



# Flow matching and diffusion models via PointNet for generating fluid fields on irregular geometries

Ali Kashefi 

Stanford University, Stanford, 94305, CA, USA

## ARTICLE INFO

Dataset link: [https://github.com/Ali-Stanford/Diffusion\\_Flow\\_Matching\\_PointNet\\_CFD](https://github.com/Ali-Stanford/Diffusion_Flow_Matching_PointNet_CFD)

### Keywords:

Generative frameworks  
Flow matching  
Diffusion model  
PointNet  
Irregular geometries  
Incompressible flow

## ABSTRACT

We present two novel generative geometric deep learning frameworks, termed Flow Matching PointNet and Diffusion PointNet, for predicting fluid flow variables on irregular geometries by incorporating PointNet into flow matching and diffusion models, respectively. In these frameworks, a reverse generative process reconstructs physical fields from standard Gaussian noise conditioned on unseen geometries. The proposed approaches operate directly on point-cloud representations of computational domains (e.g., grid vertices of finite-volume meshes) and therefore avoid the limitations of pixelation used to project geometries onto uniform lattices, as is common in U-Net-based flow matching and diffusion models. In contrast to graph neural network-based diffusion models, Flow Matching PointNet and Diffusion PointNet do not exhibit high-frequency noise artifacts in the predicted fields. Moreover, unlike such approaches, which require auxiliary intermediate networks to condition geometry, the proposed frameworks rely solely on PointNet, resulting in a simple and unified architecture. The performance of the proposed frameworks is evaluated on steady incompressible flow past a cylinder, using a geometric dataset constructed by varying the cylinder's cross-sectional shape and orientation across samples. With the same number of trainable parameters, Flow Matching PointNet and Diffusion PointNet demonstrate more accurate predictions of velocity and pressure fields, as well as lift and drag forces, and exhibit greater robustness to incomplete geometries compared to a vanilla PointNet.

## 1. Introduction and motivation

Flow matching [1], diffusion models [2], and their advanced variants [3–5] are generative deep learning frameworks that have demonstrated success in computer graphics and computer vision, particularly for image and video generation [6], outperforming earlier generative approaches such as generative adversarial networks (GANs) [7,8] and variational autoencoders (VAEs) [9]. At a high level, in supervised flow matching [1] and diffusion [2] frameworks, a forward process adds noise to the training data, while an inverse process predicts the clean fields. Several formulations of flow matching and diffusion models operate in a conditional setting, where generation is guided by auxiliary inputs such as textual prompts [4,10] or low-resolution images [11,12]. Motivated by their success in computer graphics, these models have been extended to computational physics, enabling a range of applications including predictions conditioned on material properties [13–15], high-fidelity field reconstruction from low-resolution data [16,17], and prediction of physical fields conditioned on the geometry of the computational domain [13,18,19]. In this article, we focus on the latter case.

Flow matching and diffusion models conditioned on the geometry of the computational domain fall within the category of geometric deep learning. These approaches are critical in computational physics, as they accelerate shape-optimization studies

E-mail address: [kashefi@stanford.edu](mailto:kashefi@stanford.edu).

<https://doi.org/10.1016/j.cma.2026.119037>

Received 26 January 2026; Received in revised form 23 March 2026; Accepted 25 April 2026

Available online 5 May 2026

0045-7825/© 2026 Elsevier B.V. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

that require exploration of high-dimensional geometric parameter spaces [20]. Generative flow matching and diffusion models conditioned on geometry have been shown to outperform regression-based surrogate models for predicting physical fields (see e.g., Refs. [21–23]), such as convolutional neural networks (CNNs) [24], Fourier neural operators (FNOs) [25], and deep operator neural networks (DeepONets) [26], which directly map geometry to physical fields. In particular, traditional CNNs [24], FNOs [25], and DeepONets [26] produce a single output for a given geometry, whereas flow matching and diffusion models can generate multiple realizations for the same geometry, thereby quantifying model uncertainty. Moreover, diffusion models have demonstrated superior performance compared to GANs and VAEs in predicting physical field variables (see, e.g., Refs. [21,27]). In geometric deep learning using flow matching and diffusion models, two steps are particularly important. First, how to represent irregular geometries; and second, how to condition flow matching and diffusion models on geometry from a computer science perspective.

For the representation of irregular geometries, two common approaches are used. The first is a pixelation-based strategy, in which the geometry is projected onto uniform Cartesian grids. In this setting, Q. Liu and N. Thuerey [13] proposed a diffusion model conditioned on the geometry of airfoils to predict velocity and pressure fields. In their work [13], the airfoil geometry was represented using the pixelation approach, and a U-Net architecture [2,28] was employed as the backbone neural network. The authors [13] performed geometric conditioning by concatenating the binary image of the airfoil with the velocity and pressure fields as the input to the U-Net, while noise was added only to the velocity and pressure fields during training (see Fig. 3 of Ref. [13]). A similar approach was taken by Zheng et al. [18], where the authors employed an encoder that maps a two-dimensional binary image of the geometry to a latent feature representation, which was then used in a diffusion transformer to condition the diffusion model on the geometry (see Fig. 1 of Ref. [18]). Similarly, J. Hu et al. [19] proposed a diffusion model conditioned on the geometry of obstacles to predict incompressible flow fields. The authors [19] conditioned the diffusion model on geometry by encoding the obstacle as a binary mask through a U-Net-style encoder and injecting this geometry representation into the diffusion process via cross-attention blocks (see Fig. 1 of Ref. [19]). Their results demonstrated that the proposed diffusion model outperformed CNNs and VAEs [19].

However, pixelation strategies lose smooth boundary information (see e.g., Fig. 17 of Ref. [13] and Fig. 3 of Ref. [18]) and require high global resolution to capture local geometric details (e.g., the trailing edge of an airfoil), leading to increased computational cost. In addition, training data must be interpolated onto uniform grids, introducing additional errors [29,30]. In contrast, graph and point cloud representations enable precise geometry modeling with adaptive resolution, eliminate the need for interpolation, and thereby reduce both approximation errors and computational cost [29,30]. To avoid the limitations of the first approach, the second approach has been proposed, which represents geometries using unstructured meshes or point clouds, and leverages graph neural networks [31–33] or PointNet [29,34–38]. M. Lino et al. [39] proposed a diffusion graph network and a flow matching graph network for predicting velocity and pressure fields around airfoils with a variety of geometries [39]. However, the authors [39] faced shortcomings due to the limited performance of message passing for propagating features across the graph, as this led to high-frequency noise in the velocity and pressure field predictions. To overcome this challenge, the authors [39] introduced an alternative version in which noise was added during training in the latent space after encoding the physical fields and the geometry of the domain by a graph neural network. Consequently, their neural network architecture became more complicated than a pure graph neural network and required auxiliary neural networks, such as a variational autoencoder at intermediate levels (see Fig. 2 of Ref. [39]). In the same line of research, J. Kim et al. [40] introduced a point-wise diffusion model in which the geometry of the physical domain was represented as a set of points; however, each point was treated independently, with its own forward and inverse diffusion process, an independent positional encoding, and an independent diffusion transformer block. Although the method yielded successful results, the neural network architecture was complex (see Fig. 3 of Ref. [40]) and did not account for the fact that physical field predictions (e.g., velocity and pressure) depended on global geometric features, since noise addition and denoising were not functions of the entire geometry. Moreover, the independent treatment of points made it unclear how the reverse process could be used to generate multiple samples and quantify model uncertainty, which is a key advantage of diffusion models over other surrogate approaches. This trend motivates us to introduce a simple and elegant neural network based on flow matching and diffusion models for irregular geometries.

We propose, for the first time, Flow Matching PointNet and Diffusion PointNet, in which the segmentation branch of PointNet [34] is employed as the denoising network within flow matching [1] and diffusion [41] frameworks. PointNet [34] and its extensions [34,35,38,42,43] are used for classification and segmentation of point clouds in computer graphics. Kashefi et al. [29] were the first to adapt PointNet [34] for supervised deep learning of incompressible flow. Subsequent studies extended their framework to compressible flows [44], elasticity [45], physics-informed machine learning [36,37,46,47], and Kolmogorov–Arnold networks [30].

The Flow Matching PointNet and Diffusion PointNet frameworks consist of a forward process and an inverse denoising process. In Diffusion PointNet, the forward process incrementally corrupts ground-truth velocity and pressure fields in fluid flows by adding standard Gaussian noise. PointNet is then trained to predict the noise added at each diffusion step [2]. This prediction is conditioned on the spatial coordinates of the point cloud, which represent the domain geometry, as well as on a sinusoidal time embedding [48]. The conditioning is implemented by concatenating the spatial-coordinate vectors and the sinusoidal-embedding vector with the corrupted velocity and pressure fields. In the inverse process, generation begins from samples drawn from a standard Gaussian noise distribution and is conditioned on the spatial coordinates of the point cloud and the sinusoidal embedding. During the inverse procedure, the trained PointNet predicts the noise at each stage and progressively reconstructs the velocity and pressure fields in a manner consistent with the imposed conditioning. In the Flow Matching PointNet model, PointNet is trained to predict a target vector field that corresponds to the temporal derivative of a continuously perturbed field with respect to an auxiliary time variable [1]. In the inverse process, this learned vector field is used to numerically integrate an ordinary differential equation, progressively

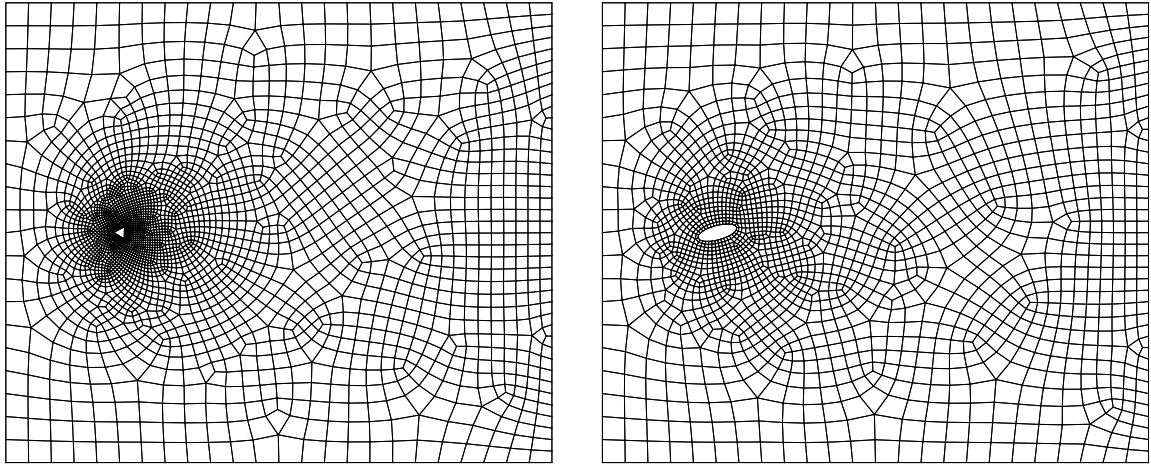


Fig. 1. Finite volume meshes used for the numerical simulation of flow over a cylinder with a triangular cross section, with 2775 vertices on the left panel, and an elliptical cross section, with 2672 vertices on the right panel.

reconstructing the velocity and pressure fields from an initial standard Gaussian noise distribution [1], using the same conditioning mechanism as in Diffusion PointNet. To investigate the performance of Flow Matching PointNet and Diffusion PointNet, the classical fluid dynamics problem of incompressible flow past a cylinder is considered, in which the cross-sectional shape of the cylinder varies across the dataset.

Note that combinations of diffusion models and point cloud neural networks in the computer graphics literature have been used in which, during the forward process, noise is added to the spatial coordinates of shapes and, during the inverse process, new three-dimensional (3D) geometries are generated [41,49]. In contrast, in the present study, noise is added to and removed from the physical fields, while the spatial coordinates remain clean and unchanged. Additionally, Y. He et al. [50] proposed a framework for semantic segmentation of 3D point clouds using a diffusion model (see Fig. 1 of Ref. [50]). In their approach, the diffusion process is conditioned on two components: first, the geometry of the input represented by the point cloud, and second, the semantic context, which provides a prior estimate of the possible label types for the points in the point cloud. Although this framework is successful in computer graphics, it is not applicable to predicting physical quantities such as velocity and pressure fields, since these quantities are continuous rather than discrete integer labels, as in semantic segmentation.

We summarize the key features and advantages of the proposed Flow Matching PointNet and Diffusion PointNet frameworks as follows. First, both Flow Matching PointNet and Diffusion PointNet predict velocity and pressure fields on unseen geometries more accurately than the baseline (vanilla) PointNet while using the same number of trainable parameters. Second, the proposed frameworks naturally handle irregular geometries and avoid the limitations of CNN-based flow matching and diffusion models that rely on pixelation techniques. Third, in contrast to graph neural network-based diffusion models, the proposed frameworks do not exhibit high-frequency noise artifacts in the predicted physical fields. Fourth, the proposed frameworks employ a simple PointNet architecture and condition the geometry through a straightforward concatenation strategy, eliminating the need for auxiliary neural networks at intermediate levels (e.g., variational autoencoders) for geometric conditioning. Finally, the proposed frameworks use a single, unified forward and inverse process for the entire point cloud, rather than independent diffusion processes for individual points, resulting in more coherent and robust frameworks.

The structure of the rest of this paper is as follows. Section 2 describes how the labeled data are generated for training the proposed deep learning configurations. Section 3 discusses the architectures of the Flow Matching PointNet, Diffusion PointNet, and baseline PointNet frameworks, along with their training strategies. Quantitative and visual results and predictions of these three models, along with a comprehensive comparison from different aspects such as velocity and pressure field predictions, lift and drag computations, and robustness in the presence of distorted point clouds, are presented in Section 4. Lastly, Section 5 concludes the paper with a summary and proposes several ideas for future research directions.

## 2. Data preparation

To train the Flow Matching PointNet and Diffusion PointNet models and evaluate their prediction accuracy, labeled data are necessary. For this purpose, we use the well-known steady-state incompressible flow past a cylinder problem as a benchmark example. The motion of an incompressible, viscous, Newtonian fluid is governed by the laws of mass and momentum conservation, together with suitable boundary conditions, as follows:

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } V, \quad (1)$$

$$\rho(\mathbf{u} \cdot \nabla)\mathbf{u} - \mu \Delta \mathbf{u} + \nabla p = \mathbf{0} \quad \text{in } V, \quad (2)$$

$$\mathbf{u} = \mathbf{u}_{\Gamma_D} \quad \text{on } \Gamma_D, \quad (3)$$

$$-pn + \mu \nabla \mathbf{u} \cdot \mathbf{n} = \mathbf{t}_{\Gamma_{Ne}} \quad \text{on } \Gamma_{Ne}, \quad (4)$$

where  $\mathbf{u}$  and  $p$  are the velocity vector and the pressure field in the domain  $V$ , respectively. The constants  $\rho$  and  $\mu$  denote the fluid density and dynamic viscosity. The Dirichlet and Neumann boundaries,  $\Gamma_D$  and  $\Gamma_{Ne}$ , are distinct and non-overlapping. The outward unit normal vector on  $\Gamma_{Ne}$  is  $\mathbf{n}$ , and the traction vector there is  $\mathbf{t}_{\Gamma_{Ne}}$ . The velocity components in the  $x$  and  $y$  directions are represented by  $u$  and  $v$ .

We seek to solve Eqs. (1)–(2) for flow past an infinite cylinder with different cross-sectional geometries. The computational domain is set as  $V = [0, 38 \text{ m}] \times [0, 32 \text{ m}]$ . The cylinder cross-section, centered at (8 m, 16 m), is modeled as a rigid body with a no-slip boundary condition. The inflow, top, and bottom boundaries are assigned a uniform velocity  $u_\infty$  parallel to the  $x$ -axis. To simulate far-field behavior [51], a stress-free condition is applied at the outflow boundary (i.e.,  $\mathbf{t}_{\Gamma_{Ne}} = \mathbf{0}$ ). All physical quantities are expressed in SI units, with  $\rho = 1.00 \text{ kg/m}^3$ ,  $u_\infty = 1.00 \text{ m/s}$ , and  $\mu = 0.05 \text{ Pa s}$ .

The computational domain  $V$  is discretized into unstructured finite volume meshes using Gmsh [52]. Fig. 1 displays two sample meshes corresponding to cylinders with triangular and elliptical cross-sections. The governing Eqs. (1)–(2) are numerically solved using the OpenFOAM package [53], which applies the Semi-Implicit Method for Pressure Linked Equations (SIMPLE) [54]. The iteration process continues until the  $L^2$  norm of the residuals becomes smaller than  $10^{-3}$ , at which point the flow is assumed steady.

To create labeled data, seven common geometric shapes are used for the cross-section, each with various scales and orientations, as summarized in Table 1. The selected shapes include circle [51,55,56], square [57], triangle [58], rectangle [59], ellipse [60], pentagon, and hexagon [61]. The characteristic length of the geometry is denoted by  $L$ , and the Reynolds number is defined as

$$\text{Re} = \frac{\rho L u_\infty}{\mu}. \quad (5)$$

As shown in Table 1, for circular, square, equilateral triangle, pentagon, and hexagon cross-sections,  $L = a$ , whereas for rectangles, ellipses, and non-equilateral triangles,  $L = b$ . The Reynolds number in the dataset ranges from 20.0 to 76.0, with a total of 2235 labeled cases. The dataset is randomly divided into training (1772 samples), validation (241 samples), and test (222 samples) subsets. To prepare point-cloud representations, we use the mesh vertices as points, following the approach of Ref. [29]. To emphasize the regions around the cylinder and its wake, the first  $N = 1024$  nearest vertices to the cylinder's center are chosen to form each point cloud. Because the grid vertices are unevenly distributed in the unstructured meshes, the coordinate ranges differ between samples:  $x_{\min} \in [0, 5.46 \text{ m}]$ ,  $x_{\max} \in [10.55 \text{ m}, 22.14 \text{ m}]$ ,  $y_{\min} \in [0, 13.41 \text{ m}]$ , and  $y_{\max} \in [18.57 \text{ m}, 32 \text{ m}]$ .

We define the dimensionless variables  $u^*$ ,  $v^*$ , and  $p^*$  for the velocity and pressure as

$$u^* = \frac{u}{u_\infty}, \quad (6)$$

$$v^* = \frac{v}{u_\infty}, \quad (7)$$

$$p^* = \frac{p - p_0}{\rho u_\infty^2}, \quad (8)$$

where  $p_0$  denotes the atmospheric pressure. Next, we normalize the spatial coordinates of the training data to the range  $[-1, 1]$  using

$$\{\phi'\} = 2 \left( \frac{\{\phi\} - \min(\{\phi\})}{\max(\{\phi\}) - \min(\{\phi\})} \right) - 1, \quad (9)$$

where  $\{\phi\}$  includes  $\{x^*, y^*\}$ . Similarly, we normalize the field variables of training data to the range of  $[0, 1]$  using









$$\{\psi'\} = \frac{\{\psi\} - \min(\{\psi\})}{\max(\{\psi\}) - \min(\{\psi\})}, \quad (10)$$

where  $\{\psi\}$  includes  $\{u^*, v^*, p^*\}$ . The scaled data are denoted by  $\{x', y', u', v', p'\}$ . The predictions are later rescaled back to their physical domain values. The reason for adopting a different scaling for the velocity and pressure fields compared to the spatial coordinates  $x$  and  $y$  is that our machine learning experiments indicate that this scaling yields more accurate and stable results than other possible choices.

### 3. Deep learning architectures

In this section, we elaborate on the architecture, training procedure, and inference process for the three models of Flow Matching PointNet, Diffusion PointNet, and the baseline PointNet. The neural network of each model is PointNet [34]; however, in each case, PointNet is used for a different purpose. All three models employ the same PointNet architecture; however, their inputs and outputs differ (see Fig. 2). The architecture of PointNet [34] itself remains unchanged unless explicitly stated, for instance, when the activation functions differ among the models. To avoid redundancy, we first describe the algorithmic formulation and the corresponding input and output definitions for each model, and then present the architecture of PointNet at the end of this section (Section 3.4).

**Table 1**  
Details of the generated data.

Shape	Schematic figure	Variation in orientation	Variation in length scale	Number of data
Circle		-	$a = 1 \text{ m}$	1
Equilateral hexagon		$3^\circ, 6^\circ, \dots, 60^\circ$	$a = 1 \text{ m}$	20
Equilateral pentagon		$3^\circ, 6^\circ, \dots, 72^\circ$	$a = 1 \text{ m}$	24
Square		$3^\circ, 6^\circ, \dots, 90^\circ$	$a = 1 \text{ m}$	30
Equilateral triangle		$3^\circ, 6^\circ, \dots, 180^\circ$	$a = 1 \text{ m}$	60
Rectangle		$3^\circ, 6^\circ, \dots, 180^\circ$	$a = 1 \text{ m}; b/a = 1.2, 1.4, \dots, 3.6$	780
Ellipse		$3^\circ, 6^\circ, \dots, 180^\circ$	$a = 1 \text{ m}; b/a = 1.2, 1.4, \dots, 3.8$	840
Triangle		$3^\circ, 6^\circ, \dots, 360^\circ$	$a = 1 \text{ m}; b/a = 1.5, 1.75$ $\gamma = 60^\circ, 80^\circ$	480

### 3.1. Generative Flow Matching PointNet

We describe the concept and architecture of the proposed Flow Matching PointNet model in this subsection. At a high level, the objective is to predict the desired physical fields such as velocity and pressure, conditioned on a given geometry represented by scattered spatial coordinates. The Flow Matching PointNet is obtained by combining PointNet [34] with the flow matching formulation [1], as shown in the top panel of Fig. 2. Let  $\mathbf{y}_{\text{clean}} \in \mathbb{R}^{N \times n_{\text{CFD}}}$  denote the clean desired physical fields, where  $n_{\text{CFD}}$  represents the number of field variables. In this study, we focus on predicting the velocity and pressure variables in two dimensions, namely  $u'$ ,  $v'$ , and  $p'$ , hence  $n_{\text{CFD}} = 3$ . Let  $\mathbf{y}_{\text{noisy}} \in \mathbb{R}^{N \times n_{\text{CFD}}}$  denote a standard Gaussian noise field. We define a temporal variable  $\tau$ . At an intermediate  $\tau \in [0, 1]$ , a linear interpolation is given by

$$\mathbf{y}_\tau = (1 - \tau)\mathbf{y}_{\text{clean}} + \tau\mathbf{y}_{\text{noisy}}. \quad (11)$$

Notably, at  $\tau = 0$ , we recover  $\mathbf{y}_{\text{clean}}$ , while at  $\tau = 1$ , we obtain  $\mathbf{y}_{\text{noisy}}$ . The derivative of  $\mathbf{y}_\tau$  with respect to  $\tau$  defines the target field as

$$\frac{d\mathbf{y}_\tau}{d\tau} = \mathbf{f}_{\text{target}}, \quad (12)$$

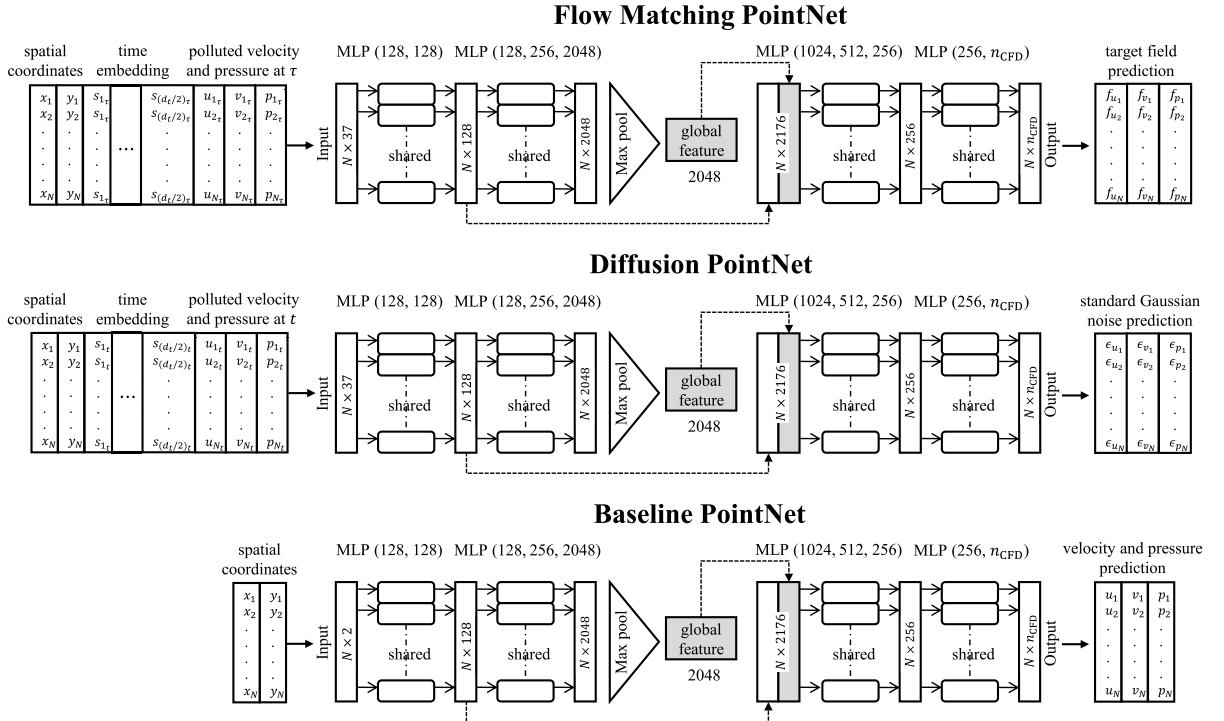
where

$$\mathbf{f}_{\text{target}} = \mathbf{y}_{\text{noisy}} - \mathbf{y}_{\text{clean}}. \quad (13)$$

and  $\mathbf{f}_{\text{target}} \in \mathbb{R}^{N \times n_{\text{CFD}}}$ .

The input of the Flow Matching PointNet consists of the spatial coordinates of each point cloud, representing the geometry of the computational domain, concatenated with the  $u'$ ,  $v'$ , and  $p'$  field variables that are perturbed by a certain level of noise according to Eq. (11) at a temporal variable  $\tau$ . In addition to the spatial coordinates (e.g.,  $x'$  and  $y'$ ) and the perturbed variables, the network must also be informed of the temporal variable, which indicates the level of noise contamination. To achieve this, we construct sinusoidal temporal embedding vectors [48] and concatenate them with the input vector of the Flow Matching PointNet. The output of the network is the predicted target field (i.e.,  $\mathbf{f}_{\text{target}}$ ). Let us mathematically formulate this procedure. Let  $\mathbf{x} \in \mathbb{R}^{N \times d}$  denote the spatial coordinates of the points in a point cloud consisting of  $N$  points that represent a computational domain, where  $d$  is the number of spatial dimensions. In this study, we set  $d = 2$ . At the temporal variable  $\tau$ , a sinusoidal time-embedding layer ( $s_\tau$ ) with dimension  $d_\tau$  is defined as

$$\mathbf{s}_\tau = [\sin(\omega_k \tau), \cos(\omega_k \tau)]_{k=1}^{d_\tau/2}, \quad (14)$$



**Fig. 2.** Architecture of Flow Matching PointNet, Diffusion PointNet, and Baseline PointNet. Labels in the form  $(B_1, B_2)$  and  $(B_1, B_2, B_3)$  are used to denote shared MLPs, as illustrated in Section 3.4.  $n_{\text{CFD}}$  denotes the number of flow variables and  $N$  represents the number of points in the point clouds. In the Flow Matching PointNet architecture, the subscript  $\tau$  is used to indicate  $s_\tau$  and  $y_\tau$  (see Section 3.1). In the Diffusion PointNet architecture, the subscript  $t$  is used to show  $s_t$  and  $y_t$  (see Section 3.2). In Flow Matching PointNet and Diffusion PointNet, the outputs are, respectively, the prediction of the target field ( $f_{\text{target}}$ ) and the noisy field ( $y_{\text{noisy}}$ ), given the polluted velocity and pressure fields at steps  $\tau$  and  $t$ , and conditioned on the geometry of the physical domain represented as a point cloud.

where, for each  $k$ ,

$$\omega_k = \exp\left(-\frac{k-1}{\frac{d_\tau}{2}-1} \ln(10^4)\right). \quad (15)$$

In this study, we set  $d_\tau = 32$ , and our machine learning experiments indicate that this choice provides sufficient accuracy. The constant  $10^4$  in Eq. (15) controls the range of frequencies, ensuring that the encoded time values span several orders of magnitude. This logarithmic scaling enables the network to capture both slow and fast temporal variations effectively across different noise levels [48].

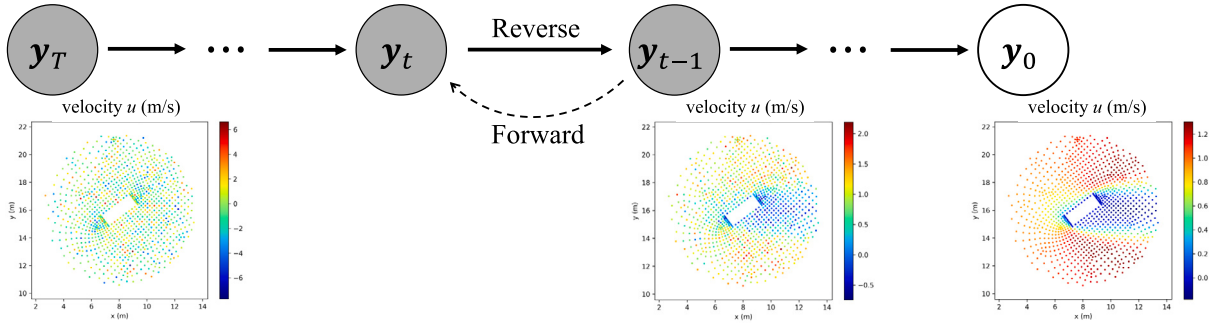
Note that  $s_\tau \in \mathbb{R}^{1 \times d_\tau}$ , simply because each point cloud at a given  $\tau$  has one defined vector  $s_\tau$ . To ensure dimensional consistency, we repeat the same vector  $s_\tau$  for every pair of  $(x', y')$  in the point cloud, thereby constructing the tensor  $S_\tau \in \mathbb{R}^{N \times d_\tau}$ . Hence, for each point cloud, the embedded tensor  $S_\tau$  is concatenated with the contaminated field  $y_\tau$  and the spatial coordinates  $x$ , forming an input tensor  $\mathcal{X} \in \mathbb{R}^{N \times (d+d_\tau+n_{\text{CFD}})}$  for the Flow Matching PointNet. The output of the Flow Matching PointNet is a tensor  $\mathcal{Y} \in \mathbb{R}^{N \times n_{\text{CFD}}}$ , representing the prediction of  $f_{\text{target}}$ . The loss function is defined as the mean squared error that measures the discrepancy between  $\mathcal{Y}$  and  $f_{\text{target}}$  and can be written as

$$\mathcal{L}_{\text{Flow Matching PointNet}} = \frac{1}{N \times n_{\text{CFD}}} \sum_{i=1}^N \left( (\tilde{f}_{u_i} - (\epsilon_{u_i} - u'_i))^2 + (\tilde{f}_{v_i} - (\epsilon_{v_i} - v'_i))^2 + (\tilde{f}_{p_i} - (\epsilon_{p_i} - p'_i))^2 \right), \quad (16)$$

where  $\tilde{f}_{u_i}$ ,  $\tilde{f}_{v_i}$ , and  $\tilde{f}_{p_i}$  denote the predicted components of the target field  $f_{\text{target}}$  produced by the Flow Matching PointNet, with the subscripts  $u_i$ ,  $v_i$ , and  $p_i$  corresponding to the  $u$ -velocity,  $v$ -velocity, and pressure components, respectively.  $\epsilon_{u_i}$ ,  $\epsilon_{v_i}$ , and  $\epsilon_{p_i}$  denote the corresponding components of  $y_{\text{noisy}}$  and the subscripts are similarly defined.

Flow Matching PointNet is trained using the Adam optimizer [62] with a constant learning rate of 0.001 and a batch size of 256. For each training sample, a temporal variable  $\tau$  is uniformly sampled from the interval  $[0, 1]$ , and three independent standard Gaussian noise fields are generated for the  $u'$ ,  $v'$ , and  $p'$  components. Note that each point cloud in the batch gets its own random  $\tau$  value, and given  $\tau$ , the noisy input  $y_\tau$  and the target field  $f_{\text{target}}$  are computed using Eqs. (11) and (13).

After training, Flow Matching PointNet generates the velocity and pressure fields conditioned on a geometry from the test set. Given  $x$  as the point cloud representing the target geometry from the test set, we generate three independent standard Gaussian



**Fig. 3.** Schematic of the reverse process in Diffusion PointNet. As an example, the denoising process for the  $u$ -velocity field is shown, while the network jointly predicts the noise for all velocity and pressure field variables.

fields representing  $\mathbf{y}_{\text{noisy}}$  and integrate Eq. (12) from  $\tau = 1$  to  $\tau = 0$  using the explicit Euler method with  $N_\tau$  steps, such that  $\Delta\tau = 1/N_\tau$ . At each step,

$$\mathbf{y}_n = \mathbf{y}_{n+1} - \Delta\tau \mathcal{Y}_{n+1 \rightarrow n}, \quad (17)$$

where  $\mathcal{Y}_{n+1 \rightarrow n}$  denotes the prediction of the Flow Matching PointNet given the concatenation of  $\mathbf{x}$ ,  $\mathbf{y}_n$ , and  $\mathcal{S}_\tau$  at  $\tau = 1.0 - n\Delta\tau$ . The final output at  $\tau = 0$  corresponds to  $\mathbf{y}_{\text{clean}}$ , which represents the predicted velocity and pressure fields ( $u'$ ,  $v'$ ,  $p'$ ) associated with the given geometry. In the present setup, we set  $N_\tau = 1000$ .

### 3.2. Generative Diffusion PointNet

In this subsection, we present the underlying concept and architectural design of the proposed Diffusion PointNet model. Similar to the previous part, the primary goal is to predict physical fields such as velocity and pressure, conditioned on a geometry represented by scattered spatial coordinates. The Diffusion PointNet framework integrates the PointNet architecture [34] with a denoising diffusion probabilistic model [2], as shown in the central panel of Fig. 2.

Similar to Section 3.1, let  $\mathbf{y}_{\text{clean}} \in \mathbb{R}^{N \times n_{\text{CFD}}}$  represent the clean field, where  $N$  and  $n_{\text{CFD}}$  are defined as before, since our objective is to predict the two-dimensional velocity and pressure variables  $u'$ ,  $v'$ , and  $p'$ . Fig. 3 illustrates the schematic of the forward and inverse processes in Diffusion PointNet. In the forward diffusion process, standard Gaussian noise is incrementally added to  $\mathbf{y}_{\text{clean}}$  over  $T$  discrete steps using a noise scheduler, producing a sequence of noisy fields  $\{\mathbf{y}_t\}_{t=1}^T$ , where  $\mathbf{y}_t \in \mathbb{R}^{N \times n_{\text{CFD}}}$ . In this study, we use a cosine noise scheduler defined as

$$\beta_t = 1 - \frac{\cos^2\left(\frac{(t/T)+r}{1+r} \frac{\pi}{2}\right)}{\cos^2\left(\frac{r}{1+r} \frac{\pi}{2}\right)}, \quad (18)$$

where  $r$  is the small offset parameter introduced in Ref. [63] to regularize the noise schedule, and set to 0.008 in our machine learning experiments. At any diffusion step  $t$ , a noisy sample is generated as

$$\mathbf{y}_t = \sqrt{\alpha_t} \mathbf{y}_{\text{clean}} + \sqrt{1 - \alpha_t} \mathbf{y}_{\text{noisy}}, \quad (19)$$

where

$$\alpha_t = 1 - \beta_t, \quad (20)$$

and

$$\bar{\alpha}_t = \prod_{i=1}^t \alpha_i, \quad (21)$$

and, as mentioned in Section 3.1,  $\mathbf{y}_{\text{noisy}}$  is standard Gaussian noise, and  $\mathbf{y}_{\text{noisy}} \in \mathbb{R}^{N \times n_{\text{CFD}}}$ .

The input to the Diffusion PointNet comprises the spatial coordinates of the point cloud, which define the geometry of the computational domain, concatenated with the field variables  $u'$ ,  $v'$ , and  $p'$  after being perturbed by noise prescribed by Eq. (19) at the diffusion step  $t$ . Similar to the Flow Matching PointNet, the Diffusion PointNet must be provided with the step  $t$ , along with the spatial coordinates (e.g.,  $x'$  and  $y'$ ) and the perturbed velocity and pressure variables. The step  $t$  identifies the associated level of noise contamination imposed through Eq. (19). To this end, we use the sinusoidal time-embedding layer [48] introduced in Eqs. (14)–(15). In terms of notation, instead of  $\tau$ , we now simply use  $t$ . For completeness, we rewrite the time-embedding layer  $s_t$  with dimension  $d_t$ , defined as

$$\mathbf{s}_t = [\sin(\omega_k t), \cos(\omega_k t)]_{k=1}^{d_t/2}, \quad (22)$$

where for every index  $k$ ,

$$\omega_k = \exp\left(-\frac{k-1}{\frac{d_t}{2}-1} \ln(10^d)\right), \quad (23)$$

and we set  $d_t = 32$ . Analogous to  $\mathcal{S}_\tau$  used in the Flow Matching PointNet framework, we define  $\mathcal{S}_t$  for the Diffusion PointNet model, where  $\mathcal{S}_t \in \mathbb{R}^{N \times d_t}$ . In this sense, for a given point cloud at diffusion step  $t$ , the embedded tensor  $\mathcal{S}_t$  is concatenated with the noisy field  $\mathbf{y}_t$  and the spatial coordinates of the point cloud  $\mathbf{x} \in \mathbb{R}^{N \times d}$ , resulting in the input tensor  $\mathcal{X} \in \mathbb{R}^{N \times (d+d_t+n_{\text{CFD}})}$  for the Diffusion PointNet. Accordingly, the Diffusion PointNet outputs a tensor  $\mathcal{Y} \in \mathbb{R}^{N \times n_{\text{CFD}}}$ , which represents the prediction of the noise  $\mathbf{y}_{\text{noisy}}$ . The loss function is defined as the mean squared error between  $\mathcal{Y}$  and  $\mathbf{y}_{\text{noisy}}$  and can be written as

$$\mathcal{L}_{\text{Diffusion PointNet}} = \frac{1}{N \times n_{\text{CFD}}} \sum_{i=1}^N \left( (\tilde{\epsilon}_{u_i} - \epsilon_{u_i})^2 + (\tilde{\epsilon}_{v_i} - \epsilon_{v_i})^2 + (\tilde{\epsilon}_{p_i} - \epsilon_{p_i})^2 \right), \quad (24)$$

where  $\epsilon_{u_i}$ ,  $\epsilon_{v_i}$ , and  $\epsilon_{p_i}$  denote the ground-truth standard Gaussian noise, as similarly defined in Section 3.1, and  $\tilde{\epsilon}_{u_i}$ ,  $\tilde{\epsilon}_{v_i}$ , and  $\tilde{\epsilon}_{p_i}$  denote the corresponding noise components predicted by the Diffusion PointNet. Diffusion PointNet is optimized with the Adam algorithm [62] using a constant learning rate of 0.001 and a batch size of 256. For each training iteration, a random step  $t$  is sampled independently for every point cloud in the training batch, and three independent standard Gaussian noises are generated, corresponding to the  $u'$ ,  $v'$ , and  $p'$  variables, to construct the tensor  $\mathbf{y}_{\text{noisy}}$  presented in Eq. (19). The corresponding noisy tensor  $\mathbf{y}_t$  is then computed using Eq. (19).

Once training is complete, the Diffusion PointNet generates the corresponding velocity and pressure variables for a geometry selected from the test set by simulating the reverse diffusion process, as shown in Fig. 3. For a point cloud  $\mathbf{x}$  representing a geometry from the test set, the reconstruction begins by generating three independent standard Gaussian noise fields to construct the tensor  $\mathbf{y}_T$ . The fields are then iteratively denoised from  $t = T$  to  $t = 0$  using

$$\mathbf{y}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( \mathbf{y}_t - \frac{\beta_t}{\sqrt{1-\alpha_t}} \mathcal{Y}_{t \rightarrow t-1} \right) + \sqrt{\beta_t} \mathbf{z}, \quad (25)$$

where  $\mathbf{z}$  denotes three independent standard Gaussian noise fields, and  $\mathbf{z} \in \mathbb{R}^{N \times n_{\text{CFD}}}$ . For  $t = 0$ , the term  $\sqrt{\beta_t} \mathbf{z}$  is omitted from Eq. (25) in accordance with the denoising diffusion probabilistic formulation [2].  $\mathcal{Y}_{t \rightarrow t-1}$  denotes the prediction of the Diffusion PointNet given the concatenation of  $\mathbf{x}$ ,  $\mathbf{y}_t$ , and  $\mathcal{S}_t$  at  $t$ . The final output  $\mathbf{y}_0$  at  $t = 0$  yields the predicted velocity and pressure fields ( $u'$ ,  $v'$ ,  $p'$ ) consistent with the input geometry. In our experiments, we set  $T = 1000$ .

### 3.3. Baseline PointNet

Now that the Flow Matching PointNet and Diffusion PointNet architectures have been described, the baseline PointNet can be introduced in a straightforward manner. The baseline PointNet for predicting fluid flow variables was first proposed by Kashefi et al. [29] and has since been adopted in several subsequent frameworks (see e.g., Refs. [30,64,65]). The version considered in this work is identical to that originally presented by Kashefi et al. [29]. In the baseline PointNet, the input consists solely of the spatial coordinates of the point cloud representing the geometry of the domain, and the output corresponds to the physical fields of interest, as shown in the bottom panel of Fig. 2. Following the notation introduced in Section 3.1, the input is given by  $\mathbf{x} \in \mathbb{R}^{N \times d}$ , and consequently, the input tensor is  $\mathcal{X} \in \mathbb{R}^{N \times d}$ . The output consists of the velocity components and pressure field, forming an output tensor  $\mathcal{Y} \in \mathbb{R}^{N \times n_{\text{CFD}}}$ . Hence, PointNet learns a direct mapping from  $\mathcal{X}$  to  $\mathcal{Y}$ . The loss function is the mean squared error and can be written as

$$\mathcal{L}_{\text{Baseline PointNet}} = \frac{1}{N \times n_{\text{CFD}}} \sum_{i=1}^N \left( (\tilde{u}'_i - u'_i)^2 + (\tilde{v}'_i - v'_i)^2 + (\tilde{p}'_i - p'_i)^2 \right), \quad (26)$$

where  $\tilde{u}'_i$ ,  $\tilde{v}'_i$ , and  $\tilde{p}'_i$  denote, respectively, the predicted  $u$ -velocity,  $v$ -velocity, and pressure fields obtained using the baseline PointNet. Training is performed using a batch size of 256 and the Adam optimizer [62] with a constant learning rate of 0.001.

### 3.4. PointNet

In Sections 3.1–3.3, we explained the role of PointNet [34] in each of the three proposed models, namely Flow Matching PointNet, Diffusion PointNet, and the baseline PointNet. In this part, we describe the architecture of PointNet itself. In general, the pair  $\mathcal{X}$  and  $\mathcal{Y}$  denote the input and output (or the quantity to be predicted) of PointNet [34], respectively. In the earlier sections, we clarified what  $\mathcal{X}$  and  $\mathcal{Y}$  represent for each of the three models. In summary, in Flow Matching PointNet,  $\mathcal{X} \in \mathbb{R}^{N \times (d+d_t+n_{\text{CFD}})}$  and  $\mathcal{Y} \in \mathbb{R}^{N \times n_{\text{CFD}}}$ ; in Diffusion PointNet,  $\mathcal{X} \in \mathbb{R}^{N \times (d+d_t+n_{\text{CFD}})}$  and  $\mathcal{Y} \in \mathbb{R}^{N \times n_{\text{CFD}}}$ ; and in the baseline PointNet,  $\mathcal{X} \in \mathbb{R}^{N \times d}$  and  $\mathcal{Y} \in \mathbb{R}^{N \times n_{\text{CFD}}}$ .

The original PointNet model developed for computer graphics applications consists of two branches: a classification branch and a segmentation branch [34]. In this work, we employ the segmentation branch of PointNet [34]. Fig. 2 illustrates the overall architecture of the segmentation branch of PointNet [29,34]. At a high level, the architecture can be viewed as an encoder–decoder framework. A key advantage of PointNet [34] and its variants (see e.g., Ref. [35]) is their ability to process unordered sets of points that constitute a point cloud. In particular, PointNet is invariant under the  $N!$  permutations of an input vector representing a point cloud with  $N$  points [34]. This permutation invariance is achieved through two main mechanisms. First, in both the encoder and

**Table 2**

Comparison between the performance of the Flow Matching PointNet, Diffusion PointNet, and baseline PointNet models for prediction of the velocity and pressure fields for the test set containing 222 unseen geometries. Number of sampling for the Flow Matching PointNet and Diffusion PointNet is 100. In numbers reported in the form  $\alpha_1 \pm \alpha_2$ ,  $\alpha_1$  denotes the mean error and  $\alpha_2$  represents the corresponding standard deviation. Note that  $\|\dots\|$  indicates the  $L^2$  norm.

	Flow Matching PointNet	Diffusion PointNet	Baseline PointNet
Average $\ \bar{u} - u\ /\ u\ $	1.39556E-2 $\pm$ 1.52319E-4	1.25420E-2 $\pm$ 1.14052E-4	3.19833E-2
Maximum $\ \bar{u} - u\ /\ u\ $	1.70648E-1 $\pm$ 9.32820E-3	1.70626E-1 $\pm$ 4.10404E-3	1.22576E-1
Minimum $\ \bar{u} - u\ /\ u\ $	5.86007E-3 $\pm$ 3.41877E-4	3.94060E-3 $\pm$ 2.04423E-4	1.12399E-2
Average $\ \bar{v} - v\ /\ v\ $	4.80994E-2 $\pm$ 3.42805E-4	6.16094E-2 $\pm$ 3.59036E-4	1.32724E-1
Maximum $\ \bar{v} - v\ /\ v\ $	5.44083E-1 $\pm$ 2.99690E-2	5.31338E-1 $\pm$ 1.79378E-2	4.25843E-1
Minimum $\ \bar{v} - v\ /\ v\ $	2.29497E-2 $\pm$ 1.92305E-3	3.19955E-2 $\pm$ 1.40590E-3	6.49750E-2
Average $\ \bar{p} - p\ /\ p\ $	3.66295E-2 $\pm$ 3.52680E-4	3.28203E-2 $\pm$ 2.33396E-4	1.08271E-1
Maximum $\ \bar{p} - p\ /\ p\ $	1.82688E-1 $\pm$ 7.92950E-3	1.65267E-1 $\pm$ 4.97312E-3	2.53676E-1
Minimum $\ \bar{p} - p\ /\ p\ $	1.63747E-2 $\pm$ 1.79350E-3	1.51713E-2 $\pm$ 6.72365E-4	3.41443E-2
Number of trainable parameters	3 554 179	3 554 179	3 554 179

decoder, each neural network layer uses a shared weight matrix and a shared bias vector across all neurons, forming what is known as a shared layer. Second, a symmetric function, one that is itself invariant to input ordering, is applied to encode global features from the input. As shown in Fig. 2, PointNet uses the max function as its symmetric function, as it has been shown to outperform alternatives such as the average function [34].

In Fig. 2, the notation  $(B_1, B_2)$  refers to two consecutive hidden shared layers with  $B_1$  and  $B_2$  neurons, whereas the triplet  $(B_1, B_2, B_3)$  follows the same convention. As shown in Fig. 2, the initial feature extraction stage consists of two shared MLP blocks with layer sizes (128, 128) and (128, 256, 2048). A symmetric max pooling operation is then applied to obtain a global feature vector of size 2048. This global feature is concatenated with an intermediate tensor of size  $N \times 128$ , producing a combined representation of dimension  $N \times 2176$ . The subsequent processing stage employs two additional shared MLP blocks with sizes (1024, 512, 256) and (256,  $n_{\text{CFD}}$ ). Consequently, the final output of PointNet is a tensor of size  $N \times n_{\text{CFD}}$ , as depicted in Fig. 2. After each shared MLP layer, a rectified linear unit activation function,

$$\sigma(\lambda) = \max(0, \lambda), \quad (27)$$

is applied and is followed by batch normalization, except for the final layer. In Flow Matching PointNet and Diffusion PointNet, the last layer does not use any activation function or batch normalization. In contrast, the baseline PointNet applies a sigmoid activation function in the final layer, defined as

$$\sigma(\lambda) = \frac{1}{1 + e^{-\lambda}}. \quad (28)$$

Further details about the architecture of PointNet can be found in Refs. [29,30,34].

## 4. Results and discussion

### 4.1. Overall results

In this section, we present a comprehensive analysis of the results using tables and histograms. Table 2 reports the relative pointwise errors ( $L^2$  norm) for predicting the velocity components  $u$  and  $v$  and the pressure  $p$  using the three models: Flow Matching PointNet, Diffusion PointNet, and the baseline PointNet. The results are computed across the test set, which contains 222 unseen geometries. Specifically, for each geometry in the test set, 100 samples are generated using both the Flow Matching PointNet and Diffusion PointNet models, meaning that 100 random noise realizations are conditioned on the same geometry. The reported values in Table 2 are expressed in the form  $\alpha_1 \pm \alpha_2$ , where  $\alpha_1$  denotes the mean error and  $\alpha_2$  represents the corresponding standard deviation. According to the values listed in Table 2, the average relative pointwise errors ( $L^2$  norm) over the 222 geometries for the  $u$ -velocity component are 1.40%, 1.26%, and 3.20% for the Flow Matching PointNet, Diffusion PointNet, and baseline PointNet models, respectively. For the  $v$ -velocity component, the corresponding errors are approximately 4.81%, 6.17%, and 13.28%. These results demonstrate that the Flow Matching and Diffusion models, particularly for the  $v$  variable, achieve substantially lower prediction errors. In comparison with the baseline PointNet, they are capable of reducing the relative pointwise error ( $L^2$  norm) to below 10%, which is a remarkable improvement from an engineering standpoint. Predicting the  $v$  variable is more challenging than predicting  $u$  because the flow field exhibits both positive and negative velocities in the  $y$ -direction due to vortex formation behind the cylinder. Although the  $u$  variable may also become negative in some regions, its spatial gradients are typically less steep than those of the  $v$  variable.

Examining the results for the pressure variable ( $p$ ) reveals a similar trend. Both the Flow Matching PointNet and Diffusion PointNet models reduce the average relative pointwise error ( $L^2$  norm) to below 10%, whereas the baseline PointNet exceeds this

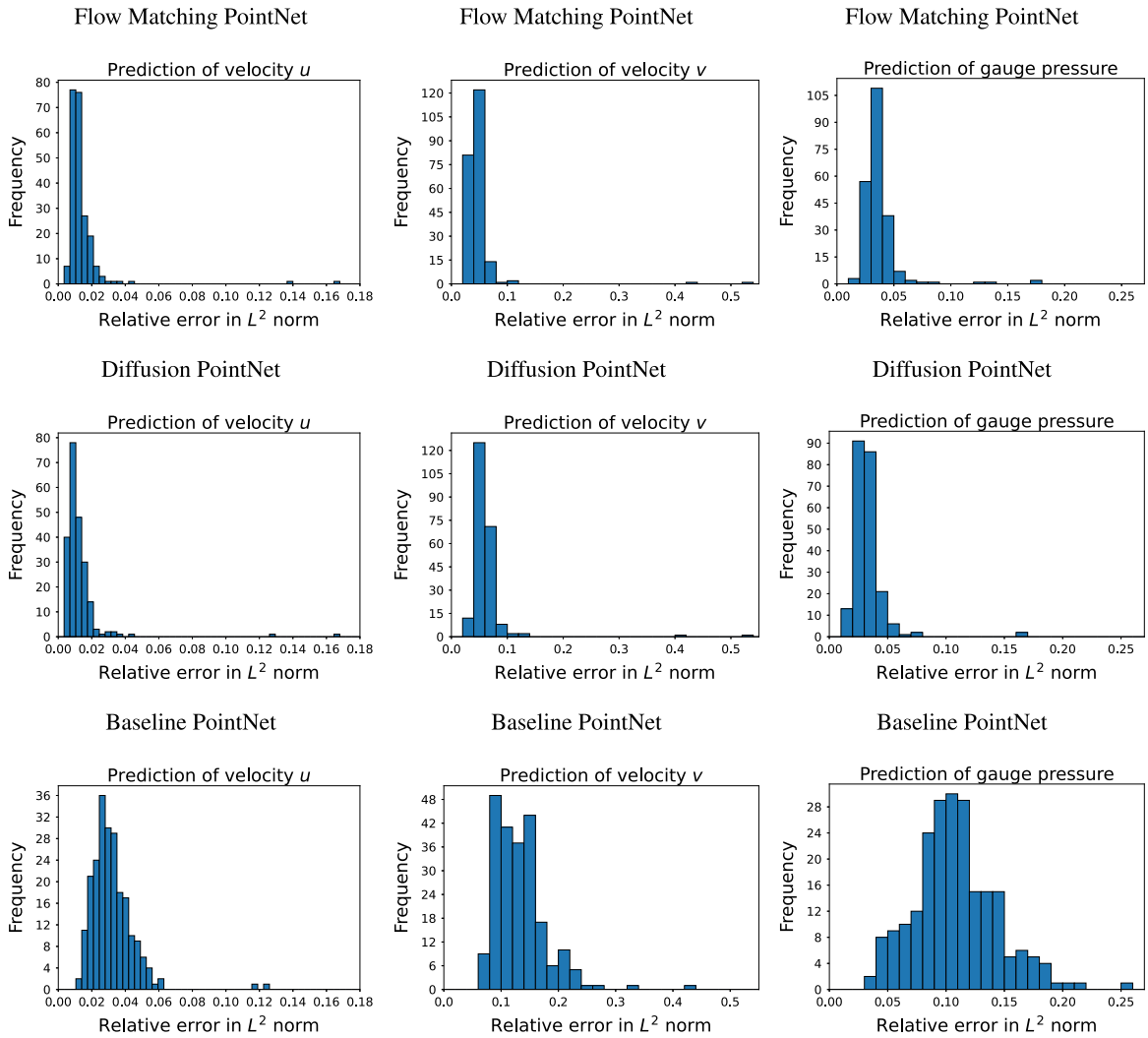


Fig. 4. Comparison of histograms of the relative pointwise error in  $L^2$  norm for the velocity and pressure fields predicted by the Flow Matching PointNet, Diffusion PointNet, and baseline PointNet models.

limit. According to Table 2, the relative pointwise errors ( $L^2$  norm) for predicting the pressure field are approximately 3.67%, 3.29%, and 10.83% for the Flow Matching PointNet, Diffusion PointNet, and baseline PointNet, respectively. The accuracy of the pressure field is particularly important because it directly influences the computation of lift and drag forces, which will be discussed in detail in Section 4.5. Furthermore, the uncertainty quantification capabilities of the proposed generative Flow Matching PointNet and Diffusion PointNet models provide valuable insights for practical engineering applications where safety factors must be considered in design processes. Finally, it is worth emphasizing that, as reported in Table 2, the number of trainable parameters, and consequently the model size, is identical across all three frameworks.

In addition to reporting the mean, minimum, and maximum errors across the 222 test geometries in Table 2, the distribution of these errors is illustrated in the histogram shown in Fig. 4. As observed in Fig. 4, Flow Matching PointNet and Diffusion PointNet demonstrate higher accuracy than the baseline PointNet in predicting the  $v$  and  $p$  variables. However, in all three models, there exist a few outlier cases for which the prediction error increases noticeably, as can be seen in Fig. 4. When comparing the Flow Matching PointNet and Diffusion PointNet in terms of prediction accuracy, the difference between the two is not substantial. Based on the overall results presented in Table 2 and Fig. 4, the Flow Matching PointNet model predicts the  $v$ -velocity component on average about 1.3% more accurately than the Diffusion PointNet model.

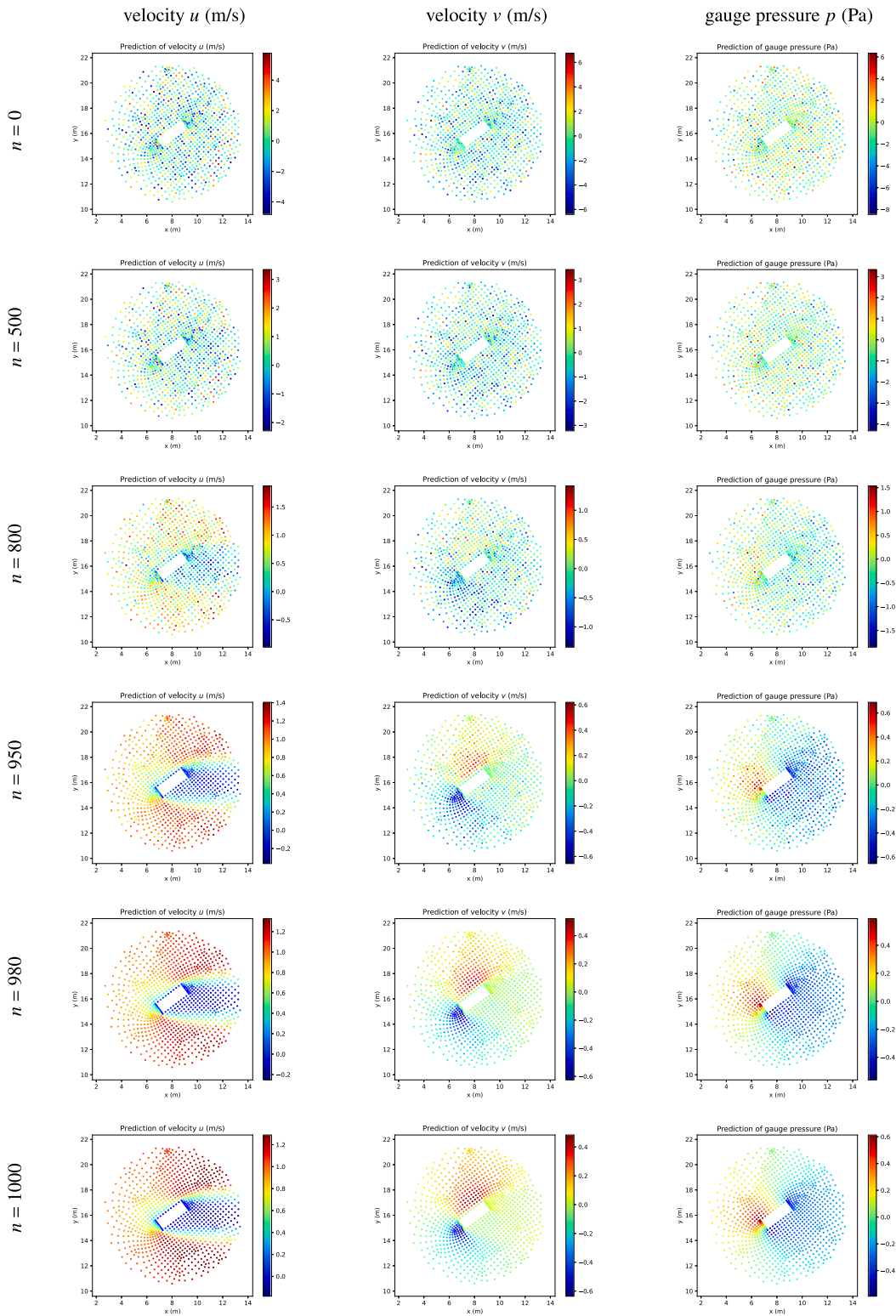


Fig. 5. Evolution of velocity and pressure fields during the reverse process of Flow Matching PointNet at  $n = 0$  (i.e.,  $n = N_s$ , standard Gaussian noise),  $n = 500$ ,  $n = 800$ ,  $n = 950$ ,  $n = 980$ , and  $n = 1000$  (i.e., the clean field).

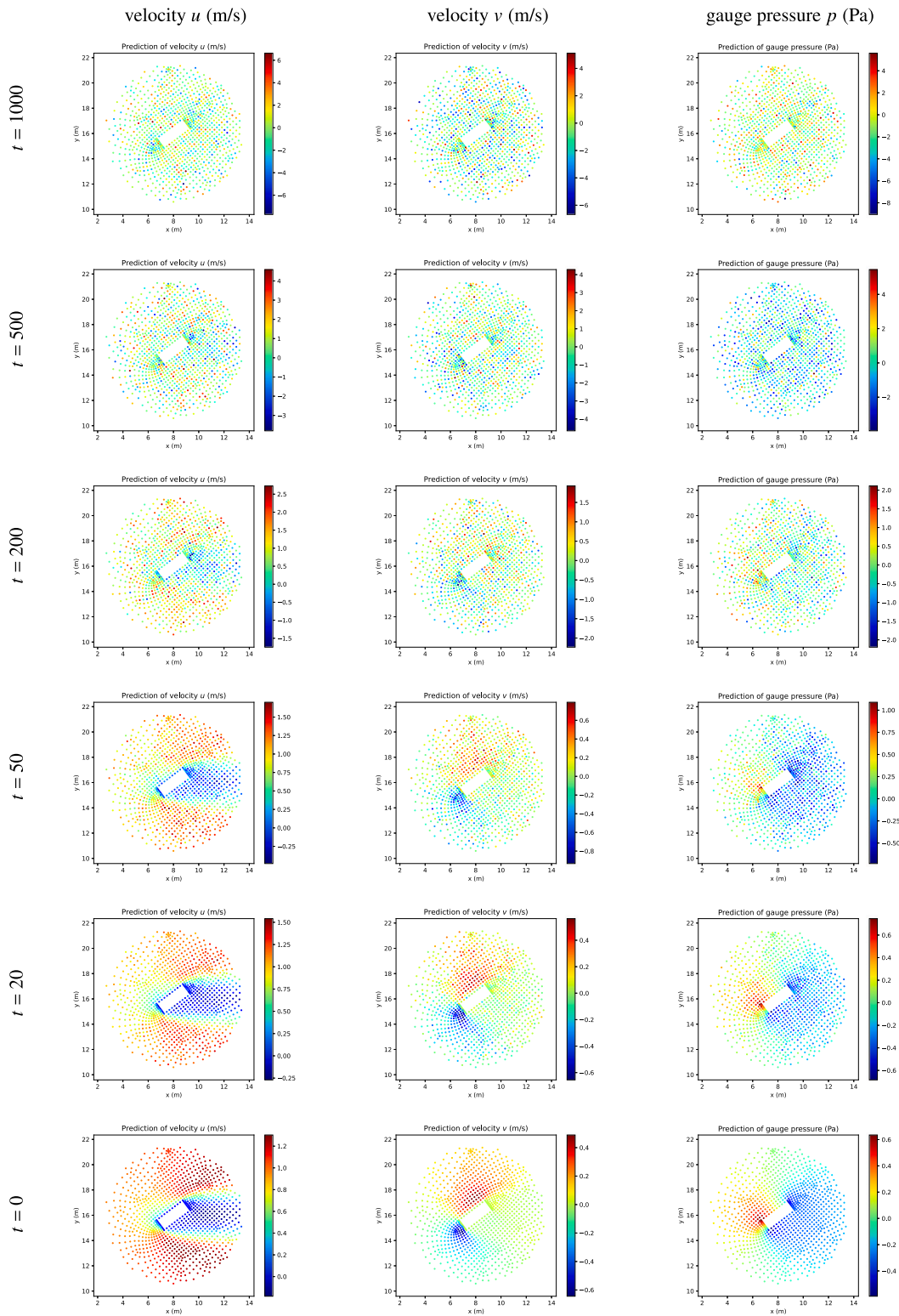
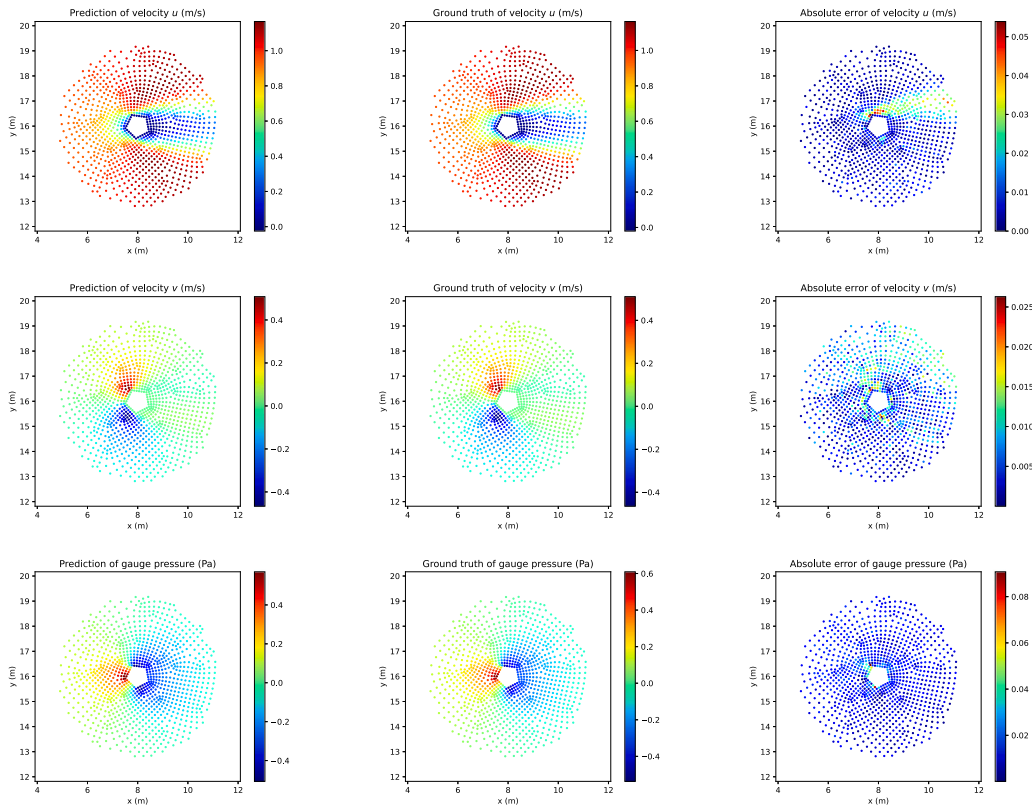


Fig. 6. Evolution of velocity and pressure fields during the reverse process of Diffusion PointNet at  $t = 1000$  (i.e.,  $t = T$ , standard Gaussian noise),  $t = 500$ ,  $t = 200$ ,  $t = 50$ ,  $t = 20$ , and  $t = 0$  (i.e., the clean field).



**Fig. 7.** The first set of examples comparing the ground truth to the predictions of Flow Matching PointNet for the velocity and pressure fields from the test set.

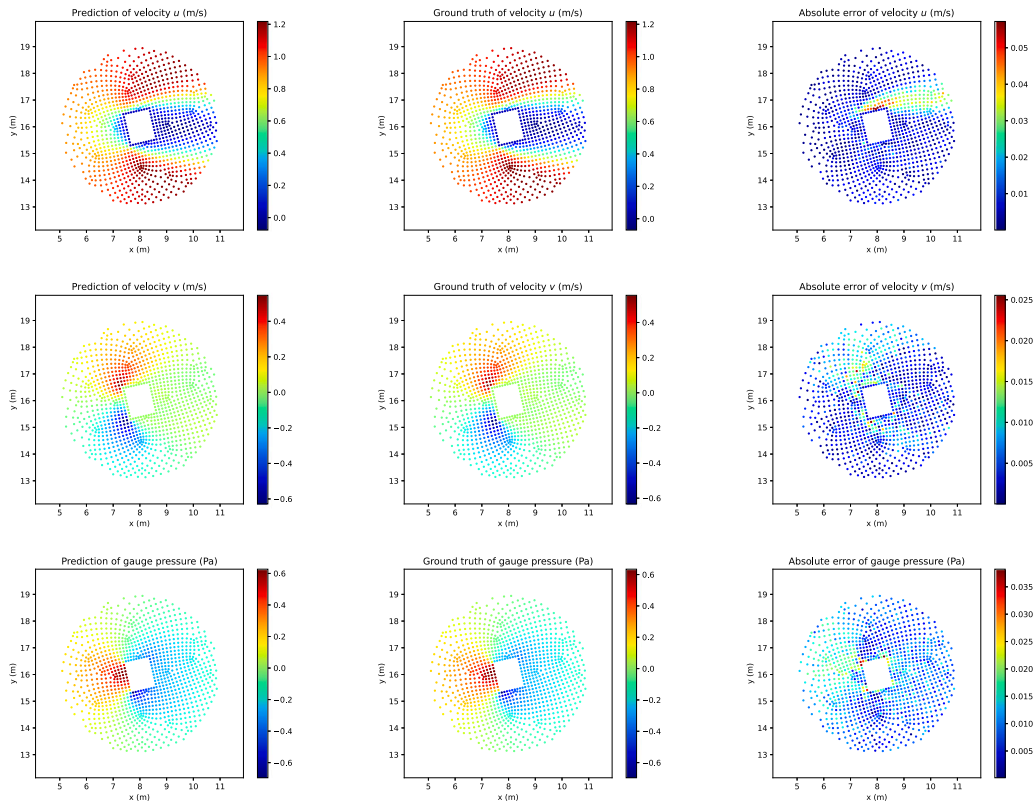
#### 4.2. Reverse generative process

Figs. 5 and 6 respectively illustrate the reverse processes of the Flow Matching PointNet and the Diffusion PointNet for fluid flow fields around a cylinder with a rectangular cross section, shown as a representative example from the test set. More specifically, Fig. 5 presents the velocity and pressure fields at different reverse steps, including  $n = 0$  (i.e., where all three fields are sampled from standard Gaussian noise),  $n = 500$ ,  $n = 800$ ,  $n = 950$ ,  $n = 980$ , and  $n = 1000$ , which corresponds to the predicted clean velocity and pressure fields. As can be seen from Fig. 5, at  $n = 500$ , the overall structure of the velocity and pressure fields is still largely visually obscured; however, the magnitudes of the variables are approximately within the range of  $[-3, 3]$ , compared to the case  $n = 0$ , where the variables lie roughly in the range of  $[-6, 6]$  due to the standard Gaussian noise. Similarly, at  $n = 800$ , the noise level decreases such that the flow patterns become partially visible. At  $n = 950$ , the velocity and pressure fields are largely reconstructed, and during the remaining reverse steps, they are further refined, with the predicted values converging to physically acceptable ranges. Quantitative error metrics have already been reported in Table 2.

Fig. 6 shows similar trends for the reverse process of the Diffusion PointNet, starting from  $t = 1000$  (i.e., the standard Gaussian noisy field) and progressing to  $t = 0$ , which corresponds to the predicted velocity and pressure fields. By comparing the outputs shown in Figs. 5 and 6, it can be observed that the underlying mechanisms of the Flow Matching PointNet and the Diffusion PointNet differ. For example, consider the velocity and pressure field predictions obtained by the Flow Matching PointNet at  $n = 950$  and by the Diffusion PointNet at  $t = 50$ , corresponding to the same number of reverse steps in the two models. Given the results shown in Figs. 5 and 6, a visual comparison indicates that the fluid flow patterns obtained by the Flow Matching PointNet exhibit a lower noise level and more closely resemble the predicted clean flow fields at this step (i.e.,  $n = 950$ ) than those obtained by the Diffusion PointNet at the corresponding step (i.e.,  $t = 50$ ). This qualitative observation highlights a fundamental difference in the reverse dynamics of the two models.

#### 4.3. Visual evaluation

Following the numerical analysis, a visual comparison of the results is conducted. Figs. 7–10 present four examples of the Flow Matching PointNet model predictions for velocity and pressure fields, alongside the corresponding ground truth and absolute pointwise error distributions. These figures include cylinders with pentagonal, rectangular, and elliptical cross-sections. Similarly,

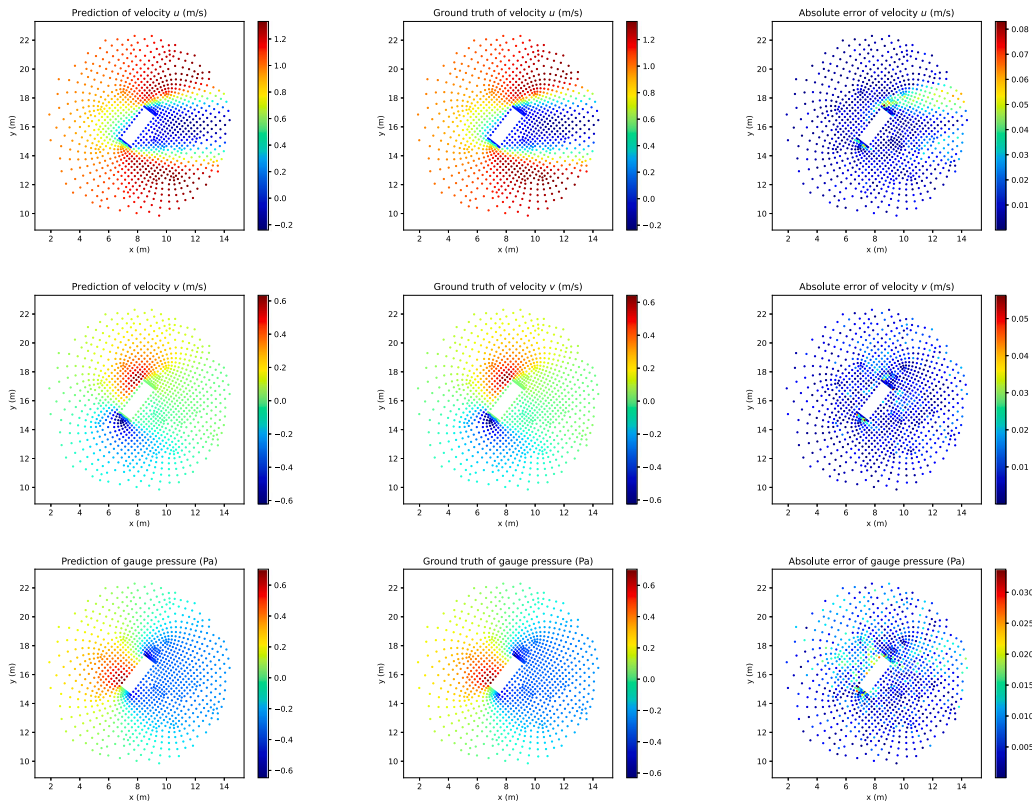


**Fig. 8.** The second set of examples comparing the ground truth to the predictions of Flow Matching PointNet for the velocity and pressure fields from the test set.

Figs. 11–14 illustrate the results of the Diffusion PointNet model for cylinders with hexagonal, triangular, rectangular, and elliptical cross-sections. In all cases, the predictions demonstrate excellent accuracy and a strong agreement with the ground truth. The majority of the absolute errors occur on or near the cylinder surface, where the geometric variations among different test cases are most significant. In Figs. 9 and 13, the  $x$  component of the velocity vector ( $u$ ) becomes negative, indicating the formation of secondary flow regions, and both the Flow Matching PointNet and Diffusion PointNet models successfully capture these phenomena. It should be noted that the test set is identical for all three models: Flow Matching PointNet, Diffusion PointNet, and the baseline PointNet. For visual diversity, however, we intentionally select cylinders with different cross-sectional shapes to provide a more comprehensive visual representation in Figs. 7–14. These results highlight the models' success in handling diverse geometries with varying shapes, orientations, and sizes. Another notable observation is that the computational domain size differs for each point cloud. For example, as can be seen in Fig. 7, the approximate minimum and maximum  $x$ -coordinates are 5 m and 11 m, respectively, while the corresponding  $y$ -coordinates range from 13 m to 19 m. In contrast, as shown in Fig. 9, the  $x$ -coordinates vary approximately from 2 m to 14 m, and the  $y$ -coordinates from 10 m to 22 m. A similar comparison can be made between Figs. 11 and 14. This adaptability reflects a fundamental strength of the PointNet architecture, which can process irregular geometries with varying domain sizes, unlike conventional CNNs that require fixed domain dimensions across the entire dataset. Additionally, the prediction of  $f_{\text{target}}$  by Flow Matching PointNet or  $y_{\text{noisy}}$  by Diffusion PointNet at each point depends on both the spatial coordinates of that point and the global geometry of the point cloud that contains it. Since the spatial ranges of the  $x$  and  $y$  coordinates vary across different point clouds in the dataset, accurately predicting  $f_{\text{target}}$  or  $y_{\text{noisy}}$  is a challenging task. The results presented here demonstrate that the proposed Flow Matching PointNet and Diffusion PointNet models successfully handle this challenge.

#### 4.4. Sampling diversity

As discussed in Section 3, when standard Gaussian noise conditioned on a given geometry (point cloud) is provided as input to the Flow Matching PointNet and Diffusion PointNet models, they generate distinct realizations of the velocity and pressure fields. Here, we compare several of these generated samples. Fig. 15 presents five samples generated by the Flow Matching PointNet model. To better illustrate the differences among the generated samples, instead of visualizing the velocity and pressure fields directly, we show the absolute pointwise error along with the corresponding ground truth. As can be seen from Fig. 15, it is evident that each sample differs from the others, even though the regions with the highest errors appear in roughly the same locations. However, the



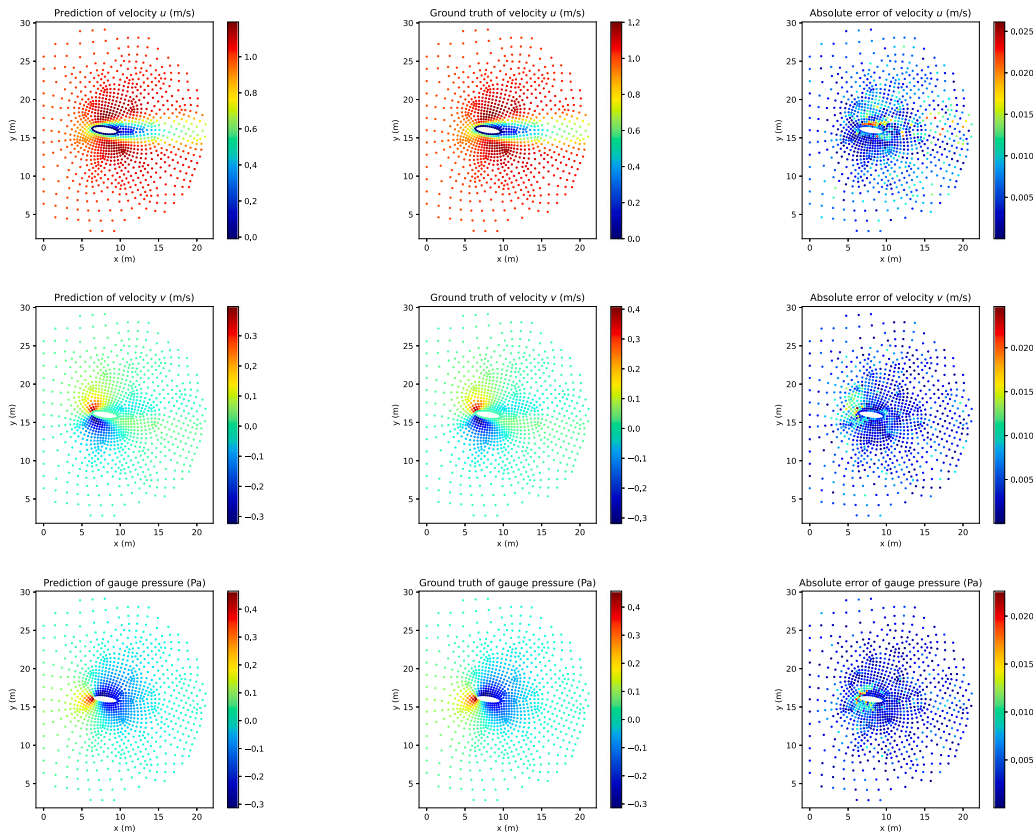
**Fig. 9.** The third set of examples comparing the ground truth to the predictions of Flow Matching PointNet for the velocity and pressure fields from the test set.

maximum absolute error for the  $y$  component of the velocity vector ( $v$ ) and the pressure field ( $p$ ) still varies noticeably across samples. For instance, in the case of the pressure field, the maximum absolute error occurs near the cylinder surface, with approximate values of 0.035, 0.04, 0.05, 0.06, and 0.07 Pa, as visualized in Fig. 15. In Section 4.5, we demonstrate that variations in these pressure values ultimately influence the predicted lift and drag forces induced by the pressure field. By sampling within the frameworks of Flow Matching PointNet and Diffusion PointNet, we estimate the maximum potential prediction error and assess model reliability. Consequently, when these models are applied in engineering design, such uncertainty estimates inform the selection of appropriate safety factors.

Similarly, Fig. 16 visualizes five samples generated by the Diffusion PointNet model. Differences among these samples are evident, particularly in the maximum absolute errors of both the velocity and pressure fields. For example, examining the generated  $u$ -velocity reveals maximum errors of approximately 0.04, 0.20, 0.25, and 0.35 m/s. Notably, in the second row of Fig. 16, or equivalently the sixth row of the same figure, the maximum absolute error of 0.35 occurs at only a single point among all 1024 points in the point cloud, while the remaining points exhibit near-zero absolute error. From a computer graphics perspective, this phenomenon resembles labeling inconsistencies encountered in part-segmentation tasks when using PointNet [34,38]. In such cases, more advanced models such as PointNet++ [35], DGCNN [66], and PointMLP [67] have been proposed and could potentially be employed in future studies for generative modeling in computational physics. These models place greater emphasis on capturing local features, thereby reducing the occurrence of outlier-like labels. Hence, these observations provide motivation for exploring more sophisticated point cloud-based architectures for generative modeling.

#### 4.5. Computation of pressure drag and lift forces

In addition to performing an error analysis over the entire computational domain, we now focus specifically on evaluating the predictions of the velocity and pressure fields along the surface of the object. A comparative analysis is conducted to determine which of the three models, Flow Matching PointNet, Diffusion PointNet, and baseline PointNet, achieves superior performance. In particular, Fig. 17 presents the velocity and pressure profiles on the surface of one test object, which corresponds to a cylinder with an elliptical cross-section. As illustrated in Fig. 17, since the flow is viscous, the components of the velocity vector (i.e.,  $u$  and  $v$ ) must vanish at the surface. It is evident from Fig. 17 that the predictions obtained by Flow Matching PointNet and Diffusion PointNet are more accurate than those of baseline PointNet for both the velocity and pressure fields. When comparing Flow Matching



**Fig. 10.** The fourth set of examples comparing the ground truth to the predictions of Flow Matching PointNet for the velocity and pressure fields from the test set.

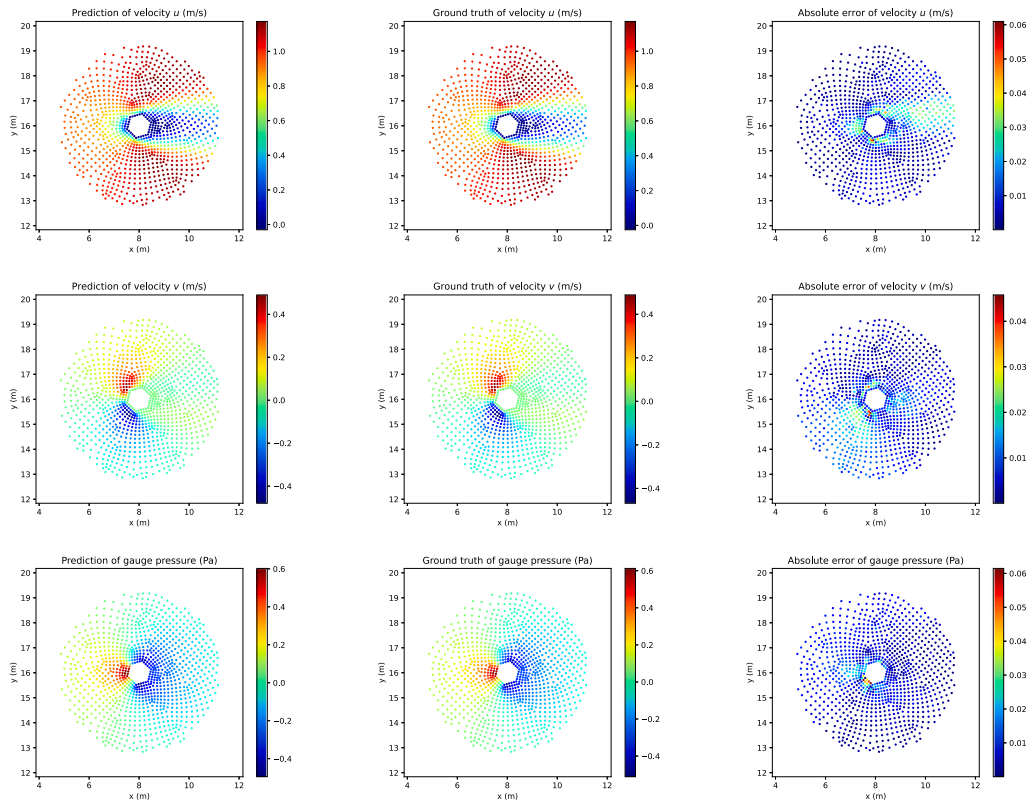
**Table 3**

Comparison between the performance of the Flow Matching PointNet, Diffusion PointNet, and baseline PointNet models for predicting pressure drag and pressure lift on a test set containing 222 unseen geometries. Lift and drag are computed from the predicted pressure fields. Number of samples is equal to 100. In numbers reported in the form  $\alpha_1 \pm \alpha_2$ ,  $\alpha_1$  and  $\alpha_2$  show the mean error and the corresponding standard deviation, respectively.

	Flow Matching PointNet	Diffusion PointNet	Baseline PointNet
Average absolute error of drag	1.65256E-2 ± 4.70271E-4	8.82595E-3 ± 3.33781E-4	1.04599E-1
Maximum absolute error of drag	1.32432E-1 ± 2.13175E-2	1.34004E-1 ± 9.27641E-3	3.48611E-1
Minimum absolute error of drag	1.40302E-4 ± 1.30478E-4	4.42719E-5 ± 4.15782E-5	1.06129E-2
Average absolute error of lift	1.28349E-2 ± 4.41792E-4	1.06831E-2 ± 3.10677E-4	4.77564E-2
Maximum absolute error of lift	6.84403E-2 ± 1.65813E-2	1.17770E-1 ± 1.18856E-2	1.90848E-1
Minimum absolute error of lift	8.33265E-5 ± 9.27361E-5	6.78547E-5 ± 5.77296E-5	1.75967E-4
Number of trainable parameters	3 554 179	3 554 179	3 554 179

PointNet and Diffusion PointNet, both models demonstrate excellent performance; however, the results shown in Fig. 17 indicate that Flow Matching PointNet provides slightly more accurate predictions of the zero-velocity condition in the y-direction compared to Diffusion PointNet.

Following the comparison of velocity and pressure distributions on the cylinder surface, performed on one of the test set, we now turn our attention to the prediction of drag and lift forces derived from the pressure field. Accurate estimation of drag and lift forces has always been of great importance in engineering and plays a critical role in aerodynamic applications and geometric design processes. Table 3 summarizes a quantitative comparison of the three models of Flow Matching PointNet, Diffusion PointNet, and baseline PointNet. According to Table 3, an examination of the average absolute error in drag reveals that the baseline PointNet model exhibits an error of approximately 10%, whereas the Flow Matching PointNet and Diffusion PointNet achieve substantially lower errors of below 1.7% and 0.9%, respectively. These results indicate that both Flow Matching PointNet and Diffusion PointNet considerably outperform the baseline PointNet model, yielding more reliable predictions for drag-based design tasks. For lift prediction, a similar trend is observed. The Diffusion PointNet achieves the highest accuracy, followed by the Flow



**Fig. 11.** The first set of examples comparing the ground truth to the predictions of Diffusion PointNet for the velocity and pressure fields from the test set.

Matching PointNet and the baseline PointNet. The Flow Matching PointNet also provides more accurate predictions than the baseline model. In this case, the prediction error of the baseline PointNet for lift is approximately 4.8%, which is lower than its corresponding drag prediction error.

Finally, as discussed in Section 4.1, 100 samples are drawn by injecting standard Gaussian noise for each geometry from the test set for both Flow Matching PointNet and Diffusion PointNet. Accordingly, the values reported in Table 3 in the form of  $\alpha_1 \pm \alpha_2$  indicate  $\alpha_1$  as the average value and  $\alpha_2$  as the standard deviation. This representation enables the quantification of uncertainty in the predictions, which is particularly valuable for design-oriented applications. For instance, the average over the maximum absolute error of drag for Flow Matching PointNet is approximately 13.25%, accompanied by a standard deviation of 2.14%, as listed in Table 3. Similar information for the maximum absolute error of lift using Diffusion PointNet is provided in Table 3. In contrast, the baseline PointNet model does not provide such uncertainty intervals for error measurements, which represents a key limitation of the model. In summary, both Flow Matching PointNet and Diffusion PointNet not only deliver more accurate predictions but also offer valuable uncertainty information about the model outputs that can assist design engineers, especially when safety margins or worst-case design scenarios must be considered.

#### 4.6. Robustness

Another parameter we aim to investigate is the case in which the target geometry is partially corrupted, meaning that a fraction of the points in a point cloud is missing, such as 5%, 10%, or 15% of the total points. This scenario is realistic from both industrial and experimental perspectives. While training datasets for deep learning models are typically geometrically complete, with all points available or reconstructed prior to training, real-time applications during inference might involve incomplete data.

The objective of this subsection is to evaluate how robust the three deep learning frameworks, Flow Matching PointNet, Diffusion PointNet, and baseline PointNet, are in handling such incomplete data. Table 4, together with Figs. 18 and 19, illustrate the results of this analysis. When 5%, 10%, and 15% of the points in each point cloud (e.g., 15% of 1024 points) are randomly removed, the corresponding errors in the  $x$ - and  $y$ -components of the velocity vector (i.e.,  $u$  and  $v$ ) and in the pressure variable (i.e.,  $p$ ) are listed in Table 4. As expected, increasing the percentage of missing points leads to higher prediction errors. Among the three models considered, the Flow Matching PointNet demonstrates the highest robustness, followed by the Diffusion PointNet, while the baseline PointNet exhibits the lowest robustness. When 10% of the points are removed, the average relative pointwise error ( $L^2$  norm) for

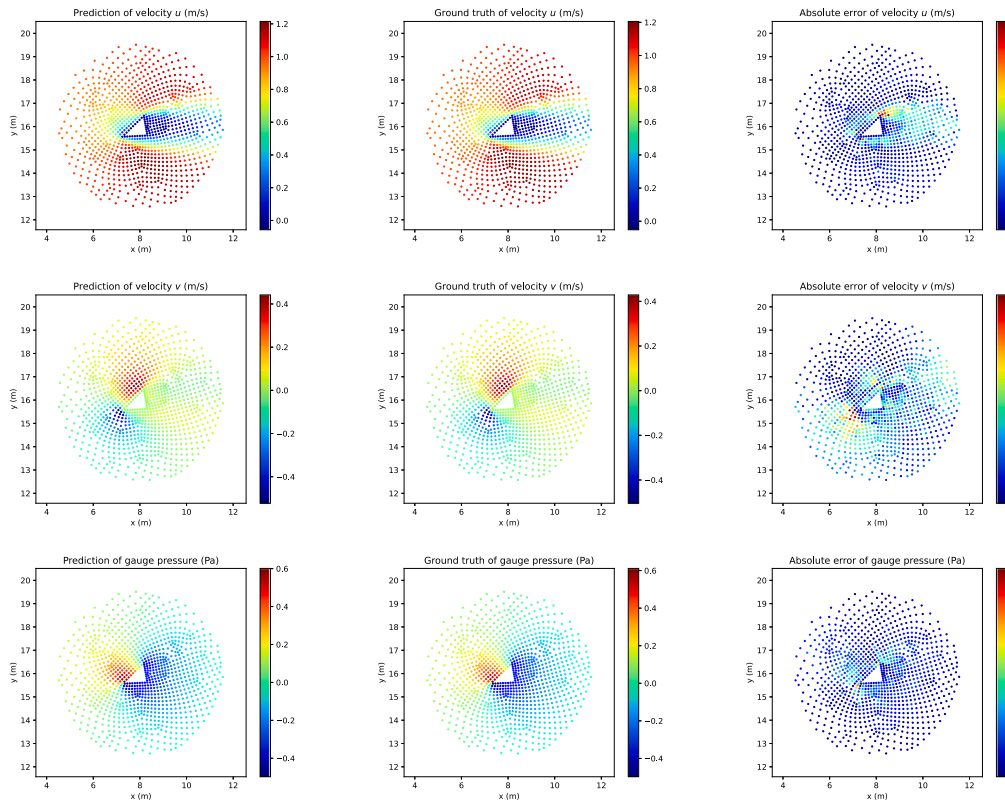


Fig. 12. The second set of examples comparing the ground truth to the predictions of Diffusion PointNet for the velocity and pressure fields from the test set.

Table 4

Robustness test of Flow Matching PointNet, Diffusion PointNet, and baseline PointNet by evaluating the error in predicted velocity and pressure fields when a certain percentage of points is randomly removed from point clouds in the test set, containing 222 unseen geometries. The  $L^2$  norm is indicated by  $|| \dots ||$ .

Model	Percentage	5%	10%	15%
Flow Matching PointNet	Average $  \bar{u} - u  /  u  $	1.46687E-2	1.81087E-2	2.27024E-2
	Average $  \bar{v} - v  /  v  $	5.25402E-2	6.84943E-2	8.20785E-2
	Average $  \bar{p} - p  /  p  $	4.27335E-2	5.48017E-2	6.71761E-2
Diffusion PointNet	Average $  \bar{u} - u  /  u  $	1.73480E-2	2.15721E-2	2.99512E-2
	Average $  \bar{v} - v  /  v  $	7.35410E-2	8.53514E-2	1.09470E-1
	Average $  \bar{p} - p  /  p  $	4.15597E-2	5.14183E-2	6.95332E-2
Baseline PointNet	Average $  \bar{u} - u  /  u  $	3.32840E-2	3.59088E-2	4.23352E-2
	Average $  \bar{v} - v  /  v  $	1.39502E-1	1.48369E-1	1.71268E-1
	Average $  \bar{p} - p  /  p  $	1.10115E-1	1.14691E-1	1.23513E-1

predicting the  $v$  variable is approximately 6.9% for Flow Matching PointNet, 8.6% for Diffusion PointNet, and 14.9% for baseline PointNet. Notably, the error for baseline PointNet exceeds 10%, which is generally considered unacceptable in engineering design contexts. When 15% of the points are removed, the prediction error for the  $v$  increases to about 8.3% for Flow Matching PointNet, nearly 11% for Diffusion PointNet, and over 17% for baseline PointNet. Under these conditions, Flow Matching PointNet is the only model maintaining an error below 10%.

Figs. 18 and 19 show, respectively, the predictions obtained by the Flow Matching PointNet and Diffusion PointNet for the  $u$ ,  $v$ , and  $p$  variables when 15% of the points are removed. It is important to note that some of the missing points are located on the surface of the cylinder. Mathematically, the velocity and pressure fields are functions of the cylinder's cross-sectional geometry. Consequently, the learning process of the deep network inherently depends on this geometry. Kashefi et al. [29] demonstrated that such boundary points are, in fact, critical within the PointNet framework and play an essential role in geometric deep learning. This finding further highlights the importance of robustness and underscores the superior performance of the Flow Matching PointNet in handling geometries with missing boundary points.

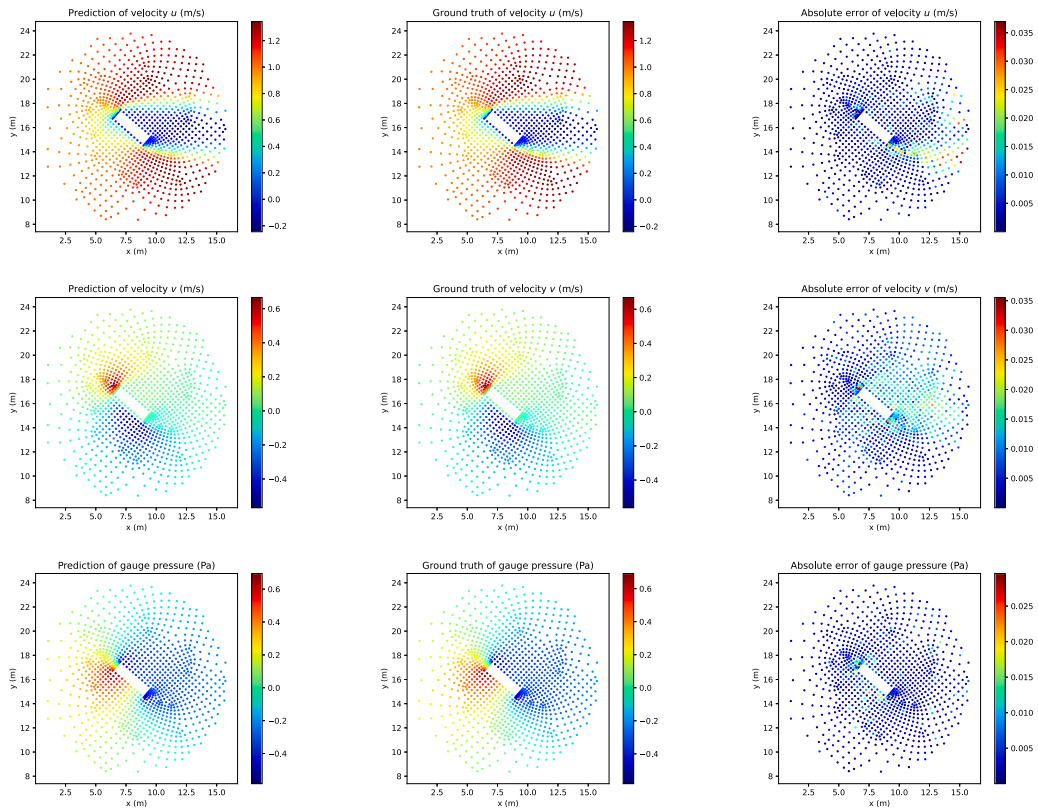


Fig. 13. The third set of examples comparing the ground truth to the predictions of Diffusion PointNet for the velocity and pressure fields from the test set.

Table 5

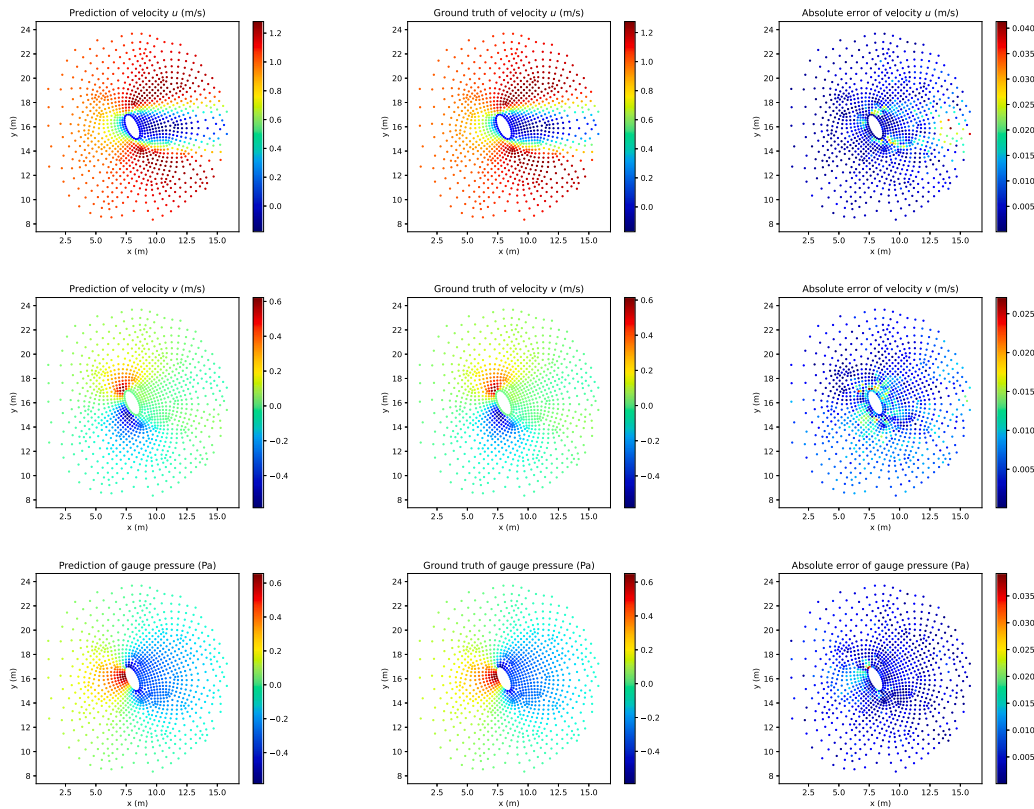
Comparison of the performance of the Flow Matching PointNet model for sampling flows using different numbers of steps ( $N_\tau$ ) in the prediction of velocity and pressure fields on a test set containing 222 unseen geometries. The number of samples generated by Flow Matching PointNet is 100. For values reported in the form  $\alpha_1 \pm \alpha_2$ ,  $\alpha_1$  denotes the mean error and  $\alpha_2$  denotes the corresponding standard deviation.

$N_\tau$	Average $\ \bar{u} - u\ /\ u\ $	Average $\ \bar{v} - v\ /\ v\ $	Average $\ \bar{p} - p\ /\ p\ $
1000	$1.39556\text{E-}2 \pm 1.52319\text{E-}4$	$4.80994\text{E-}2 \pm 3.42805\text{E-}4$	$3.66295\text{E-}2 \pm 3.52680\text{E-}4$
100	$1.28839\text{E-}2 \pm 1.60753\text{E-}4$	$4.66403\text{E-}2 \pm 4.23433\text{E-}4$	$3.38772\text{E-}2 \pm 4.02414\text{E-}4$
50	$1.23977\text{E-}2 \pm 1.55730\text{E-}4$	$4.65132\text{E-}2 \pm 3.97360\text{E-}4$	$3.30066\text{E-}2 \pm 4.03075\text{E-}4$
25	$1.20465\text{E-}2 \pm 1.29122\text{E-}4$	$4.64750\text{E-}2 \pm 3.91735\text{E-}4$	$3.36662\text{E-}2 \pm 2.86857\text{E-}4$
20	$1.20174\text{E-}2 \pm 1.30897\text{E-}4$	$4.64054\text{E-}2 \pm 3.82539\text{E-}4$	$3.41983\text{E-}2 \pm 3.99324\text{E-}4$
10	$1.20473\text{E-}2 \pm 1.13537\text{E-}4$	$4.88882\text{E-}2 \pm 3.45820\text{E-}4$	$3.79881\text{E-}2 \pm 3.32502\text{E-}4$
5	$1.31928\text{E-}2 \pm 1.04309\text{E-}4$	$5.60721\text{E-}2 \pm 2.99492\text{E-}4$	$5.18359\text{E-}2 \pm 2.94535\text{E-}4$

#### 4.7. Impact of the number of steps on inference in Flow Matching PointNet

In this section, we investigate the effect of the number of steps ( $N_\tau$ ) on inference in Flow Matching PointNet (see Eqs. (12) and (17)). In previous sections, we set  $N_\tau = 1000$ . The main reason for this choice was to ensure a fair comparison between the performance of Flow Matching PointNet and Diffusion PointNet (e.g., see Figs. 5 and 6). However, when considering Flow Matching PointNet independently, we perform an ablation study on the number of steps at inference time. Ideally, a smaller number of steps leads to faster sampling and inference, provided that prediction accuracy is maintained. Table 5 presents the results of this ablation study. Based on the results in Table 5, the choices  $N_\tau = 20$  or  $N_\tau = 25$  appear to be ideal and yield slightly improved accuracy relative to using  $N_\tau = 1000$ . We conjecture that this improvement is due to the reduced accumulation of solver-induced numerical errors (e.g., finite-precision effects) that arise when a large number of updates are applied.

Generally speaking, one of the advantages of the baseline PointNet compared to Flow Matching PointNet and Diffusion PointNet is that it requires only a single step to generate the output, resulting in faster inference time. This is particularly important for real-time applications and for accelerating shape optimization through faster exploration of the parameter space. Specifically, for



**Fig. 14.** The fourth set of examples comparing the ground truth to the predictions of Diffusion PointNet for the velocity and pressure fields from the test set.

Flow Matching PointNet, we showed that 25 steps are sufficient in the current application; however, both Flow Matching PointNet and Diffusion PointNet remain slower at inference time compared to the baseline PointNet. To address this limitation, and from a broader computer science perspective, several successful methods have been proposed to mitigate the limitation of requiring a large number of steps at inference time in flow matching and diffusion models. These include diffusion model distillation [68], consistency models [69], and fast solvers for guided sampling [70], all of which enable generation with only a few steps. Hence, a direction for our future work is to develop alternative versions of Flow Matching PointNet and Diffusion PointNet that incorporate these techniques.

#### 4.8. Correlation between predictive variance and local error

In this part of the article, we address the question of whether higher variance across samples generated by Flow Matching PointNet and Diffusion PointNet correlates with higher local prediction error. To this end, we employ the Pearson correlation coefficient. For each point cloud in the test set, we generate 100 samples. For each spatial point, we compute the mean prediction across samples, as well as the corresponding variance across these samples, which serves as a measure of predictive uncertainty. We also compute the pointwise absolute error between the mean prediction and the ground truth. The Pearson correlation coefficient is then evaluated across all spatial points within each point cloud, measuring the relationship between predictive variance and local error. A positive correlation indicates that regions with higher predicted uncertainty tend to correspond to larger local prediction errors, suggesting that the uncertainty estimates are informative.

The results of this analysis are presented as histograms for the 222 unseen geometries in the test set in Fig. 20, for the velocity and pressure variables in both Flow Matching PointNet and Diffusion PointNet. More specifically, the positive ratios (defined as the fraction of cases with a positive Pearson correlation coefficient) are 0.928, 0.977, and 1.000 for the  $u$ ,  $v$ , and  $p$  variables, respectively, in Flow Matching PointNet. Similarly, these ratios are 1.000, 0.982, and 0.990 for Diffusion PointNet. Hence, a large fraction of geometries exhibits positive correlation. Additionally, the average Pearson correlation coefficients across the 222 test geometries are 0.318, 0.254, and 0.563 for the  $u$ ,  $v$ , and  $p$  variables, respectively, in Flow Matching PointNet. Correspondingly, these values are 0.549, 0.283, and 0.319 for Diffusion PointNet. Based on this information and Fig. 20, it is concluded that there is a clear positive correlation between the variance across samples and the local prediction error, indicating that regions with higher predictive variance tend to coincide with larger local prediction errors.

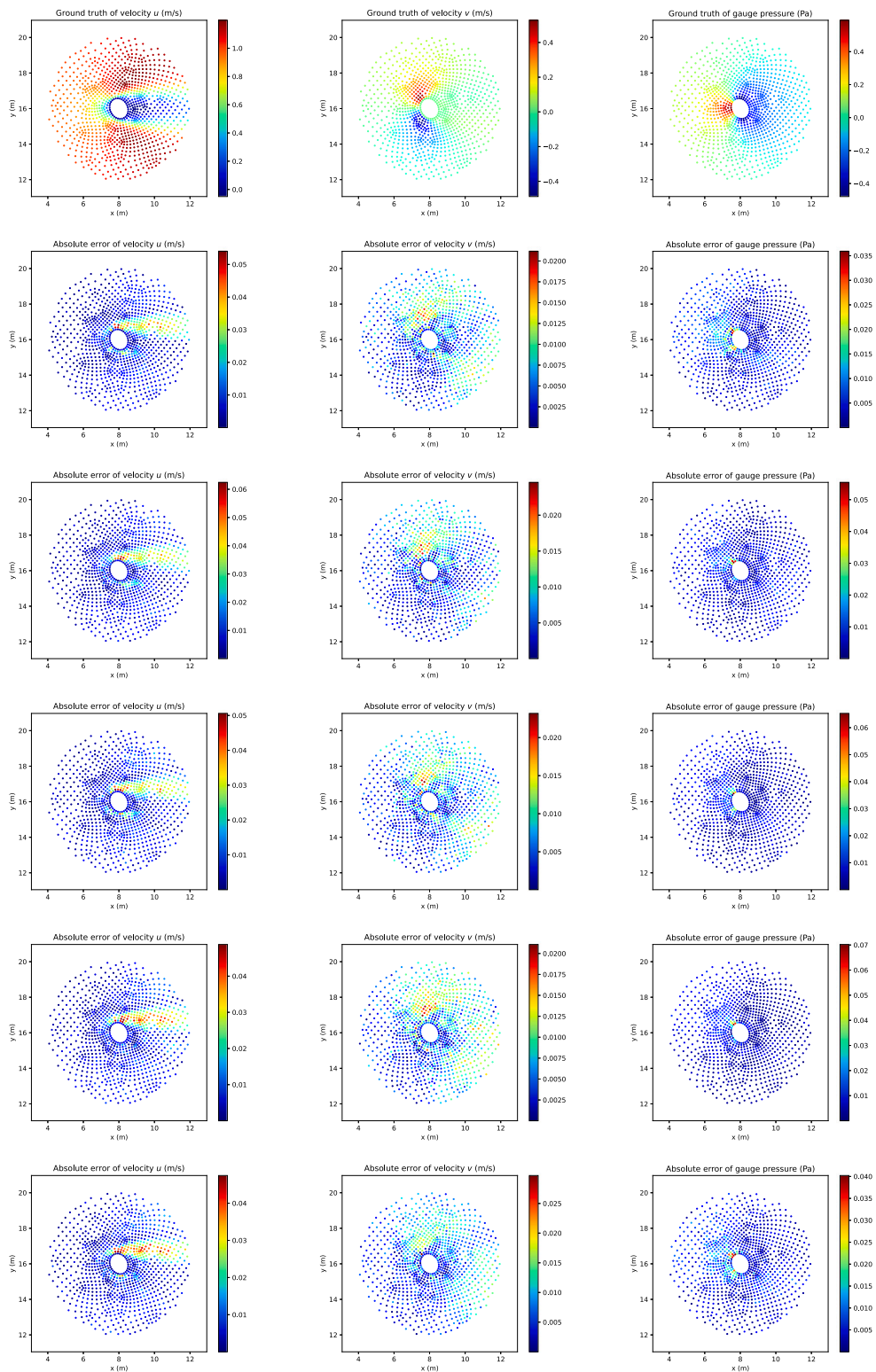


Fig. 15. Flow Matching PointNet realizations; the first row shows the ground truth, and the remaining rows show the absolute pointwise error for five different realizations.

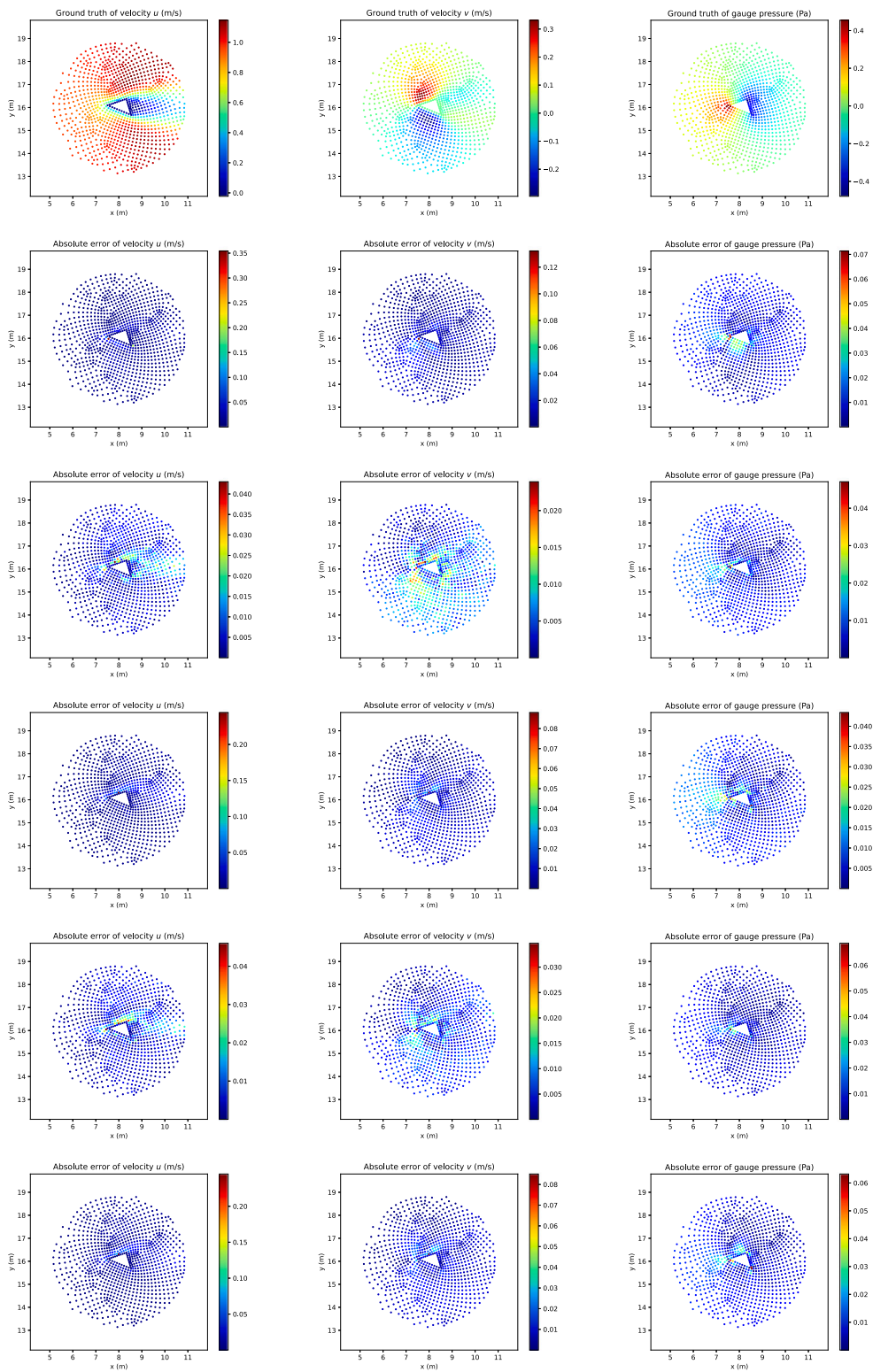
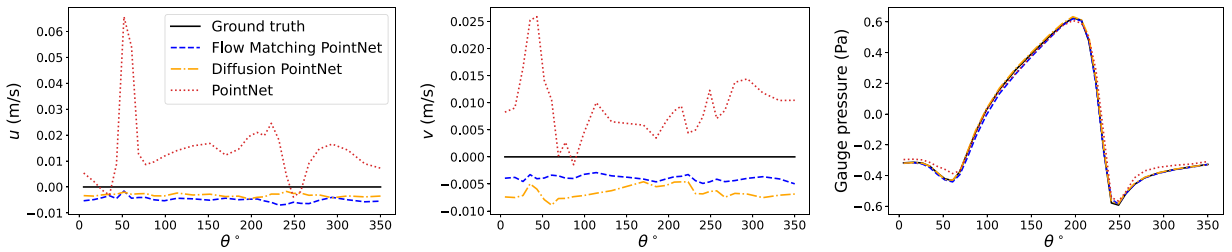
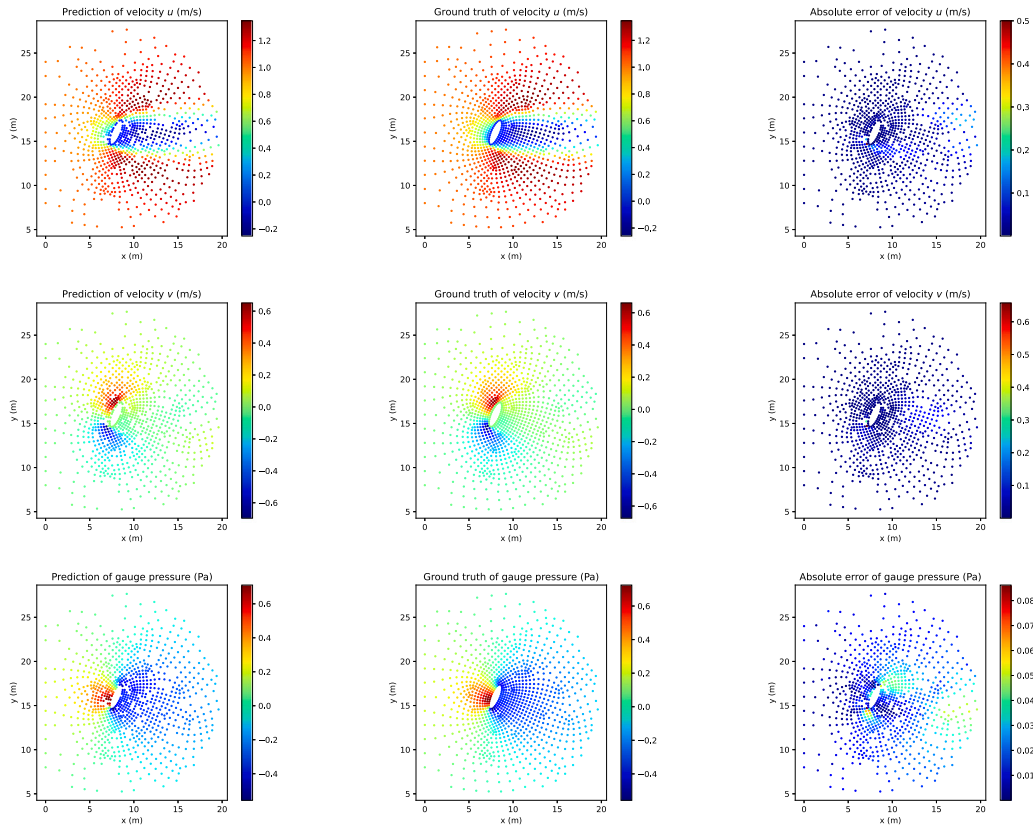


Fig. 16. Diffusion PointNet realizations; the first row shows the ground truth, and the remaining rows show the absolute pointwise error for five different realizations.



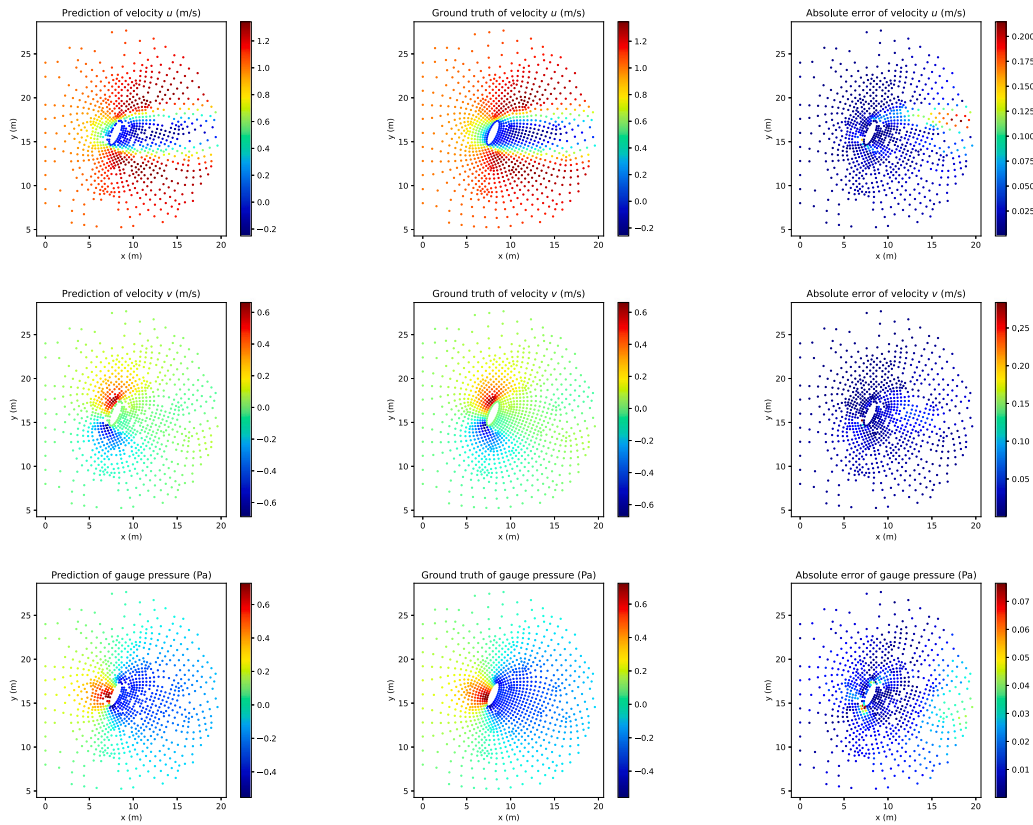
**Fig. 17.** An example of velocity and pressure distributions predicted by the Flow Matching PointNet, Diffusion PointNet, and baseline PointNet models along the surface of a cylinder with an elliptical cross-section from the test set. The angle is measured counterclockwise from the positive  $x$ -axis by connecting each surface point to the center of the ellipse at (8, 16).



**Fig. 18.** An example comparing the ground truth to the predictions of Flow Matching PointNet for the velocity and pressure fields from the test set, where 15% of points are randomly removed from the point clouds, and an imperfect geometry of the test set is fed into the network. Note that the ground truth is shown with all 1024 points.

## 5. Summary and outlook

In this study, we introduced, for the first time, the Flow Matching PointNet and Diffusion PointNet, which integrated the formulations of flow matching [1] and diffusion models [2] into PointNet [29,30,34] to generate velocity and pressure fields of fluid flows conditioned on the geometry of the computational domain. In these generative frameworks, irregular geometries were represented as point clouds. Independent standard Gaussian noise fields, corresponding to the velocity components in the  $x$ - and  $y$ -directions as well as the pressure field, were concatenated with the spatial coordinates. In addition, sinusoidal positional encodings were concatenated with the spatial coordinates and noise fields to form the input features. A forward processing phase was then initiated, during which the Diffusion PointNet was trained to predict the noise components added to the velocity and pressure fields, while the Flow Matching PointNet, in a similar manner, predicted the derivatives of the noise with respect to a temporal variable. After training, inference was performed via a reverse processing phase, in which independent standard Gaussian noise

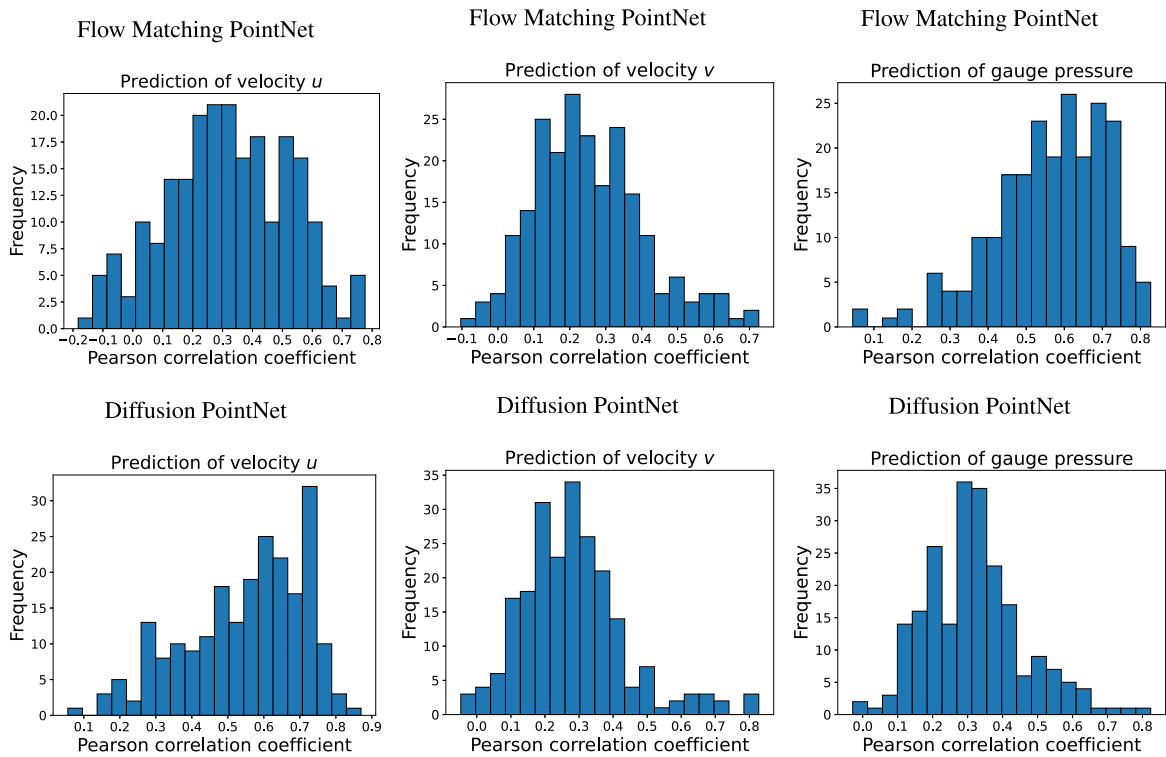


**Fig. 19.** An example comparing the ground truth to the predictions of Diffusion PointNet for the velocity and pressure fields from the test set, where 15% of points are randomly removed from the point clouds, and an imperfect geometry of the test set is fed into the network. Note that the ground truth is shown with all 1024 points.

fields associated with the velocity components and pressure were conditioned on an unseen geometry to generate the corresponding velocity and pressure fields. In Flow Matching PointNet, we used the explicit Euler numerical scheme during the reverse process.

To demonstrate the effectiveness of the proposed frameworks, the classical problem of fluid flow around a cylinder was employed as a benchmark. In this setup, the cross-section of the cylinder varied across different geometries, resulting in corresponding variations in the velocity and pressure distributions. These variations constituted the dataset that was used for training. The analysis of the predictions from the Flow Matching PointNet and Diffusion PointNet models revealed that their performance was significantly superior to that of the baseline PointNet framework. In particular, employing the Flow Matching PointNet and Diffusion PointNet models reduced the average relative pointwise error in the  $L^2$  norm to below 10%, whereas the error for the baseline PointNet model exceeded 10%. Flow Matching PointNet and Diffusion PointNet demonstrated improved accuracy in predicting pressure-induced lift and drag forces. Furthermore, since the proposed frameworks generated predictions based on standard Gaussian noise inputs, multiple realizations were produced for any given geometry, enabling estimation of the uncertainty in model predictions. This capability was especially valuable for engineering design and safety factor considerations. Specifically, we demonstrated the uncertainty range associated with the maximum possible errors in pressure-induced drag and lift predictions. Additionally, we investigated the robustness of the Flow Matching PointNet and Diffusion PointNet models in cases where the geometry was incomplete (i.e., when a percentage of points in a point cloud was missing). The results showed that both models exhibited greater robustness compared to the baseline PointNet, with the Diffusion PointNet capable of maintaining the average relative pointwise error in the  $L^2$  norm of the velocity and pressure fields below 10% even when up to 15% of the point cloud data were missing. Finally, although both the Flow Matching PointNet and Diffusion PointNet significantly outperformed the baseline PointNet, the overall difference between the two was not substantial. The Diffusion PointNet performed slightly better in predicting pressure, leading to more accurate pressure-induced drag and lift computations, whereas the Flow Matching PointNet demonstrated greater robustness when the geometry was partially missing.

We provide several directions for future extensions of the current research. One potential avenue is to investigate turbulent flow regimes, where uncertainty in the solution plays a critical role from both scientific and engineering perspectives [13,39]. Another direction involves studying time-dependent flows, particularly those in which the geometry of the computational domain itself varies with time. Moreover, the proposed generative models can be further applied to the super-resolution and reconstruction of turbulent flow fields on unstructured grids. Extending the Flow Matching PointNet and Diffusion PointNet framework to three-dimensional



**Fig. 20.** Histograms of the Pearson correlation coefficient between predictive variance and pointwise absolute error for the 222 unseen geometries in the test set, computed separately for the  $u$ ,  $v$ , and  $p$  variables. Results are shown for both Flow Matching PointNet and Diffusion PointNet.

cases with industrial applications, such as predicting Reynolds-Averaged Navier–Stokes (RANS) turbulent flow on the surfaces of aircraft wings and vehicles, is another important direction for future research. Additionally, while we introduced novel flow matching and diffusion models conditioned on the geometry of the computational domain, these models can be further extended to a hybrid conditioning approach, simultaneously conditioned on both geometry and textual prompts, thereby enabling their integration into foundation and large language/vision models [71–76] for computational physics.

### 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.

### Acknowledgment

The author wishes to thank the Stanford Research Computing Center. Additionally, the author thanks the reviewers for their constructive comments and suggestions, which have helped improve the quality of this work.

### Data availability

The Python codes and data are publicly accessible on the following GitHub repository: [https://github.com/Ali-Stanford/Diffusion\\_Flow\\_Matching\\_PointNet\\_CFD](https://github.com/Ali-Stanford/Diffusion_Flow_Matching_PointNet_CFD).

### References

- [1] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, Matthew Le, Flow matching for generative modeling, in: *The Eleventh International Conference on Learning Representations*, 2023.
- [2] Jonathan Ho, Ajay Jain, Pieter Abbeel, Denoising diffusion probabilistic models, *Adv. Neural Inf. Process. Syst.* 33 (2020) 6840–6851.
- [3] Xingchao Liu, Chengyue Gong, Qiang Liu, Flow straight and fast: Learning to generate and transfer data with rectified flow, 2022, arXiv preprint arXiv:2209.03003.
- [4] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, Björn Ommer, High-resolution image synthesis with latent diffusion models, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10684–10695.

- [5] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, Ben Poole, Score-based generative modeling through stochastic differential equations, 2020, arXiv preprint [arXiv:2011.13456](https://arxiv.org/abs/2011.13456).
- [6] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, David J Fleet, Video diffusion models, *Adv. Neural Inf. Process. Syst.* 35 (2022) 8633–8646.
- [7] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, Generative adversarial nets, *Adv. Neural Inf. Process. Syst.* 27 (2014).
- [8] Prafulla Dhariwal, Alexander Nichol, Diffusion models beat gans on image synthesis, *Adv. Neural Inf. Process. Syst.* 34 (2021) 8780–8794.
- [9] Diederik P. Kingma, Max Welling, Auto-encoding variational Bayes, in: Yoshua Bengio, Yann LeCun (Eds.), 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14–16, 2014, Conference Track Proceedings, 2014.
- [10] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, Mark Chen, Point-e: A system for generating 3d point clouds from complex prompts, 2022, arXiv preprint [arXiv:2212.08751](https://arxiv.org/abs/2212.08751).
- [11] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, Mohammad Norouzi, Image super-resolution via iterative refinement, *IEEE Trans. Pattern Anal. Mach. Intell.* 45 (4) (2022) 4713–4726.
- [12] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, Ming-Hsuan Yang, Diffusion models: A comprehensive survey of methods and applications, *ACM Comput. Surv.* 56 (4) (2023).
- [13] Qiang Liu, Nils Thuerey, Uncertainty-aware surrogate models for airfoil flow simulations with denoising diffusion probabilistic models, *AIAA J.* 62 (8) (2024) 2912–2933.
- [14] Claudio Zeni, Robert Pinsler, Daniel Zügner, Andrew Fowler, Matthew Horton, Xiang Fu, Zilong Wang, Aliaksandra Shysheya, Jonathan Crabbé, Shoko Ueda, et al., A generative model for inorganic materials design, *Nature* 639 (8055) (2025) 624–632.
- [15] Nikita Kartashov, Nikolaos N. Vlassis, A large language model and denoising diffusion framework for targeted design of microstructures with commands in natural language, *Comput. Methods Appl. Mech. Engrg.* 437 (2025) 117742.
- [16] Dule Shu, Zijie Li, Amir Barati Farimani, A physics-informed diffusion model for high-fidelity flow field reconstruction, *J. Comput. Phys.* 478 (2023) 111972.
- [17] N Benjamin Erichson, Vinicius Mikuni, Dongwei Lyu, Yang Gao, Omri Azencot, Soon Hoe Lim, Michael W Mahoney, FLEX: A backbone for diffusion-based modeling of spatio-temporal physical systems, 2025, arXiv preprint [arXiv:2505.17351](https://arxiv.org/abs/2505.17351).
- [18] Chunyang Wang, Biyue Pan, Zhibo Dai, Yudi Cai, Yuhao Ma, Hao Zheng, Dixia Fan, Hui Xiang, AeroDiT: Diffusion transformers for Reynolds-averaged Navier–Stokes simulations of airfoil flows, *Phys. Fluids* 37 (12) (2025).
- [19] Jiajun Hu, Zhen Lu, Yue Yang, Generative prediction of flow fields around an obstacle using the diffusion model, *Phys. Rev. Fluids* 10 (9) (2025) 094903.
- [20] Grégoire Allaire, Charles Dapogny, François Jouve, Shape and topology optimization, in: *Handbook of Numerical Analysis*, Vol. 22, Elsevier, 2021, pp. 1–132.
- [21] Yayati Jadhav, Joseph Berthel, Chunshan Hu, Rahul Panat, Jack Beuth, Amir Barati Farimani, StressD: 2D stress estimation using denoising diffusion model, *Comput. Methods Appl. Mech. Engrg.* 416 (2023) 116343.
- [22] Anthony Zhou, Zijie Li, Michael Schneider, John R Buchanan Jr., Amir Barati Farimani, Text2pde: Latent diffusion models for accessible physics simulation, 2024, arXiv preprint [arXiv:2410.01153](https://arxiv.org/abs/2410.01153).
- [23] Zijie Li, Anthony Zhou, Amir Barati Farimani, Generative latent neural PDE solver using flow matching, 2025, arXiv preprint [arXiv:2503.22600](https://arxiv.org/abs/2503.22600).
- [24] Saakaar Bhatnagar, Yaser Afshar, Shaowu Pan, Karthik Duraisamy, Shailendra Kaushik, Prediction of aerodynamic flow fields using convolutional neural networks, *Comput. Mech.* 64 (2019) 525–545.
- [25] Zongyi Li, Nikola Kovachki, Kamyar Aizzadenedeshli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, Anima Anandkumar, Fourier neural operator for parametric partial differential equations, 2020, arXiv preprint [arXiv:2010.08895](https://arxiv.org/abs/2010.08895).
- [26] Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, George Em Karniadakis, Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators, *Nat. Mach. Intell.* 3 (3) (2021) 218–229.
- [27] Claudia Drygala, Edmund Ross, Francesca di Mare, Hanno Gottschalk, Comparison of generative learning methods for turbulence modeling, 2024, arXiv preprint [arXiv:2411.16417](https://arxiv.org/abs/2411.16417).
- [28] Olaf Ronneberger, Philipp Fischer, Thomas Brox, U-net: Convolutional networks for biomedical image segmentation, in: *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, Part III* 18, Springer, 2015, pp. 234–241.
- [29] Ali Kashefi, Davis Rempe, Leonidas J. Guibas, A point-cloud deep learning framework for prediction of fluid flow fields on irregular geometries, *Phys. Fluids* 33 (2) (2021).
- [30] Ali Kashefi, Kolmogorov–Arnold PointNet: Deep learning for prediction of fluid fields on irregular geometries, *Comput. Methods Appl. Mech. Engrg.* 439 (2025) 117888.
- [31] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, Gabriele Monfardini, The graph neural network model, *IEEE Trans. Neural Netw.* 20 (1) (2008) 61–80.
- [32] Keyulu Xu, Weihua Hu, Jure Leskovec, Stefanie Jegelka, How powerful are graph neural networks? in: *International Conference on Learning Representations*, 2019.
- [33] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, Philip S Yu, A comprehensive survey on graph neural networks, *IEEE Trans. Neural Networks Learn. Syst.* 32 (1) (2020) 4–24.
- [34] Charles R. Qi, Hao Su, Kaichun Mo, Leonidas J. Guibas, Pointnet: Deep learning on point sets for 3d classification and segmentation, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 652–660.
- [35] Charles Ruizhongtai Qi, Li Yi, Hao Su, Leonidas J Guibas, Pointnet++: Deep hierarchical feature learning on point sets in a metric space, *Adv. Neural Inf. Process. Syst.* 30 (2017).
- [36] Ali Kashefi, Tapan Mukerji, Physics-informed PointNet: A deep learning solver for steady-state incompressible flows and thermal fields on multiple sets of irregular geometries, *J. Comput. Phys.* 468 (2022) 111510.
- [37] Ali Kashefi, Leonidas J. Guibas, Tapan Mukerji, Physics-informed PointNet: On how many irregular geometries can it solve an inverse problem simultaneously? Application to linear elasticity, *J. Mach. Learn. Model. Comput.* 4 (4) (2023).
- [38] Ali Kashefi, Pointnet with kan versus pointnet with mlp for 3d classification and segmentation of point sets, *Comput. Graph.* (2025) 104319.
- [39] Mario Lino Valencia, Tobias Pfaff, Nils Thuerey, Learning distributions of complex fluid simulations with diffusion graph networks, in: *The Thirteenth International Conference on Learning Representations*, 2025.
- [40] Jiyong Kim, Sunwoong Yang, Namwoo Kang, Point-wise diffusion models for physical systems with shape variations: Application to spatio-temporal and large-scale system, 2025, arXiv preprint [arXiv:2508.01230](https://arxiv.org/abs/2508.01230).
- [41] Shitong Luo, Wei Hu, Diffusion probabilistic models for 3d point cloud generation, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2837–2845.
- [42] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, Vladlen Koltun, Point transformer, in: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 16259–16268.

- [43] Zekun Qi, Runpei Dong, Shaochen Zhang, Haoran Geng, Chunrui Han, Zheng Ge, Li Yi, Kaisheng Ma, Shapellm: Universal 3d object understanding for embodied interaction, in: *European Conference on Computer Vision*, Springer, 2024, pp. 214–238.
- [44] Yang Shen, Wei Huang, Zhen-guo Wang, Da-fu Xu, Chao-Yang Liu, A deep learning framework for aerodynamic pressure prediction on general three-dimensional configurations, *Phys. Fluids* 35 (10) (2023) 107111.
- [45] Junyan He, Seid Koric, Diab Abueidda, Ali Najafi, Iwona Jasiuk, Geom-DeepONet: A point-cloud-based deep operator network for field predictions on 3D parameterized geometries, *Comput. Methods Appl. Mech. Engrg.* 429 (2024) 117130.
- [46] Ali Kashefi, Tapan Mukerji, Prediction of fluid flow in porous media by sparse observations and physics-informed PointNet, *Neural Netw.* 167 (2023) 80–91.
- [47] Ali Kashefi, Tapan Mukerji, Physics-informed KAN PointNet: Deep learning for simultaneous solutions to inverse problems in incompressible flow on numerous irregular geometries, 2025, arXiv preprint arXiv:2504.06327.
- [48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, Illia Polosukhin, Attention is all you need, *Adv. Neural Inf. Process. Syst.* 30 (2017).
- [49] Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, Karsten Kreis, et al., Lion: Latent point diffusion models for 3d shape generation, *Adv. Neural Inf. Process. Syst.* 35 (2022) 10021–10039.
- [50] Yong He, Hongshan Yu, Mingtao Feng, Tongjia Chen, Zechuan Li, Anwaar Ulhaq, Saeed Anwar, Ajmal Saeed Mian, PointDiffuse: A dual-conditional diffusion model for enhanced point cloud semantic segmentation, 2025, arXiv preprint arXiv:2503.06094.
- [51] Ali Kashefi, A coarse grid projection method for accelerating free and forced convection heat transfer computations, *Results Math.* 75 (1) (2020).
- [52] Christophe Geuzaine, Jean-François Remacle, Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities, *Internat. J. Numer. Methods Engrg.* 79 (11) (2009) 1309–1331.
- [53] Henry G Weller, Gavin Tabor, Hrvoje Jasak, Christer Fureby, A tensorial approach to computational continuum mechanics using object-oriented techniques, *Comput. Phys.* 12 (6) (1998) 620–631.
- [54] L.S. Caretto, A.D. Gosman, S.V. Patankar, D.B. Spalding, Two calculation procedures for steady, three-dimensional flows with recirculation, in: *Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics*, Springer, 1973, pp. 60–68.
- [55] M. Behr, D. Hastreiter, S. Mittal, T.E. Tezduyar, Incompressible flow past a circular cylinder: Dependence of the computed flow field on the location of the lateral boundaries, *Comput. Methods Appl. Mech. Engrg.* 123 (1) (1995) 309–316.
- [56] Ali Kashefi, Anne E. Staples, A finite-element coarse-grid projection method for incompressible flow simulations, *Adv. Comput. Math.* 44 (4) (2017-11-27) 1063–1090.
- [57] Subhankar Sen, Sanjay Mittal, Gautam Biswas, Flow past a square cylinder at low Reynolds numbers, *Internat. J. Numer. Methods Fluids* 67 (9) (2011) 1160–1174.
- [58] Arnab Kumar De, Amaresh Dalal, Numerical simulation of unconfined flow past a triangular cylinder, *Internat. J. Numer. Methods Fluids* 52 (7) (2006) 801–821.
- [59] Wenjie Zhong, Lu Deng, Zhiying Xiao, Flow past a rectangular cylinder close to a free surface, *Ocean Eng.* 186 (2019) 106118.
- [60] Rajat Mittal, S. Balachandar, Direct numerical simulation of flow past elliptic cylinders, *J. Comput. Phys.* 124 (2) (1996) 351–367.
- [61] Z Abedin, Nafiz A Khan, MST Mousumi Rizia, Md Quamrul Islam, Simulation of wind flow over square, pentagonal and hexagonal cylinders in a staggered form, in: *AIP Conference Proceedings*, AIP Publishing LLC, 2017, 020004, Number 1.
- [62] Diederik P. Kingma, Jimmy Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.
- [63] Alexander Quinn Nichol, Prafulla Dhariwal, Improved denoising diffusion probabilistic models, in: *International Conference on Machine Learning*, PMLR, 2021, pp. 8162–8171.
- [64] Fernando Tejero, Sanjeeth Sureshbabu, Luca Boscaqli, David MacManus, Point-enhanced convolutional neural network: A novel deep learning method for transonic wall-bounded flows, *Aerosp. Sci. Technol.* 155 (2024) 109689.
- [65] Wontae Hwang, Sooyoung Kim, Donghyun Park, Seongim Choi, A point cloud-based mesh-independent convolutional neural network frameworks for flow field prediction on variable geometries, *Phys. Fluids* 37 (5) (2025).
- [66] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, Justin M Solomon, Dynamic graph cnn for learning on point clouds, *ACM Trans. Graph. (Tog)* 38 (5) (2019) 1–12.
- [67] Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, Yun Fu, Rethinking network design and local geometry in point cloud: A simple residual MLP framework, in: *International Conference on Learning Representations*, 2022.
- [68] Tim Salimans, Jonathan Ho, Progressive distillation for fast sampling of diffusion models, in: *International Conference on Learning Representations*, 2022.
- [69] Yang Song, Prafulla Dhariwal, Mark Chen, Ilya Sutskever, Consistency models, 2023, arXiv preprint arXiv:2303.01469.
- [70] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, Jun Zhu, Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models, *Mach. Intell. Res.* 22 (4) (2025) 730–751.
- [71] Rishi Bommasani, On the opportunities and risks of foundation models, 2021, arXiv preprint arXiv:2108.07258.
- [72] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, Mark Chen, Hierarchical text-conditional image generation with clip latents, 2022, arXiv preprint arXiv:2204.06125, 1 (2) 3.
- [73] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al., Gpt-4 technical report, 2023, arXiv preprint arXiv:2303.08774.
- [74] Ali Kashefi, Tapan Mukerji, Chatgpt for programming numerical methods, *J. Mach. Learn. Model. Comput.* 4 (2) (2023).
- [75] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al., Gemini: A family of highly capable multimodal models, 2023, arXiv preprint arXiv:2312.11805.
- [76] Ali Kashefi, A misleading gallery of fluid motion by generative artificial intelligence, *J. Mach. Learn. Model. Comput.* 5 (2) (2024) 113–144.