water

MDPI

Article

High-Fidelity Simulation of Turbulence in the Piscataqua River Using a Novel Neural Network Surrogate

Samin Shapour Miandouab [1], Mustafa Meriç Aksen [1], Mehrshad Gholami Anjiraki [1], Fotis Sotiropoulos [2], SeokKoo Kang [3] and Ali Khosronejad [1,*]

[1] Department of Civil Engineering, Stony Brook University, Stony Brook, NY 11794, USA; samin.shapourmiandouab@stonybrook.edu (S.S.M.); mehrshad.gholamianjiraki@stonybrook.edu (M.G.A.)
[2] Department of Mechanical Engineering, Penn State University, State College, PA 16802, USA
[3] Department of Civil and Environmental Engineering, Hanyang University, Seoul 04763, Republic of Korea
* Correspondence: ali.khosronejad@stonybrook.edu

Abstract

Accurate three-dimensional characterization of turbulent flows in natural waterways is essential for the effective design of tidal farms and other critical infrastructure situated along or across rivers. High-fidelity predictions based on the large-eddy simulation (LES) method capture the necessary physics but incur computational costs that hinder rapid scenario testing. Statistically, a relatively long history of instantaneous flow fields is required to generate reliable turbulence statistics, e.g., mean velocity and Reynolds stresses, of river flow. Such a requirement often incurs high simulation runtime and data storage costs. This study seeks to develop a neural network surrogate model that learns from a limited number of instantaneous flow realizations and approximates the outputs of the corresponding time-averaged fields with LES-level accuracy. Such a surrogate would eliminate the need to accumulate extensive ensembles, enabling faster hydrodynamic assessment and making LES-informed analyses more accessible for practical engineering decisions.

Keywords: natural river; turbulent flow; high-fidelity modeling; neural network; large eddy simulation

1. Introduction

In the design phase of infrastructure projects involving riverine systems, such as marine energy farms, power plants, and bridge construction, predicting turbulence statistics of the river flow is essential. Computational models such as Large-Eddy Simulation (LES) provide high-fidelity flow physics; however, they incur considerable computational cost, limiting rapid and frequent exploration of operating scenarios, including changes in inflow Boundary Conditions (BC), tidal array configurations, and environmental factors. The engineering quantities of interest for turbulent river flows are generally time-averaged statistics, which, when obtained via LES, often require lengthy integrations and many instantaneous realizations, which leads to substantial computational and storage overhead.

To address the computational cost of high-fidelity modeling of natural rivers, we aim to implement a machine-learning (ML) model to enable three-dimensional (3D), field-scale river-flow modeling at a fraction of the cost of LES. In that light, we have developed a Convolutional Neural Network (CNN) surrogate model to map a finite set of instantaneous LES snapshots to the corresponding time-averaged flow field, i.e., first- and second-order turbulence statistics. The proposed framework is a data-driven CNN-based spatial mapping

Check for updates

Academic Editor: Jueyi Sui

Received: 28 April 2026
Revised: 12 June 2026
Accepted: 14 June 2026
Published: 18 June 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

model that learns the relationship between instantaneous LES fields and the corresponding time-averaged target fields. The first- and second-order statistics constitute the mean flow velocity field and the Reynolds stress tensor, respectively. Given this objective, we selected the Piscataqua River (see Figure 1), which is the main river that flows through Portsmouth, New Hampshire, marking the border between New Hampshire and Maine before discharging into the Gulf of Maine at Portsmouth Harbor.

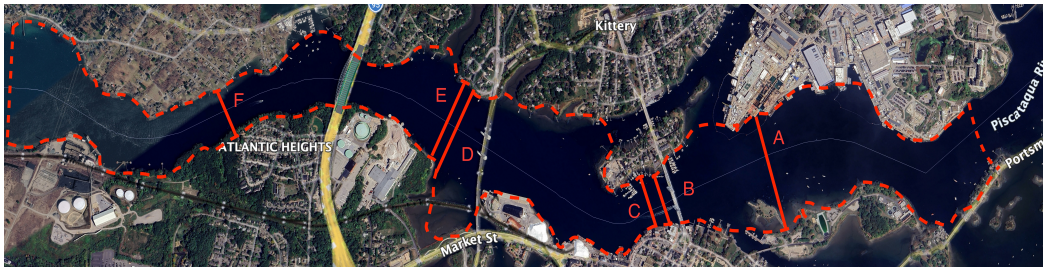


Figure 1. Top view of the study area (marked by red dashed line) within the Piscataqua River at Portsmouth, NH. The cross sections marked in red depict the locations where the ML model's mean flow field predictions are compared with LES results. From right to left, the cross-sections A, B, C, D, E, and F are 800, 995, 1025, 1600, 1630, and 2500 m, respectively, downstream from the inlet cross plane of the study area. Image source: Google Earth.

Conventionally, turbulent river flow modeling relies primarily on Reynolds Averaged Navier Stokes (RANS) models and, more recently, the LES method, both of which are computationally expensive, especially when dealing with the highly complex geometry of natural rivers at high Reynolds numbers [1,2]. The prohibitive computational cost of these methods, especially as the complexity of flow dynamics and riverine geometry increases, has motivated the development of Reduced-Order Modeling (ROM) strategies, which are grounded in data-driven approaches. Since 2010, CNN and autoencoder-based architectures have been employed to learn compact latent representations of high-dimensional flow fields and to reconstruct turbulent velocity fields [3,4]. Studies show that machine-learned inflow generators based on autoencoder-type CNNs can reproduce the spatio-temporal evolution of turbulent channel flows and maintain fully developed turbulence at a fraction of the cost of direct simulations [5,6]. Recent convolutional autoencoders combined with convolutional LSTM networks provide efficient models for 3D turbulence and pass stringent physics-based statistical tests [4,7]. Moreover, convolutional autoencoders and LSTM-based ROMs have been shown to outperform Proper Orthogonal Decomposition (POD) methods and have become a common choice for non-intrusive unsteady flow modeling [8]. These models confirm the ability to reconstruct turbulent velocity fields and develop coherent structures through learned latent dynamics.

More recently, transformer-based architectures have extended this paradigm by directly learning mappings between function spaces or long-range spatio-temporal dependencies, thus improving stability and long-horizon predictive capacity in turbulent regimes [1,9]. Despite these advances, the majority of studies have been validated primarily on canonical configurations, such as low-Reynolds-number turbulent channel flows with geometrically simple domains and periodic or otherwise idealized BCs [4,5]. Similarly, foundational studies in ML-assisted turbulence modelings focused on controlled settings that isolate specific modeling challenges [6,10,11]. While such configurations are essential for methodological development, they generally underrepresent the geometric irregularity, boundary heterogeneity, and multiscale hydrodynamic interactions characteristic of natural riverine systems. Therefore, the demonstrated success of ML-based surrogates in canonical flows does not automatically translate to predictive reliability in environmentally complex domains. These early studies should instead be viewed as methodological stepping stones

that establish algorithmic feasibility while leaving open the question of robustness under realistic environmental complexities.

Past investigations of tidal flow dynamics within the Piscataqua River [12,13] have documented several salient features, including a pronounced tidal asymmetry that modulates river flow during ebb and flood events, nonlinear evolution of tidal waveforms, and substantial variability in current magnitude under changing sea-level conditions. These dynamics reflect strong nonlinear hydrodynamic interactions and intense sensitivity to boundary forcing.

Recently, complementary feasibility and site-selection studies [14] have adopted a hybrid strategy that combines regional-scale hydrodynamic modeling with targeted LES to explain turbulence-dependent processes and evaluate sensitivity to turbulence content and other inflow conditions. While LES provides a physically consistent solution of energy-containing turbulent flow structures, its computational cost grows rapidly with Reynolds number and grid resolution [15], limiting its practical use for systematic parametric or ensemble exploration of BCs [16,17]. This limitation is especially consequential in tidal river systems, where engineering assessments and sea-level rise projections require repeated simulations in a wide range of forcing scenarios to quantify uncertainty and system sensitivity [18]. The surrogate ML framework presented here is motivated by this methodological bottleneck: approximating brute-force LES-resolved mean-flow statistics while reducing computational expense enough to enable tractable BC sensitivity analyses and engineering-scale ensemble studies of real-life waterways.

Attempts to extend ML-based surrogates to model realistic turbulent flows have mostly focused on atmospheric boundary layers and urban canopy configurations [19,20], while riverine environments remain comparatively underexplored. Unlike canonical boundary-layer flows, natural rivers exhibit pronounced spatial heterogeneity due to multiscale geometric constrictions, bathymetric variability, and flow-structure interactions. Flow accelerates near contractions, diverges around bridge piers, and generates complex wake flow structures and substantial gradients in second-order turbulence statistics [21–23]. These quantities may vary by orders of magnitude between upstream, shear-layer, and wake regions. Such variability poses a fundamental challenge for surrogate ML modeling: without appropriate normalization, sampling strategies, or loss weighting, models may become biased toward energetically dominant regions or suffer from unstable gradient dynamics during the training process.

Another limitation concerns generalization across boundary conditions. Many ML surrogates are trained under a single inflow specification and evaluated under identical or weakly perturbed conditions. However, practical river engineering applications require robustness to significant variations in discharge, velocity profiles, and turbulence intensity. Cross-boundary evaluations remain scarce, and studies that explicitly test out-of-distribution conditions report significant performance degradation when training and testing distributions differ [24,25]. This limitation raises concerns regarding the deployment of surrogate ML models in real-world riverine hydrodynamics scenarios.

In parallel with advances in data-driven ML-based turbulence modeling, considerable effort has been devoted to embedding physical constraints into machine learning architectures. Physics-informed neural networks (PINNs) [26,27] and related loss-augmentation strategies enforce conservation laws directly within training objectives and have shown promise in hydrodynamic applications [28,29]. In the context of CNN-based river-flow prediction, Zhang et al. [30] incorporated a divergence-free constraint into the loss function and reported only modest improvements in the predicted flow field compared with a purely data-driven mapping network.

Nonetheless, most demonstrations remain confined to lower-dimensional or idealized configurations, including simplified geometries or depth-averaged flow representations. Applications to fully three-dimensional, geometrically complex river domains remain scarce. More broadly, state-of-the-art ML surrogates have demonstrated strong performance in reproducing canonical turbulence datasets, including channel and isotropic flows [31]. Yet significant challenges persist when extending these approaches to strongly heterogeneous natural systems. In particular, existing models struggle with (i) spatially variable turbulence intensities spanning multiple orders of magnitude, (ii) generalization across substantially different boundary conditions, and (iii) stable training when targeting second-order turbulence statistics. Recent efforts in hydrodynamic and energy-systems modeling have leveraged reduced-order ML surrogates to improve computational efficiency and cross-boundary robustness [28,32]. While effective for predicting bulk quantities such as water surface elevation, discharge, or power output, these models typically do not reconstruct fully resolved three-dimensional turbulent flow fields. Consequently, they do not explicitly capture wake–bathymetry interactions, fine-scale turbulence structure, or large spatial variability in Reynolds stresses characteristic of realistic riverine environments.

This study addresses these gaps by curating LES data from a real-life riverine system on planform geometry and bathymetry, and systematically evaluating preprocessing and training strategies to stabilize surrogate learning under strongly heterogeneous, three-dimensional hydrodynamics.

This paper is organized as follows. In Section 2, we describe the methodology, followed by the results and discussion in Section 3. Finally, in Section 4, we conclude the study's findings.

2. Methodology

The high-fidelity data used to train the model were obtained from our in-house hydrodynamics model, the Virtual Flow Simulator (VFS-Geophysics) model with a wall-model treatment to resolve near-bed dynamics. The solver produced three-dimensional snapshots of the velocity components u, v, and w in the x, y, and z directions, respectively, from which, by time averaging, the first- and second-order turbulence statistics are computed. The first-order statistics, i.e., mean flow velocity field, include $\overline{u}$, $\overline{v}$, and $\overline{w}$, while the second-order statistics, i.e., Reynolds stresses, consist of $\overline{uu}$, $\overline{vv}$, $\overline{ww}$, $\overline{uv}$, $\overline{vw}$, and $\overline{uw}$. The dataset included both the instantaneous unsteady turbulence and the time-averaged field (i.e., mean flow and turbulence intensities).

This richness in spatial and temporal characteristics, however, comes with its own challenges. Namely, the second-order statistics vary by nearly two orders of magnitude across the domain. For instance, near the inlet region, turbulence intensity could be moderate, whereas around wall-mounted infrastructures or near mixing layers within the river, Reynolds stresses could spike sharply. This heterogeneity proved to be a major obstacle to ML model training, as unscaled datasets led to unstable optimization and biased the loss function toward high-magnitude regions.

One solution we employed to address this issue was normalizing the data to mitigate the impact of this diversity. Then, to test the model's generalization power, we split the train and test data based on two grounds. Once into instantaneous and time-averaged within one BC to assess model stability. Then, by BC, to test the cross-boundary generalization for different flow regimes within the same river system. The rationale behind this decision is that instantaneous fields capture unsteady turbulence, while time-averaged fields capture more steady mean flow and turbulence intensities. On that account, we followed two main scenarios for our surrogate models: (i) mean flow velocity predictor, which is trained on instantaneous velocity as input and time-averaged velocity as target for the RF boundary

condition and tested on instantaneous velocity as input and time-averaged velocity magnitude as target for the straight channel and log-law profile inflow boundary conditions; and (ii) TKE predictor, which is trained on instantaneous velocity as input and TKE as target for the RF boundary condition and tested on instantaneous velocity as input and TKE as target for the straight channel and log-law profile inflow boundary conditions. Further details on the boundary conditions and their implementation are provided in the subsequent sections. It is worth mentioning that we deliberately chose a similar BC assignment for training and testing scenarios to keep the experiment as controlled as possible.

We then discretized the computational domain of our study area to $4500 \times 500 \times 100$ computational nodes in the x, y, and z directions, respectively. The time step of the computations was set equal to 0.05 s to ensure that the Courant–Friedrichs–Lewy number stays less than 1.0 at all times. It should be noted that, given the computational resources at our disposal, this resolution was chosen to capture the turbulent structures of interest while remaining computationally feasible.

2.1. Hydrodynamic Model

The hydrodynamics model solves the spatially filtered Navier–Stokes equations for incompressible flow in non-orthogonal generalized curvilinear coordinates. In compact Newton notation, with repeated indices indicating summation, the equations are expressed as follows [33]:

$$J\frac{\partial U^j}{\partial \xi^j} = 0 \tag{1}$$

$$\frac{\partial U^i}{\partial t} = \frac{\xi_l^i}{J}\left(\frac{\partial}{\partial \xi^j}\left(U^j u_i\right) + \frac{1}{\rho}\frac{\partial}{\partial \xi^j}\left(\mu\frac{G^{jk}}{J}\frac{\partial u_i}{\partial \xi^k}\right) - \frac{1}{\rho}\frac{\partial}{\partial \xi^j}\left(\frac{\xi_i^j p}{J}\right) - \frac{1}{\rho}\frac{\partial \tau_{ij}}{\partial \xi^j}\right) \tag{2}$$

where the Jacobian of the geometric transformation,

$$J = \left|\partial\left(\xi^1, \xi^2, \xi^3\right)/\partial(x_1, x_2, x_3)\right|$$

is used for transforming the Cartesian coordinate system to curvilinear. $U^i = (\xi_m^i/J)\, u_m$, shows the contravariant volume flux, where $\xi_l^i = \partial \xi^i/\partial x_l$. The i-th filtered velocity component in the Cartesian coordinate is shown as u_i, μ represents the dynamic viscosity of the fluid (i.e., water), and also $G^{jk} = \xi_l^j\, \xi_l^k$, shows the contravariant metric tensor, the background density (i.e., water density) is shown as ρ (=1000 kg/m^3). The pressure term is defined as p.

The subgrid-scale stresses are modeled with dynamic Smagorinsky in the LES turbulence model and defined as [34,35]:

$$\tau_{ij} = -2\mu_t \overline{S}_{ij} + \frac{1}{3}\tau_{kk}\delta_{ij} \tag{3}$$

$$\mu_t = C_s \Delta^2 \left|\overline{S}\right| \tag{4}$$

where μ_t is the viscosity of eddies, the tensor of filtered strain-rate is represented as $\overline{S}_{ij}$, and δ_{ij} is the Kronecker delta. The Smagorinsky constant is shown as C_s, and $\left|\overline{S}\right| = \sqrt{2(\overline{S_{ij}S_{ij}})}$. The filter size ($\Delta$) is defined as the cube root of the cell volume. More details about the hydrodynamic solver can be found elsewhere [36,37]. Finally, it should be noted that the LES results for river flow within the study area were validated against field data measured with an Acoustic Doppler Velocity Profiler (ADCP), as reported by others [14].

Importantly, the arbitrarily complex geometry of the river bathymetry is resolved using an immersed boundary method [38,39]. The boundary conditions of the hydrodynamics computations of the study area consist of a no-slip condition for solid surfaces (e.g., river

banks, bed, and surfaces of bridge piers). Since the computations lacked sufficient resolution to capture the viscous sublayer near the solid surfaces, we employed a wall model to account for the hydrodynamic effects of the wall on the flow. The free surface of the river is described using the rigid-lid assumption. A Neumann boundary condition is imposed at the outlet cross-plane, while the inlet cross-plane of the computational domain is prescribed using various inlet boundary conditions [40], as follows:

- Straight channel: This boundary condition is calculated by simulating the flow in a 20-m-long straight channel extension from the inlet cross-plane of the study area. The cross-plane of the background mesh along the length of the straight channel is identical to the study area's inlet cross-plane. In a separate simulation, we used periodic boundary conditions in the streamwise direction to obtain fully developed turbulent flow in the channel. The instantaneous flow field from a cross-plane in the middle of this channel was saved and later fed into the study area as the 'straight channel' inflow boundary conditions.
- River inflow: This boundary condition is obtained from a separate LES that modeled the flow field of the 600-m-long reach of the river located immediately upstream of the study area. The instantaneous flow field at the outlet cross plane of this precursor simulation was saved and then imposed at the inlet of the study area as the "river inflow" boundary conditions. Since this boundary condition is derived from site-specific river bathymetry upstream of the study site, the so-obtained inflow constitutes a more realistic inlet condition marked by strong vertical shears, lateral variations, and turbulence intensities that are relatively more consistent with those observed in the field.
- Log-law profile: This boundary condition imposes a velocity profile, $u(z)$, at the inlet cross plane that follows the classic wall law form:

$$u(z) = \frac{u_*}{\kappa} \ln\left(\frac{z}{z_0}\right) \tag{5}$$

where u_* is the friction velocity, $\kappa \approx 0.41$ is the von Karman constant [41], and z_0 is the roughness length.

2.2. Network Architecture

The NN surrogate model maps the instantaneous LES-computed flow field database of the Piscataqua River to its corresponding time-averaged quantities. Instead of running hundreds of thousands of LES time steps—depending on the turbulent flow and eddies' frequency range—to obtain the statistically-converged mean flow quantities, the surrogate model learns the relationship between a small number of instantaneous snapshots and their ensemble-mean representation. Two complementary model configurations were developed to explore the effect of input dimensionality on predictive capability:

Single-channel network: This architecture receives a single instantaneous scalar field, typically the velocity magnitude or the streamwise velocity component u, and predicts its time-averaged counterpart $\overline{u}$. The objective of this configuration is to evaluate whether the instantaneous spatial structures in a single velocity component alone contain sufficient information for the network to infer the statistically converged mean flow field. It provides a minimal, computationally inexpensive baseline.

Tri-channel network: In the second approach, the instantaneous velocity vector (u, v, w) is supplied as a three-channel input tensor. This configuration allows the model to capture cross-component correlations and the combined influence of streamwise, spanwise, and vertical motions on the time-averaged flow field. Since the magnitudes of u, v, and w can

vary greatly across the river domain, especially near bridge-induced wakes, a cube-root transformation is applied to reduce this disparity prior to normalization.

Both models are implemented as compact 3D CNNs following an encoder-decoder topology (see Figure 2). Convolutions were performed in three spatial dimensions to preserve volumetric context within each full-domain sample. The 1-channel and 3-channel variants share the same architectural backbone, differing only in the number of input channels and in the inclusion of the cube-root preprocessing step. The end goal of both models is to establish a scalable framework that can learn the statistical imprint of turbulence from a limited set of instantaneous flow fields and serve as an efficient data-driven surrogate for brute-force LES.

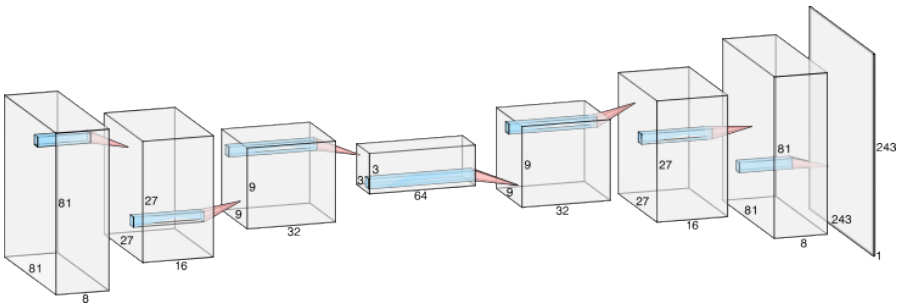


Figure 2. Schematic of the CNN architecture used as the surrogate.

2.2.1. Encoder

The encoder progressively compresses the spatial domain while enhancing the representation of turbulence features. Each encoding stage consists of a 3D convolution followed by a nonlinear activation (Rectified Linear Unit, ReLU). The network begins with a small number of filters that increase with depth to capture features at multiple spatial scales:

$$C_{\text{in}} \rightarrow 8 \rightarrow 16 \rightarrow 32 \rightarrow 64 \tag{6}$$

where $C_{\text{in}} = 1$ for the single-channel model and $C_{\text{in}} = 3$ for the tri-channel model. Each convolution uses a kernel size of $5 \times 5 \times 5$, a stride of $3 \times 3 \times 3$, and padding of $2 \times 2 \times 2$. This kernel size allows the network to aggregate information from neighboring voxels while progressively reducing the spatial resolution. The stride of 3 was empirically selected as a midpoint between preserving sufficient spatial detail (e.g., wake gradients) and maintaining GPU efficiency for 3D data. Eventually, to maintain numerical stability, a 2-pixel padding is applied to ensure that receptive fields overlap smoothly between layers.

We used the ReLU activation function to introduce nonlinearity while preserving nonzero gradients for positive activations. This choice improves training efficiency and reduces the likelihood of vanishing-gradient issues in deeper 3D convolutional architectures.

From the perspective of fluid dynamics, the encoder functions as a hierarchy of spatial filtering operations that become progressively coarser. As network depth increases, the receptive field expands, enabling the learned representations to capture flow structures at multiple spatial scales. These scales range from localized velocity gradients and wake regions to broader flow patterns determined by channel geometry and river-scale circulation.

2.2.2. Decoder

The decoder mirrors the encoder, using 3D transpose convolutions (i.e., deconvolutions) to upsample the compressed latent representation back to the original spatial resolution. The feature channels are arranged in reverse order:

$$64 \rightarrow 32 \rightarrow 16 \rightarrow 8 \rightarrow 1 \tag{7}$$

Each transpose-convolution layer uses a stride of $3 \times 3 \times 3$ to upsample the compressed feature maps. The first three decoder layers use a kernel size of $5 \times 5 \times 5$, while the final output layer uses a kernel size of $5 \times 5 \times 4$ to recover the required output dimensions. Unlike the encoder, which uses uniform $2 \times 2 \times 2$ padding, the decoder uses layer-specific padding values to match the target spatial resolution after upsampling. Specifically, the decoder padding values are $1 \times 1 \times 1$, $2 \times 2 \times 2$, $3 \times 3 \times 3$, and $4 \times 4 \times 3$ for the four transpose-convolution layers, respectively. ReLU activations are applied after each convolutional and transpose-convolutional layer, including the output layer. Figure 3 summarizes the above-mentioned configurations on the network. It should be noted that the present framework predicts statistically converged flow quantities non-iteratively; therefore, prediction errors do not recursively propagate across temporal states as in autoregressive surrogate models.

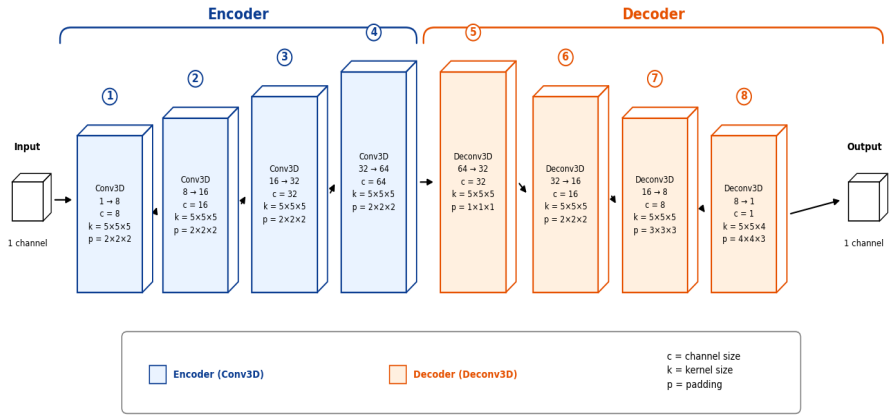


Figure 3. Detailed topology of the CNN encoder–decoder surrogate model.

Architectural hyperparameters and the symmetric encoder–decoder arrangement were determined through preliminary numerical experiments designed to balance reconstruction accuracy, preservation of turbulent flow structures, numerical stability, GPU memory usage, and computational efficiency. Smaller kernels and strides enhanced the retention of fine spatial details but significantly increased memory consumption and training costs. In contrast, larger strides or overly deep architectures led to excessive smoothing and reduced reconstruction quality. Although it lacks direct skip connections (as in a full U-Net), the combination of wide kernels and overlapping receptive fields helps retain both coarse flow organization and moderate-scale spatial variations within the learned feature space.

2.2.3. Training Procedure

The training dataset comprised 21 instantaneous LES realizations from three inflow boundary-condition scenarios: 10 snapshots from the Straight-channel case (timesteps 105 to 195, sampled every 10 timesteps) and 11 from the Log-law case (timesteps 180 to 200, sampled every 2 timesteps). For model testing, 10 snapshots from the River-flow case (timesteps 105 to 195, sampled every 10 timesteps) were available as an alternative. In each scenario, a single LES time-averaged velocity field was used as the supervisory target, establishing a many-to-one mapping between instantaneous flow realizations and the corresponding statistically averaged flow field.

During this process, we noticed that training the NN on raw LES data is prone to instability. This could be due to the underlying turbulence quantities that span several orders of magnitude. In the Piscataqua River LES dataset, near-inlet velocities were fairly uniform ($\approx$0.5–1 ms^{-1}), whereas regions close to the mixing layers mark strong shear and velocity spikes exceeding 3–4 ms^{-1}. Second-order quantities, such as Reynolds stresses and TKE, exhibited even greater variability. To mitigate these disparities and promote balanced

gradient propagation during training, we applied several preprocessing and normalization strategies before feeding data to the network.

The most effective stabilization step was a cube root transform applied to the relevant input or target field. For a signed velocity quantity q,

$$\widetilde{q} = \mathrm{sign}(q)\,|q|^{1/3} \tag{8}$$

This nonlinear mapping compresses the dynamic range without destroying the relative ordering of values. Because the transformation is odd-symmetric, it preserves the sign of velocity fluctuations and therefore the directionality of motion. In the untransformed data, extreme gradients near mixing layers dominated the loss function, leading the network to overlook quieter regions, such as mid-channel or near-bed areas. After the cube-root transform, the distribution of magnitudes across the domain became more homogeneous, allowing the optimizer to treat each region of the river with comparable weight. For non-negative targets such as TKE, the transform simplifies to $k^{1/3}$. We empirically verified that regressing the transform after prediction (i.e., cubing the outputs) restores the variables' physical scale with negligible numerical bias.

Following the cube-root transform, the transformed input or target field was standardized using the training-set statistics:

$$q_{\mathrm{norm}} = \frac{\widetilde{q} - \mu_q}{\sigma_q} \tag{9}$$

where μ_q and σ_q are the mean and standard deviation computed over all training samples. The same transformation parameters were applied to validation and test data to prevent information leakage. This normalization ensures that the transformed fields contribute more consistently to the total loss and avoids scale-imbalance effects during the training procedure. It also speeds convergence of gradient-based optimization by centering the data around zero.

The study area, i.e., the LES domain of $(4500 \times 500 \times 100)$ computational cells, is computationally demanding for direct full-volume processing in GPU memory. During development, tiled processing was evaluated as a possible memory-management strategy. For the present grid, a natural candidate tile size is $100 \times 100 \times 100$ cells because it spans the full vertical direction while partitioning the streamwise and spanwise directions. However, tiled processing increases the number of subdomains and requires an additional reconstruction step to average overlapping predictions. Therefore, the final workflow did not treat tile size or overlap ratio as model parameters, and tiled processing is left as a future extension for larger domains or higher-resolution simulations.

These preprocessing operations, especially the cube-root scaling, proved crucial for achieving stable convergence. Training runs performed without this preprocessing exhibited instability and abnormal losses in the first few epochs, whereas the transformed or preprocessed datasets exhibited smooth, monotonic loss decay. In qualitative terms, cube-root preprocessing enabled the network to see both energetic regions of the flow and quiescent zones on an equal footing, yielding spatially coherent predictions across the river's study area.

The model was trained using the AdamW optimizer [42] with an initial learning rate of 10^{-4} and a decoupled weight-decay coefficient of 10^{-2}. A StepLR scheduler was used with a step size of 400 epochs, a multiplicative decay factor of $\gamma = 0.7$, and a batch size of 2 due to memory constraints.

The loss function used for the reported results was defined as the mean squared error (MSE) between the predicted and reference time-averaged fields obtained from

LES. Although the implementation was developed with the capability to incorporate additional physics-based penalty terms, such as divergence-free constraints following physics-informed formulations [30], these terms were not activated in the final training configuration. This choice was motivated by the previous work of Zhang et al. [30], where the addition of a divergence-free constraint produced only limited gains relative to the unconstrained data-driven CNN surrogate. The final model was therefore trained using only the data-driven MSE objective, which provided satisfactory predictive accuracy while avoiding the additional computational cost of repeatedly evaluating physics residuals during optimization. Accordingly, all results reported in this study correspond to the MSE-based CNN surrogate model. As a future extension, the currently inactive physics-based penalty terms, including divergence-free and momentum-residual constraints, can be incorporated into the training objective to evaluate their impact on physical consistency, predictive accuracy, and cross-boundary-condition generalization.

The numerical experiments conducted during the training process revealed several consistent behaviors and practical lessons regarding the network architecture and preprocessing strategies. These findings can be summarized as follows:

Single-variable network: We consider the ML model trained with only one input vector, e.g., the instantaneous streamwise velocity u, as a baseline, owing to its simplicity. Despite its limited input, this configuration demonstrated that the spatial coherence of an instantaneous u field contains information about long-term mean structures, such as the high-velocity core, shear layers near the banks, and wake regions behind bridge piers. Because the input field was relatively smooth compared with multi-component data, convergence was stable even without special transformations. However, the network's predictions occasionally underrepresented recirculation zones and failed to reproduce fine-scale secondary flows in highly curved or sheared river sections. This shortfall is consistent with the physical limitation of single-component information. In other words, the instantaneous streamwise velocity component, u, alone cannot describe the divergence-free conditions and/or momentum conservation and thus impacts the predicted target vector, i.e., the mean flow field.

Tri-variable network: Considering the 3D instantaneous velocity field, (u, v, w), provided a richer physical basis. In principle, the three components carry the anisotropic signatures of turbulence, allowing the surrogate to infer coupling between streamwise and cross-stream motions. In practice, however, raw multi-component training proved unstable, gradients exploded within the first few epochs, and the optimizer quickly diverged. This instability stemmed from the fact that u, v, and w span different magnitude ranges and dynamic behaviors, while u dominates the bulk flow and v and w fluctuate around lower means but exhibit intense local bursts near shear layers. Employing the cube-root transformation equalized these magnitudes and effectively "linearized" the tails of the distribution, enabling the network to learn balanced filters across all components. After transformation, the tri-channel network produced more accurate reconstructions of turbulence-intense regions, particularly the shear layers along the banks and the wake recovery zones behind the Memorial and Sarah-Mildred Bridges (see Figure 1). In general, the predictions preserved the large-scale topology better than those of the one-channel model.

A comparison of the two approaches highlights some key insights, as follows:

- Information content vs. stability: Adding input channels increased representational capacity but required more robust normalization and regularization. The baseline single-variable network, trained on fewer variables, was easier to stabilize but less expressive.
- Importance of preprocessing: Pre-training data transformation of the multi-channel network exerted a greater influence on convergence than architectural depth or num-

ber of filters. Without cube-root scaling, even more complicated network structures failed to converge.

- Physical consistency of learned features: Visualization of latent features revealed that deeper layers in both ML models responded to coherent flow structures such as jet cores and vortex–shear interfaces, suggesting that the networks learned physically meaningful encodings rather than purely statistical correlations.
- Cross-boundary robustness: Both models generalized reasonably well when the test scenarios (of river flow) resembled the training one, but deteriorated under a strong distribution change. This limitation underscores the need for domain adaptation or transfer learning techniques to develop surrogates across hydrodynamic regimes.

Overall, our numerical experiments confirmed that cube-root preprocessing plays a key role in stabilizing training when the target fields exhibit strong spatial heterogeneity, particularly for second-order turbulence quantities such as TKE. The final surrogate framework captures the essential topology of the mean flow field while remaining computationally lightweight. The findings also reinforce a broader lesson that the preprocessing choices grounded in physical reasoning can be as important as architectural factors in determining ML model performance.

Furthermore, several challenges were identified during training and evaluation phases that warrant further investigation:

- Heterogeneity: Orders-of-magnitude differences in second-order turbulence statistics quantities made raw training challenging.
- Dead-end trial: Training directly on Reynolds stresses (i.e., second-order turbulence statistics), without transformation, caused exploding gradients.
- Key breakthrough: Cube-root preprocessing of TKE reduced skewness and stabilized the learning process.
- Cross boundary condition generalization: Performance degraded significantly when inflow profiles, used as inlet boundary conditions, differed strongly.

We measured model performance using the mean squared error (MSE) between the predicted and reference time-averaged target fields, including mean velocity and TKE. MSE was chosen for its direct penalization of deviations in physical magnitude and its smooth, well-behaved gradients. For an instantaneous input y_n and LES-computed mean flow target of $\overline{y}_n$, the loss is obtained as:

$$L_{\mathrm{MSE}} = \frac{1}{N}\sum_{n=1}^{N}(\hat{y}_n - \overline{y}_n)^2 \tag{10}$$

where $\hat{y}_n$ is the ML model's predicted mean flow quantity, and N is the number of voxels per batch. The loss is computed in the normalized space (after cube-root and standardization transforms), and predictions were converted to the physical scale for visualization and error metric calculations.

Herein, the ML models were trained using the AdamW optimizer with a step-based learning-rate scheduler. AdamW builds on the Adam optimization framework [43] by decoupling weight-decay regularization from the gradient-based update [42], while the scheduler progressively adjusted the learning rate to support stable convergence.

Each training run consisted of 80–120 epochs, depending on the complexity of the river flow scenario and the type of the input vector. The training process was monitored by validating the loss and by consistent visual inspection of the predicted mean flow field (e.g., every 5 epochs). Early stopping terminated training if the validation loss did not improve for 20 epochs. Moreover, no explicit dropout or batch-normalization layers were used. training was performed using mini-batches with a batch size of 2, with the available full-

domain training samples randomly shuffled before mini-batch construction. In addition, a cube-root transformation was applied to the target variables, followed by standardization. These preprocessing and training procedures reduced scale imbalance within the target fields, helped mitigate potential small-batch training instability, and improved convergence stability without requiring spatial tiling in the final workflow.

3. Results and Discussion

In this section, we examine the predictive performance of the trained surrogate CNN model qualitatively and quantitatively for the mean-flow velocity fields (first-order turbulence statistics) and the turbulence kinetic energy field (second-order turbulence statistics). It should be noted that the CNN model was first trained using the LES-computed mean flow field data associated with the "river inflow" boundary conditions. For brevity, this section focuses on validation (test) cases, i.e., the river flow field with the "log-law profile" and "straight channel" boundary conditions, which the CNN model did not encounter during training.

In Figure 4, we plot the LES-computed and CNN predictions for the contours of mean velocity magnitude, $\overline{U}$, within the study area at the water surface. As seen, Figure 4a,b correspond to the LES-computed velocity magnitude obtained using the log-law and straight-channel boundary conditions, respectively, while panels (c) and (d) show the corresponding CNN predictions. Overall, the surrogate CNN model successfully reproduces the low and high-velocity fields throughout the study area. In particular, the model accurately identifies zones of high-velocity flow where the river narrows or near the inner bends, as indicated by the red and orange regions in the color maps. The predicted mean velocity fields closely resemble the LES-computed mean flow patterns under both the "straight channel" and "log-law profile" boundary conditions, demonstrating that the trained model generalizes well to inflow configurations that differ from those used during training (i.e., "river flow" boundary condition).

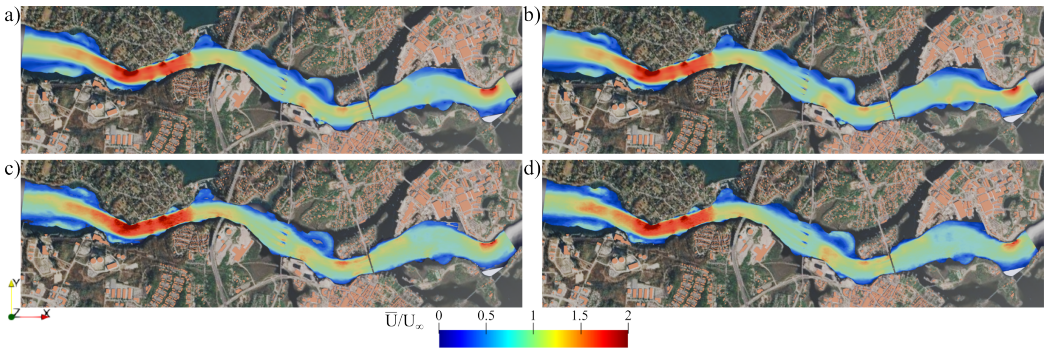


Figure 4. Color maps of the normalized time-averaged velocity magnitude $\overline{U}/U_\infty$, at the free surface for different turbulent inflow boundary conditions. $U_\infty = 2.0$ ms^{-1} is the bulk velocity of the river. Panels (**a**,**b**) show LES results for the log-profile and straight-channel inflows, respectively, and panels (**c**,**d**) present the corresponding CNN predictions. Flow is from right to left.

In addition, Figure 5 provides a more detailed quantitative comparison of velocity magnitude at the six representative cross-sections (A–F) within the study area (see Figure 1 for the locations of cross-sections). The solid and dashed curves denote the LES results and CNN predictions, respectively. As shown, the ML-predicted velocity magnitude distributions closely match those of the LES, capturing both the magnitudes and peaks of the mean velocity field. The agreement is particularly strong in sections characterized by smooth shear layers and gradual velocity transitions. In regions with stronger lateral variability, minor discrepancies appear near the channel boundaries, where sharp gradients and complex secondary flow structures are present.

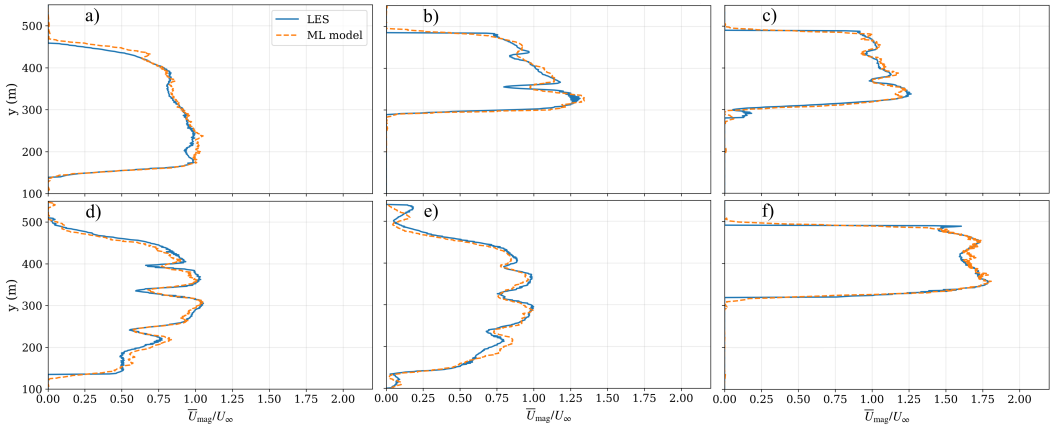


Figure 5. Cross-stream profiles of the normalized time-averaged velocity magnitude, $\overline{U}/U_\infty$, at the free surface for the straight channel inlet boundary condition. Solid lines denote LES results, while dashed lines represent CNN predictions. Panels (**a**–**f**) correspond to cross-sections A–F, as marked in Figure 1.

Figure 6 further examines the model performance under the log-law profile. As seen, despite the change in the inflow specification, the surrogate model continues to reproduce the LES profiles with high accuracy across all sections. The predicted velocity distributions capture the primary lateral gradients and peak velocity regions associated with the channel geometry, indicating that the model retains its ability to represent the dominant flow features under unseen boundary conditions. Compared with the straight-channel case, slightly larger deviations are observed in sections exhibiting stronger asymmetry or sharper velocity gradients, particularly near the channel boundaries. The observed discrepancies are consistent with the increased complexity introduced by the log-law inflow, which induces more pronounced shear and non-uniform momentum distribution. Nevertheless, the CNN predictions remain well aligned with the LES reference profiles, demonstrating robust generalization of the surrogate model across different inlet boundary conditions.

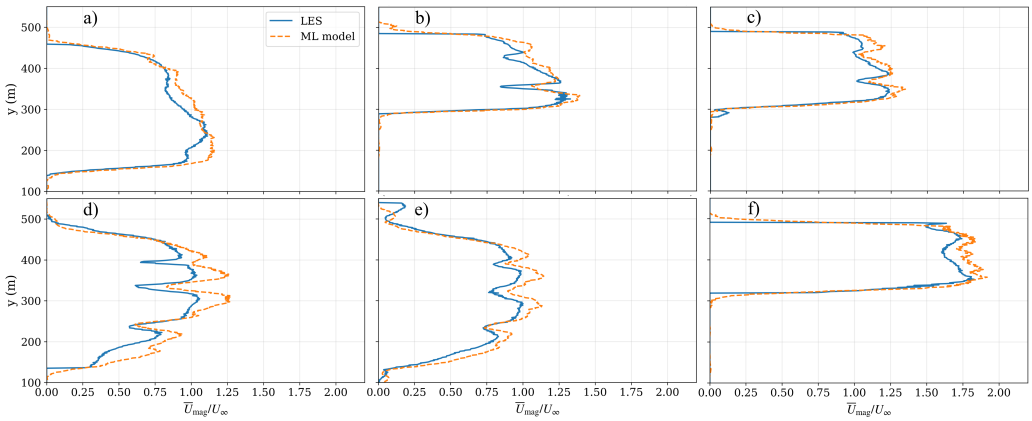


Figure 6. Cross-stream profiles of the normalized time-averaged velocity magnitude, $\overline{U}/U_\infty$, at the water surface for the log-law profile inlet boundary condition. Solid lines denote LES results, while dashed lines represent CNN predictions. Panels (**a**–**f**) correspond to cross-sections A–F, as marked in Figure 1.

Figures 7–9 illustrate the predictive performance of the surrogate model for the time-averaged velocity magnitude at mid-depth. Figure 7 presents the spatial distribution of the velocity field for both inlet boundary conditions. The CNN predictions closely reproduce the LES velocity patterns across the river reach, capturing the high-velocity core and the gradual velocity decay toward the channel banks. In addition, the model accurately

represents the redistribution of momentum induced by channel curvature, with localized acceleration zones forming along the outer banks of bends.

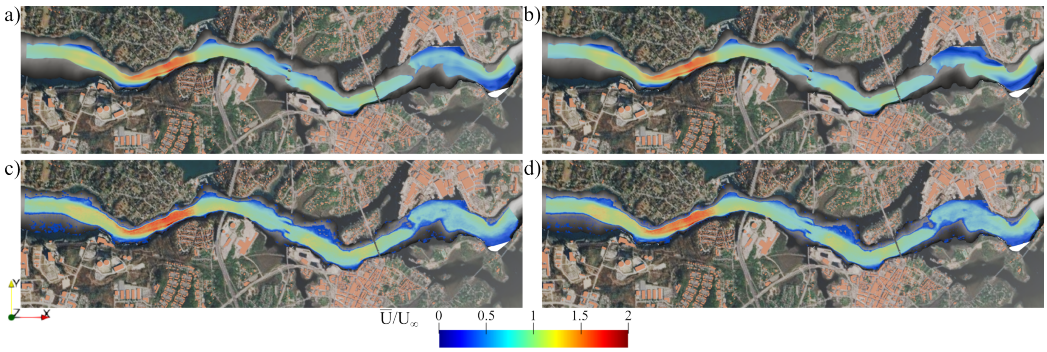


Figure 7. Color maps of the normalized time-averaged velocity magnitude $\overline{U}/U_\infty$, at the mid-depth for different turbulent inflow boundary conditions. Panels (**a**,**b**) show LES results for the log-profile and straight-channel inflows, respectively, and panels (**c**,**d**) present the corresponding CNN predictions.

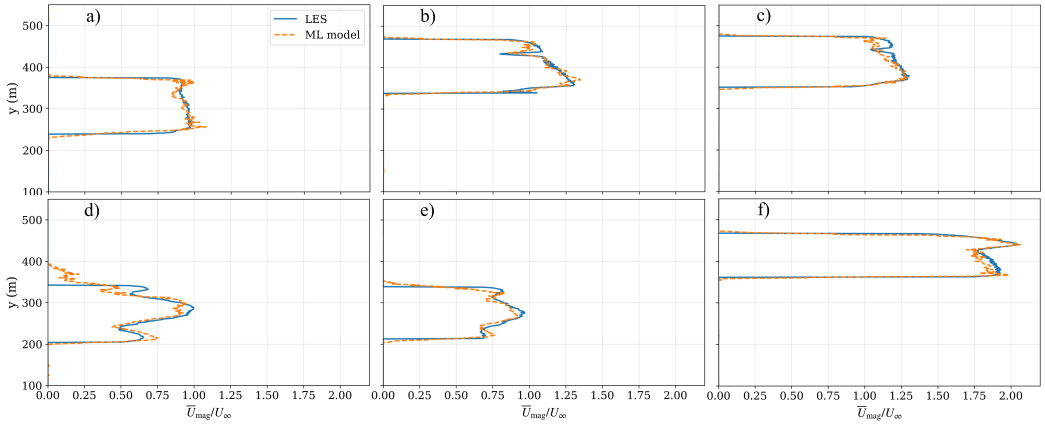


Figure 8. Cross-stream profiles of the normalized time-averaged velocity magnitude, $\overline{U}/U_\infty$, at the mid-depth for the straight-channel inlet boundary condition. Solid lines denote LES results, while dashed lines represent CNN predictions. Panels (**a**–**f**) correspond to cross-sections A–F, as shown in Figure 1.

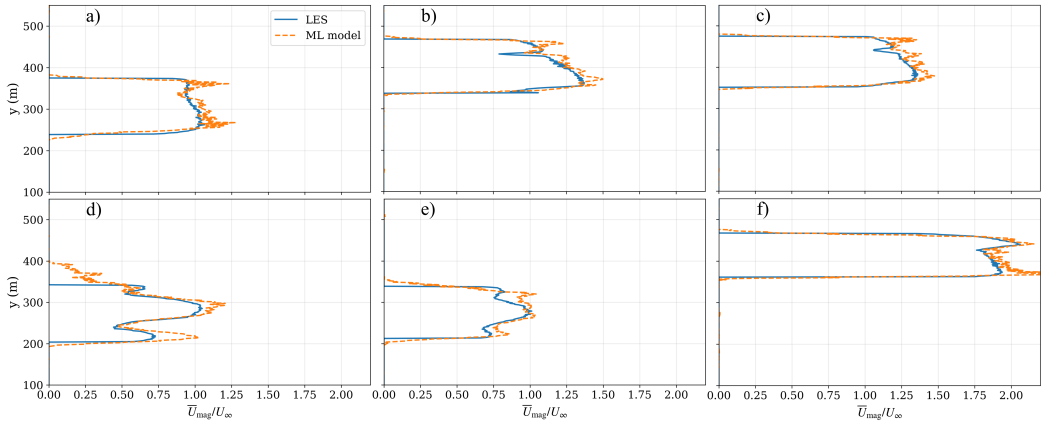


Figure 9. Cross-stream profiles of the normalized time-averaged velocity magnitude, $\overline{U}/U_\infty$, at the mid-depth for the log-profile inlet boundary condition. Solid lines denote LES results, while dashed lines represent CNN predictions. Panels (**a**–**f**) correspond to cross-sections A–F, as shown in Figure 1.

We plot in Figure 8 velocity profiles for the straight-channel inlet condition at mid-depth of the river. The predicted profiles show strong agreement with the LES results,

capturing both the magnitudes and the overall shapes of the mid-depth lateral velocity distributions. As expected, this alignment is especially pronounced in sections with smooth velocity gradients, while minor discrepancies appear in regions of sharper shear near the channel boundaries.

Figure 9 further evaluates the model under the log-law profile as the inlet boundary condition. Despite the increased complexity associated with non-uniform inflow profiles, the CNN predictions remain well aligned with the LES reference data. The model successfully captures the primary velocity gradients and peak locations across the channel, although slightly larger deviations are observed in sections exhibiting stronger asymmetry and sharper gradients.

Now we focus our attention on the CNN predictions for the second-order turbulence statistics, namely, the turbulence kinetic energy. As shown in Figures 10–12, the predictive performance of our model for the TKE field at the water surface is presented. Figure 10 compares the spatial distribution of TKE obtained from LES and CNN predictions for the two unseen inlet boundary conditions: log-law profile and straight channel inflow. The CNN model seems to have captured the large-scale spatial organization of turbulence intensity. More specifically, CNN has done a good job capturing the peaks of TKE along regions of strong shear and flow acceleration. However, compared with the mean velocity field, the TKE predictions exhibit an elevated background TKE pattern, visibly indicating a reduced ability to resolve localized turbulent structures and sharp spatial variations. We argue that such discrepancies are due to the elevated turbulence content (i.e., high background turbulence intensity) of the "river flow" inlet boundary conditions, with which the ML model was trained.

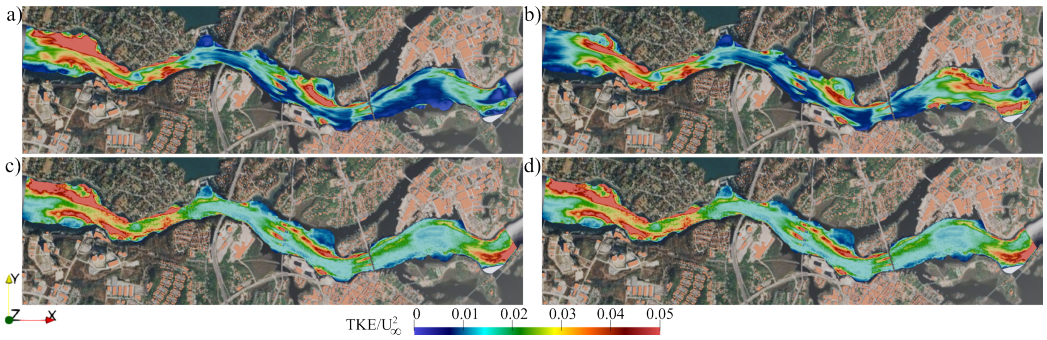


Figure 10. Color maps of $TKE/U_\infty{}^2$, at the water surface for different turbulent inflow boundary conditions. Panels (**a**,**b**) show LES results for the log-profile and straight-channel inflows, respectively, and panels (**c**,**d**) present the corresponding CNN predictions.

A quantitative analysis of the TKE computations of the two models at the free surface (see Figure 11) provides a more precise comparison throughout the study area. As seen, the CNN predictions generally follow the overall trends of the LES profiles, with discrepancies more noticeable than those observed in the velocity field. The CNN model captures the locations of major peaks; however, it tends to misrepresent the magnitude of localized TKE peaks and smooths out smaller-scale fluctuations, especially in regions with strong lateral gradients.

Figure 12 further evaluates the ML model under the log-law profile inlet boundary condition. As seen, the predicted TKE distributions across the cross sections resemble those of the LES prediction, although with larger deviations in sections exhibiting strong asymmetry and intermittent turbulent flow structures. Overall, these results indicate that while the surrogate model effectively captures the large-scale behavior of the TKE field, it has limited capability to resolve fine-scale turbulence features.

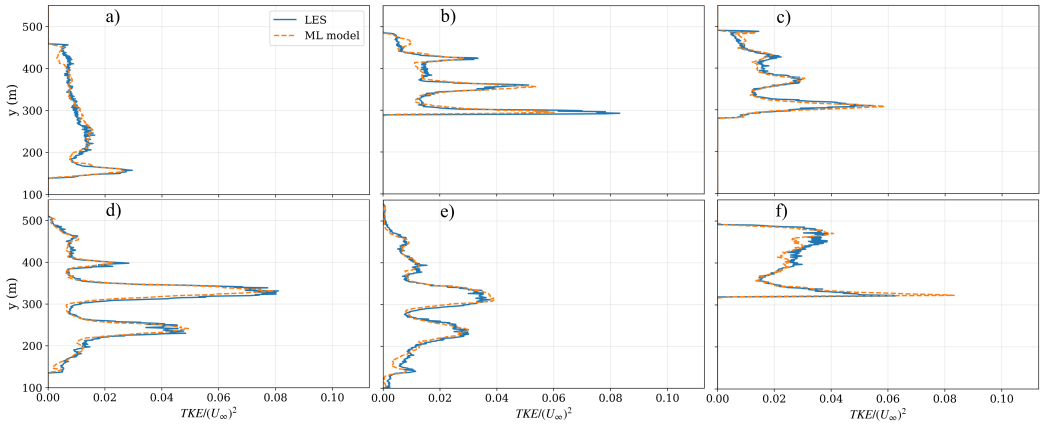


Figure 11. Cross-stream profiles of the normalized turbulent kinetic energy, TKE/U_∞^2, at the water surface for the straight-channel inlet boundary condition. Solid lines denote LES results, while dashed lines represent CNN predictions. Panels (**a**–**f**) correspond to cross-sections A–F defined in Figure 1.

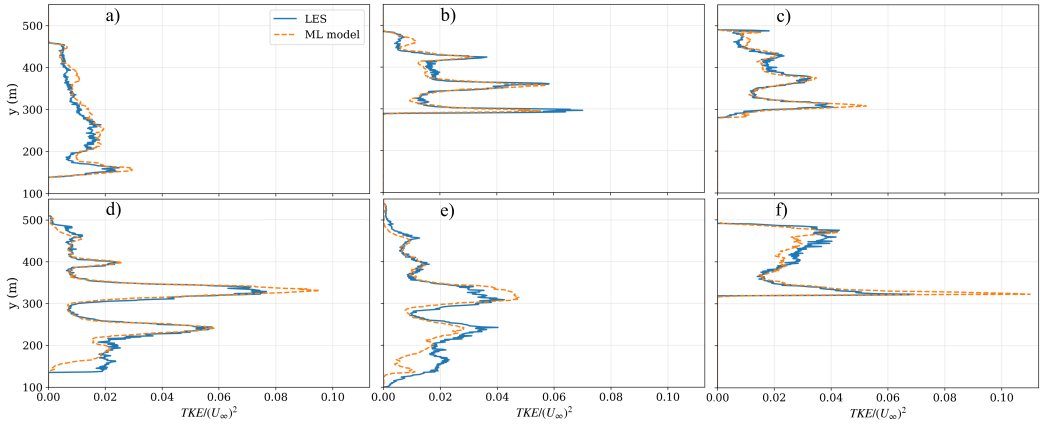


Figure 12. Cross-stream profiles of the normalized turbulent kinetic energy, TKE/U_∞^2, at the water surface for the log-profile inlet boundary condition. Solid lines denote LES results, while dashed lines represent CNN predictions. Panels (**a**–**f**) correspond to cross-sections A–F defined in Figure 1.

In Figure 13, we present the predictive performance of the surrogate model for the TKE field at mid-depth of the river flow. As seen, the ML model has successfully captured the overall spatial organization of the turbulence kinetic energy within the study area. However, similar to the results at the free surface, the predicted field appears to slightly underestimate the background TKE within the river, indicating a limited ability to resolve fine-scale turbulence content of the river.

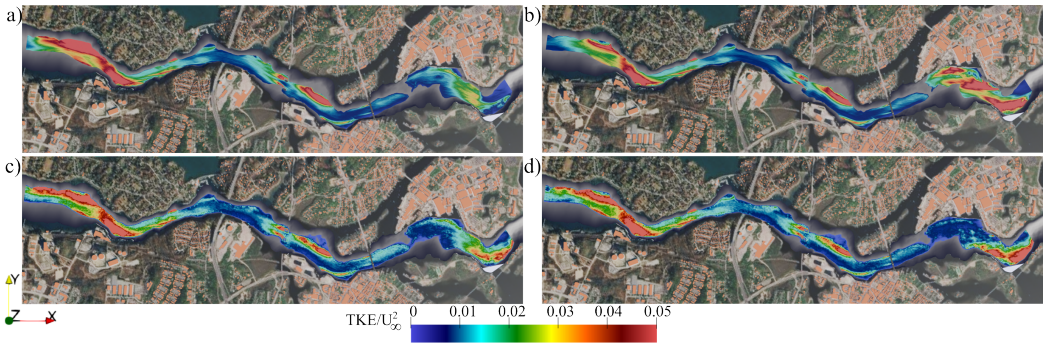


Figure 13. Color maps of $TKE/{U_\infty}^2$, at the mid-depth for different turbulent inflow boundary conditions. Panels (**a**,**b**) show LES results for the log-law profile and straight channel inflow conditions, respectively, and panels (**c**,**d**) present the corresponding CNN predictions.

On the other hand, as seen in Figures 14 and 15, the transverse profiles of the TKE at mid-depth show that the ML model has successfully captured the trend in the TKE distribution across the river. Nevertheless, discrepancies are more evident in regions with sharp gradients and localized peaks. The model tends to misrepresent the magnitude of peak TKE values, reflecting the difficulty of learning highly intermittent turbulence features from instantaneous inputs. We should note that the ML model's predictions have successfully produced the general TKE distributions, including the relative positioning of high- and low-energy regions across the channel.

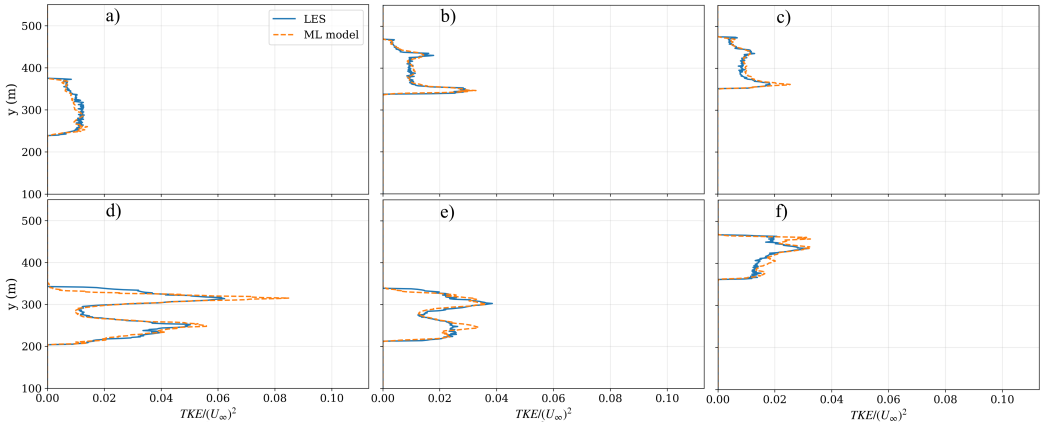


Figure 14. Cross-stream profiles of the normalized turbulent kinetic energy, TKE/U_∞^2, at the mid-depth for the straight-channel inlet boundary condition. Solid lines denote LES results, while dashed lines represent CNN predictions. Panels (**a**–**f**) correspond to cross-sections A–F, as described in Figure 1.

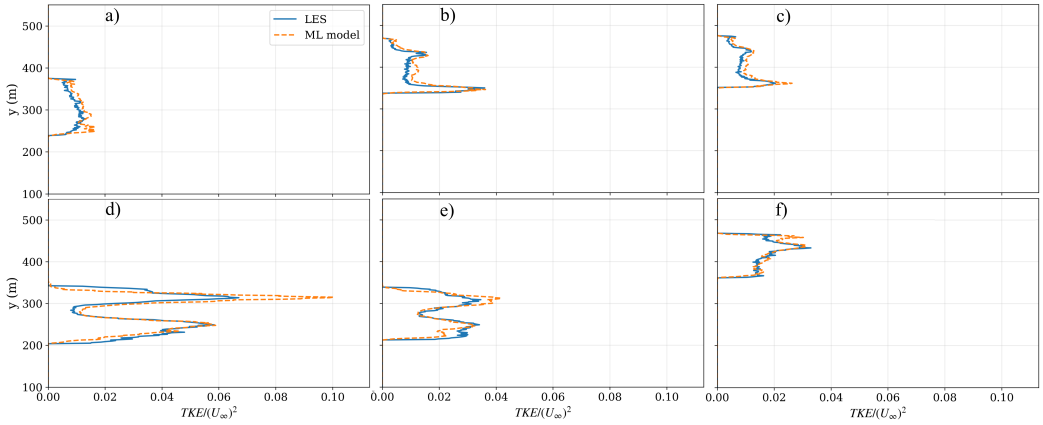


Figure 15. Cross-stream profiles of the normalized turbulent kinetic energy, TKE/U_∞^2, at the mid-depth for the log-law profile inlet boundary condition. Solid lines denote LES results, while dashed lines represent CNN predictions. Panels (**a**–**f**) correspond to cross-sections A–F, as defined in Figure 1.

Deviations are observed in sections with pronounced asymmetry and strong localized turbulence. These results suggest that while the CNN predictions capture the general spatial distribution of high- and low-TKE regions, the relatively low R^2 values demonstrate limited quantitative accuracy in TKE prediction. As a result, the current TKE outputs should be regarded as qualitative or semi-quantitative indicators of large-scale turbulence organization, rather than as fully reliable predictions of local turbulence intensity. To further assess the predictive performance of the trained neural network models, we used standard regression error metrics for both test cases, straight-channel and log-law profile boundary conditions.

Table 1 presents the error analysis results for the velocity magnitude prediction results (MLUavg), while Table 2 summarizes the performance of the ML model for TKE predictions

(MLTKE). The error analysis of the ML model includes the average loss, measured as mean squared error (MSE), root mean squared error (RMSE), and mean absolute error (MAE), as well as the coefficient of determination (R^2). It is worth mentioning that the present surrogate framework is deterministic and does not explicitly quantify predictive uncertainty, and that the above-mentioned assessments do not fully characterize predictive confidence for safety-critical or risk-informed hydrodynamic applications.

Table 1. Error metrics of the ML model prediction error for time-averaged velocity magnitude. The model was trained using the LES-computed flow field with "river flow" inflow boundary conditions and tested against an unseen LES-computed flow field with "straight channel" and "log-law" inflow boundary conditions.

Metric	Straight Channel	Log-Law
AVG Loss (MSE)	0.010018	0.005767
RMSE	0.100091	0.075942
MAE	0.037113	0.031087
R^2	0.954517	0.974122

As seen in Table 1, the MLUavg error analysis results indicate strong predictive capability across both boundary-condition scenarios. The RMSE and MAE remain relatively small, and the coefficient of determination exceeds 0.95 in both cases, suggesting that the model successfully captures the dominant spatial variability of the time-averaged velocity field even when tested under boundary conditions different from those used during training. The slightly improved performance observed with the log-law profile boundary condition, reflected in lower MSE, RMSE, and MAE values, as well as a higher R^2 score, indicates that the learned mapping generalizes well to this inflow configuration.

Similar to the time-averaged velocity prediction results in Table 1, the TKE prediction also performs better for the log-law profile boundary condition than for the straight-channel case, as shown in Table 2. Specifically, compared with the straight-channel case, the log-law case reduces the MSE by approximately 28.0%, the RMSE by 15.3%, and the MAE by 19.5%, while increasing the R^2 value by approximately 44.2%. This moderately improved performance under the log-law profile suggests that, although TKE prediction remains more challenging than mean-velocity prediction, the surrogate retains partial generalization capability across different inflow specifications. In contrast, the MLTKE error analysis (Table 2) exhibits larger prediction errors and substantially lower R^2 values compared with the MLUavg prediction errors. Although the MSE, RMSE, and MAE values for TKE are relatively small in absolute magnitude, the substantially lower R^2 values indicate that the model explains less of the spatial variance of the TKE field than it does for the time-averaged velocity field. This behavior suggests that the CNN captures the broad spatial organization of TKE but has difficulty reproducing localized high-gradient regions and peak values. This is consistent with previous CNN-based turbulence-prediction studies. Santoni et al. [44] reported larger prediction errors for TKE than for velocity fields when using a CNN and attributed this difference to the stronger nonlinearities of turbulent-flow quantities. They also suggested that additional instantaneous snapshots or deeper architectures may be required for improved TKE reconstruction. Similarly, Zhang et al. [30] showed that second-order turbulence statistics in riverine flows can be highly heterogeneous, differing by nearly two orders of magnitude across the domain. They also noted that relative or normalized error metrics can be inflated when reference quantities are very small in magnitude. Therefore, the lower quantitative accuracy of the present TKE predictions reflects both model limitations and the inherent difficulty of learning small-magnitude,

intermittent, and spatially heterogeneous second-order turbulence statistics from limited instantaneous inputs.

Several factors likely contribute to this behavior. First, TKE exhibits sharper spatial gradients and localized high-magnitude extrema, especially within shear layers, wake regions, and areas of intense turbulence production. These localized extrema are difficult to reconstruct from a limited number of instantaneous LES snapshots. Second, the encoder–decoder architecture may introduce smoothing because strided convolutions reduce the spatial resolution of feature maps, improving computational efficiency but potentially suppressing small-scale, high-frequency turbulent structures. Third, the MSE loss penalizes errors across the entire computational domain, which tends to favor spatially smooth predictions that minimize the average error rather than preserve localized extrema. Consequently, the model captures the large-scale spatial distribution of TKE but does not consistently reproduce the magnitude of localized extrema.

Insufficient temporal sampling may further contribute to the observed discrepancies in localized TKE extrema. Because TKE is derived from turbulent fluctuations, it is more sensitive to rare and intermittent flow events than the mean velocity field. A limited set of instantaneous snapshots may not fully capture the temporal variability of peak turbulence regions, causing the model to learn dominant spatial patterns while missing less frequent high-intensity events. Thus, the observed TKE discrepancies likely result from a combination of architectural smoothing, MSE-based loss averaging, and limited temporal sampling. Future work may explore peak-aware loss functions-such as weighted MSE, percentile-weighted losses, or hybrid approaches that assign greater penalties to cells with large values, increased snapshot sampling, and architectures incorporating skip connections [37] to improve the prediction of localized TKE extrema.

Table 2. Error metrics of the ML model prediction errors for the TKE field. The model was trained using the LES-computed flow field with "river flow" inflow boundary conditions and tested against an unseen LES-computed flow field with "straight channel" and "log-law" inflow boundary conditions.

Metric	Straight Channel	Log-Law
AVG Loss (MSE)	0.000082	0.000059
RMSE	0.009053	0.007664
MAE	0.002385	0.001919
R^2	0.455691	0.656978

We should note that in the multivariate model, cube-root preprocessing of the input vector was shown to be crucial for stable convergence and improved reconstruction of energetic zones, despite its detrimental effect on predicting local peak velocity magnitudes and TKE. Single-variable models, on the other hand, demonstrated that even limited information from instantaneous velocity fields can yield physically consistent mean-flow estimates.

Figure 16 shows the evolution of the training and testing losses over the course of model training on a logarithmic scale. Both losses decrease rapidly during the early epochs, indicating effective initial learning, and then gradually level off as training progresses. As expected for unseen data, the testing loss remains higher than the training loss throughout training; however, the testing curve follows the same decreasing trend as the training curve and does not exhibit late-epoch divergence. By the final epoch, the training and testing losses reach approximately 2.5×10^{-3} and 8.0×10^{-3}, respectively, corresponding to a final generalization gap of approximately 5.5×10^{-3}. Although this gap indicates that the model performs better on the training data than on unseen test data, the stable test loss behavior suggests that the model does not suffer from severe overfitting and retains acceptable generalization capability.

The computational advantage of the proposed surrogate model was assessed by comparing the cost of generating LES-based reference fields with the cost of the CNN-based prediction workflow. In the present study, each LES-based numerical result required approximately 32,000 CPU-hours on AMD EPYC CPU nodes. In comparison, the complete CNN training procedure required approximately 30 GPU-hours on an NVIDIA A100 GPU node. Although CPU-hours and GPU-hours are not directly hardware-equivalent units, the comparison highlights the substantially lower workflow-level resource requirement of the surrogate approach: the LES benchmark required tens of thousands of CPU-hours, whereas the CNN training required only tens of GPU-hours, followed by sub-second inference after training. After training, the inference time required to generate a predicted time-averaged field was approximately 0.8 s. This indicates that the trained surrogate model can provide near-instantaneous predictions once the offline training is complete. The resulting acceleration is particularly valuable for repeated evaluations, sensitivity studies, and scenario testing, where performing a separate LES simulation for each new condition would be computationally prohibitive. It should be noted that CPU-hours and GPU-hours are not identical hardware-normalized units; therefore, the above ratio should be interpreted as a workflow-level reduction in allocated computational cost rather than a direct processor-to-processor floating-point speed-up.

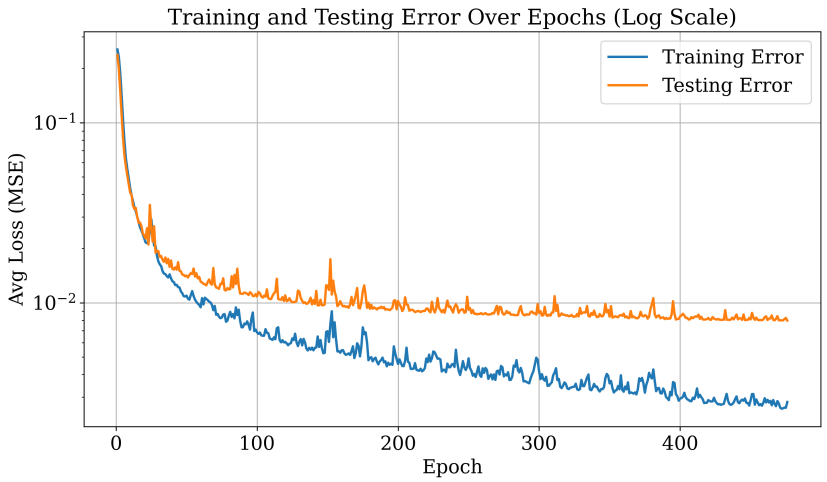


Figure 16. Training and testing MSE histories over training epochs, plotted on a logarithmic scale.

4. Conclusions

We developed and evaluated convolutional neural network surrogates to predict time-averaged mean velocity and TKE fields in a real-life riverine environment. Training the ML model on high-fidelity LES data for the Piscataqua River, we demonstrated that even a small number of instantaneous flow snapshots can be used to reconstruct the river's mean flow field at a fraction of the cost of high-fidelity LES. The cube-root preprocessing of multi-component inputs was crucial for stabilizing training and improving predictions in shear-dominated regions. The models showed sensitivity to distributional shifts in the inlet boundary conditions, underscoring the need for domain adaptation strategies in future work. The observed sensitivity to strong inflow-distribution shifts indicates that the present surrogate is most reliable for boundary conditions close to the training distribution and should be applied cautiously outside the range of boundary conditions represented in the training data. Future work will improve generalization by training the model on a wider range of inflow conditions, adapting it to new boundary conditions, and quantifying prediction uncertainty. Overall, our study confirms that data-driven surrogates can approximate first- and second-order turbulence statistics from a small number of

instantaneous fields, thereby enabling fast, LES-informed hydrodynamic assessments of natural river flows. Although the proposed surrogate achieved a high quantitative accuracy for time-averaged velocity prediction, its performance for TKE was more limited. Therefore, the present TKE predictions should be interpreted as qualitative to semi-quantitative indicators of large-scale turbulence organization rather than fully reliable local predictions of turbulent intensity. Future improvements may include TKE prediction through weighted loss functions, increased temporal sampling, enhanced CNN architectures, and physics-informed regularization to better preserve localized turbulent structures and improve physical consistency.

The current study focused on river flow prediction in the absence of human-made wall-mounted energy infrastructure. In a future study, we plan to include arrays of tidal hydrokinetic turbines in the study area and assess how well the model predicts the velocity field, TKE, and power production of the marine energy devices. Additionally, we plan to explore more advanced neural network architectures to assess whether they improve prediction accuracy and generalization. Ultimately, we will explore transfer learning techniques to leverage pre-trained models for similar fluid dynamics problems, potentially reducing the amount of training data required for effective surrogate modeling and facilitating the framework's adaptation to new river systems beyond the Piscataqua River.

Author Contributions: Conceptualization, S.S.M. and A.K.; methodology, S.S.M.; software, S.S.M. and M.G.A.; validation, S.S.M., M.M.A. and M.G.A.; formal analysis, S.S.M. and M.G.A.; investigation, S.S.M. and M.G.A.; resources, A.K.; data curation, S.S.M. and M.G.A.; writing—original draft preparation, S.S.M.; writing—review and editing, S.S.M., M.M.A., M.G.A., F.S., S.K. and A.K.; visualization, S.S.M.; supervision, A.K.; project administration, A.K.; funding acquisition, A.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the U.S. National Science Foundation (NSF) grant number 2233986 and the U.S. Department of Energy's Office of Critical Minerals and Energy Innovation under the Hydropower and Hydrokinetic Office (H2O), Award No. DE-EE0009450.

Data Availability Statement: The LES model's code is openly available at https://doi.org/10.5281/zenodo.15002824. The hydrodynamic simulation results can be accessed at https://doi.org/10.5281/zenodo.18331415, and the ML code is available at https://github.com/samin75/Simulation-of-Turbulence-in-the-Piscataqua-River.git, accessed on 15 April 2026.

Acknowledgments: The views expressed herein do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ADCP	Acoustic Doppler Velocity Profiler
BC	Boundary condition
CNN	Convolutional neural network
LES	Large-eddy simulation
LSTM	Long short-term memory
MAE	Mean absolute error
ML	Machine learning
MSE	Mean squared error
NN	Neural network
PINN	Physics-informed neural network
POD	Proper orthogonal decomposition

RANS	Reynolds-averaged Navier–Stokes
ReLU	Rectified linear unit
RF	River flow
RMSE	Root mean squared error
ROM	Reduced-order modeling
SC	Straight channel
TKE	Turbulent kinetic energy
VFS	Virtual Flow Simulator
3D	Three-dimensional

References

1. Hemmasian, A.; Farimani, A.B. Reduced-order modeling of fluid flows with transformers. *Phys. Fluids* **2023**, *35*, 057126 [CrossRef]
2. Chanson, H.; Toi, Y.H. Physical modeling of breaking tidal bores: Comparison with prototype data. *J. Hydraul. Res.* **2015**, *53*, 264–273. [CrossRef]
3. Murata, T.; Fukami, K.; Fukagata, K. Nonlinear mode decomposition with convolutional neural networks for fluid dynamics. *J. Fluid Mech.* **2020**, *882*, A13.
4. Nakamura, T.; Fukami, K.; Hasegawa, K.; Nabae, Y.; Fukagata, K. Convolutional neural network and long short-term memory based reduced order surrogate for minimal turbulent channel flow. *Phys. Fluids* **2021**, *33*, 025116. [CrossRef]
5. Fukami, K.; Nabae, Y.; Kawai, K.; Fukagata, K. Synthetic turbulent inflow generator using machine learning. *Phys. Rev. Fluids* **2019**, *4*, 064603. [CrossRef]
6. Fukami, K.; Fukagata, K.; Taira, K. Super-resolution reconstruction of turbulent flows with machine learning. *J. Fluid Mech.* **2019**, *870*, 106–120. [CrossRef]
7. Mohan, A.; Daniel, D.; Chertkov, M.; Livescu, D. Compressed Convolutional LSTM: An Efficient Deep Learning framework to Model High Fidelity 3D Turbulence. *arXiv* **2019**, arXiv:1903.00033.
8. Halder, R.; Ataei, M.; Salehipour, H.; Fidkowski, K.; Maki, K. Reduced-order modeling of unsteady fluid flow using neural network ensembles. *Phys. Fluids* **2024**, *36*, 077140. [CrossRef]
9. Li, Z.; Peng, W.; Yuan, Z.; Wang, J. Long-term predictions of turbulence by implicit U-Net enhanced Fourier neural operator. *Phys. Fluids* **2023**, *35*, 075145. [CrossRef]
10. Ling, J.; Kurzawski, A.; Templeton, J. Reynolds averaged turbulence modeling using deep neural networks with embedded invariance. *J. Fluid Mech.* **2016**, *807*, 155–166. [CrossRef]
11. Duraisamy, K. Perspectives on machine learning-augmented Reynolds-averaged and large eddy simulation models of turbulence. *Phys. Rev. Fluids* **2021**, *6*, 050504. [CrossRef]
12. Lippmann, T.C.; Cook, S.; Simpson, A. The Piscataqua River Currents with Sea Level Rise. In Proceedings of the New Hampshire Coastal Adaptation Workgroup (NHCAW), Greenland, NH, USA, 20 June 2018.
13. Cook, S.E.; Lippmann, T.C.; Irish, J.D. Modeling nonlinear tidal evolution in an energetic estuary. *Ocean Model.* **2019**, *136*, 13–27. [CrossRef]
14. Aksen, M.M.; Gholami Anjiraki, M.; Craig, J.; Choi, J.G.; Lippmann, T.; Wosnik, M.; Khosronejad, A. Feasibility study of a tidal energy site in the Piscataqua River, New Hampshire, using large-eddy simulation. *Adv. Water Resour.* **2026**, *213*, 105302. [CrossRef]
15. Gopalan, H.; Heinz, S.; Stöllinger, M.K. A unified RANS–LES model: Computational development, accuracy and cost. *J. Comput. Phys.* **2013**, *249*, 249–274. [CrossRef]
16. Pope, S.B. *Turbulent Flows*; See Chapter 7 for Wall-Bounded Turbulence and Discussion of the von Kármán Constant; Cambridge University Press: Cambridge, UK, 2000. [CrossRef]
17. Zhang, Y.; Zhang, D.; Jiang, H. Review of Challenges and Opportunities in Turbulence Modeling: A Comparative Analysis of Data-Driven Machine Learning Approaches. *J. Mar. Sci. Eng.* **2023**, *11*, 1440. [CrossRef]
18. IPCC. *Climate Change 2021: The Physical Science Basis*; Cambridge University Press: Cambridge, UK, 2021. [CrossRef]
19. Maulik, R.; Fukami, K.; Ramachandra, N.; Fukagata, K.; Taira, K. Probabilistic neural networks for fluid flow surrogate modeling and data recovery. *Phys. Rev. Fluids* **2020**, *5*, 104401. [CrossRef]
20. Hoptopphya, A.; Hassanzadeh, P.; Pasha, S. Predicting clustered weather patterns: A test case for applications of convolutional neural networks to spatio-temporal climate data. *Sci. Rep.* **2020**, *10*, 1317. [CrossRef] [PubMed]
21. Buffin-Bélanger, T.; Rice, S.; Reid, I.; Lancaster, J. Spatial heterogeneity of near-bed hydraulics above a patch of river gravel. *Water Resour. Res.* **2006**, *42*, W04413. [CrossRef]
22. Legleiter, C.J. A geostatistical framework for quantifying the reach-scale spatial structure of river morphology: 2. Application to restored and natural channels. *Geomorphology* **2014**, *205*, 85–101. [CrossRef]

23. Trinci, G.; Harvey, G.L.; Henshaw, A.J.; Bertoldi, W. Turbulence signatures of natural river morphology in four dimensions. *River Res. Appl.* **2023**, *39*, 122–133. [CrossRef]
24. Ye, N.; Li, K.; Bai, H.; Yu, R.; Hong, L.; Zhou, F.; Li, Z.; Zhu, J. OoD-Bench: Quantifying and Understanding Two Dimensions of Out-of-Distribution Generalization. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 21–24 June 2022; pp. 7937–7948. [CrossRef]
25. Brunton, S.L.; Noack, B.R.; Koumoutsakos, P. Machine learning for fluid mechanics. *Annu. Rev. Fluid Mech.* **2020**, *52*, 477–508. [CrossRef]
26. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **2019**, *378*, 686–707. [CrossRef]
27. Karniadakis, G.E.; Kevrekidis, I.G.; Lu, L.; Perdikaris, P.; Wang, S.; Yang, L. Physics-informed machine learning. *Nat. Rev. Phys.* **2021**, *3*, 422–440. [CrossRef]
28. Donnelly, J.; Daneshkhah, A.; Abolfathi, S. Physics-informed neural networks as surrogate models of hydrodynamic simulators. *Sci. Total Environ.* **2024**, *912*, 168814. [CrossRef] [PubMed]
29. Wang, S.; Teng, Y.; Perdikaris, P. Understanding and Mitigating Gradient Flow Pathologies in Physics-Informed Neural Networks. *SIAM J. Sci. Comput.* **2021**, *43*, A3055–A3081. [CrossRef]
30. Zhang, Z.; Flora, K.; Kang, S.; Limaye, A.B.; Khosronejad, A. Data-Driven Prediction of Turbulent Flow Statistics Past Bridge Piers in Large-Scale Rivers Using Convolutional Neural Networks. *Water Resour. Res.* **2022**, *58*, e2021WR030163. [CrossRef]
31. Fukami, K.; Fukagata, K.; Taira, K. Assessment of supervised machine learning methods for fluid flows. *Theor. Comput. Fluid Dyn.* **2020**, *34*, 497–519. [CrossRef]
32. Ghafariasl, P.; Mahmoudan, A.; Mohammadi, M.; Nazarparvar, A.; Hoseinzadeh, S.; Fathali, M.; Chang, S.; Zeinalnezhad, M.; Astiaso Garcia, D. Neural network-based surrogate modeling and optimization of a multigeneration system. *Appl. Energy* **2024**, *364*, 123130. [CrossRef]
33. Moin, P.; Mahesh, K. Direct Numerical Simulation: A Tool in Turbulence Research. *Annu. Rev. Fluid Mech.* **1988**, *30*, 539–578. [CrossRef]
34. Germano, M.; Piomelli, U.; Moin, P.; Cabot, W.H. A dynamic subgrid-scale eddy viscosity model. *Phys. Fluids A Fluid Dyn.* **1991**, *3*, 1760–1765. [CrossRef]
35. Smagorinsky, J. General Circulation Experiments with the Primitive Equations: I. The Basic Experiment. *Mon. Weather. Rev.* **1963**, *91*, 99–164. [CrossRef]
36. Lilly, D.K. A proposed modification of the Germano subgrid-scale closure method. *Phys. Fluids A Fluid Dyn.* **1992**, *4*, 633–635. [CrossRef]
37. Gholami Anjiraki, M.; Santoni, C.; Shapourmiandouab, S.; Seyedzadeh, H.; Craig, J.; Sotiropoulos, F.; Khosronejad, A. Wind farm layout optimization using a novel machine learning approach. *Phys. Fluids* **2026**, *38*, 055124. [CrossRef]
38. Peskin, C.S. The immersed boundary method. *Acta Numer.* **2002**, *11*, 479–517. [CrossRef]
39. Gholami Anjiraki, M.; Aksen, M.M.; Shapourmiandouab, S.; Craig, J.; Khosronejad, A. Computational Study of a Utility-Scale Vertical-Axis MHK Turbine: A Coupled Approach for Flow–Sediment–Actuator Modeling. *Fluids* **2025**, *10*, 304. [CrossRef]
40. Poinsot, T.J.; Lelef, S.K. Boundary conditions for direct simulations of compressible viscous flows. *J. Comput. Phys.* **1992**, *101*, 104–129. [CrossRef]
41. Schlichting, H.; Gersten, K. *Boundary-Layer Theory*, 8th ed.; Springer: Berlin/Heidelberg, Germany, 2000. [CrossRef]
42. Loshchilov, I.; Hutter, F. Decoupled Weight Decay Regularization. *arXiv* **2017**, arXiv:1711.05101. [CrossRef]
43. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980. [CrossRef]
44. Santoni, C.; Zhang, Z.; Sotiropoulos, F.; Khosronejad, A. A data-driven machine learning approach for yaw control applications of wind farms. *Theor. Appl. Mech. Lett.* **2023**, *13*, 100471. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.