer model alongside autoregressive training. The learning strategy is assessed based on its ability to predict various aspects of the flow, including mean turbulent statistics, two-point correlations, quadrant events, and the dominant flow structures.

INTRODUCTION

Air quality in urban environments is a pressing contemporary issue, with significant socio-economic implications. The geometric complexity of built areas and the interaction of numerous thermodynamic processes challenge our comprehension of the urban climate. Turbulence plays a fundamental role in the instantaneous dynamics of airflow. Specifically, the atmospheric flow, combined with the complex geometry of the urban canopy, exhibits pronounced multi-scale characteristics, both in space and time. The canyon geometry and upstream roughness are significant parameters, wherein non-linear amplitude modulations manifest between the large scales in the boundary layer and the separated low-frequency flapping shear layer at roof level. Hence, the critical aspect lies in the comprehension of the spatio-temporal structure of such flows, particularly in the examination of transient phenomena like accidental pollutant releases or in the prediction of flow states with a limited number of sensors.

Recently, machine learning has been applied to study urban flow dynamics. This includes computing pollution concentrations from mobile field data (Alas *et al.*, 2022), examining inter-scale turbulent interactions over obstacle arrays (Liu *et al.*, 2023), determining drag coefficients on buildings using large eddy simulations (LES) (Lu *et al.*, 2023), and even developing reduced-order models of flow dynamics (Xiao *et al.*, 2019). However, there are few works leverage machine learning (ML) algorithms systematically to predict spatial-temporal turbulent physics, especially when using simplified idealized experimental data for training to predict results in more complex conditions.

Several studies have focused on spatial and temporal reconstruction, as well as spatial supersampling (Schmidt *et al.*, 2021). Hybrid deep neural network architectures have been designed to capture the spatial-temporal features of unsteady flows (Han *et al.*, 2019), and machine learning—based reduced-order models have been proposed for three-dimensional flows (Nakamura *et al.*, 2021). For instance, a deep learning framework that combines long short-term memory networks and convolutional neural networks has been employed to predict the temporal evolution of turbulent flames (Ren *et al.*, 2021).

New deep learning architectures, such as transformers, are emerging for temporal problems in structured and unstructured data. Inspired by convolutional neural networks, transformers build input features using self-attention to assess the relevance of other data points in the dataset, without relying on recurrence. They excel in natural language processing tasks and are replacing traditional recurrent neural networks like long short—term memory networks. Transformers have also been applied in spatio-temporal contexts, such as video analysis. However, they have not been used for spatio-temporal prediction in experimental flow fields involving turbulent flows in urban street canopies.

In this work, we use a convolutional encoder-decoder transformer model along with autoregressive training to make spatio-temporal predictions using an experimental particle image velocimetry (PIV) dataset of a street canyon flow. The model is used to predict mean turbulent and two-point statistics, quadrant events, and the modal decomposition of the flow data, highlighting the potential of data-driven techniques for addressing complex problems related to urban flows.

EXPERIMENTAL DATA

The experimental data, as referenced in Jaroslawski *et al.* (2019, 2020), was acquired from wind tunnel experiments at École Centrale de Nantes, France. These experiments utilized a low-speed boundary-layer wind tunnel with dimensions of 2 meters in width, 2 meters in height, and 24 meters in length, featuring a 5:1 inlet contraction ratio. Figure 1 shows the experimental setup. A simulated suburban atmospheric boundary

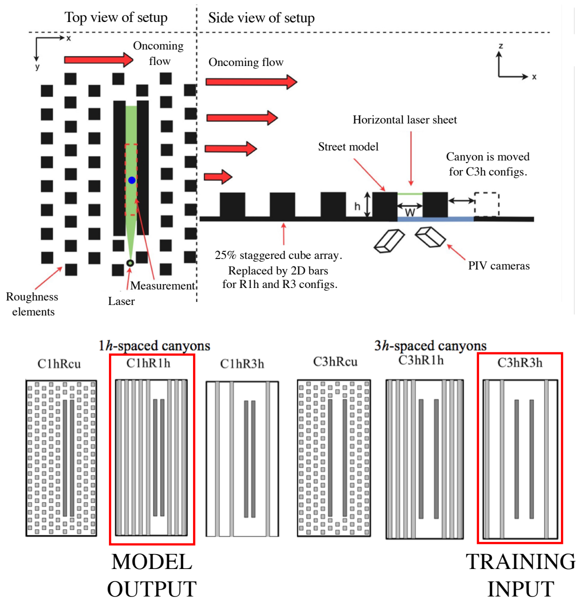

Figure 1. (Experimental setup (top) and configurations (bottom).

layer at a 1:200 scale was created at the model location using a combination of three vertical, tapered spires at the inlet, a solid fence 300 mm in height located 1.5 meters downstream of the inlet, and an 18.5-meter fetch of either 2D or 3D roughness elements. The study investigated street canyons with width-to-height aspect ratios of 1 and 3. Stereoscopic particle image velocimetry (PIV) measurements were conducted horizontally ($x-z$ plane) at a height of $0.9h \pm 0.05h$. The PIV setup, positioned beneath the wind tunnel floor, utilized a Litron double cavity laser and DANTEC Dynamic Studio software to produce vector fields with a spatial resolution of 1.6 mm and a sampling frequency of 7 Hz. Velocity vector fields were computed using an iterative cross-correlation analysis with a window size of 64×64 pixels and an interrogation window of 32×32 pixels, overlapped by 50%, and a pulse interval of 500 μs. Measurement uncertainties for mean velocity, standard deviation, and turbulent shear stress were estimated at 0.9%, 1.4%, and 3.9%, respectively, based on 2551 independent samples from 10,000 velocity field recordings. The freestream velocity of $U_e = 5.9$ ms^{-1}, measured using a pitot-static tube, remained constant across experiments, yielding a Reynolds number of 1.9×10^4 based on this speed and canyon height, h.

MACHINE LEARNING FRAMEWORK

Transformers, when combined with convolutional encoder-decoder models, offer optimal performance for spatio-temporal data analysis. This fusion is particularly effective for tasks like video frame prediction in computer vision. The self-attention mechanism embedded within convolutional layers enhances spatial representation by prioritizing essential features while suppressing less relevant ones. In Figure 2, we illustrate this integrated framework, which we utilize in our current study.

In our spatio-temporal learning task, we are given a time-series comprising N sequential snapshots $\left[x_t, x_{t+\Delta t}, \ldots, x_{t+(N-1)\Delta t}\right]$, with the objective of predicting the same quantity of interest M steps ahead in time. The input X of our deep learning model consists of a sequence T_{in} of snapshots $\left[x_t, x_{t+\Delta t}, \ldots, x_{t+(N-1)\Delta t}\right]$, while the output Y comprises a sequence T_{out} of snapshots $\left[x_{t+N\Delta t}, \ldots, x_{t+N+(M-1)\Delta t}\right]$. Each snapshot x_t can either be a scalar field or a vector field containing multiple features.

The encoder processes input snapshots with $H \times W$ resolutions, extracting pertinent information and mapping it to a high-dimensional representation. Meanwhile, the decoder converts this representation into target output tensors through up-sampling and convolutions. Together, the weight matrices of the encoder and decoder facilitate the mapping of input to output, thereby enabling small-scale feature learning. At $x_{t+\Delta t}$, the decoder transforms the latent space back to the original spatial dimensions. When a transformer block follows a convolutional layer, the model learns to emphasize significant features across channels and spatial dimensions.

Initially, input sequences are concatenated channel-wise to the input layer, followed by convolutional operations in the encoder. Within the convolutional layers, intermediate feature maps $F \in R^{C \times H \times W}$ with C intermediate channels from a specific layer pass through the self-attention convolutional transformer layer. This layer considers the spatial representation and positional embeddings of input sequence channels, employing a 3×3 kernel and incorporating convolutional features. The combination of convolutional neural networks with self-attention enhances the learning of spatio-temporal structures.

In addition to the convolutional transformer layer, the model undergoes training in an autoregressive manner. Autoregressive models, in formal terms, predict future sequences by leveraging previously predicted sequences in a cyclical manner. In this context, "auto" signifies the regression of the variable sequence against itself. For a trained model M as shown in Figure 2, multi-step training is performed for quantity X_t in an auto-regressive manner, $X_{t+\Delta t}$ is predicted from previously predicted X_t, where t is some non-dimensional time. In other words, an initial condition X_t is inputted to the model to learn $\widehat{X}_{t+\Delta t}$, after this predicted $\widehat{X}_{t+\Delta t}$ is then fed back to the model again to learn $\widehat{X}_{t+2\Delta t}$ and so on:

$$\begin{cases} \widehat{X}_{t+\Delta t} = M(X_t), \\ \widehat{X}_{t+2\Delta t} = M(\widehat{X}_{t+\Delta t}), \\ \quad \cdots \\ \widehat{X}_{t+(n-1)\Delta t} = M(\widehat{X}_{t+(n-2)\Delta t}), \end{cases} \tag{1}$$

where t is the time step and $X \in R^{C \times H \times W}$ is the input tensor snapshot at instant t. In the following, the autoregressive training sequence length is set equal to two in order to limit the computational cost. We train the model by employing the Adam optimizer (Kingma & Ba, 2014) to iteratively minimize the total equi-weighted mean squared error (MSE) loss defined by:

$$\begin{aligned} \mathscr{L} = \frac{1}{n_s} \Bigg[& \sum_{i=1}^{n_s} \left((X_{t+\Delta t})^i - \left(\widehat{X}_{t+\Delta t}\right)^i \right)^2 + \cdots \\ & + \sum_{i=1}^{n_s} \left((X_{t+2\Delta t})^i - \left(\widehat{X}_{t+2\Delta t}\right)^i \right)^2 + \cdots \\ & + \sum_{i=1}^{n_s} \left(\left(X_{t+(n-1)\Delta t}\right)^i - \left(\widehat{X}_{t+(n-1)\Delta t}\right)^i \right)^2 \Bigg]. \end{aligned} \tag{2}$$

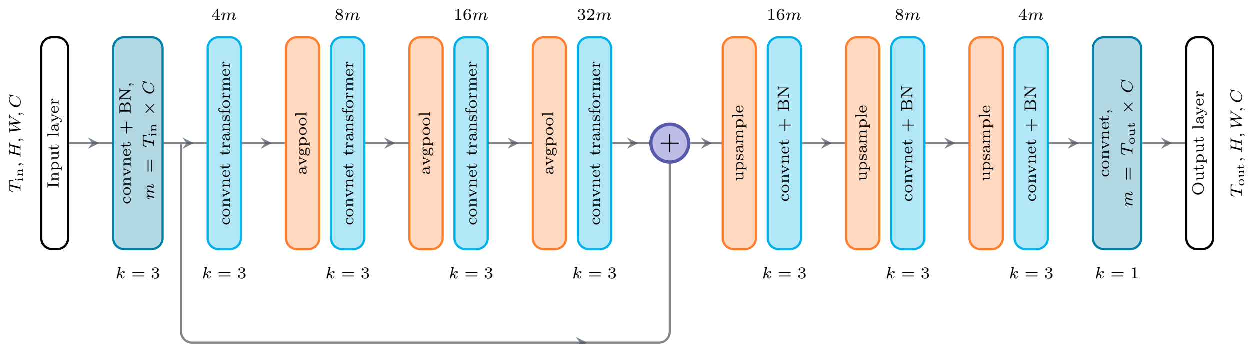

Figure 2. Convolutional encoder-decoder transformer deep learning architecture : Model architecture of the convolutional encoder-decoder transformer to process low and high level features. The canonical four-stage design is utilized in addition to the convolutional transformer blocks or layers. H, W are the input resolutions for each snapshot in T_{in} sequence and T_{out} sequence, k is the kernel size, and m, the number of filters.

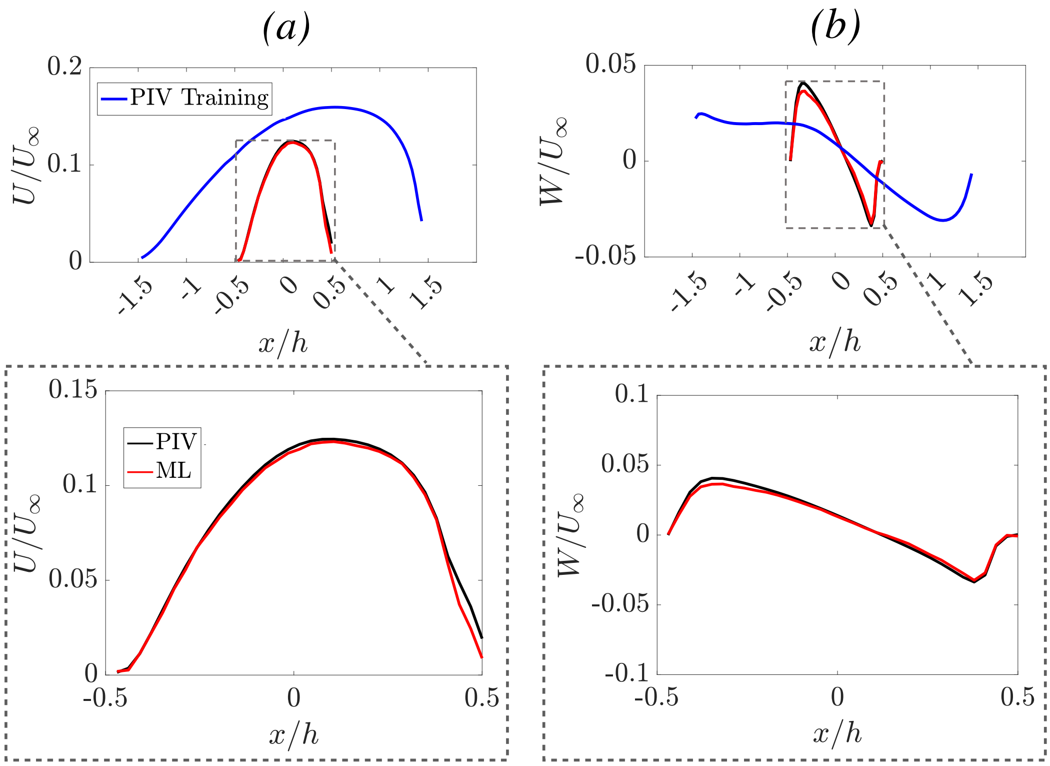

Figure 3. Time- and spatially-averaged mean velocity profiles for (a) streamwise velocity and (b) vertical velocity. The blue line represents the training configuration (Ch3R3h), while the red and black lines denote the test case (C1hC1h).

The neural network employs the ReLU activation function, chosen for its documented efficacy in stabilizing weight updates during training (Nair & Hinton, 2010). Training involves repeated presentation of the entire dataset to the network, with shuffling applied, and each complete iteration termed as an epoch. To halt the training process, an early stopping criterion is implemented, supplemented by a reduction in the learning rate if significant learning progress was not observed after every 100 epochs. Implementation of the deep learning architecture utilizes the TensorFlow library (Abadi *et al.*, 2016), with training executed on an Nvidia RTX A4500 GPU.

In this study, we utilize a training dataset, as shown in Figure 1, consisting of a canyon with a width of 3h and upstream roughness characterized by 2D bars spaced at intervals of 3h. This configuration represents the wake-interference flow regime (Oke, 1988), where closely spaced roughness elements enhance each other's wakes. We train the ML model using the complete set of 10,000 snapshots extracted from the PIV experiments. Subsequently, we evaluate the model's performance in the skimming flow regime, which features a canyon spacing of 1h, as shown in Fig. 1. In this regime, the density of packing is such that the flow "skips" over the top of the elements and is significantly different from the wake-interference flow regime. The ML model generated 500 snapshots. This reduction in the number of snapshots resulted from significant error propagation observed afterward.

RESULTS

This section presents the results, comparing mean and two-point turbulent statistics, quadrant analysis, and proper orthogonal decomposition between the PIV reference and the ML predictions for the C1hR1h configuration.

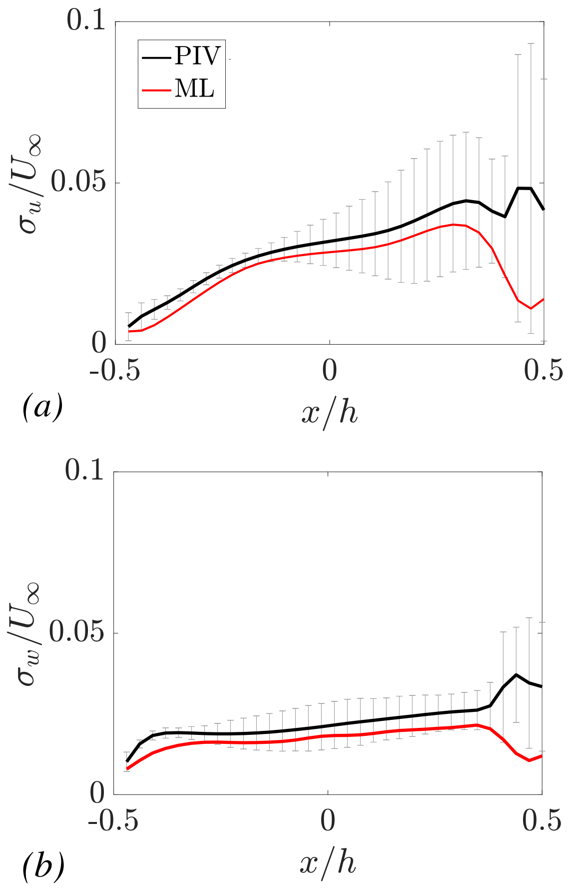

Figure 4. Time- and spatially-averaged standard deviation profiles of (a) streamwise velocity and (b) vertical velocity components for the C1hR1h test case.

Mean turbulence statistics

Figure 3 presents a comparison between the model's predictions and the experimental data for the time- and spanwise-averaged streamwise and vertical velocity profiles as functions of the x-direction. The mean flow profiles in the training data, presented in blue in Fig. 3, differ significantly from those in the test case. This difference arises because the canyon width-to-height ratio is reduced to 1 in the test case compared to 3 in the training data. Additionally, the test case is in a skimming flow regime, which differs from the wake interaction flow regime present in the training data (Oke, 1988). Notably, in the training dataset, the W velocity component shown in Fig. 3b, reveals a larger and asymmetric recirculation skewed towards the windward portion of the canyon, unlike the test case profile. The streamwise and vertical mean velocity profiles shown in Fig. 3 demonstrate a strong agreement between the model's predictions and the experimental data for mean flow.

Figure 4 shows the profiles of the spanwise-averaged standard deviation at the streamwise roof level for both the streamwise (σ_u) and vertical (σ_w) velocity components. The error bars indicate the spanwise variation of σ_u and σ_w calculated from the experimental data. We can observe a general agreement with the experiment in the spatial evolution of the profiles. The ML prediction exhibits lower values of σ_u and σ_w; however, they mostly lie within the spanwise variation of these statistics in the experimental data.

Two-point statistics

To further assess the model's capacity in predicting turbulence structure, we conducted two-point spatial correlation analysis on the streamwise velocity fluctuation, represented as R_{uu}. Two-point spatial fluctuating velocity correlations offer important information regarding the structure of the flow field that single-point measurements are unable to provide. A two-point spatial correlation was conducted using the middle of the street canyon as the reference point. The two-point correlation coefficient was computed using

$$R_{uu} = \frac{\overline{u'(x_{ref}, y_{ref})u'(x,y)}}{\sqrt{\overline{u'(x_{ref}, y_{ref})}}\sqrt{\overline{u'(x,y)}}}. \tag{3}$$

Figure 5a shows contour fields of R_{uu} for both the ML model and the PIV data. The ML model effectively captures the general spatial structure, particularly near the reference point $x_{ref} = 0, y_{ref} = 0$. In Figure 5b, a spanwise slice is presented at the streamwise position of $x = 0$. This further highlights the model's capability to predict fluctuations' decorrelation at smaller spatial lags, indicating its potential in forecasting the spanwise turbulence structure. At greater spatial lags, the correlation decreases more rapidly than in the experimental data, possibly due to energy dissipation by the model. It is worth noting that the noise observed in the ML results may stem from the utilization of 500 snapshots for computations, whereas the experimental dataset incorporated 10,000 snapshots.

Quadrant analysis

We employ quadrant analysis (Wallace, 2016) to assess the model's capability in predicting turbulent events within the flow. This evaluation allows us to gauge the model's performance in capturing the behavior of coherent structures near the roof level of the canyon. The turbulent momentum flux, $u'w'$, is decomposed into four quadrants: Q1 (outward interaction, $u' > 0, w' > 0$), Q2 (ejection, $u' < 0, w' > 0$), Q3 (inward interaction, $u' < 0, w' < 0$), and Q4 (sweep, $u' > 0, w' < 0$). A hole analysis is then applied to measure the extent of these contributions to $u'w'$ in each quadrant by further limiting the data to consider values above a certain amplitude threshold. The threshold level, T_H, is determined from a multiple (H) of the root-mean-square (r.m.s.) stress: $T_H = H(u'w')$.

The results of the hole analysis are shown in Figure 6a, illustrating the contribution of each quadrant to the total shear stress as a function of hole size, H. The ML dataset exhibits agreement with the experimental outcomes, a conclusion further supported by examining the probability density function of the streamwise and vertical fluctuations, $P(u', w')$, shown in Figure 6b. Nonetheless, at lower threshold levels (H), opportunities for enhancement of the model emerge, highlighted by a slightly less precise overlap of the ML and PIV data. This divergence is, at least in part, presumably due to the truncated dataset size within the ML analysis vis-à-vis the exhaustive experimental dataset, indicating that future model refinement should focus on enhancing the model's ability to generate more snapshots.

Proper orthogonal decomposition

We conduct a snapshot Proper Orthogonal Decomposition (POD) analysis (Sirovich, 1987) on the fluctuating velocity fields from both datasets. This assessment evaluates the ML

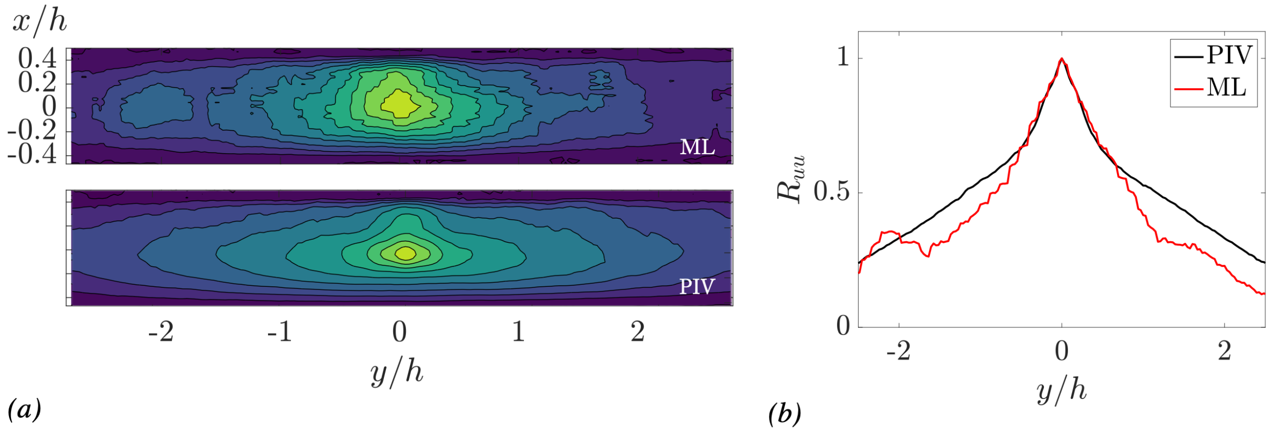

Figure 5. (a) Two-point correlation fields, R_{uu}, of the ML model and PIV dataset. (b) Spanwise slice at $x = 0$ of the R_{uu} field.

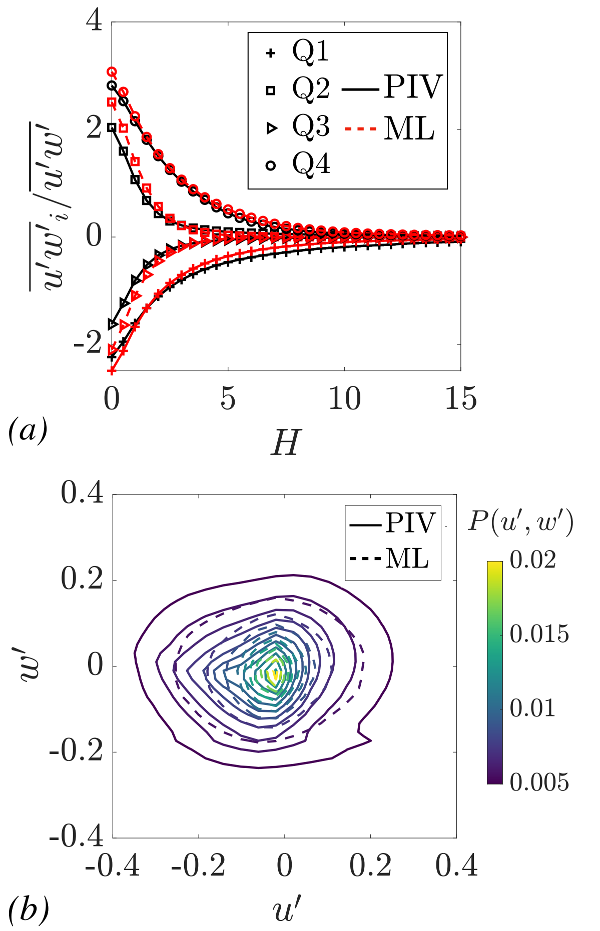

Figure 6. (a) Q1-Q4 contributions to the total shear stress versus the hole size, H (b) Joint-Probability Function, $P(u',w')$

model's proficiency in capturing and forecasting the most energetically significant flow structures, as well as the distribution of kinetic energy across different scales of motion identified within the data.

In Figure 7a,b, we show the spatial structures of POD modes 1, 2, 4, and 20 for the streamwise fluctuating velocity field. In the first two modes, we observe structural agreement between the experimental and ML-derived POD modes. These modes are associated with the large-scale structure related to the low-frequency dynamics of the separated shear layer. Conversely, the higher-order modes exhibit a reduction in spanwise wavelength and display periodicity along the canyon's span. Of particular interest is the performance of the ML model in Mode 20. Despite the small-scale dynamics, there is a qualitative resemblance in the spatial organization of this mode, suggesting that the model can replicate the finer-scale structures present within the shear layer at the canyon's roof level. However, we note a discrepancy in the periodicity of Mode 4, which could be associated with variations in the eigenvalues. Figure 7c displays the eigenvalues for the initial 30 modes. Although the trend of eigenvalues generally matches, the eigenvalues of the ML model are smaller across all modes.

CONCLUSION AND IMPROVEMENTS

In this study, we employed a machine learning framework that integrates a convolutional encoder-decoder transformer with autoregressive training to predict spatio-temporal dynamics within a street canyon. The model was trained using wind tunnel PIV measurements at the roof level of a street canyon in the x-z plane. The training dataset configuration comprised a canyon with a width-to-height ratio of 3 and an upstream roughness fetch consisting of 2D transverse bars designed to induce a wake interference flow regime. Subsequently, we evaluated the model's performance using a configuration with a smaller canyon width-to-height ratio of 1, where the upstream roughness consisted of 2D transverse bars spaced to generate a skimming flow regime.

The ML model predicted the mean turbulent statistics and captured the spanwise structure of roof-level turbulence in the street canyon, as evidenced by comparisons with two-point statistics. The frequency of turbulent events was also analyzed using quadrant analysis. Additionally, results from a POD analysis indicated that the model's predictions were consistent with the experimental observations, particularly in the structure and organization of the most energetic modes.

Future refinements will focus on training the model using fluctuations snapshots rather than velocity snapshots, augmenting the number of convolution layers, and broadening training to include additional configurations. Additionally, we will explore the integration of diffusion model strategies to generate more flow field snapshots.

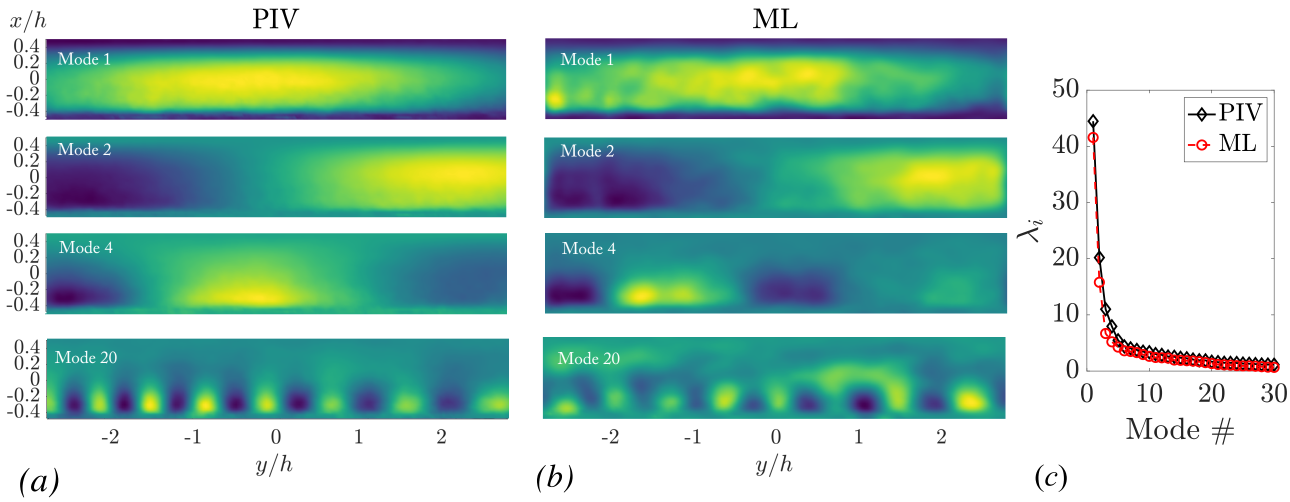

Figure 7. POD modes 1,2,4 and 20 of the PIV (a) and ML (b) data. (c) First 30 eigenvalues.

ACKNOWLEDGEMENTS

Partial support of ONR under grant N00014-22-1-2150 is gratefully acknowledged. The data was obtained by T.M. Jaroslawski during his stay at École Centrale de Nantes, under the supervision of L. Perret and E. Savory.

REFERENCES

Abadi, Martín, Barham, Paul, Chen, Jianmin, Chen, Zhifeng, Davis, Andy, Dean, Jeffrey, Devin, Matthieu, Ghemawat, Sanjay, Irving, Geoffrey, Isard, Michael *et al.* 2016 Tensorflow: a system for large-scale machine learning. In *12th USENIX Sym Oper Syst Des Implement (OSDI 16)*, pp. 265–283.

Alas, Honey Dawn, Stöcker, Almond, Umlauf, Nikolaus, Senaweera, Oshada, Pfeifer, Sascha, Greven, Sonja & Wiedensohler, Alfred 2022 Pedestrian exposure to black carbon and pm2. 5 emissions in urban hot spots: new findings using mobile measurement techniques and flexible bayesian regression models. *Expo Environ Epidemiol* **32**, 604–614.

Han, Renkun, Wang, Yixing, Zhang, Yang & Chen, Gang 2019 A novel spatial-temporal prediction method for unsteady wake flows based on hybrid deep neural network. *Phys Fluids* **31**.

Jaroslawski, Thomas, Perret, Laurent, Blackman, Karin & Savory, Eric 2019 The spanwise variation of roof-level turbulence in a street-canyon flow. *Bound-Layer Meteorol* **170**, 373–394.

Jaroslawski, Thomas, Savory, Eric & Perret, Laurent 2020 Roof-level large-and small-scale coherent structures in a street canyon flow. *Environ Fluid Mech* **20**, 739–763.

Kingma, Diederik P & Ba, Jimmy 2014 Adam: Method for stochastic optimization. *arXiv Preprint* .

Liu, Wanting, Zou, Yajun & Li, Xuebo 2023 Study of interscale interactions for turbulence over the obstacle arrays from a machine learning perspective. *Phys Fluids* **35**.

Lu, Yanle, Zhou, Xu-Hui, Xiao, Heng & Li, Qi 2023 Using machine learning to predict urban canopy flows for land surface modeling. *Geophys Res Lett* **50**, e2022GL102313.

Nair, Vinod & Hinton, Geoffrey E 2010 Rectified linear units improve restricted boltzmann machines. In *Proc 27th ICML-10*, pp. 807–814.

Nakamura, Taichi, Fukami, Kai, Hasegawa, Kazuto, Nabae, Yusuke & Fukagata, Koji 2021 Convolutional neural network and long short-term memory based reduced order surrogate for minimal turbulent channel flow. *Phys Fluids* **33**.

Oke, Tim R 1988 Street design and urban canopy layer climate. *Ener Build* **11**, 103–113.

Ren, Jiahao, Wang, Haiou, Chen, Guo, Luo, Kun & Fan, Jianren 2021 Predictive models for flame evolution using machine learning: *apriori* assessment in turbulent flames without and with mean shear. *Phys Fluids* **33**.

Schmidt, David, Maulik, Romit & Lyras, Konstantinos 2021 Machine learning accelerated turbulence modeling of transient flashing jets. *Phys Fluids* **33**.

Wallace, James M 2016 Quadrant analysis in turbulence research: history and evolution. *Annual Review of Fluid Mechanics* **48**, 131–158.

Xiao, Dunhui, Heaney, CE, Mottet, L, Fang, F, Lin, W, Navon, IM, Guo, Y, Matar, OK, Robins, AG & Pain, CC 2019 A reduced order model for turbulent flows in the urban environment using machine learning. *Build Environ* **148**, 323–337.