Ocean Engineering 348 (2026) 124137

Contents lists available at ScienceDirect

Ocean Engineering

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

Research paper

A data-driven approach to real-time reconstruction of turbulent flow fields above a coarse-grain bed using sparse observational boundary data

Yifan Yang [a], Yushu Xie [b,*], Julia C. Mullarney [c], Dong Shao [d]

[a] State Key Laboratory of Water Resources Engineering and Management, Wuhan University, Wuhan, China
[b] Department of Civil and Environmental Engineering, The University of Auckland, Auckland, New Zealand
[c] Coastal Marine Group, School of Science, The University of Waikato, Hamilton, New Zealand
[d] Key Laboratory for Mechanics in Fluid Solid Coupling Systems, Institute of Mechanics, Chinese Academy of Sciences, Beijing, China

ARTICLE INFO

Keywords:
Turbulence
Rough bed
Data-driven
Real-time reconstruction
Convolutional neural network

ABSTRACT

Reconstructing turbulent flow fields from sparse or partial measurements has become increasingly feasible with advances in data-driven machine learning. This study introduces a convolutional autoencoder-based neural network for reconstructing real-time velocity and turbulent kinetic energy (TKE) fields at the same sampling time as the boundary input over coarse-grain beds in a 100 mm $\times$ 100 mm region, solely using upstream boundary profiles. A scalable model architecture is established to handle various input/output data formats, and a physics-informed training strategy is implemented to enhance training efficiency and accuracy. Assessments demonstrate that the model effectively captures spatiotemporal flow patterns, particularly horizontal and total velocities, while time-averaged fields further reduce errors in real-time outputs. However, accuracy decreases for vertical velocity and high-TKE regions. Robustness tests indicate that the model generally remains stable under incomplete or corrupted inputs, whereas TKE reconstructions are more sensitive to those disturbances and require cleaner data and improved training. This study highlights the importance of embedding more physical laws within the model. Overall, the proposed model demonstrates promising potential for real-time flow field reconstruction that requires fewer input data, though further refinement is necessary to improve the generalizability, physical consistency, and application to more complex flow environments.

1. Introduction

Flow measurements in open channel flows, encompassing laboratory flumes, rivers, and artificial conduits, provide essential data of both mean speeds and turbulent characteristics for use in hydrological or engineering applications. This information is crucial for understanding sediment transport dynamics, especially for flows over beds with significant wall roughness (Nakagawa, 1993). However, capturing measurements in energetic, and highly spatially and temporally heterogeneous environments is often logistically challenging and costly. Typical applications may use sparsely distributed sensors at only a few locations, which prohibits comprehensive knowledge of the heterogeneity of the flow-field. Digital surrogate-like tools are promising new solutions to those difficulties in water management. There is a growing demand among researchers and practitioners for sophisticated flow-field analytical tools that can offer rapid response times, real-time monitoring capabilities, and the ability to process vast amounts of flow-field data concurrently to reconstruct comprehensive flow information from limited measurements. In this study, the specific task is to reconstruct domain-wide turbulent fields from sparse upstream boundary observations in a rough-bed flow setting, with the capacity for data automation.

In practice, people are often familiar with performing fixed-point or transactional measurements using instruments like acoustic Doppler current profilers (ADCPs) for open channel flows (Gotvald and Oberg, 2009), but this approach is laborious and does not enable comprehensive and real-time data feedback. Therefore, the ability to efficiently transform (or upsample) measured data into more comprehensive flow field with higher dimensions in a data-driven manner is invaluable. In these scenarios, measurements provide crucial boundary conditions that are directly related to, and inform the inference of, the broader flow field downstream. This measurement-driven “boundary-to-field” reconstruction is complementary to (and distinct from) standard CFD simulation; rather than solving governing equations for a prescribed case, the aim is to infer the instantaneous/updated field directly from streaming

* Corresponding author.
E-mail address: yxie399@aucklanduni.ac.nz (Y. Xie).

https://doi.org/10.1016/j.oceaneng.2025.124137
Received 3 November 2025; Received in revised form 23 December 2025; Accepted 29 December 2025
Available online 31 December 2025
0029-8018/© 2025 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

observations, particularly when acquiring measurements is difficult. Although RANS modelling is good at simulating large-scale channel flows with better efficiency than large-eddy (LES) or detached-eddy (DES) simulations, more complex time-space structures such as streamwise-oriented vortices are often not resolved (e.g., Zeng et al., 2008, Déjeans et al., 2022); LES and DES are good at simulating complex time-dependent turbulence fields but are much more computationally demanding (Koken et al., 2013). Moreover, repeatedly re-running CFD to reflect frequent measurement updates is typically impractical for real-time monitoring, motivating surrogate reconstruction approaches. Data-driven approaches offer promise towards resolving this “accuracy or efficiency” dilemma through combining machine learning with vast data streams (Brunton et al., 2020).

In recent years, the rapid advancement of machine learning-based data-driven approaches has made a significant impact on hydro-environmental research. This progress is evident in flood mapping (Zahura et al., 2020; Bentivoglio et al., 2022), urban water system modelling (Fu et al., 2022), bridge scour monitoring (Yousefpour et al., 2021), and rainfall-runoff prediction (Kao et al., 2020). A majority of these studies used machine learning models to process, transform, and predict observed data, often working with tabular-to-tabular or sequence-to-sequence structures.

However, reconstructing serial data fields (i.e., the task of this study) is more complicated due to the inherent spatiotemporal complexities. In general, we identified two primary approaches to this challenge: field dimensionality reduction using POD (Proper Orthogonal Decomposition) or DMD (Dynamic Mode Decomposition), and machine learning using convolutional neural network-based autoencoders. Here, POD/DMD are representatives of a class of mathematical techniques used for dimensionality reduction, modal analysis, and reduced-order modelling. These methods operate by decomposing data into a set of orthogonal modes, capturing the most significant structures or dynamics in the data, thereby facilitating fast field reconstruction from predominant modes. For example, Scherl et al. (2020) used robust principal component analysis to improve the quality of PIV-measured flow-field data by leveraging global coherent structures to repair corrupted data points. However, traditional POD/DMD methods do not learn from data, despite the potential for integration with neural networks or physical laws for enhanced analysis (Baddoo et al., 2023). Importantly, in their standard form, POD/DMD methods typically rely on a modal basis extracted from full-field snapshots and are not designed to infer a downstream 2D turbulent field solely from sparse upstream boundary profiles in an online, frame-by-frame manner; therefore, this study adopts the neural network-based approach for the proposed task.

Machine learning, particularly deep learning-based autoencoders, can assist with augmenting flow-field measurements to restore or enhance data quality and improve visualization (Raissi et al., 2020; Vinuesa et al., 2023), leveraging their good ability to discern spatial patterns through the encoding-decoding pipeline, which is particularly suitable for the task of the present study. Such techniques has been verified across many applications, including subsurface flow (Wang et al., 2022; Zhou et al., 2022a,b), enhance flow fields based on obscure (Zhang et al., 2022a; Yang et al., 2023), sparse (Callaham et al., 2019; Erichson et al., 2020; Xu et al., 2023; Zhang et al., 2023), or noisy data (Gao et al., 2021), and, notably, prediction of open channel flow-fields (Zhang et al., 2022b). However, none of these studies have addressed the “boundary-to-field” challenge proposed in this study.

The present study utilizes the capabilities of convolutional neural networks (CNNs) with a customized autoencoder architecture for processing of flow-field data. Our primary objective is to develop a new model framework to address two main gaps identified in the existing literature: first, the absence of an effective model for flow-field reconstruction based on upstream boundary measurements, and second, the lack of a workflow designed for real-time feedback that enables data updating and automation. Different from many prior “sparse sensors to field” studies that use interior-point sensors or partial-field masking, our input is constrained to a single upstream boundary transect, making the inverse mapping more ill-posed and therefore a stronger test of generalizability and robustness. For this objective, we designed a simplified quasi-2D coarse-bed flow scenario to acquire data for model development, which is detailed in the following section. Based on those considerations, we have streamlined the basic workflow of real-time flow-field reconstruction using boundary measurements, as illustrated in Fig. 1. This conceptual approach features four key steps: (1) real-time data acquisition, (2) data preprocessing and feeding, (3) field reconstruction, and (4), data integration and time-averaging. It should be noted that Step (4) is an automated aggregation step that stabilizes streaming reconstructions and updates mean statistics as new frames arrive, rather than being the sole/ultimate goal of the workflow. In addition to velocity, we also reconstruct TKE (in a quasi-2D sense in this study), using a model trained separately, because it provides an application-relevant measure of turbulence intensity and energetic near-bed conditions, which is informative for interpreting rough-bed flow structures and related sediment-transport processes. More details of this workflow and model architecture can be found in the subsequent sections.

It is also worth mentioning that a main innovation of this study is we avoid predicting time-dependent flow fields using traditional recurrent neural networks, such as LSTM models, which may require substantial computational resources for training. Our approach, as shown in Fig. 1, focuses on real-time reconstruction using small-scale measurements processed efficiently by CNN models. Although this approach may not inherently support multi-step-ahead forecasting as some LSTM models in hydrology applications (e.g., Kao et al., 2020), the workflow ensures that the time-averaged state remains current, refining continuously as new instantaneous outputs are incorporated over time. This feature is particularly advantageous for digital-twin systems, which require large-scale data exchange in real-time and fast feedback loops with their real-world counterparts (Li et al., 2023; Yang et al., 2024).

The subsequent sections of this paper are aligned with the conceptual approach illustrated in Fig. 1. A brief introduction to the observational data used for model development is provided in Section 2, Section 3 outlines the model design and the integration of physical constraints during training, while the model's ability and efficacy in reconstructing both quasi-instantaneous and time-averaged velocity (and hence TKE fields) over a coarse-grain bed are discussed in Section 4. An analysis of the robustness of the model in scenarios with incomplete or corrupted data is given in Section 5. Lastly, the paper concludes with a discussion of potential applications and future research directions (Section 6).

2. Observational data

Laboratory flume experiments were undertaken to gather flow-field measurements essential for model training and validation. Measurements were collected of steady flow over a rough bed in a recirculating flume of 12-m length and 0.44-m width. The experimental design is detailed in Xie et al. (2023), and further validation is not presented here to conserve paper length; the present study primarily focuses on the flow-field reconstruction framework. The schematic flume setup is shown in Fig. 2.

The flume bottom was covered by a single layer of spherical elements, each of diameter D = 40 mm, representing a regular rough-bed environment. Those rough elements were densely organized in a hexagonal pattern, spanning a total length of 4.5 m, with gravel-formed sections at both ends to ensure a smooth flow transition from flat bottom to rough bottom (Fig. 2a). A 30-mm diameter target particle, representing a protruding coarse grain, was positioned to maintain contact with surrounding elements in a way that the overall layout was symmetric about the flume's centerline, as illustrated in Fig. 2b-c.

Five different flow depths (h = 0.135 m, 0.15 m, 0.165 m, 0.18 m and 0.195 m) were used. Each experiment was undertaken twice to augment the flow-field dataset. A two-dimensional particle tracking velocimeter

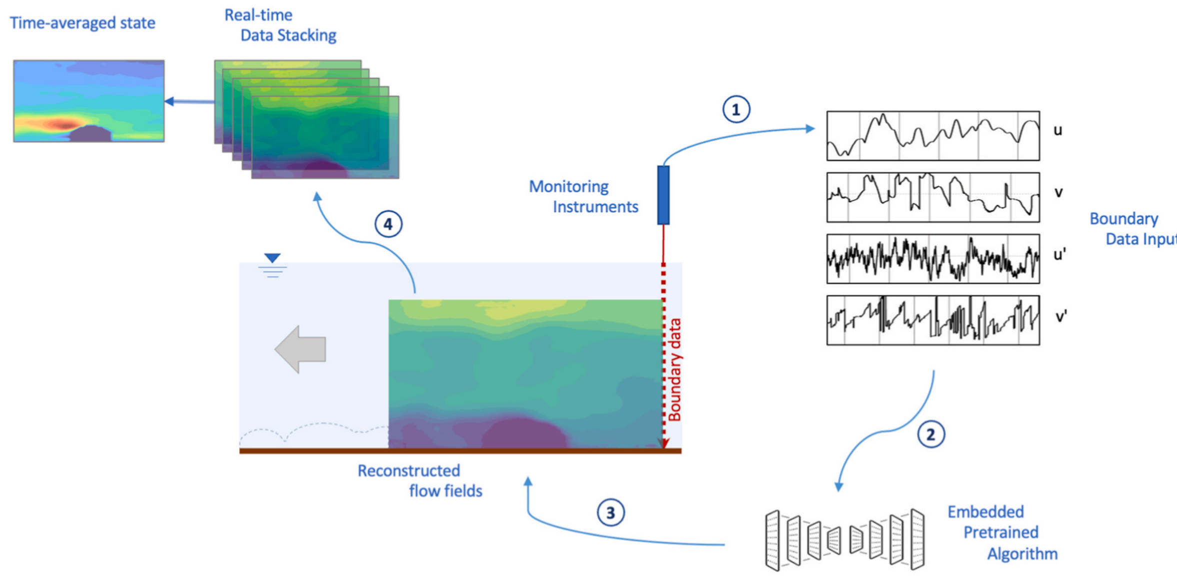

Fig. 1. Schematic illustration of the concept of real-time flow-field reconstruction based on sparse boundary measurement. Step 1 is the acquisition of real-time flow data at the boundary of the domain of interest; Step 2 integrates and preprocesses measured data and feeds them into the trained model; Step 3 reconstructs the real-time flow fields in the domain; Step 4 uses the real-time outputs to compute and update the time-averaged fields with a growing window.

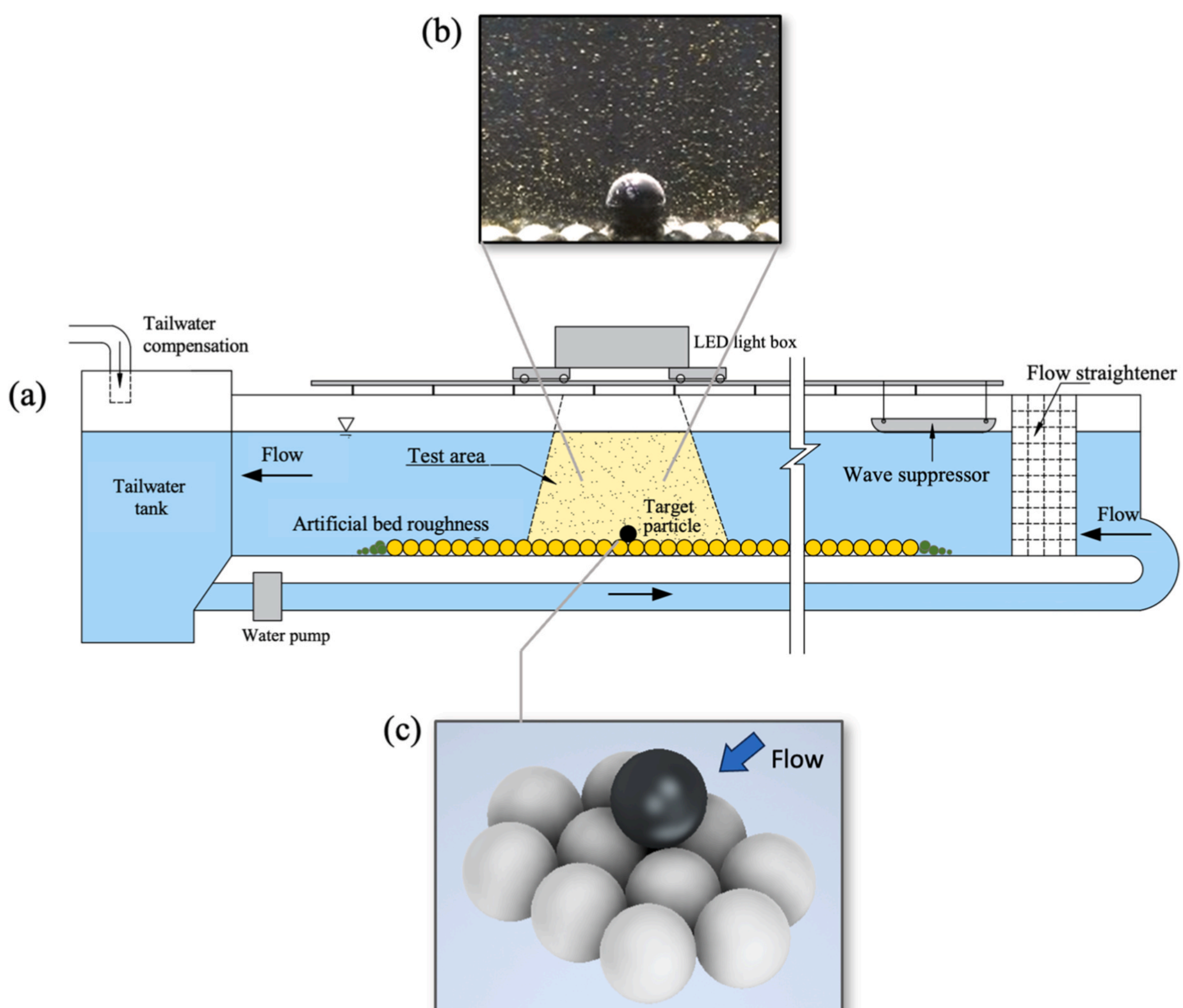

Fig. 2. Flume set-up for flow-field data acquisition: a) side view of the flume including artificial bed and light sheet; b) target particle, rough bed and illuminated tracing particles; (c) placement of target particle.

(PTV) was used to measure the flow field. A lightbox consisting of LED light strips was placed above the flume centerline as the light source for illumination. Tracing particles, with a median diameter of 10 μm, were monitored using a high-speed camera operating at 100 Hz. Each measurement period lasted for 10 s, resulting in a total of 1000 frames of flow-field data. The observation window was cropped to a 10 cm square sitting on the equivalent bed elevation (0.2 D below the top of bed particles) while symmetrically aligned with the center of the target particle. A summary of experimental parameters is provided in Table 1.

The PTV-recorded raw images were processed using the software *Streams* (Nokes, 2019) to generate four velocity fields over the measurement periods of 10 s. These fields include the along-channel (hereafter referred to as horizontal) and vertical velocity fields (i.e. u and v) and their respective velocity fluctuation components (i.e. u' and v') at a 2-mm spatial resolution covering a 100mm × 100 mm area. In particular, u' is the instantaneous u minus the mean velocity over the entire experimental period (10s), i.e., $u' = u - \overline{u}$. The total dataset used for model development comprised 10,000 recorded frames.

It should be noted that the data fields in this study were taken to be quasi-2D, as given the symmetrical bed layout, the lateral velocities were small, and the plane of measurement (i.e., the light sheet) overlapped with the longitudinal centerline of the target particle. Furthermore, the measurement cells occupied by the target particle were assigned with NaN values (no-flow), and this pattern was learnt automatically by the model during various convolution operations. This approach is a cost-effective method to incorporating bed geometry learning without further data processing work.

3. Model design

3.1. Model architecture

Convolutional autoencoders are widely used as a foundational architecture for processing data fields, with many successful attempts in flow-field prediction (Fukami et al., 2019; Zhang et al., 2022a, 2023). A convolutional autoencoder fuses the principles of convolutional neural networks with the encoding-decoding pipeline, learning a compressed representation of the input fields by first encoding the field data into a lower-dimensional space (i.e., the latent vector). The decoder part upsamples and reconstruct the original data from the latent state. The final output of the decoder is either the reconstructed input fields or other alternative fields associated with the input data.

Classical convolutional autoencoders can mitigate information lost during data compression by adding "skip connections", which is used in the well-known U-Net framework (Ronneberger et al., 2015). While U-Net was initially developed for semantic segmentation of medical images, this framework has been successful in many other "flow-to-field" reconstruction tasks (Thuerey et al., 2020; Xu et al., 2023). These features suit the purpose of this study very well, and therefore, the traditional U-Net was used as the base model for further development.

The architecture of the proposed model is shown in detail in Fig. 3, with additional details on layer operations and field resolutions available in Table 2. The measured boundary data are firstly interpolated to the desired size, e.g., a column vector of size 1024 in this study (A1), to facilitate reshaping to square arrays for subsequent operations. From B1-B3 onwards, each block features double 3 × 3 convolutions with a stride of 1, each followed by a batch normalization (for regularization and better training efficiency) and ReLU activation. ReLU (Rectified Linear Unit) is a widely used activation function in neural networks that outputs the input directly if the values are positive, otherwise, function outputs zero. Average pooling is employed for downsampling to preserve localized features in data fields with intermittent negative values; this operation is different from the max pooling operation adopted by traditional U-Net models. Upsampling utilizes 2 × 2 convolutional transpose layers with a stride of 2 to increase field resolution by a factor of two. Specifically, the "Upsampling 0" step in Fig. 3 represents enlarging reshaped boundary data to suit the desired output field size, which is eventually realized by the block G1-G3. This procedure ensures that the general patterns in the boundary data are not distorted given a desired output resolution significantly higher than the measurement. The spatial resolutions in blocks H1-H3 and G1-G3 can be adjusted to meet varying task requirements. The final output undergoes a 1 × 1 point convolution with a stride of 1 and a designated activation function for further refinement.

Notably, separate models are trained for instantaneous velocities (i.e. u and v) and their fluctuating components (i.e. u' and v', then yielding quasi-2D TKE) of velocity, each requiring different final activation functions. Leaky ReLU is used for velocity fields to accommodate negative values, while standard ReLU is employed for turbulent kinetic energy (TKE) reconstruction, as TKE values are inherently positive. Skip connections (denoted as SC), as used in traditional U-Net models, are integrated between blocks of the same spatial resolution to preserve spatial features during compression and facilitate smooth gradient backpropagation, mitigating issues related to vanishing or exploding gradients.

3.2. Physics-informed training strategy

Traditional deep neural networks often require large datasets of high quality for training and may lack interpretability. Physics-Informed Neural Networks (PINNs) offer a solution to these challenges by incorporating physical constraints into the training process (e.g., customized loss functions with physics-based terms), which can improve physical consistency and training stability by restricting the hypothesis space (Karniadakis et al., 2021; Cai et al., 2021). Depending on the form/weighting of the constraints and implementation, this may reduce the number of epochs needed to reach a target accuracy. PINNs have been successful in various areas of hydro-environmental research, such as fluvial turbulence (Zhang et al., 2022a), hydro-thermal modelling (Wang et al., 2023), and weather forecasting (Kashinath et al., 2021).

In this study, two different physics-informed loss functions were constructed for the reconstruction of velocity fields (i.e., u and v) and TKE fields, respectively. The basic non-physical term in both loss functions is the mean-absolute-percentage error (MAPE), which reflects the general magnitude of relative errors by comparing all pixels in the output and target (measured) fields. Unlike the more commonly used mean absolute error (MAE) or root-mean-square error (RMSE), use of MAPE can mitigate the scaling effect when applying the trained model to problems with spatiotemporal scales different from the training data. This principle of utilizing relative errors was extended to other physics-

Table 1
Parameter range of experimental observations.

Parameter	Test Group 1	Test Group 2	Test Group 3	Test Group 4	Test Group 5
Flow depth (m)	0.135	0.150	0.165	0.180	0.195
Mean flow velocity (m/s)	0.21	0.21	0.22	0.22	0.22
Re × 10^4	1.77	1.85	2.00	2.08	2.21
Fr	0.18	0.17	0.17	0.16	0.15
Duration of measurement (s)	10	10	10	10	10
Data volume per test (frames)	1000	1000	1000	1000	1000
Repetitions	2	2	2	2	2

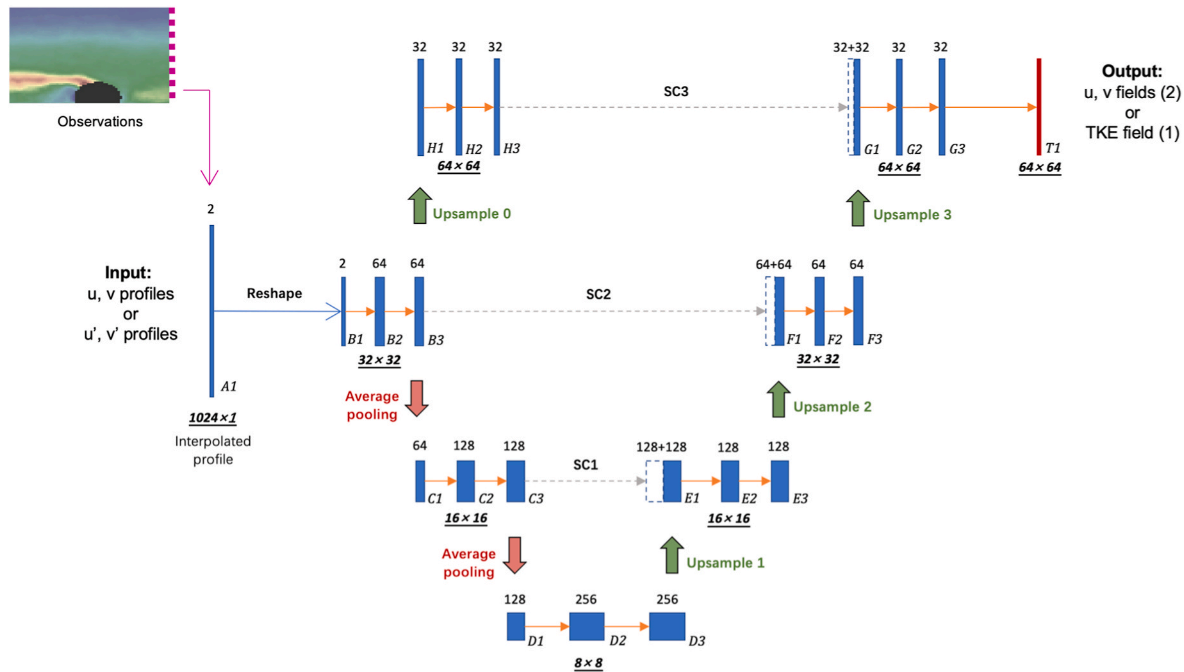

Fig. 3. Architecture of the proposed model. Solid arrows are convolution operations followed by batch normalization, activation and dropout; dashed arrows are skip connections. Numbers above tensors are the channel numbers; underlined numbers are the field resolution. The velocity-trained model takes u and v profile as input and then ouputs u and v fields (2 layers); the TKE-trained model takes u' and v' profile as input and then ouputs a TKE field (1 layer).

related terms when designing the loss functions.

The loss function for reconstructing horizontal and vertical velocity fields (i.e., u and v) is defined as follows:

$$\mathscr{L}_{uv}(\xi_o,\xi_t)=\gamma_1\mathscr{L}_{mape}(\xi_o,\xi_t)+\gamma_2\mathscr{L}_{mape_r}(\xi_o,\xi_t)+\gamma_3\mathscr{L}_{div}(\xi_o,\xi_t) \tag{1}$$

where ξ_o and ξ_t represent the output and target tensors, respectively, and γ_1, γ_2, γ_3 are constant coefficients (weights) that require fine-tuning. Besides the standard MAPE term involving $\mathscr{L}_{mape}$, two extra terms are added to improve training efficiency and ensure physical consistency. Specifically, $\mathscr{L}_{mape_r}$ calculates the MAPE for a select percentage of pixels in the output tensor that exhibits the highest mean relative error when compared to the corresponding pixels in the target tensor. In this study, this percentage was fine-tuned and set at 20 %. The inclusion of $\mathscr{L}_{mape_r}$ enhances model fidelity by focusing on pixels where large residuals are present but might otherwise be diluted by the bulk of the data. This approach is particularly useful for fields that lack a distinct and consistent spatial pattern, such as vertical velocity fields. Additionally, $\mathscr{L}_{div}$ computes the MAPE of pixel-wise flow-field divergence, promoting that the reconstructed velocity fields are physically plausible, which in another way aids in accelerating the training convergence by excluding unrealistic results. The concept of using divergence as a physical constraint has been supported by many other studies using deep neural networks for flow-field predictions, such as Zhang et al. (2022a). Eqs (2)–(5) provide detailed definitions of each term in the combined loss function:

$$\mathscr{L}_{mape}(\xi_o,\xi_t)=\frac{1}{Bt\times Ch\times H\times W}\sum_{i=1}^{Bt\times Ch\times H\times W}\left|\frac{y_i-\widehat{y}_i}{y_i}\right| \tag{2}$$

$$\mathscr{L}_{mape_r}(\xi_o,\xi_t)=\frac{1}{0.2(Bt\times Ch\times H\times W)}\sum_{i=1}^{0.2(Bt\times Ch\times H\times W)}\left|\frac{y_i-\widehat{y}_i}{y_i}\right|_{\text{top 20\% per u+v}} \tag{3}$$

$$\mathscr{L}_{div}(\xi_o,\xi_t)=\frac{1}{Bt\times Ch\times H\times W}\sum_{i=1}^{Bt\times Ch\times H\times W}\left|\frac{\mathrm{div}_i-\widehat{\mathrm{div}}_i}{\mathrm{div}_i}\right| \tag{4}$$

$$\mathrm{div}=\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y} \tag{5}$$

where Bt, Ch, H and W represent the batch number, channel number, field height and field width of the training dataset, respectively; y_i and $\widehat{y}_i$ are pixel-wise values in the target and output tensor, respectively; div_i and $\widehat{\mathrm{div}}_i$ are the calculated flow-field divergence at specific pixels in the target and output tensor, respectively. It should be noted that the flow divergence is treated as the two-dimensional partial differential (Eq. (5)), as transverse velocity is not taken into consideration in this study.

A "quasi-instantaneous TKE" field is reconstructed separately using measured velocity fluctuations (i.e., u' and v') at the upstream boundary, a different loss function is required for model training, which is defined as follows:

$$\mathscr{L}_{TKE}(\theta_o,\theta_t)=\lambda_1\mathscr{L}_{mape}(\theta_o,\theta_t)+\lambda_2\mathscr{L}_{mape_r}(\theta_o,\theta_t)+\lambda_3\mathscr{L}_{k}(\theta_o,\theta_t)+\lambda_4\mathscr{L}_{dist}(\theta_o,\theta_t) \tag{6}$$

where θ_o and θ_t represent the output and target tensors, respectively, and λ_1, λ_2, λ_3, λ_4 are constant coefficients (weights) that require fine-tuning in a similar manner to $\mathscr{L}_{mape}$ and $\mathscr{L}_{mape_r}$ (Equation (1)). Two unique terms, $\mathscr{L}_k$ and $\mathscr{L}_{dist}$, are incorporated to help capture high-TKE zones, which are often sparsely distributed and time-dependent. Specifically, $\mathscr{L}_k$ calculates the MAPE of pixels in a kernel with given size, and the kernel location is determined by scanning the output or target field while maximizing the mean TKE of the kernel-covered pixels. The kernel size is also a parameter that requires fine-tuning, after which the size of 25 × 25 was selected for the final training. $\mathscr{L}_{dist}$ computes the mean distance between the kernels for the output and target field divided by the kernel width. The combination of $\mathscr{L}_k$ and $\mathscr{L}_{dist}$ facilitates the model generation of TKE fields that are physically meaningful. This constraint is crucial given the spatiotemporal distribution of TKE values is closely related to flow energy distribution and bed mobility. The detailed definitions of the terms in Eq. (6) are as below:

Table 2
Model configurations.

Layers	Operation	Field size	Number of output channels	Note
A1 → B1	Reshape	1024 × 1	2	Preprocessing
B1 → B3	$\begin{pmatrix} 3\times3 \text{ Conv stride 1} \\ \text{Batch normalization} \\ \text{ReLU} \end{pmatrix} \times 2$	32 × 32	64	Convolution
B3 → C1	2 × 2 Average Pooling	16 × 16	64	Down-sampling
C1 → C3	$\begin{pmatrix} 3\times3 \text{ Conv stride 1} \\ \text{Batch normalization} \\ \text{ReLU} \end{pmatrix} \times 2$	16 × 16	128	Convolution
B3 → H1	2 × 2 ConvTranspose stride 2	64 × 64	32	Upsampling
H1 → H3	$\begin{pmatrix} 3\times3 \text{ Conv stride 1} \\ \text{Batch normalization} \\ \text{ReLU} \end{pmatrix} \times 2$	64 × 64	32	Convolution
C3 → D1	2 × 2 Average Pooling	8 × 8	128	Down-sampling
D1 → D3	$\begin{pmatrix} 3\times3 \text{ Conv stride 1} \\ \text{Batch normalization} \\ \text{ReLU} \end{pmatrix} \times 2$	8 × 8	256	Convolution
D3 → E1	2 × 2 ConvTranspose stride 2	16 × 16	128	Upsampling
E1 → E3	Skip connection from C3 $\begin{pmatrix} 3\times3 \text{ Conv stride 1} \\ \text{Batch normalization} \\ \text{ReLU} \end{pmatrix} \times 2$	16 × 16	128	Convolution
E3 → F1	2 × 2 ConvTranspose stride 2	32 × 32	64	Upsampling
F1 → F3	Skip connection from B3 $\begin{pmatrix} 3\times3 \text{ Conv stride 1} \\ \text{Batch normalization} \\ \text{ReLU} \end{pmatrix} \times 2$	32 × 32	64	Convolution
F3 → G1	2 × 2 ConvTranspose stride 2	64 × 64	32	Upsampling
G1 → G3	Skip connection from H3 $\begin{pmatrix} 3\times3 \text{ Conv stride 1} \\ \text{Batch normalization} \\ \text{ReLU} \end{pmatrix} \times 2$	64 × 64	32	Convolution
G3 → T1	1 × 1 Conv stride 1 Leaky ReLU / ReLU	64 × 64	2/1	Refinement

$$\mathscr{L}_{mape}(\theta_o, \theta_t) = \frac{1}{Bt \times Ch \times H \times W} \sum_{i=1}^{Bt\times Ch\times H\times W} \left| \frac{y_i - \widehat{y}_i}{y_i} \right| \tag{7}$$

$$\mathscr{L}_{mape_r}(\theta_o, \theta_t) = \frac{1}{0.2(Bt \times Ch \times H \times W)} \sum_{i=1}^{0.2(Bt\times Ch\times H\times W)} \left| \frac{y_i - \widehat{y}_i}{y_i} \right|_{\text{top 20\% per channel}} \tag{8}$$

$$\mathscr{L}_{k}(\theta_o, \theta_t) = \frac{1}{B \times C} \sum_{j=1}^{B\times C} \left(\frac{1}{Kl^2} \sum_{i=1}^{Kl^2} \left| \frac{y_i - \widehat{y}_i}{y_i} \right|_{\text{in kernel}} \right)_j \tag{9}$$

$$\mathscr{L}_{dist}(\theta_o, \theta_t) = \frac{1}{B \times C} \sum_{j=1}^{B\times C} \left(\frac{\sqrt{(I_{max} - \widehat{I}_{max})^2 + (J_{max} - \widehat{J}_{max})^2}}{Kl} \right)_j \tag{10}$$

where Kl is the kernel width (i.e. 25 in this study), I_{max} and J_{max} are the row and column indices of the kernel's centroid in the target field, and $\widehat{I}_{max}$ and $\widehat{J}_{max}$ are the indices in the output field. A schematic illustration of the construction of loss functions is shown in Fig. 4.

Among the loss function components in Eq. (1) and Eq. (6), the divergence penalty corresponds most directly to a conservation-based physical constraint in the measurement plane; the remaining terms encode physically motivated priors that emphasize coherent flow structures and energetically important regions (e.g., high-TKE zones associated with shear layers), which improves the physical plausibility of reconstructions from limited boundary information.

After finalizing the loss functions, the model underwent adaptive training to ensure a consistent and continuous descent of both training and validation losses and to mitigate the risk of overfitting. The entire dataset was partitioned into training, validation, and test sets in a 7:2:1 ratio. The test set was reserved exclusively for evaluating the model performance post-training.

A few regularization techniques were adopted to prevent overfitting and enhance the generalizability of the model. These techniques include a 20 % dropout following each ReLU activation (except in the final output layer) and weight decay with a L2 penalty coefficient of 4×10^{-5}. The constant coefficients (weights) in Eq. (1) and Eq. (6) were fine-tuned to optimize the training process. For the reconstruction of u and v fields based on $\mathscr{L}_{uv}$, the model training was implemented in three sequential steps, each with different combinations of weights and learning rates, as detailed in Table 3. Initially, relatively higher weights were assigned to terms corresponding to significantly deviated pixels (γ_2) and flow-field divergence (γ_3) to enforce a fast generation of physically meaningful data fields. Subsequent phases focused on refining the output fields by emphasizing their overall similarity to the target fields (larger γ_1/γ_2 and γ_1/γ_3). The learning rate also incrementally increased in each step to counterbalance the growing magnitude of $\mathscr{L}_{uv}(\xi_o, \xi_t)$. The model was trained for 300 epochs (iterations) in each step until no significant improvements were observed. This adaptive training strategy balanced the needs for output accuracy with computational efficiency, outperforming models trained with a single set of weight configurations. However, this approach did not yield significant improvements for the reconstruction of TKE fields using $\mathscr{L}_{TKE}(\theta_o,\theta_t)$. Consequently, a fixed set of weights and learning rate was used for TKE field reconstruction, as also outlined in Table 3. The model was trained for 400 epochs until training and validation losses converged and ceased to improve.

The models were developed using Jupyter Notebooks for better interactivity, readability, and shareability in the public domain. Model training and evaluation were conducted using Google Colaboratory, a cloud-based platform that allows users to write and execute Python code in a web-based, interactive environment and offers flexible access to GPU resources. The source code of the models developed in the present study, together with the original datasets, can be accessed using the link provided at the end of the paper.

4. Results and assessment of model performance

The evaluation of the performance of the trained model is organized around three key criteria: the accuracy of reconstructed quasi-instantaneous fields, the fidelity of time-averaging in longer periods using quasi-instantaneous reconstructions, and the robustness of the model when using incomplete or corrupted boundary measurements, which often occurs when collecting data in challenging environments. As the models for velocity and TKE fields are trained independently, their respective discussions are presented in different subsections.

4.1. Reconstruction of velocity fields

The model performance in reconstructing velocity fields is first evaluated by comparing the output horizontal, vertical, and the resultant total velocity fields to their respective target (measured) fields. For the horizontal velocity field, i.e. the dominant velocity component in typical open channel flows, the model aims to accurately capture key features such as flow separation over protruding particles, the spatial characteristics of sheltered wake zones, and representative velocity shear patterns over the flow depth. Vertical velocity fields, on the other hand, generally exhibit significantly smaller magnitudes and more

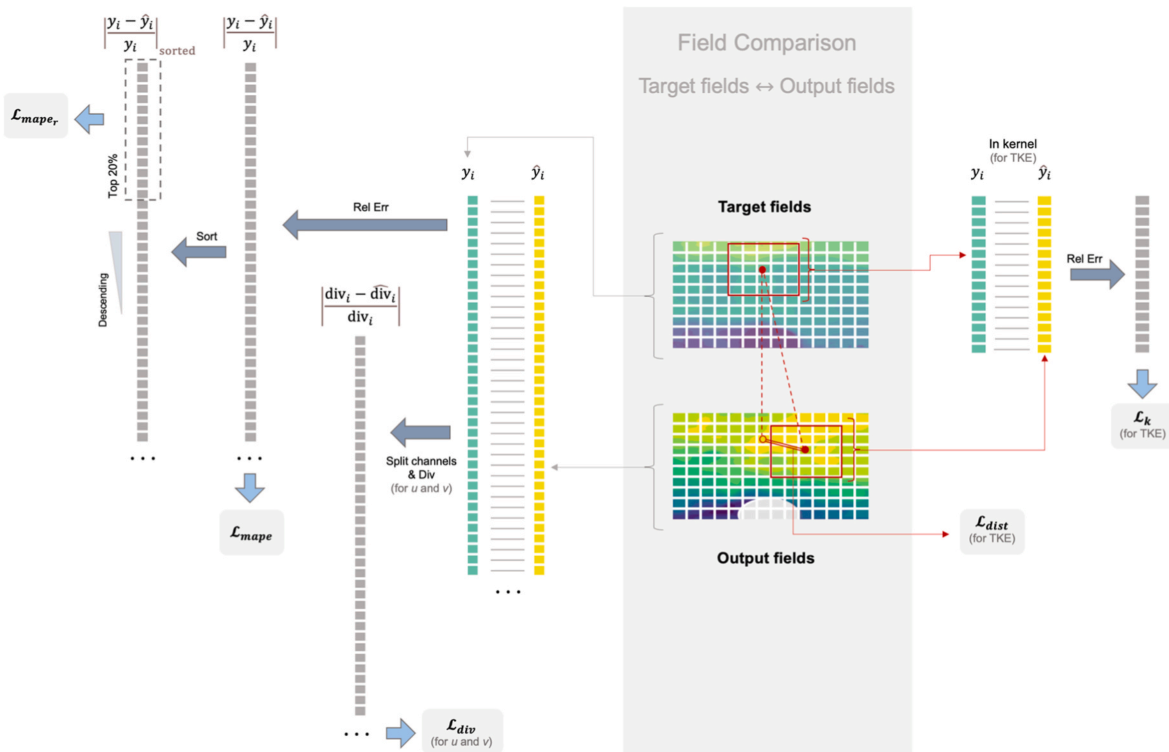

Fig. 4. Schematic illustration of the physics-informed training strategy and the construction of loss function components. Mesh grid size is not to scale.

Table 3
Summary of physics-informed training steps and the selection of training parameters.

$\mathscr{L}_{uv}(\xi_o, \xi_t)$				–	$\mathscr{L}_{TKE}(\theta_o, \theta_t)$	
Weights	Step 1	Step 2	Step 3		Weights	Single-step
γ_1	1.0	2.0	4.0		λ_1	5.0
γ_2	0.3	0.2	0.1		λ_2	1.0
γ_3	0.3	0.2	0.1		λ_3	1.0
–	–	–	–		λ_4	1.0
Learning rate	5×10^{-5}	2×10^{-4}	4×10^{-4}		Learning rate	5×10^{-5}
Epochs	300	300	300		Epochs	400

variability. Consequently, the evaluation of vertical velocity fields focuses more on statistical comparability across the region of interest.

Fig. 5 provides a side-by-side comparison of target and output fields for three distinct quasi-instantaneous scenarios (“snap-shots”). Specifically, Fig. 5a features a fast-velocity zone approaching from the upper region, with a range of vertical velocities including both upward and downward flows. Fig. 5b and c shows more distinct conditions with predominant upward and downward flow patterns, respectively, near the protruded particle. Across all three scenarios, the model demonstrates satisfactory accuracy in capturing both the primary spatial patterns and velocity magnitudes at various locations. Here, “similarity” for quasi-instantaneous fields is assessed primarily by whether the output reproduces the same dominant flow structures, especially the wake footprint, the separated shear-layer signature, and the approach-flow distribution, rather than strict pixel-by-pixel coincidence of turbulent details. The average domain-wide MAPE is less than 3 % for horizontal and total velocity fields, and less than 5 % for vertical velocity fields. These low domain-wide errors indicate that the remaining differences are mainly confined to fine-scale, high-frequency details rather than changes in the dominant structures. Notably, the location, outer boundary, and core region of the wake zone are well-represented in both the horizontal and total velocity output fields. While the reconstructed vertical velocity fields exhibit good spatial fidelity, some small-scale variations appear slightly smoother (i.e., mildly “blurred”) in the output, possibly owing to the model limitations in handling complex spatial features across various spatiotemporal scales. This limitation can possibly be addressed in the future by incorporating the multiresolution algorithm (Liu et al., 2023) commonly used in the areas of computer vision and image processing. In general, the proposed model is proven capable of reconstructing velocity fields with an accuracy (less than 5 % error), which is a complementary tool for rapid reconstruction/monitoring when boundary measurements are available. The three scenarios illustrated in Fig. 5 are representative examples of the general performance of the model; additional details and examples from the test set can be explored with the source code.

The model developed in the present study creates cumulative stacking of quasi-instantaneous velocity fields, enabling rapid derivation of time-averaged velocity fields over any specified averaging window as required. It is also expected that spatial inaccuracies during reconstruction can be improved through time averaging. Here, similarity is interpreted primarily in terms of the agreement in dominant zones (upper fast-velocity band, shear-layer onset, and wake footprint) and their magnitudes, with time-averaging providing a stricter check on persistent, non-transient features. This feature is particularly advantageous for intelligent surrogate systems aiming to provide customized flow monitoring and analysis.

To further verify the trained model and validate the fidelity of these time-averaged fields, we employ an additional time series of boundary measurements of u and v that are not included in the original dataset, avoiding the issue of data leakage. New data consist of consecutive boundary measurements taken over 10 s (i.e., 1000 frames) under the same flow conditions listed in Table 1, with a flow depth of 0.15 m. Time-averaged resultant velocity fields are generated using output velocity component fields, cumulatively averaged over frame counts of 5 (0.05s), 50 (0.5s), 500 (5s), and 1000 (10s). The results are presented in Fig. 6. Due to observable inaccuracies in the quasi-instantaneous fields,

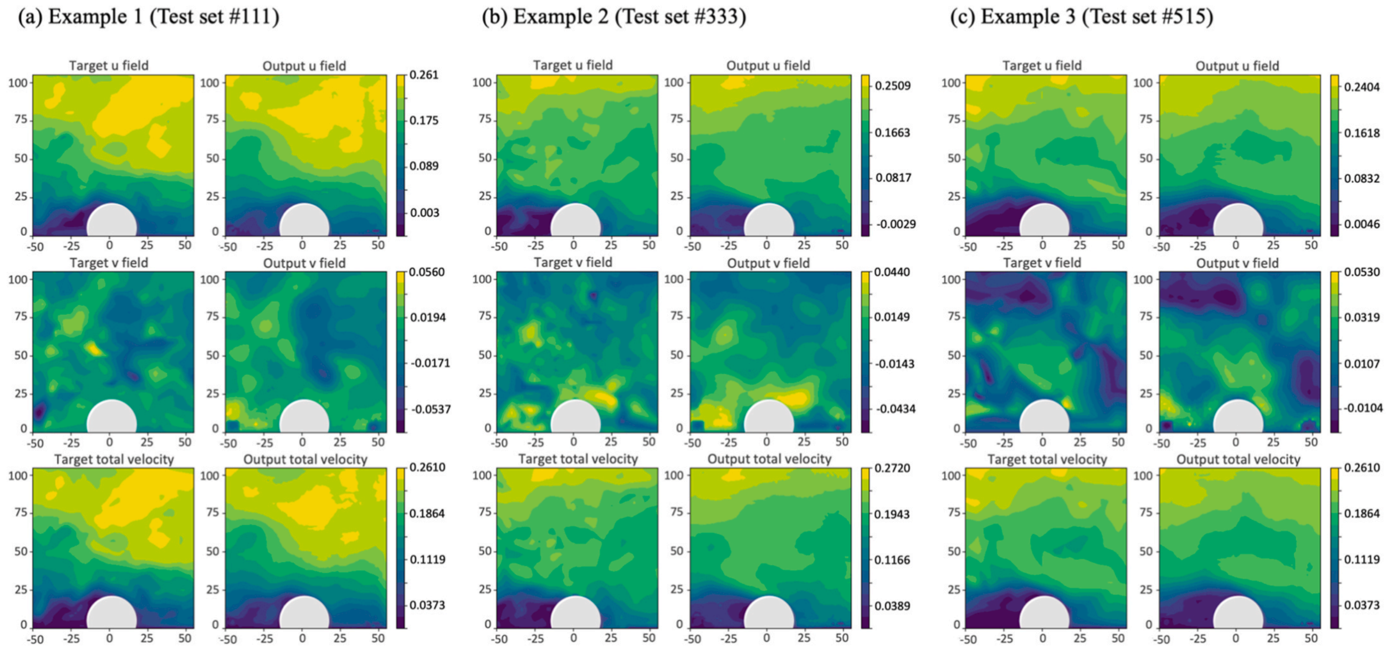

Fig. 5. Examples of reconstructed snapshots of horizontal (u), vertical (v) and total velocity fields using boundary measurements and comparisons with target fields. Group (a) represents typical scenarios with fast horizontal velocity in the upper region; Group (b) represents typical scenarios with significant upward flow around the bed particle; Group (c) represents typical scenarios with significant downward flow patterns. The horizontal and vertical axes are in millimeters, and the color bar units are m/s. Flow is from right to left. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

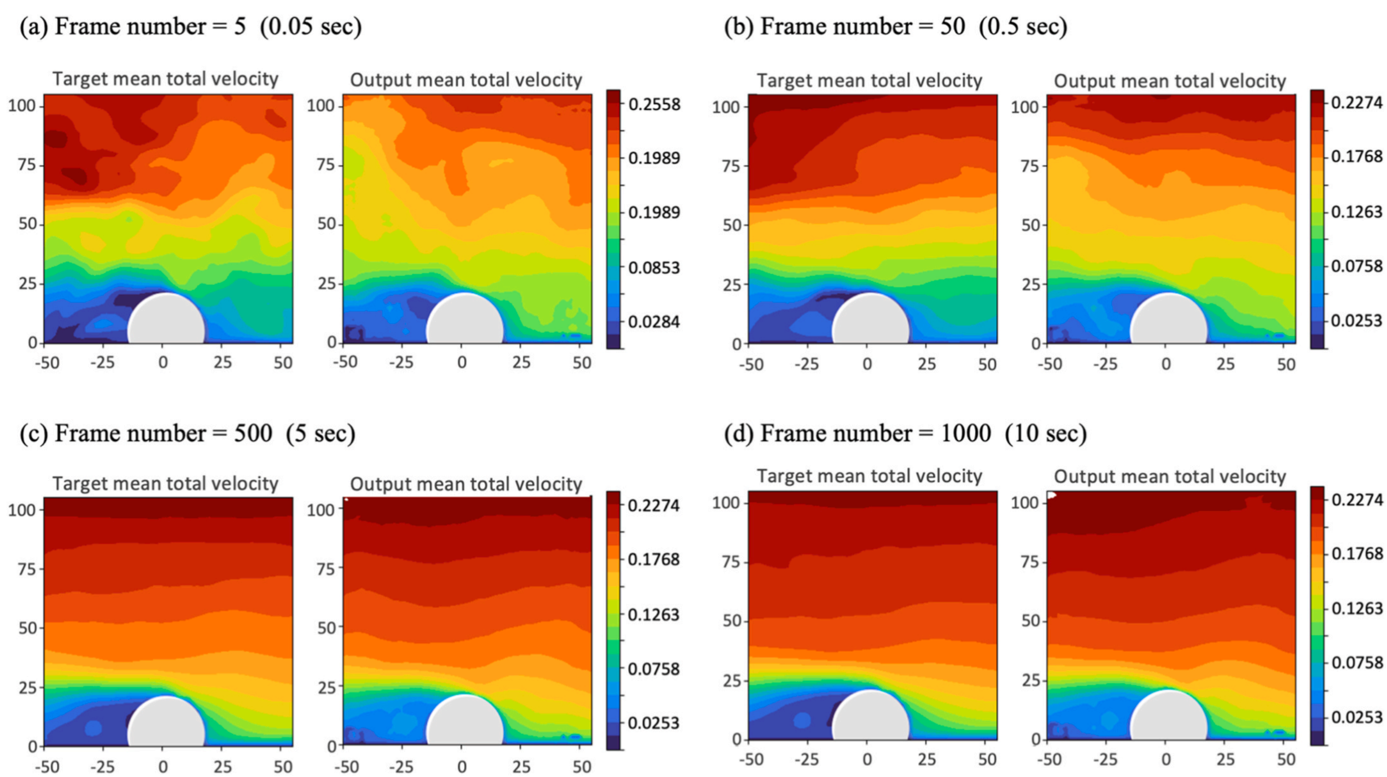

Fig. 6. Stabilization of mean velocity field with output fields augmented with time: (a) frame number = 5 (i.e. 0.05 s in total); (b) frame number = 50 (i.e. 0.5 s in total); (c) frame number = 500 (i.e. 5 s in total); (d) frame number = 1000 (i.e. 10 s in total). The horizontal and vertical axes are in millimeters, and the color bar units are m/s. Flow is from right to left. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

time-averaging over a small number of frames (see Fig. 6a) results in significant distortions in the spatial velocity distribution, particularly in the upper fast-velocity zone and the upstream section near the bed. However, the location and strength of the separated shear layer and wake zone generally align with the target field, indicating a level of accuracy comparable to that of quasi-instantaneous reconstruction. As the number of frames used for time-averaging increases (see Fig. 6b-c), the spatial accuracy improves rapidly, and both the target and reconstructed mean velocity fields stabilize. Beyond approximately 500 frames (5 s), further averaging produces only marginal changes in the mean patterns. A small localized deviation may still be observed immediately upstream of the protruded particle, which is expected in a region of strong velocity gradients and intermittent near-bed/turbulent fluctuations where time-averaged estimates are most sensitive to sampling variability and measurement uncertainty. Importantly, this localized difference is confined to a limited area and does not change the overall flow organization (approach flow distribution and downstream wake structure) that underpins the subsequent analyses; therefore, it does not affect the reliability of the model for the study's objectives. Additionally, the velocity shear near the bed is also reconstructed with satisfactory accuracy. The results support future efforts to relate general bed mobility to sparse flow measurements using an enhanced version of the existing model. While the velocity in the wake zone is slightly overestimated, likely due to the overestimation of vertical velocities in that region, this discrepancy does not significantly compromise the model's overall performance, especially when applied to similar scenarios involving densely packed bed elements.

A comprehensive analysis of model performance also necessitates assessing the temporal velocity evolution of at different locations. This approach helps examine the model's point-wise reliability over time, which may not be apparent in quasi-instantaneous and averaged fields. For this purpose, we used the same set of 1000 consecutive frames mentioned earlier and extracted time-series data for both horizontal and vertical velocities at four locations above the target particle and positioned 25 mm, 50 mm, 75 mm, and 100 mm above the mean bed elevation (P1-P4). Time series of velocities from these points are shown in Fig. 7. The model generally captures the temporal evolution of horizontal velocity well, and the output signals have almost the same upper and lower data limits and variation trends as the target series. This point-wise temporal agreement provides direct evidence of similarity between target and output that is less sensitive to visual ambiguity in instantaneous contour plots. While some transient fluctuations are not perfectly replicated and slight phase lags are observable, these discrepancies are likely due to the limitation of the model in capturing and learning flow patterns with very small temporal resolution (e.g., less than 5 mm in length). Nonetheless, the overall performance is satisfactory. However, major differences between the target and output time series are observed in the vertical velocity. The output signals show considerably narrower oscillation ranges compared to the measured data. This limitation may stem from the current model's limitation in effectively learning and representing fields with highly randomized and dispersed spatial patterns that lack a dominant, recognizable structure, as is the case for the vertical velocity fields in Fig. 5. More efforts should be made in the future to address this limitation. Fortunately, these discrepancies in the vertical velocity fields do not substantially impact the fidelity of the resultant velocity time series, given that the magnitudes of vertical velocities are typically much smaller than horizontal velocities.

Spectral analysis was also performed to inspect the consistency between target and output fields. Because spectra summarize the distribution of turbulent energy across frequencies, the spectral agreement provides an additional demonstration that the output reproduces the essential instantaneous dynamics even if some small-scale spatial details appear smoother in snapshot comparisons. The time series of total velocities at the four points (P1-P4) in Fig. 7 were extracted to implement Fast Fourier Transformation analysis. The results, as shown in Fig. 8, show very good agreements between the target and output fields, all of which follow the −5/3 slope of Kolmogorov's law, indicating that homogenous turbulence in the inertial subrange was present in the measurements and then reconstructed accurately by the data-driven model.

4.2. Reconstruction of fluctuating components and TKE fields

The model performance in reconstructing fluctuating components and turbulent kinetic energy (TKE) fields is evaluated and discussed in this subsection. In this study, we define a parameter which (by analogy with TKE) we have called ‘quasi-instantaneous TKE’ in 2D flow as below:

$$TKE_{2D} = \frac{1}{2}\left(u'^2 + v'^2\right) \tag{11}$$

where u' and v' represent the instantaneous horizontal and vertical flow velocity fluctuating components, respectively (by subtracting mean values from a 10-s window to match the measurement period from the laboratory experiments). The absence of transverse velocity is unlikely to substantially impact the model's output considering the nature of experimental data in this study. However, future work should aim to address this limitation by incorporating the three-dimensional flow-field, which would be achievable through minor modifications to the model's channel numbers in each block (see Fig. 3) and the input data structure.

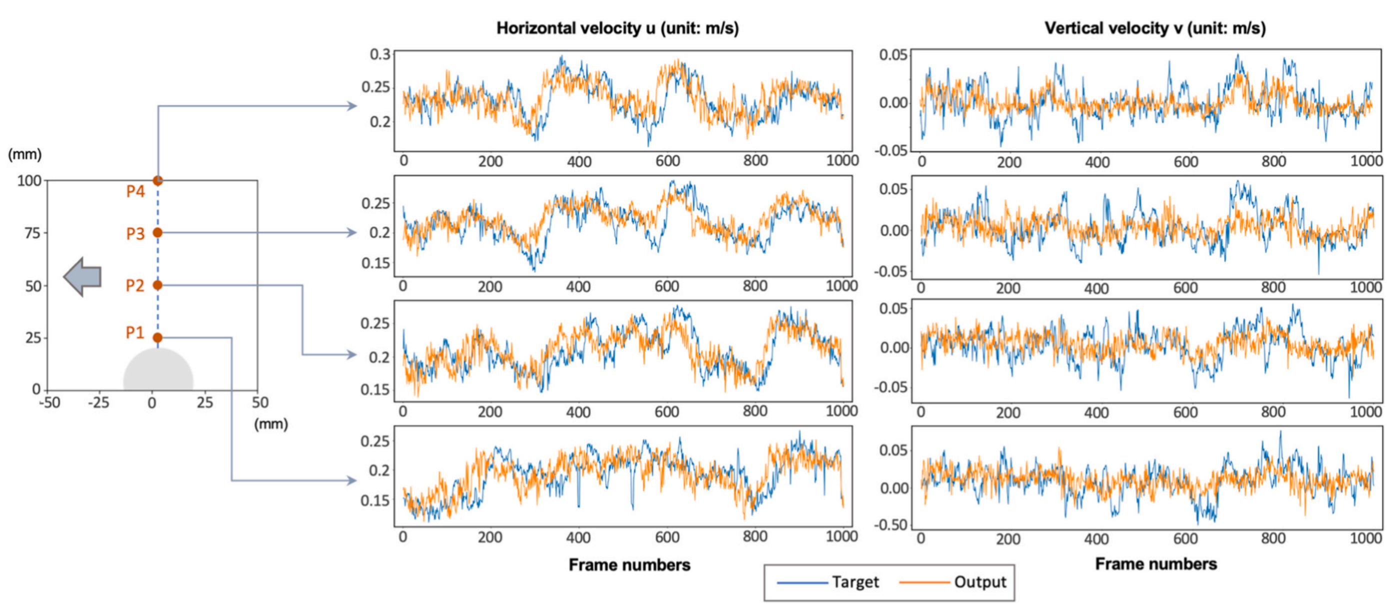

Fig. 7. Comparison of the temporal evolution of target and output velocity components (u and v) at different depths above the particle using 1000 consecutive measurements.

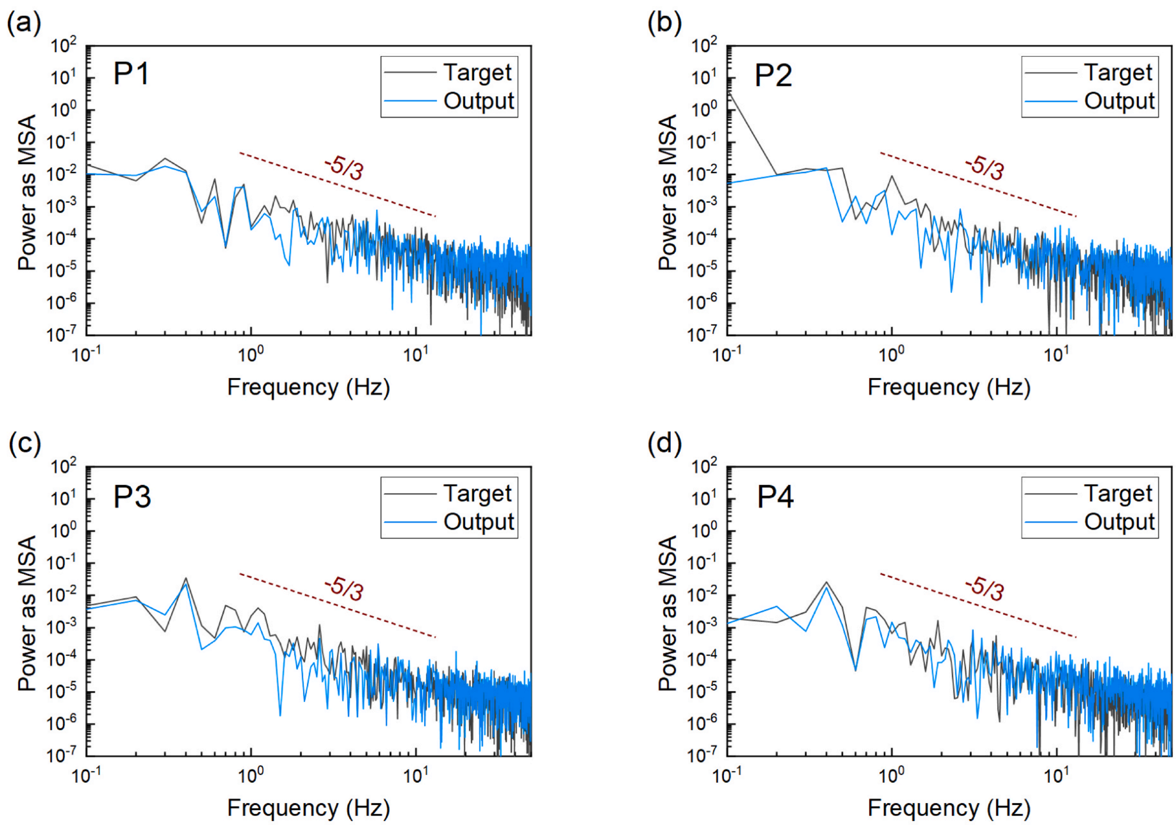

Fig. 8. Power spectra of resultant velocities obtained at four points above the target particle. Note that power is represented by the mean square amplitude (MSA), and the unit is $m^2/s^2/Hz$.

The choice of TKE (i.e. the "quasi-instantaneous TKE" defined above), instead of other second-order flow parameters, as a focus for model design and development is motivated by several factors. Firstly, TKE has direct implications for sediment transport, especially in the near-bed regions of water bodies, and is widely used as a metric for assessing flow-field strength. Secondly, TKE avoids representing multiple directional components using different individual formulas, especially for 3D flow fields, and thus involves less complication in model design. Furthermore, TKE fields inherently contain only positive values, which streamlines the model training process by only using ReLU activation functions across different convolution blocks (as detailed in Table 2), while also simplifying the learning of field patterns.

The model's accuracy in reconstructing quasi-instantaneous TKE fields is shown in Fig. 9, which presents six distinct scenarios: two

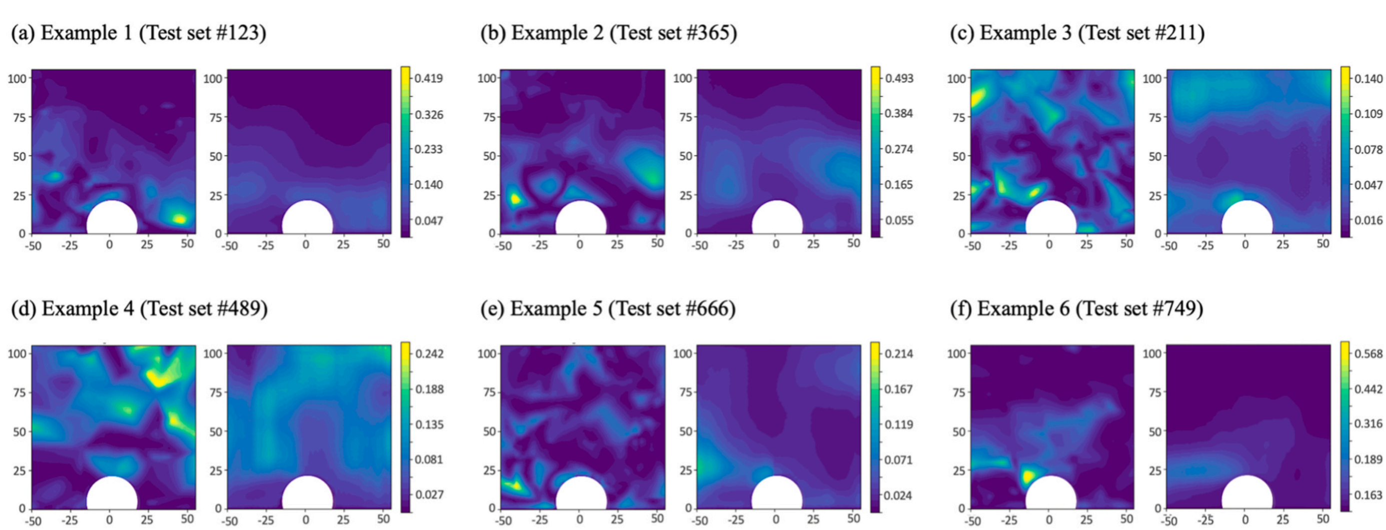

Fig. 9. Examples of quasi-instantaneous TKE fields reconstructed using measured u' and v' at the upstream boundary at single sampling times. Group (a) and (b) represent scenarios with multiple randomly scattered high-TKE nuclei; Group (c) and (d) represent scenarios with extensively distributed high TKE regions in the domain; Group (e) and (f) represent scenarios with high-TKE values around the separated shear layer. The horizontal and vertical axes are in millimeters, and the color bar units are 10^{-2} m^2/s^2. Flow is from right to left. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

featuring scattered high-TKE nuclei near the bed (panels a and b), two with extensively distributed high-TKE regions (panels c and d), and two highlighting high-TKE values around the separated shear layer (panels e and f). Similar to the velocity fields shown in Fig. 5, the examples in Fig. 9 represent only a small subset of the test set, and additional details can be found in the source code. A unique aspect of the model's performance to reconstruct the TKE fields is the ability to maintain domain-wide fidelity while slightly compromising spatial resolution and accuracy in localized small high-TKE areas. For quasi-instantaneous TKE, "similarity" is most appropriately judged by whether the output preserves the dominant energetic organization (where high-TKE regions occur and how they relate to the particle, separation point, and shear layer), rather than reproducing every intermittent peak pixel-by-pixel. As shown in Fig. 9a and b, while the model blurs the scattered high-TKE nuclei near the target particle, their overall spatial footprint and placement relative to the protruded particle remain consistent with the target, the overall spatial characteristics are still effectively captured. Similar results are also observed in Fig. 9c and d, where the model approximates domain-wide TKE distributions in a manner similar to down-sampling through pooling operations. In Fig. 9e and f, the model, drawing experience in learning extensive target samples, selectively omits certain details outside the separated shear layer. Importantly, this does not shift the location or extent of the dominant shear-associated high-TKE region, which is the key physically meaningful feature discussed later in the time-averaged results. This feature, although somewhat arbitrary, enhances the generation of time-averaged TKE conditions, which will be discussed later. This performance is attributed to the design of the integrated loss function, which employs a kernel to scan high-TKE sub-regions and aims to minimize the distance between kernels in the target and output fields. Sensitivity analysis was also conducted to optimize the kernel size, revealing that optimal performance is achieved when the kernel width is approximately one-third of the domain size; deviating from this size in either direction compromises field accuracy.

It is also noteworthy that a separate trial was conducted to rectify the "blurring" issue and to explore capturing subdomains with extreme TKE values more accurately. This test involved modifying the training loss functions to include the MAPE for a select percentage (e.g., 2 %) of the highest TKE pixels across the domain. However, results showed that such an approach will, as opposed to our initial expectation, significantly destabilize the model's converging during training and therefore compromise the overall accuracy of TKE field reconstruction. This result was largely attributed to the presence of severely overfitted singular pixel points with extremely high TKE values, which misled the model during the process of parameter optimization. As a result, this percentage was set to 20 % in the formal training, the same value as for velocity fields. Care should therefore be taken to ensure that overfitting does not occur when developing a convolutional autoencoder-based regression model for physically meaningful fields.

To assess whether the blurring issue persists in time-averaging with more quasi-instantaneous TKE fields accumulated, we conducted an analysis similar to that presented in Figs. 6 and 7 for velocity fields. The fidelity of these time-averaged fields is examined using a similar extra test set including 1000 consecutive boundary measurements of horizontal and vertical velocity fluctuation components (i.e. u' and v'). Fig. 10 shows the TKE fields time-averaged over longer periods, with frame numbers of 5 (i.e. 0.05s), 50 (i.e. 0.5s), 500 (i.e. 5s), and 1000 (i.e. 10s), respectively. Results show that an insufficient number of frames (as seen in Fig. 10a) still suffer a similar blurring issue, leading to a shift in large-TKE patterns and the model doesn't capture some regions of localized large TKE values. However, the key patterns of interest, such as upper large-TKE regions and those at the point of flow separation, are largely preserved. As the number of frames increases, the blurring diminishes, and the time-averaged field begins to emphasize more stable, non-transient TKE distributions. As shown in Fig. 10b-c, the separate shear layer becomes stable and predominant after 500 frames, and the output field shows good accuracy in the location, extent and core magnitude of this region, which is roughly situated above and

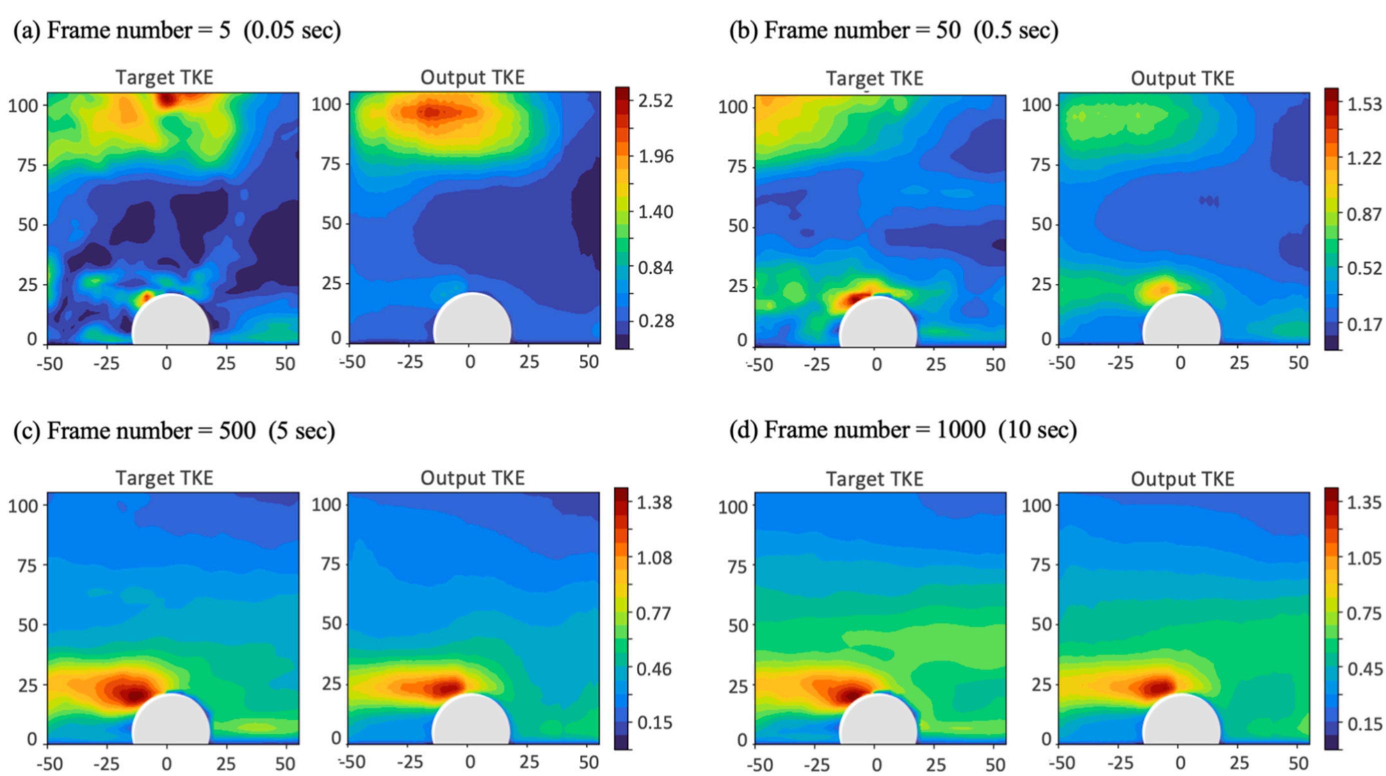

Fig. 10. Stabilization of TKE field with output fields augmented with time: (a) frame number = 5 (i.e. 0.05 s in total); (b) frame number = 50 (i.e. 0.5 s in total); (c) frame number = 500 (i.e. 5 s in total); (d) frame number = 1000 (i.e. 10 s in total). The horizontal and vertical axes are in millimeters, and the color bar units are 10^{-3} m^2/s^2. Flow is from right to left. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

downstream of the protruded particle. The general vertical pattern along the depth in the approach section is also well-recovered, which is consistent with the results in Fig. 6 for resultant velocity fields. The field pattern shown in Fig. 10d correspond to the standard definition of TKE, which necessitates time-averaging over a specific time window until the field converges, showing the general distribution of kinetic energy. Combining Figs. 6 and 10 suggests that the model has a more stable performance in time-averaging over longer periods than solely reconstructing quasi-instantaneous fields (i.e. over 0.01s), provided a sufficient length of observation window is used.

Finally, the temporal evolution of the model-outputted TKE at key locations of interest is analyzed using the same set of 1000 consecutive boundary measurements (10s) as above. Here, we slightly adjusted the vertical profile for data extraction, with the lowest point positioned at the approximate centroid of the high-TKE core. The four points of data extraction were located 25 mm, 45 mm, 75 mm and 100 mm above the mean equivalent rough-bed elevation, respectively. Fig. 11 illustrates the locations of these points and compares the extracted TKE time series with their corresponding target values, all of which are quasi-instantaneous and time-averaged over 0.01s. As expected, the separate shear layer exhibited significantly greater TKE values and oscillation ranges compared to the upper regions and showed dense and prominent signal peaks. At the elevation of 25 mm (i.e. the lowest point), an apparent mismatch between target and output signals is observed showing time lags and excessive suppression of some peaks. This difference explains the lack of high-TKE cores in certain cases. Despite this limitation, the overall trend in the data was adequately captured by the model. These discrepancies diminish at higher elevations (45 mm, 75 mm, and 100 mm), where the model demonstrates a satisfactory ability to reconstruct both the general temporal trends and major peaks in TKE. Additionally, minor peaks arising from transient flow variations, as well as potential outliers (spikes) in the original measurements, are appropriately filtered out. This approach serves the dual purpose of preliminary data post-processing and reducing the data cleaning workload in an integrated intelligent flow monitoring system.

Generally, the model performance assessed in this section so far has proven the efficacy of the proposed model architecture and training methodology. The model successfully achieves reliable domain-wide quasi-instantaneous field reconstruction, point-wise time-series prediction, and control in producing accurate time-average fields, applicable to both flow velocity and TKE fields. However, intelligent flow monitoring systems are often deployed in challenging environments with extreme velocities, low temperatures, or significant signal disturbances. Consequently, the boundary measurements used for model input may not be as clean and complete as those used at the training stage. Therefore, we undertake a comprehensive model robustness analysis to evaluate the impact of incomplete or corrupted input data in the subsequent section.

5. Model robustness analysis

5.1. Incomplete boundary measurement

The first robustness test examines the adaptability of the model to scenarios with incomplete boundary measurements, particularly when only a single point measurement is available at a specific depth. This limitation is common when using instrumentation such as acoustic Doppler velocimeters lacking profiling capabilities. Two approaches were devised to transform this point measurement into a format compatible with the model's input requirements, i.e., an interpolated line vector. The two approaches are as follows:

1. Logarithmic Law of the Wall: The first method utilizes the logarithmic law of the wall to derive the vertical profile of horizontal flow velocity in turbulent flow near-bed, based on the point measurement

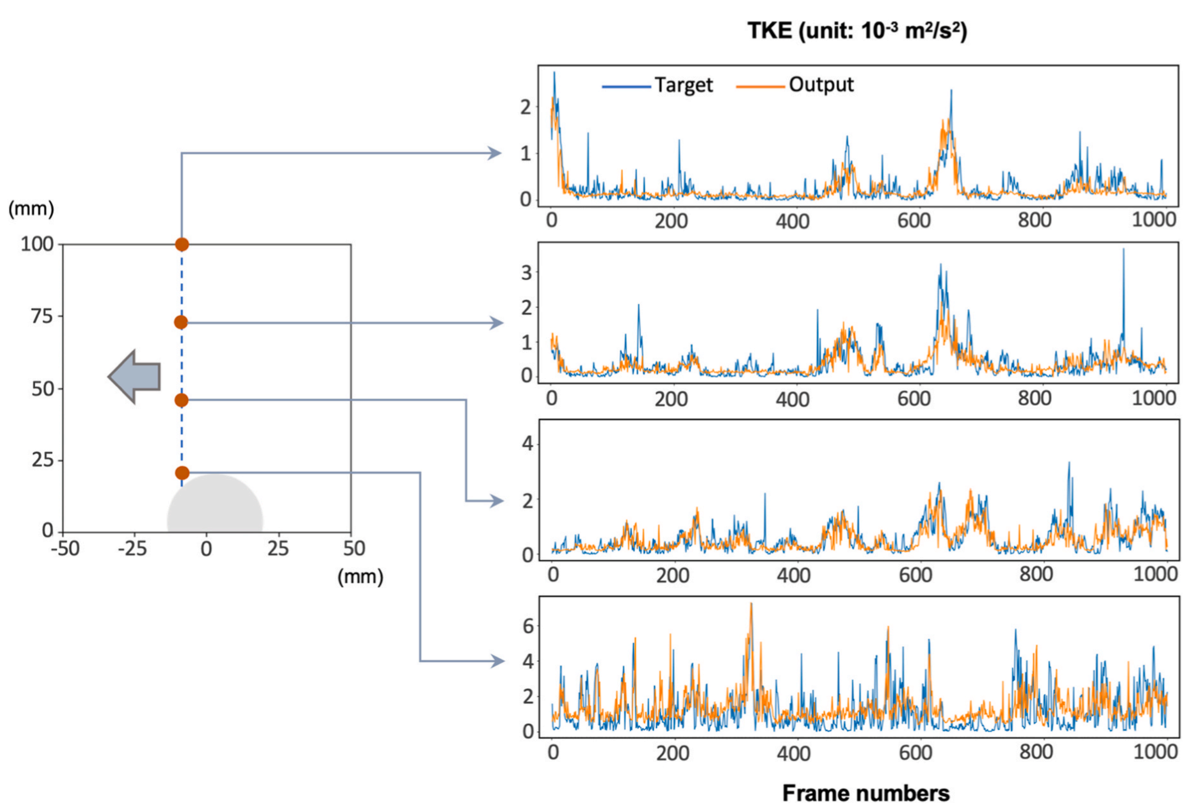

Fig. 11. Comparison of the evolution of target and output TKE values at different depths along the vertical profile passing the high-TKE region of the separated shear layer using 1000 consecutive measurements. Each TKE value is quasi-instantaneous and averaged over instruments' sampling interval (0.01s).

at a known elevation. Concurrently, vertical velocity is assumed to be constant throughout the depth, matching the point measurement.

2. Constant Depth Distribution: The second method is tailored TKE measurements and assumes both horizontal and vertical TKE distributions to be constant throughout the depth.

The logarithmic law of the wall equation used by the first method can be expressed as:

$$u(z) = \frac{u_*}{\kappa} \ln\left(\frac{z}{z_0}\right) \tag{12}$$

where κ is the von Kármán constant taken as 0.4 here, z_0 is the bed roughness length, and u_* is the shear velocity. The determination of z_0 is based on multiple factors (e.g., particle arrangement, flow conditions, etc) that may affect the general bed roughness level, and $z_0 \approx 30/\ d$ is usually adopted as an approximation before further calibration (Schlichting and Gersten, 2016), where d is the bed particle diameter. In this study, three z_0 values were selected, i.e. $z_0 \approx 1.33$ mm (i.e. $30/\ d$), 0.1 mm and 5.0 mm, to generate three different horizontal velocity profiles. For each profile, u_* was back-calculated using the given κ and z_0 values, and the horizontal velocity value at the point of measurement. Here, it is assumed that the vertical distance of the point of measurement above the mean bed level (i.e. z in Eq. (12)) is 60 mm. Additionally, the vertical velocity value is set to the value measured at $z = 60$ mm. Here, 60 mm above the bed equals to 40 % of the total flow depth for the test case (i.e. 150 mm), which is usually where one-point measurement is undertaken to estimate mean-flow information by convention. Through this approach, three different alternative combinations of model input are constructed. Fig. 12 shows the three output resultant velocity fields and the corresponding target and original outputs. Results show that the near-bed velocity field is relatively insensitive to variations in the input data; specifically, no monotonic or consistent trend between the input perturbation level and the near-bed velocity error is observed. All three alternative inputs produced wake zones with extents and magnitudes comparable to the original output and target, despite slight differences in lower-region velocities. However, discrepancies were evident in the upper flow region (e.g., for z > 60 mm), likely due to reduced u' values compared to the point measurement. In general, the model demonstrates robustness in handling various alternative inputs derived from point measurements. No severely distorted regions were observed in the output fields. Nevertheless, caution should still be exercised when applying the model to scenarios where bed conditions significantly distort the velocity profile so that the profile does not fit a logarithmic shape.

The second method for handling TKE fields assumes that both horizontal and vertical velocity fluctuations (i.e. u' and v') are constant throughout the depth. Consequently, the model input consists of two line vectors filled with these constant values. Here, three different elevations were chosen for the point measurements of u' and v', resulting in three sets of alternative model inputs. Fig. 13 displays the locations of these measurements along with the corresponding model outputs. Results show that the turbulent velocities exhibit more variability and severely oscillatory behavior in their vertical profiles. As a result, the reconstructed TKE field is highly sensitive to deviations in the modified input from the actual measurements. Alternative inputs 2 and 3, despite having similar TKE magnitudes but taken at different elevations, produced substantially different output TKE fields, as shown in Fig. 13e and f. The latter significantly overestimates both the extent and magnitude of the TKE within the separated shear layer. Interestingly, although alternative inputs 1 and 2 differ considerably, their outputted TKE fields are quite similar, both outlining a weak shear layer while omitting other spatial patterns, as shown in Fig. 13d and e. In contrast, the original output (see Fig. 13c) apparently has a better performance than all alternative outputs, capturing multiple high-TKE regions of interest effectively. The average domain-wide MAPE for the original output is around 12 %, and those for the three alternative outputs are in the range

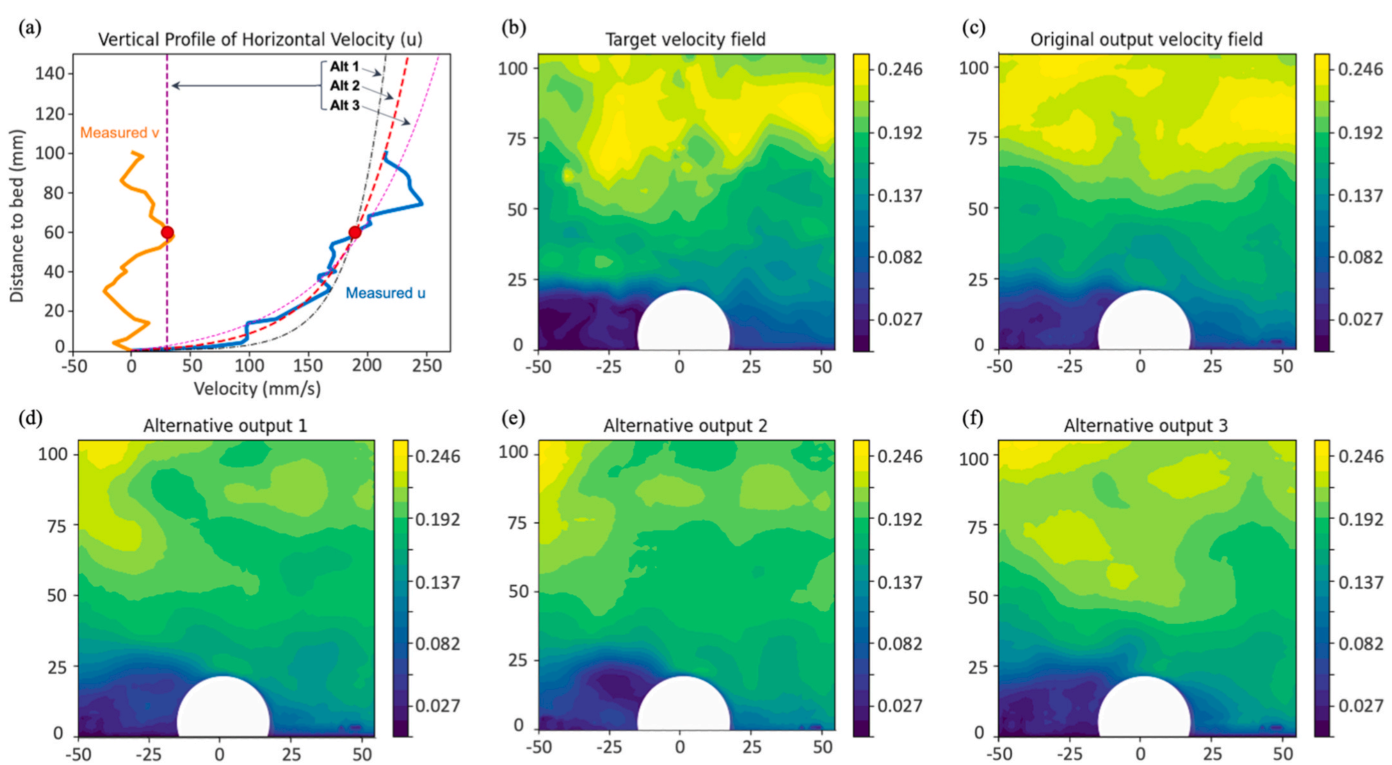

Fig. 12. Model robustness analysis using single-point velocity measurement. Input horizontal velocity (u) fields for alternative outputs are derived by fitting the function of logarithmic law, and the corresponding input vertical velocity (v) fields use the v values measured at the point. Red circles in (a) represent the measured data at the point. In (b)–(f), the horizontal and vertical axes are in millimeters, and the color bar units are m/s. Flow is from right to left. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

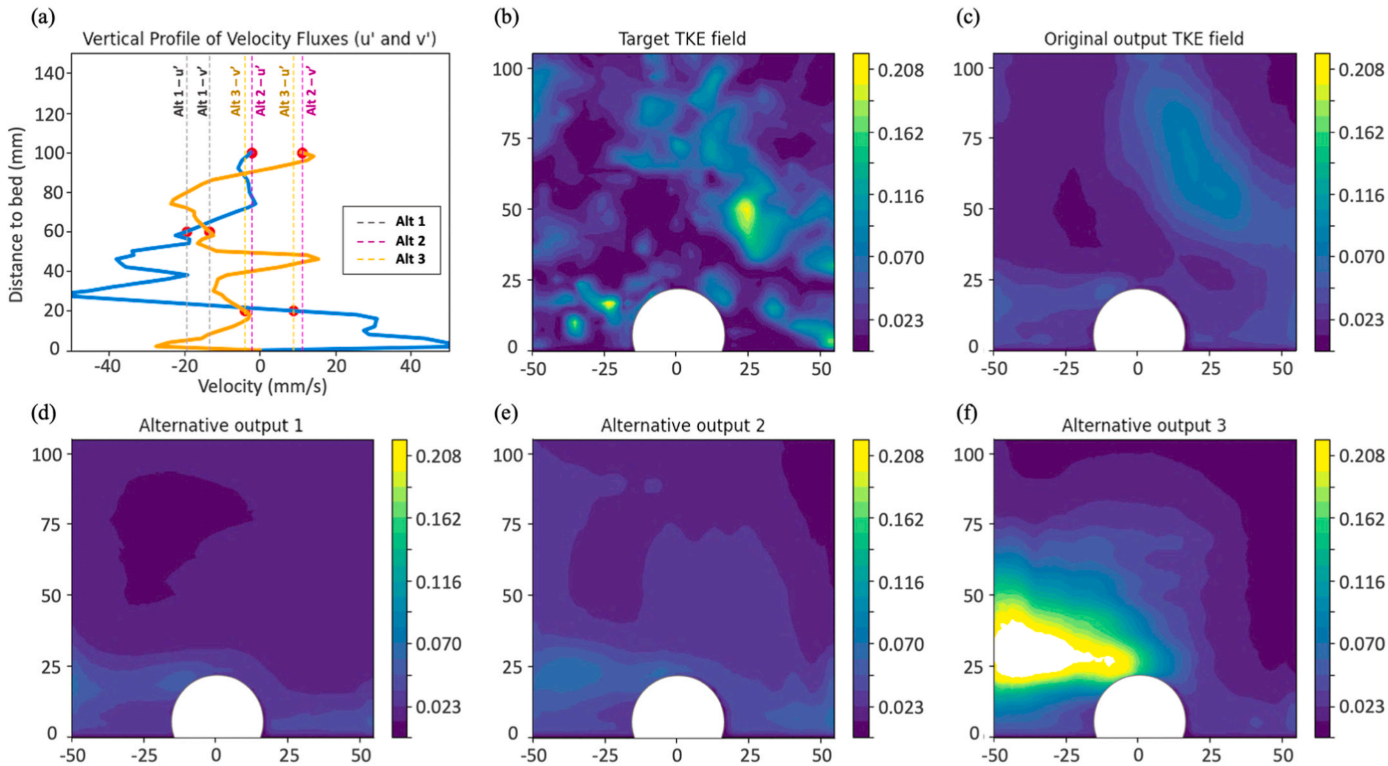

Fig. 13. Model robustness analysis 1 for quasi-instantaneous TKE reconstructed with single-point measurement. The input horizontal and vertical velocity fluctuation components (u' and v') use the values measured at the point extended to the full depth. Red circles in (a) represent the measured data at three alternative points. In (b)–(f), the horizontal and vertical axes are in millimeters, and the color bar units are 10^{-2} m^2/s^2. Flow is from right to left. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

of 18 %–28 %. The analysis for the second method suggests that the trained model may be overly sensitive to and excessively concerned with the separated shear layer, potentially leading to significant distortions in the reconstructed TKE field when slight modifications are made to the input data. This result highlights the need for future work to diversify the training dataset and revise the training strategy to enhance the model's robustness, particularly in the context of TKE field reconstruction.

5.2. Corrupted boundary measurement

The second analysis of model robustness focuses on the effect of input data corruption on the fidelity of output velocity and TKE fields. The previous subsection has explored the model's robustness to incomplete boundary measurements, particularly when only single-point measurement is available. However, the measurements used in those tests were still clean and free from significant unwanted corruption. In real-world applications, data corruption is common and can be attributed to multiple external factors (e.g., acoustic disturbance, suspended microparticles, air bubbles, abnormal vibrations) and internal sources (e.g., internal sensor noises, turbulence-induced fluctuations, algorithmic errors, etc). External factors may cause intermittent noise occurrences with bursting and uncategorizable patterns; in contrast, internal factors tend to show noise with persistent and regular characteristics. Therefore, different methods should be used for artificial noise generation. Given the characteristics of these corruptions, the following steps are used to superimpose artificial noise onto clean boundary measurements.

1. Add Gaussian noise to represent persistent random errors.
2. Introduce a systematic bias to represent a known issue with the measurement setup.
3. Include occasional spikes or outliers to represent external physical disturbances.

The general equation for corrupted data can be expressed as:

$$S_{noisy}(i) = S(i) + N_g(i) + M(i) \times N_s(i) \tag{13}$$

for $i = 1, 2, \ldots,$ n where n is the length of the measurement; besides, S_{noisy} is the corrupted measurement with noise, S is the original clean measurement, N_g is the Gaussian noise, N_s is the spiky noise representing abnormal data points, and M is a binary mask that indicates the presence (1) or absence (0) of a spike at each data point. This mask is generated based on a certain probability of occurrence. Specifically, the Gaussian noise can be mathematically expressed as

$$N_g(i) = f\left(x \middle| \mu, \sigma^2\right) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{14}$$

where $f\left(x \middle| \mu, \sigma^2\right)$ is the probability density function of x (the original data), μ is the mean of the distribution representing any bias from the original mean, and σ is the standard deviation determining the spread or intensity of the Gaussian noise. As a result, it is expected that the noise N_g is sampled from a Gaussian distribution with mean μ and variance σ^2. Adding Gaussian noise to a signal is essentially adding a random value sampled from this Gaussian distribution to each data point in the signal. Unlike Gaussian noise, the spiky noise N_s can be seen as sudden, intense bursts or spikes in the data which are usually caused by the abnormal vibration of instruments, Doppler noise, etc. The detailed equation can be expressed mathematically as:

$$N_s(i) = \begin{cases} R(i) \times m & \text{if } M(i) = 1 \\ 0 & \text{if } M(i) = 0 \end{cases} \tag{15}$$

where $R(i)$ is a random number between 0 and 1, and m is the spike multiplier. Combining the Gaussian and spiky noise using Eqs (13)–(15) ensures the creation of a more realistic scenario for underwater flow measuring devices. This approach allows for a comprehensive evaluation of the model robustness against various types of data corruption. The parameters used for constructing these corrupted inputs for flow velocity and velocity fluctuations are detailed in Tables 4 and 5, respectively.

The first evaluation employs noise parameters outlined in Table 4 to compare alternative total velocity fields with the target and original output fields. The original clean boundary measurements used are the same as those in Fig. 12, which offer a representative vertical velocity profile for flows over rough beds. The detailed results are shown in Fig. 14. The noise types 1–3 represent different bias directions, increasing regular noise magnitudes and more frequent and intense spikes, as illustrated by the fine lines in Fig. 14c–e. The comparison of the three alternative outputs suggests that the model is fairly robust in preserving essential features of the wake zone, even when subjected to noisy data. As expected, the fidelity of the field deteriorates with an increase in the Gaussian standard deviation and the frequency and intensity of spikes. The average domain-wide MAPE for the original output is around 3 %, and those for the three alternative outputs are in the range of 6 %–18 %. Interestingly, the direction of noise bias does not appear to significantly affect the wake and near-bed flow regions. However, the upper flow region is noticeably more sensitive to data corruption, showing severe underestimation or overestimation of velocities for depths above an elevation of 25 mm. Moreover, both positive and negative mean biases can result in excessively large-velocity values in the elevation range of 75 mm–100 mm (see Fig. 14c and e). This phenomenon underscores the need for further scrutiny of model interpretability. Given these findings, caution should be exercised when using the existing velocity-trained model with severely corrupted inputs. Future work could involve additional, comprehensive training that incorporates artificially interfered noisy data or includes built-in signal processing functions in the source code. This investigation would aim to enhance the model's robustness, interpretability and generalizability. Although such extensions are beyond the scope of this paper, these explorations are considered valuable directions for future research.

The second evaluation employs the noise parameters outlined in Table 5, focusing on TKE fields. A new set of boundary velocity measurements is utilized for this analysis (see Fig. 15). The corrupted profiles of fluctuating components are depicted using dashed fine lines in Fig. 15c-e. The results indicate that the TKE-trained model is considerably more sensitive to data corruption than its velocity-trained counterpart. Across all three types of noise, severe overestimations are observed in both the full approach channel and the upper flow region. Interestingly, the patterns of field distortion appear to be consistent, irrespective of the noise composition, no matter the magnitude of regular noises, the direction of bias, or the spikiness of the noise. Given this heightened sensitivity to noise, the TKE-trained model, in its current state, should be deployed only when the input data are noise-free, complete, and of high quality. The current vulnerability to noise in TKE data underscores the need for further refinements to improve the robustness and reliability of the model.

Table 4
Parameters of noises used for model robustness analysis. The noises are superimposed with velocity (u and v) data inputs as artificial data corruption.

Noise parameters for u and v		Noise Type 1		Noise Type 2		Noise Type 3	
		u	v	u	v	u	v
Gaussian noise	Bias	0.2 m/s	0.1 m/s	−0.3 m/s	0.1 m/s	−0.5 m/s	−0.2 m/s
	Standard deviation	0.3 m/s	0.25 m/s	0.35 m/s	0.25 m/s	0.55 m/s	0.55 m/s
Spiky noise	Spike probability	1 %	2 %	5 %	2 %	5 %	5 %
	Spike multiplier	2.0	2.0	2.0	2.0	5.0	5.0

Table 5
Parameters of noises used for model robustness analysis. The noises are superimposed with velocity fluctuations (u' and v') data inputs as artificial data corruption.

Noise parameters for u' and v'		Noise Type 1		Noise Type 2		Noise Type 3	
		u'	v'	u'	v'	u'	v'
Gaussian noise	Bias	0	0	0	0	−0.2 m/s	−0.2 m/s
	Standard deviation	0.1 m/s	0.1 m/s	0.3 m/s	0.3 m/s	0.1 m/s	0.1 m/s
Spiky noise	Spike number	10 %	10 %	0 %	0 %	0 %	0 %
	Spike multiplier	50	50	0	0	0	0

6. Conclusions

This study introduces a convolutional autoencoder-based model designed to reconstruct flow velocity and TKE fields in a 100 mm × 100 mm area solely using quasi-instantaneous measurements of velocity and fluctuation components at the upstream boundary. The model aims to provide a scalable and efficient architecture that can adapt to various input-output configurations while maintaining model accuracy, robustness, and generalizability. The following main conclusions are drawn:

1. Regarding velocity field reconstruction, a detailed comparison between output and target velocity fields demonstrates satisfactory accuracy in capturing spatial patterns, especially for horizontal velocity components. While some deviations were observed in reconstructed quasi-instantaneous fields, time-averaging over an extended duration improved the fidelity of the model, converging efficiently with the target fields. Temporal analysis further showed the model's good ability to predict horizontal velocities at specific locations and maintain realistic power decay, following the −5/3 slope of Kolmogorov's law in spectral analysis:
2. Regarding TKE field reconstruction, the model made a trade-off between the accuracies for high-TKE regions and the overall domain. However, the model showed more consistency in reconstructing fields time-averaged in longer periods compared to quasi-instantaneous fields, particularly when provided with an extended observation window. The results of the temporal analysis also show the capacity of the model to effectively replicate point-wise TKE time series at key locations with appropriate smoothening and despiking.
3. This study proved the efficacy of physics-informed neural networks in quasi-2D flow field reconstruction. Integrating physically motivated regularization terms in training improves reconstruction fidelity and physical plausibility, particularly in preserving key flow structures (e.g., wake and shear-layer features). Although this strategy does not necessarily achieve universally faster wall-clock convergence, it provides a more reliable and physically consistent reconstruction, which is the primary advantage demonstrated in this study.
4. With incomplete boundary measurements, the velocity-trained model demonstrates impressive stability. While the TKE-trained model can capture general features from single-point measurements, the results displayed greater sensitivity to small input disturbances, thus suggesting a need to diversify the training dataset and modify training strategies. With corrupted boundary measurements, the velocity-trained model maintained key flow characteristics (e.g., magnitude and wake zone's shape) despite some noise. In contrast, the TKE-trained model was more vulnerable to noise-induced errors, with consistent overestimations regardless of the type of noise.

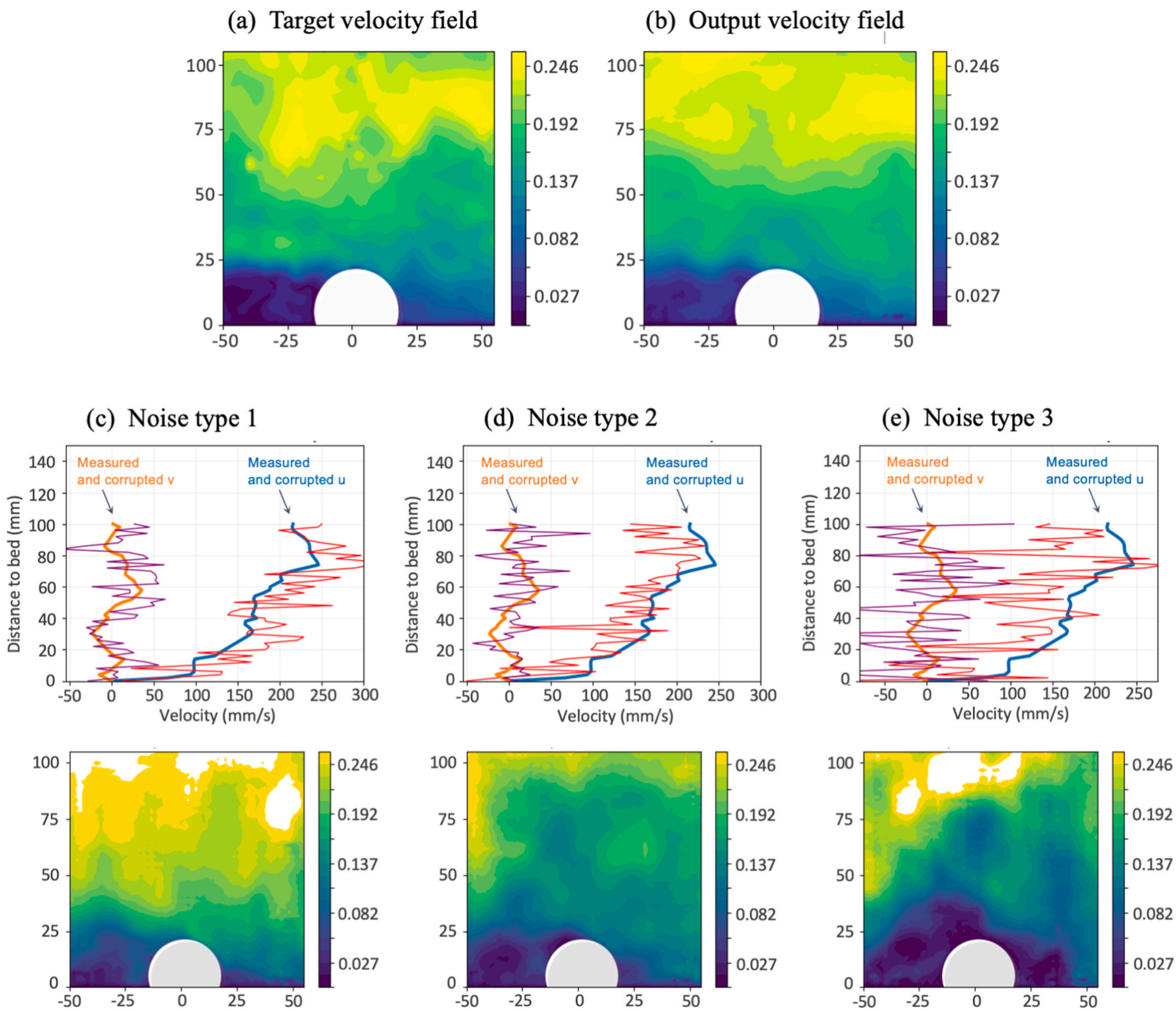

Fig. 14. Model robustness analysis 2 using corrupted velocity measurement. (a) and (b) are target velocity field and original output using clean line measurement, respectively. (c)–(e) show the profiles of corrupted boundary measurements with noises and the corresponding output velocity fields. For the contour plots, the horizontal and vertical axes are in millimeters, and the color bar units are m/s. Flow is from right to left. White regions are off-scale. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

As stated above, there are still a few issues that need to be addressed to fulfil the objective of establishing a practical and intelligent flow-field reconstruction methodology; that is, better model generalizability and interpretability, as well as expanding the scope of applicability.

One of the essential attributes of good machine learning is the ability to generalize beyond the training dataset. From the robustness analyses, it is evident that the trained model maintains acceptable performance with varied alternative inputs and some corrupted data, suggesting model generalizability to some extent. However, challenges may arise when the model is subjected to severely corrupted inputs or entirely different conditions not encountered in training. In practical deployments, the available boundary measurements and flow conditions are often more complex and less idealized than those considered here. Incorporating geomorphic factors in the data array structure can potentially solve this dilemma.

The present study validates the proposed “boundary-to-field” reconstruction framework using a controlled quasi-2D experimental configuration with a single coarse-bed arrangement; therefore, the manuscript does not provide direct evidence of scalability in the sense of transferability across different particle arrangements/bed geometries or extension to fully three-dimensional reconstruction. Extending the current framework to other particle arrangements/geometries would require additional datasets spanning different roughness configurations (and potentially adding geometry descriptors as model inputs, and adjusting the dimensions of intermediate and output tensors) to quantify cross-configuration generalization. Extension to 3D would require volumetric or multi-plane measurement/simulation data and a corresponding model adaptation (e.g., 3D convolutions or multi-plane reconstruction). These extensions are technically feasible but are beyond the scope of the current study and are reserved for future work. Despite these limitations, the robustness tests under incomplete/corrupted boundary inputs demonstrate the workflow's operational stability under realistic data-quality variations, supporting its potential for low-latency monitoring once trained.

Interpretability is also very important to models dealing with systems controlled by complex physics. The inconsistent results observed in

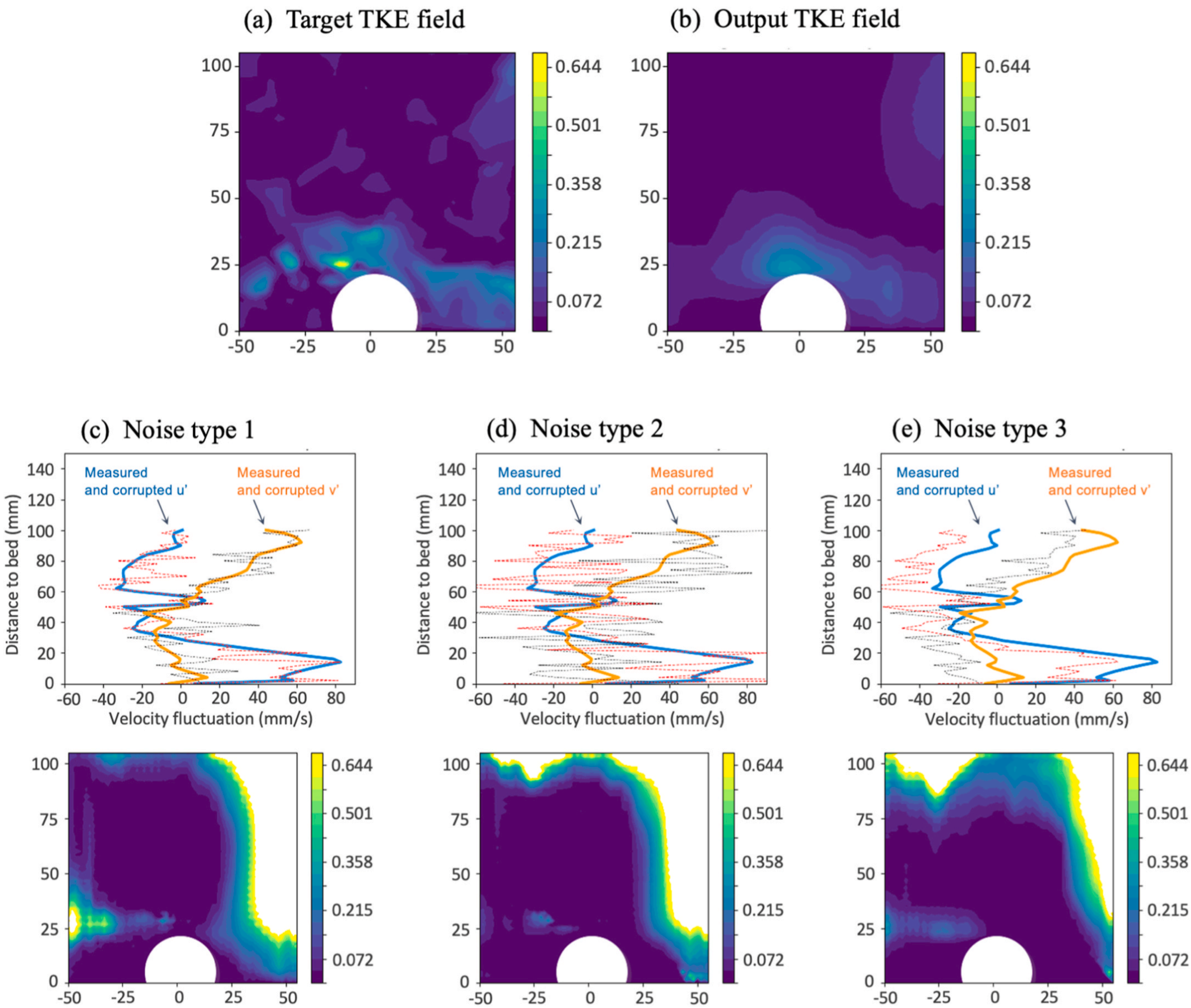

Fig. 15. Model robustness analysis 2 for quasi-instantaneous TKE reconstructed using corrupted measurement at the boundary. (a) and (b) are target TKE field and original output using clean measurement, respectively. (c)–(e) show the profiles of corrupted boundary measurements of u' and v' with noises and the corresponding output TKE fields. For the contour plots, the horizontal and vertical axes are in millimeters, and the color bar units are 10^{-2} m^2/s^2. Flow is from right to left. White regions are off-scale. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

the velocity field, particularly at elevations between 75 mm and 100 mm when subjected to both positive and negative mean biases, underscores a potential issue with model interpretability. Further improvements should aim not just for general accuracy but also for clarity in its internal mechanisms, e.g., using physics-embedded neural networks (Horie and Mitsume, 2022; Mohan et al., 2023).

The current model serves as a foundation for a series of evolutions in the future. Extending the model to accommodate a broader range of aquatic conditions (e.g., pollution or mass transport) or different fluid dynamics scenarios (e.g., sediment-laden flows and sediment suspension concentration) can widen the use across various fields. Moreover, we position the current approach as a fast and versatile "boundary-to-field" reconstructor and leave systematic baseline benchmarking against standard POD/PCA regression reconstructions and PINN-based formulations for future work.

CRediT authorship contribution statement

Yifan Yang: Writing – review & editing, Writing – original draft, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Yushu Xie:** Writing – review & editing, Investigation, Data curation. **Julia C. Mullarney:** Writing – review & editing, Methodology, Formal analysis. **Dong Shao:** Writing – review & editing, Investigation.

Data availability statement

The source code files in the format of Jupyter Notebooks and original datasets used for model development can be accessed using the following link:

https://www.kaggle.com/datasets/yyan749/flow-field-reconstruction-code.

Declaration of competing interest

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

References

Baddoo, P.J., Herrmann, B., McKeon, B.J., Nathan Kutz, J., Brunton, S.L., 2023. Physics-informed dynamic mode decomposition. Proceed. Royal Soc. A 479 (2271), 20220576.

Bentivoglio, R., Isufi, E., Jonkman, S.N., Taormina, R., 2022. Deep learning methods for flood mapping: a review of existing applications and future research directions. Hydrol. Earth Syst. Sci. 26 (16), 4345–4378.

Brunton, S.L., Noack, B.R., Koumoutsakos, P., 2020. Machine learning for fluid mechanics. Annu. Rev. Fluid Mech. 52, 477–508.

Cai, S., Mao, Z., Wang, Z., Yin, M., Karniadakis, G.E., 2021. Physics-informed neural networks (PINNs) for fluid mechanics: a review. Acta Mech. Sin. 37 (12), 1727–1738.

Callaham, J.L., Maeda, K., Brunton, S.L., 2019. Robust flow reconstruction from limited measurements via sparse representation. Phys. Rev. Fluids 4 (10), 103907.

Déjeans, B.S., Mullarney, J.C., MacDonald, I.T., 2022. Lagrangian observations and modeling of turbulence along a tidally influenced river. Water Resour. Res. 58 e2020WR027894.

Erichson, N.B., Mathelin, L., Yao, Z., Brunton, S.L., Mahoney, M.W., Kutz, J.N., 2020. Shallow neural networks for fluid flow reconstruction with limited sensors. Proceed. Royal Soc. A 476 (2238), 20200097.

Fu, G., Jin, Y., Sun, S., Yuan, Z., Butler, D., 2022. The role of deep learning in urban water management: a critical review. Water Res., 118973

Fukami, K., Nabae, Y., Kawai, K., Fukagata, K., 2019. Synthetic turbulent inflow generator using machine learning. Phys. Rev. Fluids 4 (6), 064603.

Gao, H., Sun, L., Wang, J.X., 2021. Super-resolution and denoising of fluid flow using physics-informed convolutional neural networks without high-resolution labels. Phys. Fluids 33 (7).

Gotvald, A.J., Oberg, K.A., 2009. Acoustic Doppler current profiler applications used in rivers and estuaries by the US Geological Survey (No. 2008-3096). US Geol. Survey.

Horie, M., Mitsume, N., 2022. Physics-embedded neural networks: graph neural PDE solvers with mixed boundary conditions. Adv. Neural Inf. Process. Syst. 35, 23218–23229.

Kao, I.F., Zhou, Y., Chang, L.C., Chang, F.J., 2020. Exploring a long short-term memory based encoder-decoder framework for multi-step-ahead flood forecasting. J. Hydrol. 583, 124631.

Karniadakis, G.E., Kevrekidis, I.G., Lu, L., Perdikaris, P., Wang, S., Yang, L., 2021. Physics-informed machine learning. Nat. Rev. Phys. 3 (6), 422–440.

Kashinath, K., Mustafa, M., Albert, A., Wu, J.L., Jiang, C., Esmaeilzadeh, S., et al., 2021. Physics-informed machine learning: case studies for weather and climate modelling. Phil. Transac. Royal Soc. A 379 (2194), 20200093.

Koken, M., Constantinescu, G., Blanckaert, K., 2013. Hydrodynamic processes, sediment erosion mechanisms, and Reynolds-number-induced scale effects in an open channel bend of strong curvature with flat bathymetry. J. Geophys. Res.: Earth Surf. 118 (4), 2308–2324.

Li, X., Feng, M., Ran, Y., Su, Y., Liu, F., Huang, C., Shen, H., Xiao, Q., Su, J., Yuan, S., Guo, H., 2023. Big data in Earth system science and progress towards a digital twin. Nat. Rev. Earth Environ. 1–14.

Liu, Y., Ponce, C., Brunton, S.L., Kutz, J.N., 2023. Multiresolution convolutional autoencoders. J. Comput. Phys. 474, 111801.

Mohan, A.T., Lubbers, N., Chertkov, M., Livescu, D., 2023. Embedding hard physical constraints in neural network coarse-graining of three-dimensional turbulence. Phys. Rev. Fluids 8 (1), 014604.

Nakagawa, H., 1993. Turbulence in Open Channel Flows. Routledge.

Nokes, R., 2019. Streams version 3.02 user's guide: core objects. Department of Civil and Natural Resources Engineering. University of Canterbury, Christchurch, New Zealand (accessed via: https://www.canterbury.ac.nz/engineering/schools/cnre/software/streams/.

Raissi, M., Yazdani, A., Karniadakis, G.E., 2020. Hidden fluid mechanics: learning velocity and pressure fields from flow visualizations. Science 367 (6481), 1026–1030.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: convolutional networks for biomedical image segmentation. Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18, 234–241. Springer International Publishing.

Scherl, I., Strom, B., Shang, J.K., Williams, O., Polagye, B.L., Brunton, S.L., 2020. Robust principal component analysis for modal decomposition of corrupt fluid flows. Phys. Rev. Fluids 5 (5), 054401.

Schlichting, H., Gersten, K., 2016. Boundary-Layer Theory. Springer.

Thuerey, N., Weißenow, K., Prantl, L., Hu, X., 2020. Deep learning methods for Reynolds-averaged Navier–Stokes simulations of airfoil flows. AIAA J. 58 (1), 25–36.

Vinuesa, R., Brunton, S.L., McKeon, B.J., 2023. The transformative potential of machine learning for experiments in fluid mechanics. Nat. Rev. Phys. 1–10.

Wang, N., Chang, H., Zhang, D., 2022. Inverse modeling for subsurface flow based on deep learning surrogates and active learning strategies. Water Resour. Res. e2022WR033644.

Wang, Y., Shi, L., Hu, X., Song, W., Wang, L., 2023. Multiphysics-informed neural networks for coupled soil hydrothermal modeling. Water Resour. Res. 59 (1) e2022WR031960.

Xie, Y., Melville, B.W., Shamseldin, A.Y., Whittaker, C.N., Yang, Y., 2023. Simultaneous measurements of inertial acceleration and flow structures during coarse-grain entrainment under steady currents. Ocean Eng. 276, 114165.

Xu, Y., Sha, Y., Wang, C., Wei, Y., 2023. Estimation of cavitation velocity fields based on limited pressure data through improved U-shaped neural network. Phys. Fluids 35 (8), 083323.

Yang, Y., Xie, C., Fan, Z., Xu, Z., Melville, B.W., Liu, G., Hong, L., 2024. Digital twinning of river basins towards full-scale, sustainable and equitable water management and disaster mitigation. npj Nat. Hazards.

Yang, Z., Yang, H., Yin, Z., 2023. Super-resolution reconstruction for the three-dimensional turbulence flows with a back-projection network. Phys. Fluids 35 (5).

Yousefpour, N., Downie, S., Walker, S., Perkins, N., Dikanski, H., 2021. Machine learning solutions for bridge scour forecast based on monitoring data. Transp. Res. Rec. 2675 (10), 745–763.

Zahura, F.T., Goodall, J.L., Sadler, J.M., Shen, Y., Morsy, M.M., Behl, M., 2020. Training machine learning surrogate models from a high-fidelity physics-based model: application for real-time street-scale flood prediction in an urban coastal community. Water Resour. Res. 56 (10) e2019WR027038.

Zhang, F., Hu, H., Ren, F., Zhang, H., Du, P., 2022a. Reconstructing turbulent velocity information for arbitrarily gappy flow fields using the deep convolutional neural network. Phys. Fluids 34 (12).

Zhang, Z., Flora, K., Kang, S., Limaye, A.B., Khosronejad, A., 2022a. Data-driven prediction of turbulent flow statistics past bridge piers in large-scale Rivers using convolutional neural networks. Water Resour. Res. 58 (1) e2021WR030163.

Zhang, Z., Hao, X., Santoni, C., Shen, L., Sotiropoulos, F., Khosronejad, A., 2023. Toward prediction of turbulent atmospheric flows over propagating oceanic waves via machine-learning augmented large-eddy simulation. Ocean Eng. 280, 114759.

Zhang, Z., Limaye, A.B., Khosronejad, A., 2022b. Three-dimensional realizations of flood flow in large-scale rivers using the neural fuzzy-based machine-learning algorithms. Comput. Fluid 246, 105611.

Zeng, J., Constantinescu, G., Blanckaert, K., Weber, L., 2008. Flow and bathymetry in sharp open-channel bends: experiments and predictions. Water Resour. Res. 44 (9).

Zhou, Z., Shi, L., Zha, Y., Wang, S., Xu, B., Tian, L., Yang, R., 2022a. Densely connected squeeze-and-excitation convolutional encoder-decoder networks for identifying preferential channels in highly heterogeneous porous media. Water Resour. Res. 58 (9) e2021WR031429.

Zhou, Z., Zabaras, N., Tartakovsky, D.M., 2022b. Deep learning for simultaneous inference of hydraulic and transport properties. Water Resour. Res. 58 (10) e2021WR031438.