ASME Journal of Computing and Information Science in Engineering
Online journal at:
https://asmedigitalcollection.asme.org/computingengineering

Finite-Volume Physics-Informed U-Net for Flow Field Reconstruction With Sparse Data

Tong Zhu
Department of Mechanical and Aerospace Engineering,
The Hong Kong University of Science and Technology,
Clear Water Bay,
Hong Kong, SAR, 999077, China
e-mail: tzhuaf@connect.ust.hk

Dehao Liu
Department of Mechanical Engineering,
State University of New York at Binghamton,
Binghamton, NY 13902
e-mail: dehaoliu@binghamton.edu

Yanglong Lu[1]
Department of Mechanical and Aerospace Engineering,
The Hong Kong University of Science and Technology,
Clear Water Bay,
Hong Kong, SAR, 999077, China
e-mail: maeylu@ust.hk

Fluid dynamics is governed by partial differential equations (PDEs) which are solved numerically. The limitations of traditional methods in data assimilation hinder their effective engagement with experiments. Physics-informed neural network (PINN) has emerged as a hybrid data-physics-driven model for convective problems. However, the approach suffers from low accuracy and poor efficiency due to the way of incorporating PDEs. In this work, a novel convolutional neural network framework integrating the finite volume method (FVM) is developed to address the challenge. The interface variables of the grid are predicted by the neural network for the first time, rather than a complex procedure in FVM. The physical law is then learned by minimizing the residual of the discretized conservative form of PDEs. A comparison between this model and the existing PINN models regarding prediction accuracy demonstrates the superiority of embedding PDEs through FVM. The effects of sampling strategies and quantities are studied. The result confirms the model's capability to utilize sparse measurement data within the computational domain. Furthermore, the model performs well even in scenarios where partial initial and boundary conditions are absent. [DOI: 10.1115/1.4067583]

Keywords: physics-informed machine learning, fluid dynamics, finite volume method, deep learning, flow field prediction, advanced computing infrastructure, inverse methods for engineering applications

1 Introduction

Fluid flow is a common phenomenon of interest in science. It has attracted widespread attention in aerospace [1], biomedicine [2], chemical engineering [3], etc. The flow behavior is governed by nonlinear partial differential equations (PDEs), such as Navier–Stokes (N–S) equations. Due to the complexity of the governing equations, it is nearly infeasible to obtain analytical solutions, especially in three-dimensional and unsteady cases. Previous studies have proved that numerical methods manage to address this challenge [4–6], including finite difference method (FDM), finite volume method (FVM), and finite element method (FEM). However, the traditional numerical methods still struggle when combined with real experiments because the system's boundaries and initial conditions are hard to measure completely. Intermediate measurement data cannot be utilized to correct the numerical results yet. These factors may lead to a certain difference between the numerical solutions and the experimental measurements [7–9]. Thus, there is a practical need to develop an efficient approach to leverage measurement data and refine computational models.

Physics-informed machine learning takes advantage of the data-driven model and the constraints defined by the PDEs to solve scientific problems [10]. One of the popular physics-informed machine learning models is known as physics-informed neural networks (PINN), which can be used to reconstruct full-field physics with sparse measurement data [11]. The key concept of PINN is to predict the solutions through an artificial neural network that has been trained with the integration of PDEs. Besides the data loss function in the traditional machine learning models, PINN also contains a loss function of physical laws described by the PDEs. In the original PINN, the physical loss function is constructed by an automatic differentiation technique. Automatic differentiation applies the chain rule to approximate the arbitrary order derivates of output from neural networks since the gradient of each neuron is obtained during backpropagation [12]. Although PINN has achieved acceptable predictions in some cases, many studies have reported that the derivatives obtained via automatic differentiation were inaccurate [13,14]. Therefore, even if the residual of physical loss constructed in this way is small, it is by no means a good learning for the PDEs.

The challenge of the current model has led to the development of some extended PINNs by integrating numerical methods. Ranade et al. [14] proposed a machine learning-based solver for N–S equations using FVM. The results achieved good accuracy, better stability, and faster convergence. Yadav et al. [15] combined FEM and a neural network to solve a two-dimensional convective-diffusion equation. Their approach suppressed the undershoots and overshoots that were encountered in the existing neural network-based PDE solvers. These studies demonstrated that the significant drawback of automatic differentiation in the construction of physical loss can be effectively alleviated. In addition to the aforementioned works, some studies also think of using unsupervised learning,

[1]Corresponding author.

Manuscript received August 15, 2024; final manuscript received December 19, 2024; published online April 3, 2025. Assoc. Editor: Gaurav Ameta.

which does not require labeled data. In this strategy, the total loss function is only comprised of physical loss, making it easier to converge. Zhu et al. [16] proposed a physics-constrained deep learning model that is trained without any labeled data. Their results demonstrated that the model had high accuracy for predicting high-dimensional stochastic input fields. Another data-free model was designed by Yuan et al. [17]. They incorporated the governing PDEs and an implicit Runge–Kutta (RK) integration scheme into the training process. Their model thus achieves the prediction of a dynamic system at any discrete time-step and yields up to 100 times faster than standard time-domain simulations. Although these studies claim not to require data, they all assume boundary conditions are already known or use complete initial conditions as input. The primary function of these works is to serve as surrogate models for numerical methods, which cannot be used when boundary or initial conditions are not fully known in practical problems.

In this paper, a finite volume physics-informed U-Net (FV-PIUnet) framework is developed to further improve the accuracy of prediction in fluid dynamics. A fully convolutional neural network, U-Net, has replaced the original network to enhance computational efficiency. The neural network is used to predict the interface variables of the grid for the first time, which simplifies the solving process of FVM. The proposed method is then compared with the existing PINN models by solving a convective-diffusion equation. Sensitivity analyses regarding sampling strategies and quantities are also conducted. The model's capability with the sparse internal measurement data under incomplete initial and boundary conditions is validated. The major contributions of this work are as follows:

(1) A novel hybrid model, by integrating U-Net and finite volume frameworks, is proposed to reconstruct flow fields with sparse data. The interface variables in the fluid domain after discretization can be obtained from the neural networks efficiently.
(2) Different sampling strategies are investigated and their influences on prediction accuracies are studied. The sampling strategy provides guidance for sensor selection and installation in practical applications.
(3) The proposed model is designed to exhibit flexibility in managing incomplete initial and boundary conditions, a challenge beyond the capabilities of traditional numerical methods.

In the remainder of the paper, the backgrounds of the traditional FVM to discretize PDEs and the original physics-informed machine learning model are introduced in Sec. 2. The developed FV-PIUnet framework is described in Sec. 3. The results and sensitive analyses are given in Sec. 4.

2 Background

In this section, the conventional FVM for fluid dynamics is reviewed. The original physics-informed machine learning model is also introduced.

2.1 Finite Volume Method. Various approaches have been applied to numerically solve the governing equations in fluid dynamics. The FVM is one of the most well-known techniques because it ensures the conservation of physical quantities [18]. In the FVM, the spatial domain is discretized into a finite number of small, non-overlapping control volumes or cells. Each cell encloses a single point called the cell center. The method works by integrating the PDEs written in conservative form over a control volume Ω. Here, the inviscid equation is taken as an example to demonstrate the discretization process.

$$\int_{\Omega}\left[\frac{\partial u}{\partial t}\right]d\Omega+\int_{\Omega}\left[\frac{\partial(u^2/2)}{\partial x}\right]d\Omega=0 \tag{1}$$

where u is the variable of interest such as velocity. It is a function of temporal and spatial coordinates. The expression $u^2/2$ denotes the flux function at the cell interface. Using the divergence theorem, the above equation can be written as

$$\int_{\Omega}\left[\frac{\partial u}{\partial t}\right]d\Omega+\int_{\Gamma}[(u^2/2)\cdot\hat{\boldsymbol{n}}]d\Gamma=0 \tag{2}$$

where $\hat{\boldsymbol{n}}$ is the outward pointing unit normal to the control volume boundary Γ. The integral formula then can be transformed into discrete algebraic equations that can be solved iteratively by

$$u_i^{n+1}=u_i^n-\frac{\Delta t}{2\Delta x}[u_{i+1/2}^n-u_{i-1/2}^n] \tag{3}$$

where the superscript n is the current time-step. The subscript i represents the index of the cell center, while $i+1/2$ or $i-1/2$ means the interface of the cell. The computational parameters Δt and Δx are prescribed to time and spatial steps, respectively.

To evolve the solution according to Eq. (3), the interface fluxes are required. The FVM model assumes that the value of a variable in a cell is stored at the cell center. Therefore, the fluxes need to be calculated based on the known cell-averaged values, which mainly involves two steps: reconstructing and solving the Riemann problem. The reconstruction step is to use the cell average to approximate the values on both sides of the interfaces. The reconstruction step can be done by using various techniques, such as piecewise constant (zero-order), linear (first-order), or higher-order polynomial interpolations [19]. Once the variables are reconstructed on either side of the control volume face, the interface values can be determined by the variable pairs. This is where the Riemann problem comes into play. The Riemann solver is thus crucial and relies on the nature of the PDEs being solved and the expected behavior of the solution. The Riemann solver must accurately capture the wave speeds and the associated changes in the state variables. Some commonly used Riemann solvers include the Roe solver, HLL solver, and Osher solver [20].

Although FVM is considered a robust numerical approach to solving compressible or incompressible flows, the choice of reconstruction schemes and Riemann solvers still pose many challenges [21–23]. Using low-order reconstruction schemes, such as zero or first order, introduces numerical diffusion and dissipation, diminishing computational accuracy. Especially in strong nonlinear problems like shock and contact discontinuity, low-order schemes fail to accurately capture these features [24]. High-order schemes can enhance solution accuracy but may encounter non-physical oscillations [25], which even result in divergence. Developing a Riemann solver for multidimensional problems is quite complex, and its computational efficiency is low [26]. Furthermore, the discrete equations are solved iteratively starting from an initial condition in the classical FVM [27]. This characteristic makes it difficult to apply in real-world experiments, as a complete initial condition is hard to obtain due to sensor accessibility, sensor resolution, and sensor cost. The conventional practice assumes a continuous boundary distribution based on the measured data and then calculates the entire fields through the FVM, which often introduces errors.

2.2 Physics-Informed Machine Learning. Machine learning, as a pure data-driven method, has managed to handle different tasks such as image recognition [28], semantic segmentation [29], and natural language processing [30], etc. Since it relies heavily on large and high-quality datasets, its applications can be limited when sufficient training data are unavailable especially in some physical scenarios [31]. Physics-informed machine learning offers a hybrid physics-data-driven mode to address this drawback [11]. Among the various machine learning techniques, neural networks have been widely adopted in the physics-informed model due to their ability to approximate nonlinear functions, handle high-dimensional data, and flexibly incorporate physical constraints through loss functions [32]. A typical neural network is also

known as the multilayer-perceptron comprising an input layer, one or more hidden layers, and an output layer [33]. Each layer contains nodes that apply weights $\boldsymbol{w}$ and biases $\boldsymbol{b}$ to their inputs and pass the results through an activation function $\boldsymbol{\varphi}$ to generate outputs. By training these networks on large datasets, the model learns the optimal weights and biases that map inputs to outputs [34].

The architecture of the original PINN used as the baseline is presented in Fig. 1. There are eight hidden layers between its input and output layers, each of which contains 20 neurons. The hyperbolic tangent function is employed as the activation function. The adopted configuration has been reported in previous research [35], demonstrating its ability to solve similar PDEs with high accuracy.

The PINN remains the standard neural network structure, but the loss function is augmented to include additional terms that enforce the governing physical laws of the system [36]. These physics-based constraints could involve the PDEs, initial or boundary conditions, resulting in the ability to learn not only the statistical correlations but also the underlying physical laws between the data [37]. In applications of fluid flow, PINNs take spatial coordinates x and time t as inputs and generate flow-related predictions $\tilde{\boldsymbol{u}}$ as outputs. It is represented as

$$\tilde{\boldsymbol{u}} = \mathcal{N}(x, t; \boldsymbol{w}, \boldsymbol{b}) \tag{4}$$

where $\mathcal{N}$ is the neural network operator.

Rao et al. [38] developed a PINN for modeling incompressible laminar flows. They carried out predictions of the flow around a cylinder at low Reynolds numbers by PINN. The results were then compared with numerical references. Arzani et al. [39] used PINN to obtain near-wall hemodynamics and wall shear stress data from sparse velocity measurements. Their work indicated that even without knowledge of the inlet and outlet boundaries, PINN can recover the global distributions of the concerned quantities with internal measurements. As for the compressible flows, Wassing et al. [40] solved the parametric Euler equations. They improved the convergence and reduced the training time of the neural networks by introducing an artificial viscosity. Jagtap et al. [41] proposed an extended PINN method to solve the inverse problems in supersonic flows which were difficult to solve using numerical methods. The global shock was successfully captured by the neural networks when using a limited amount of data within the domain and data on the boundary. The original PINNs have been employed to predict fluid flow using both measurement data and physical laws, but the accuracy of the learned PDEs is much lower than that attained by numerical methods. Meanwhile, the traditional neural network architecture has a lower computational efficiency due to its densely connected neurons [42].

In terms of network architecture enhancement, Li and McComb [43] introduced a modified generative adversarial network to perform super-resolution for multiphase flow. Their models outperform non-machine learning upsampling methods, effectively preserving intricate details. Meanwhile, Peng et al. [44] developed a graph neural network that served as a prediction module. They extended the geometric adaptability to irregular domains by employing unstructured meshes. The proposed model was evaluated on flow around a cylinder and demonstrated that the predicted velocity fields are closely aligned with the ground truth, even with very limited training data. As for learning strategies of governing equations, Tong et al. [45] presented a novel network inspired by the classical Roe solver to predict the solutions of hyperbolic PDEs. Comparisons with the original solver revealed that the neural network-based approach exhibited slight superiority. Ren et al. [46] developed a physics-informed convolutional-recurrent network to solve the nonlinear convective PDEs. In their study, FDM was introduced to discretize the computational domain, and the physical loss function was defined by the discrete governing equations. Their method outperformed the original one in terms of solution accuracy, extrapolability, and generalizability. Although these advancements improve the accuracy and efficiency of the original PINN, they require full-field data to train the models or rely on the entire previous state to predict subsequent solutions. This led to similar challenges as traditional numerical methods, which are unable to effectively leverage sparse scattered data during the intermediate process.

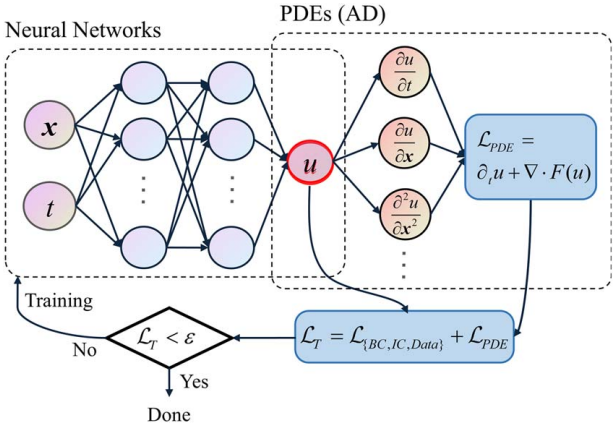

Fig. 1 The architecture of the original PINN

3 Methodology

To address the aforementioned challenges, the FV-PIUnet architecture is proposed by learning the PDEs with the FVM, where the original neural network is replaced with U-Net [47]. The diagram of FV-PIUnet is shown in Fig. 2. The process starts by discretizing the computational domain into a grid, where the interface coordinates of this grid serve as the input of FV-PIUnet. It is subsequently transformed through an embedding procedure by mapping each temporal instance onto the spatial coordinates. The spatiotemporal grid is then processed by a U-Net architecture which is an encoder–decoder framework. The encoder is comprised of a series of convolutional layers followed by activation layers. The purpose of the encoder is to compress input data to capture the latent context. The decoder has a symmetric structure of transposed convolutional layers, which are used to expand the compressed feature map to the original input size. To link the encoder and decoder, an intermediate layer with a 1×1 convolutional kernel is employed [48]. There are some skip connections between low-level feature maps (in encoder) and high-level ones (in decoder) to facilitate the stability of convergence.

There are four independent predictions $\tilde{u}_f^i$ on each interface represented by hollow circles in Fig. 3, while the solid circle denotes the cell-centered value $\tilde{u}$. It should be noted that the interface has a unit length even in one-dimensional case. To obtain a single representative variable $\tilde{u}_f$ for that interface, the predicted values are integrated over the interface using the trapezoidal quadrature and then averaged. The formula is defined as

$$\tilde{u}_f = \frac{1}{6}(\tilde{u}_f^1 + 2\tilde{u}_f^2 + 2\tilde{u}_f^3 + \tilde{u}_f^4) \tag{5}$$

Finally, the cell-centered value is obtained through the interpolation between two neighboring interfaces as

$$\tilde{u}_i = \frac{1}{2}(\tilde{u}_{i+1/2} + \tilde{u}_{i-1/2}) \tag{6}$$

and this is also the output of the entire model.

During the training process, the network parameters θ are optimized by solving

$$\theta^* = \arg\min_{\theta} \mathcal{L}_{\mathcal{T}}(\theta) \tag{7}$$

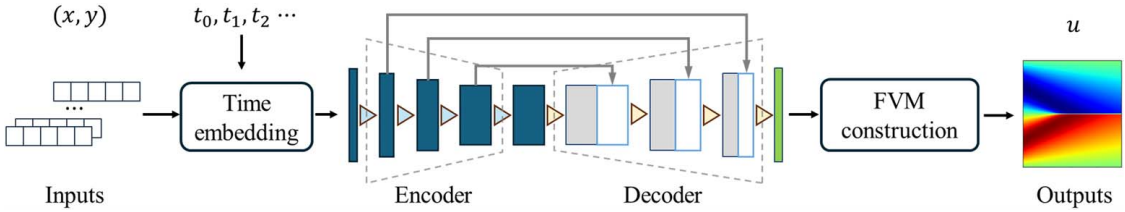

Fig. 2 The workflow of the proposed network architecture

where θ^* denotes the optimal parameters and $\mathcal{L}_{\mathcal{T}}$ is the total loss function of the neural networks which consists of the physical loss $\mathcal{L}_{PDE}$, boundary conditions (BC) loss $\mathcal{L}_{BCs}$, initial condition (IC) loss $\mathcal{L}_{IC}$, and data loss $\mathcal{L}_{data}$. Due to significant variations in the magnitudes of different loss components, simply summing all losses may lead the optimization to prioritize the largest component. This focus causes the model to neglect smaller components, resulting in their divergence. It is expected to satisfy all constraints simultaneously in the physics-informed machine learning model. To address this challenge, an adaptive weighting method is employed to design the total loss function, referred to Ref. [49]

$$\mathcal{L}_{\mathcal{T}} = \frac{1}{2\sigma_1^2}\mathcal{L}_{PDE} + \frac{1}{2\sigma_2^2}\mathcal{L}_{BCs} + \frac{1}{2\sigma_3^2}\mathcal{L}_{IC} + \frac{1}{2\sigma_4^2}\mathcal{L}_{data} + \sum_i^4 \ln(1+\sigma_i^2) \tag{8}$$

where σ_i is a series of trainable parameters that are dynamically changed according to each loss function; $\sum_i^4 \ln(1+\sigma_i^2)$ is the regularization terms that penalize excessively large σ_i and ensure the loss value to be positive even when $\sigma_i < 1$.

The physical loss function is derived by evaluating the residual of the discrete PDEs using the model outputs. The formula is shown as

$$\mathcal{L}_{PDE} = \frac{1}{N_t N_s}\sum_{n=0}^{N_t}\sum_{i=0}^{N_s}\left(\frac{\tilde{u}_i^{n+1} - \tilde{u}_i^n}{\Delta t} + \frac{F(\tilde{u}_{i+1/2}^n) - F(\tilde{u}_{i-1/2}^n)}{\Delta x}\right)^2 \tag{9}$$

where N_t and N_s are the number of temporal and spatial nodes, respectively. The term time derivative is discretized using the Euler forward scheme, while the spatial derivative is computed using the FVM. Decreasing the physical loss indicates that the predictions of neural networks are subject to the constraints of PDEs, which also means that these physical laws have been learned successfully. The boundary and initial loss functions can be defined by the mean square error (MSE) between the predictions and actual values, such as

$$\mathcal{L}_{ICs} = \frac{1}{N_{IC,s}}\sum_{i=0}^{N_{IC,s}}(\tilde{u}_i^0 - u_i^0)^2 \tag{10}$$

$$\mathcal{L}_{BCs} = \frac{1}{N_{BC,t}}\sum_{n=0}^{N_{BC,t}}(\tilde{u}_{BC}^n - u_{BC}^n)^2 \tag{11}$$

where $N_{IC,s}$ denotes the number of spatial nodes at the initial condition and $N_{BC,t}$ represents the number of temporal nodes at the boundary conditions. Similarly, the data loss function is computed by the MSE between the predicted values and interior samples, such as

$$\mathcal{L}_{data} = \frac{1}{N_t N_s}\sum_{n=0}^{N_t}\sum_{i=0}^{N_s}(\tilde{u}_i^n - u_i^n)^2 \tag{12}$$

This part of loss is consistent with the traditional machine learning strategy aiming to fit the measurement data.

FV-PIUnet is inspired by the discrete method of finite volume, but it directly predicts the interface variables via the neural networks instead of the traditional procedure. Employing U-Net achieves a node-to-node mapping from a spatiotemporal grid to physical quantities. In this way, the spatiotemporal relationships among each predicted value are preserved, which helps apply numerical schemes later. Meanwhile, the convolution operations reduce the density of connections between layers, thus enhancing computational efficiency. The details of the FV-PIUnet algorithm are listed in Algorithm 1.

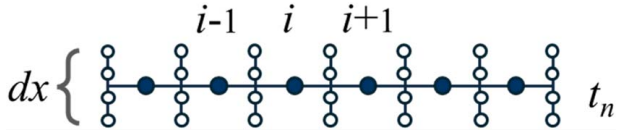

Fig. 3 The diagram of predicted interface values and cell-centered values

Algorithm 1 Finite Volume Physics-informed U-Net

Input: Spatial coordinates of grid interface x and y; Time series t; Internal sample dataset D_1; Boundary and initial conditions sample dataset D_2; Total number of epoch E; $E_0 = 0$
Output: $\tilde{u}(x,\ y,\ t;\ \theta)$

1. Initialize network parameters $\theta \in \mathbb{R}^k$ and a U-Net $\mathcal{N}$
2. Embed each time instant t_i into the spatial coordinates
3. WHILE $E_0 < E$ DO
   a. Compute the four values $\tilde{u}_f^i$ at the interface by forward propagation, $\tilde{u}_f^i = \mathcal{N}(x,\ y,\ t;\ \theta)$.
   b. Compute the variables $\tilde{u}_f$ at the interface by integrating $\tilde{u}_f^i$ and then averaging.
   c. Reconstruct the variables at the cell-centered $\tilde{u}$ based on $\tilde{u}_f$.
   d. Compute physical loss function via PDEs discretized by FVM.
   e. Employ the sample datasets to compute the corresponding loss functions.
   f. Weight each part of loss functions adaptively to form a total loss function.
   g. Minimize the total loss function and update θ by performing backpropagation.
   h. $E_0 = E_0 + 1$

END while

4 Results and Discussion

4.1 One-Dimensional Burgers' Equation. Burgers' equation is a fundamental nonlinear PDE used to model various physical phenomena in fluids, such as supersonic flow and shock waves. It is also regarded as a prerequisite to the N–S equations. In this work, a one-dimensional viscous Burgers' equation was investigated to validate the performance of the proposed method. The equation takes the form as

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2} \tag{13}$$

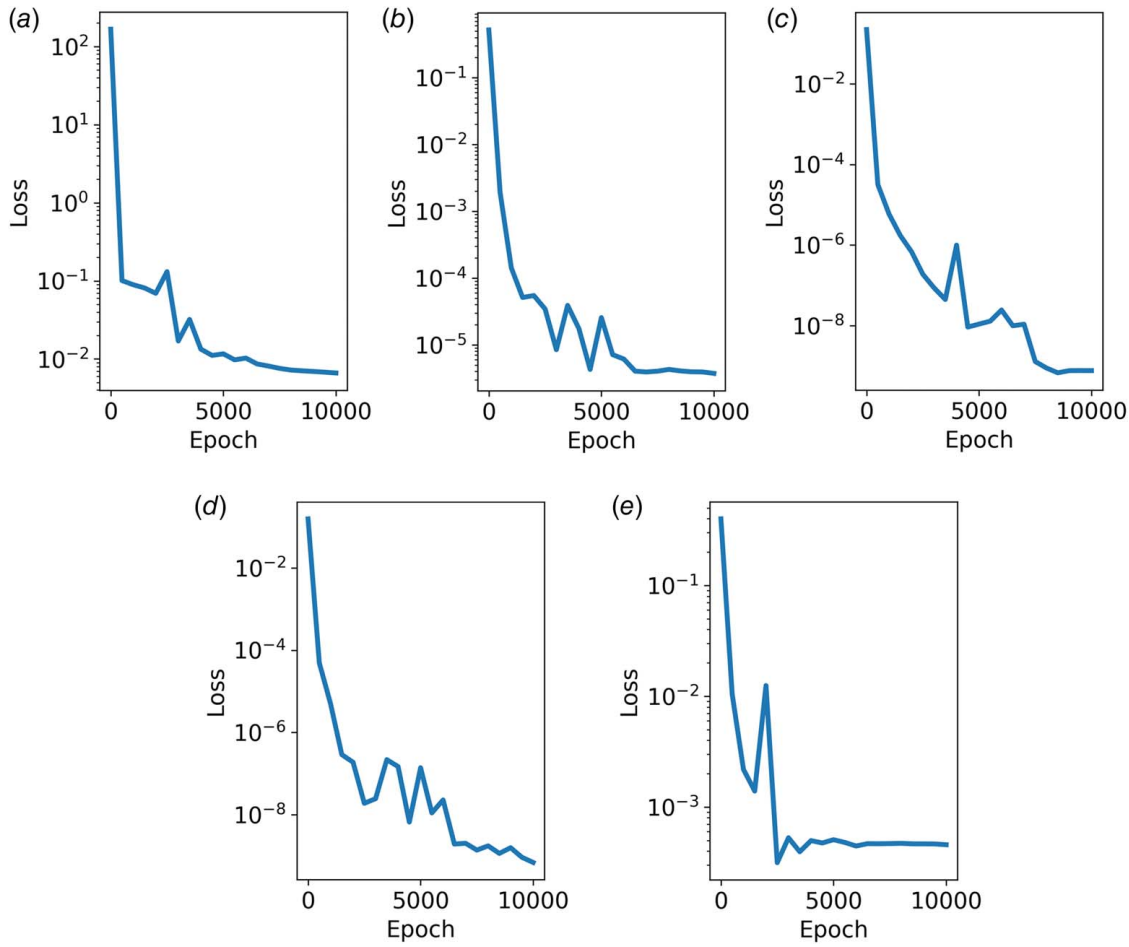

Fig. 4 The training histories of (*a*) physical loss, (*b*) initial condition loss, (*c*) bottom boundary loss, (*d*) top boundary loss, and (*e*) data loss

where u here represents velocity and ν is the constant viscosity meaning the diffusion of velocity in the medium. The system is subject to the initial condition, such as

$$u(x, t_0) = -sin(\pi x) \tag{14}$$

and the periodic boundary conditions at both ends. Given a dimensionless space-time domain $x \in [-1, 1]$ and $t \in [0, 1]$, both the temporal and spatial domains were discretized using a uniform grid. The spatial step was set to be 5×10^{-3}, resulting in 400 nodes along the x-axis. While the time-step was set to be 1×10^{-3}, which led to 1001 nodes along the t-axis. For the standard case, all the nodes on the boundaries and initial conditions were sampled to train the *I*/*BC* loss, while the interior domain was randomly sampled with 500 nodes to form data loss. The ratio η of the measurement data relative to the total data to be predicted can be calculated as

$$\eta = \frac{N_{BC} + N_{IC} + N_{data}}{N_{total}} \tag{15}$$

In the standard case, this ratio is 0.72%. Although the required measurement data seems quite sparse, they are adequate for training the proposed model. On one hand, the amount of training data required by this model is reduced by incorporating PDE constraints. On the other hand, all nodes located on the boundaries and initial

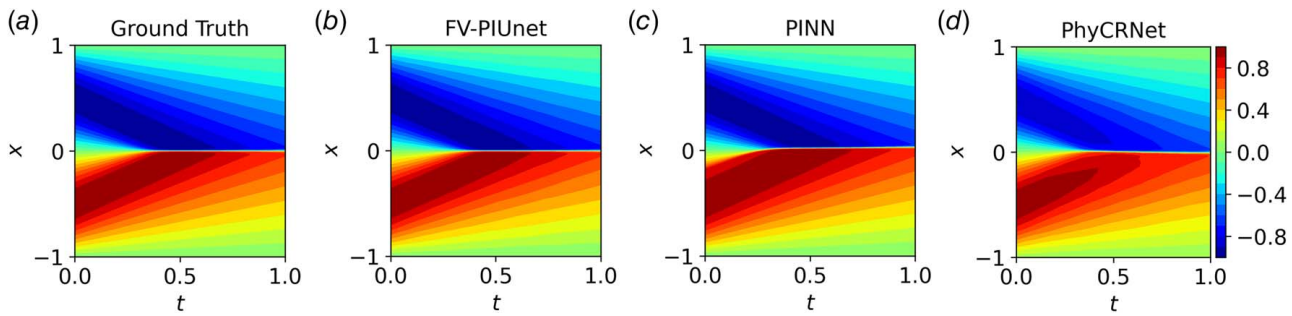

Fig. 5 Velocity distributions of (*a*) ground truth, (*b*) FV-PIUnet, (*c*) original PINN, and (*d*) PhyCRNet

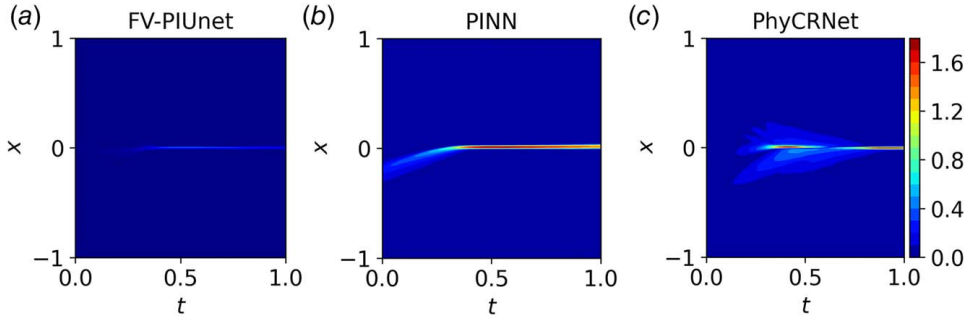

Fig. 6 Absolute error distributions between (*a*) FV-PIUnet, (*b*) original PINN, and (*c*) PhyCRNet

conditions are sampled, which are crucial for the solution of differential equations. The interior sampling data further enhance the model's prediction accuracy. In addition, the effects of sampling numbers and strategies will be discussed in detail later in the text. To evaluate the accuracy of the proposed method, the root mean square error (RMSE) is used as a metric. The smaller RMSE value indicates that the prediction is closer to the actual one and the model has a better performance. RMSE is computed by

$$RMSE = \sqrt{\sum_{i,n}^{N} \frac{(\tilde{u}_i^n - u_i^n)^2}{N}} \tag{16}$$

where N is the total number of nodes in the discrete domain.

The influence of the U-Net architecture was initially evaluated by varying the number of convolutional layers to identify the most effective configuration. As the number of layers increased from two to four, the training time correspondingly extended from 308.5 s to 476.2 s and then to 834.9 s. The RMSE values for predictions were 1.84×10^{-2}, 1.73×10^{-2}, 2.92×10^{-2}, respectively. These findings indicate that increasing model complexity beyond a certain point does not enhance prediction accuracy. Consequently, a three-layer U-Net demonstrated superior performance in this context and has been selected as the benchmark for subsequent research.

Figures 4(*a*)–4(*e*) show the evolutions of each loss with respect to the epoch during the training. The initial losses varied from the maximum value of 166.59 in the physical loss to the minimum value of 0.15 in the top boundary loss. After undergoing some fluctuations around the epoch of 5000, each loss function eventually converged to small values. The fluctuations result from the learning rate being too large for some small loss terms. All the losses decreased simultaneously which indicates that the model was learned from all the components. The adaptive weight method managed to address the uneven order of magnitude.

The velocity fields of the ground truth as well as three results from FV-PIUnet, PINN, and PhyCRNet [46] are shown in Fig. 5, respectively. The true velocity profile started with a sine shape shifting from positive to negative values smoothly. After $t=0.4$, there was a discontinuous shock at $x=0$ which means a sharp transition. The prediction from FV-PIUnet reproduced the true solution quite well in terms of the distribution and magnitude. Although the result from PINN seemed reasonable overall, it predicted the generation of the shock wave earlier at $t=0.3$. Additionally, the PhyCRNet result shows an obvious discrepancy before $t=0.5$. The overall accuracy of each model in predicting velocity was evaluated using RMSE. The RMSE values are 1.73×10^{-2}, 1.36×10^{-1}, and 4.13×10^{-2} for FV-PIUnet, PINN, and PhyCRNet, respectively. These results indicate that the proposed FV-PIUnet model achieves the highest prediction accuracy.

More details can be found in the error distributions. In Fig. 6(*a*), the error of FV-PIUnet outcome mainly focused on the region of shock where the maximum error was only 0.37. In contrast, the error of PINN was evident in both shock and non-shock regions, with a maximum error of 1.76, as shown in Fig. 6(*b*). The error distribution of PhyCRNet in Fig. 6(*c*) is relatively more widespread, but the maximum error slightly reduces to 1.62. Errors in the rest regions far from $x=0$ were small for both models because the slight change in the gradient allowed neural networks to fit measurement data easily. The comparison illustrated that incorporating the physics laws into the neural networks through the FVM is superior, especially when the gradient changes rapidly.

To further demonstrate the difference among FV-PIUnet, PINN, and PhyCRNet models, the velocity profiles at $t=0.2$ and 1.0 are plotted in Fig. 7, representing the results in non-shock region and shock region respectively. When $t=0.2$, the result from FV-PIUnet is closely aligned with the ground truth, which indicates the model's high level of accuracy. The results from PhyCRNet can generally predict the velocity profile, except for a slight underestimation near the maximum value of u. However, the PINN model exhibits

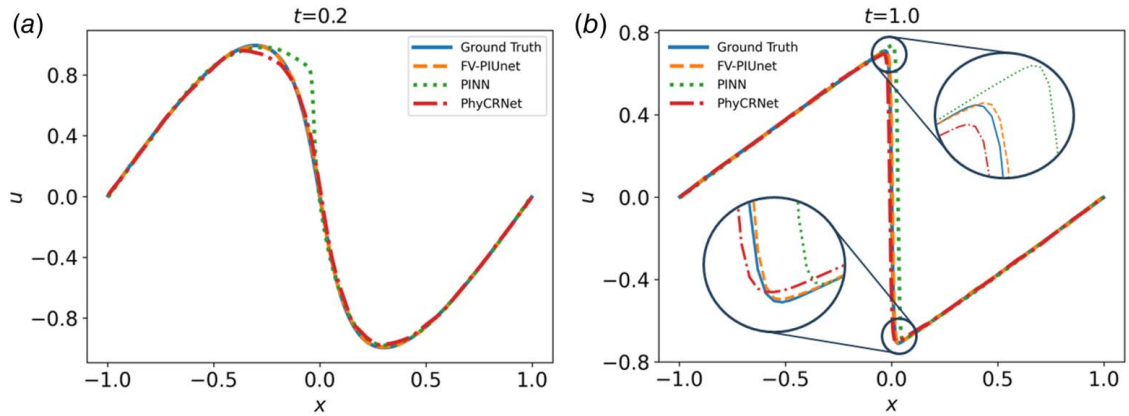

Fig. 7 The comparison of velocity profiles at (*a*) non-shock region ($t=0.2$) and (*b*) shock region ($t=1.0$)

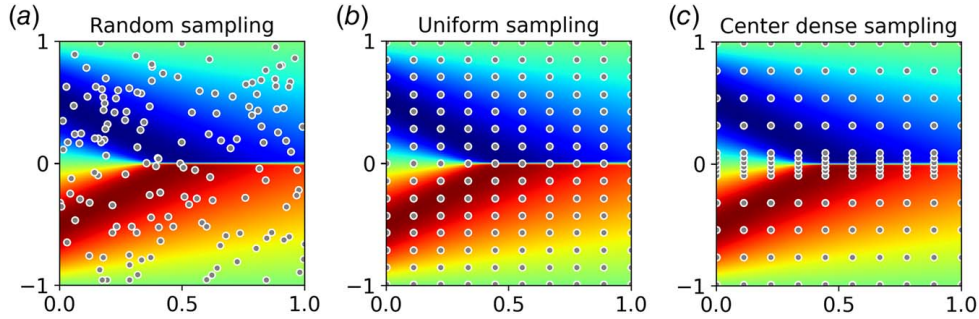

Fig. 8 Diagrams of (*a*) random sampling, (*b*) uniform sampling, and (*c*) center-dense sampling

a noticeable overshoot at the negative axis of the origin point. When $t = 1.0$, FV-PIUnet and PhyCRNet outcomes coincide with the ground truth. On the other hand, the PINN model failed to predict extreme values. For instance, there was an overshoot at the maximum point and an undershoot at the minimum point. These errors led to a wrong shift of the shockwave front, which was a critical feature in the system dynamics. The earlier results highlighted the proposed model's robustness and accuracy, both in non-shock or shock regions.

Various sensitivity analyses were conducted to study the effect of the number of interior samples and sampling strategies on the prediction accuracy with the FV-PIUnet model. Three types of sampling strategies are shown in Fig. 8. In the concept of random sampling, each node has an equal chance of being chosen from the whole spatiotemporal domain. The sampling locations in space may change at any time instant in this strategy. In real practice, it means that the positions of sensors need to be changed frequently, which is generally difficult to execute. Uniform sampling selects samples according to uniform distributions across both the spatial and temporal domains which ensures that every part of the area is equally represented in the sample set. Sensors can be easily arranged accordingly in practical applications. Center-dense sampling takes more samples from the center of the spatial domain and keeps them even in the temporal domain. This approach is a special case of gradient-oriented sampling, where more data is collected from regions with large gradient changes to aid the neural network in better learning. In this context, the central area in space is the place of interest, hence the use of center-dense sampling. Using this method requires prior knowledge or advanced judgment, which also brings certain challenges to the practical application.

The RMSE with different numbers of interior samples and different sampling strategies are listed in Table 1. The RMSE was reduced when more samples were collected in all cases. When the number of internal sampling points was below 500, the RMSE of random sampling was smaller than the others. But when the count reached 500 or more, the RMSE of center-dense sampling was the lowest. This is because random sampling can provide more global information with relatively small sample sizes. Whereas with a larger dataset, center-dense sampling can better capture the changes in gradients. Although the RMSE of uniform sampling was consistently the highest across the board, the gap narrows significantly at a sample size of 1000, such as 1.70×10^{-2}, 1.71×10^{-2}, and 1.69×10^{-2} for three sampling strategies, respectively. This suggests that uniform sampling becomes a preferable option when sufficient data are available due to its ease of implementation.

The effect of sample quantities at the boundary and initial conditions is also listed in Tables 2 and 3. The BC samples accounted for 0%, 25%, and 50% of the total BC nodes were tested while the rest settings remained the same as the standard case. It was found that the RMSE value slightly increased to 1.74×10^{-2} only when the ratio of BC samples was 0%. In fact, the periodic boundary conditions were incorporated into the physical loss function and assimilated into the learning process. This is because computing the physical loss at the boundaries required periodic padding. Hence, acceptable accuracy was maintained even in the absence of BC samples. When there were no samples on the initial conditions, the RMSE increased by 23.7% and reached 2.14×10^{-2}. A case with 100 internal random sampling nodes was also performed. Its RMSE value rose to 7.35×10^{-2} which suggested a significant impact of the absence of initial conditions on the model's accuracy. When the proportion of IC samples reached 50% of the total nodes, the RMSE value decreased to 1.74×10^{-2}, which is close to the accuracy achieved with 100% sampling. This demonstrates that FV-PIUnet can accurately predict velocity distributions even with incomplete initial conditions.

4.2 Two-Dimensional Burgers' Equation. To effectively illustrate the proposed model's capability in high-dimensional problems, a two-dimensional viscous Burgers' equation was investigated. The equation is expressed as follows:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + u\frac{\partial u}{\partial y} = v\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \tag{17}$$

where u here is a function of the spatial coordinates x, y and the temporal variable t. Both the spatial and temporal computational domains are dimensionless, specifically $x, y, t \in [0, 1]$. The discretized mesh is constructed by partitioning the spatial domain into a structured 64×64 grid and the temporal domain into 1000 time-

Table 1 RMSE of three sampling strategies with different numbers of internal sampling points

Numbers of interior samples	Random sampling	Uniform sampling	Center-dense sampling
10	1.32×10^{-1}	8.57×10^{-1}	2.18×10^{-1}
50	2.01×10^{-2}	7.69×10^{-2}	2.22×10^{-2}
100	1.78×10^{-2}	4.86×10^{-2}	1.94×10^{-2}
500	1.73×10^{-2}	1.77×10^{-2}	1.72×10^{-2}
1000	1.70×10^{-2}	1.71×10^{-2}	1.69×10^{-2}

Table 2 RMSE with different ratios of BC samples

Ratio of BC samples	0%	25%	50%
RMSE	1.74×10^{-2}	1.73×10^{-2}	1.73×10^{-2}

Table 3 RMSE with different ratios of IC samples

Ratio of IC samples	0%	25%	50%
RMSE	2.14×10^{-2}	1.75×10^{-2}	1.74×10^{-2}

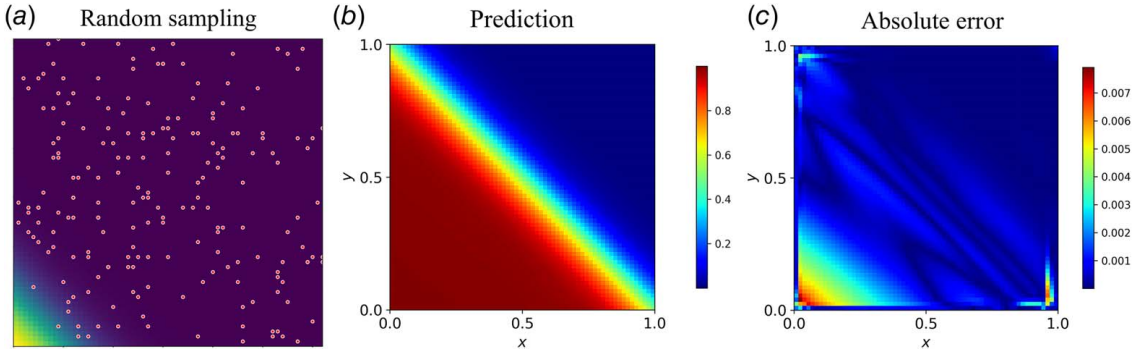

Fig. 9 (*a*) Diagram of random sampling points in space, (*b*) distribution of prediction velocity at last time-step, and (c) distribution of the absolute error

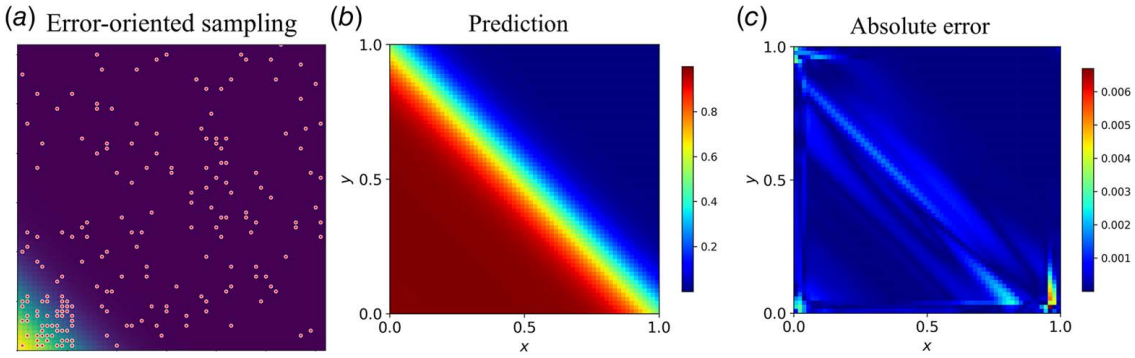

Fig. 10 (*a*) Diagram of error-oriented sampling points in space, (*b*) distribution of prediction velocity at last time-step, and (c) distribution of the absolute error

steps. The analytical solution of Eq. (17) is available, such as

$$u(x, y, t) = \frac{1}{1 + \exp\left[(x + y - t)/2v\right]} \tag{18}$$

In the 2D case, the sampling strategy also influences prediction accuracy. The random sampling process involves initially selecting 100 time instants, followed by the random sampling of 200 nodes within the spatial domain, as illustrated in Fig. 9(*a*). The predicted velocity distribution at $t = 1$ is shown in Fig. 9(*b*), as well as the absolute error distribution in Fig. 9(*c*). Overall, the predictions were consistent with the exact solution, achieving a RMSE of 1.18×10^{-3}. While the primary gradient along the diagonal is accurately captured, the model fails to predict the maximum value in the lower-left corner, resulting in concentrated errors in this region.

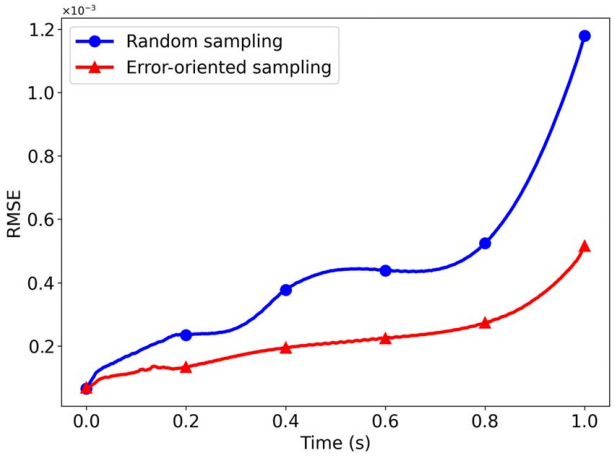

Fig. 11 Comparison of RMSE over time between random sampling and error-oriented sampling strategies

To further reduce the overall prediction errors, an error-oriented sampling strategy was introduced to mitigate these inaccuracies by allocating additional sampling points to high-error areas. It should be noted that the total number of sampling points remained unchanged. Figure 10(*a*) illustrates the sampling distribution at a specific time instant, highlighting an increased concentration of samples in the lower-left corner. As a result, Fig. 10(*c*) demonstrates that the original errors in this region were effectively eliminated, and the RMSE was reduced to 5.17×10^{-4}.

Figure 11 shows the overall accuracy of the two sampling strategies over time. The RMSE values of both methods increased with time, indicating an accumulation of prediction errors. Since complete and accurate initial conditions cannot be obtained, this error was amplified during each time-step solving. It is seen that the RMSE associated with error-oriented sampling remains consistently lower than that of random sampling throughout the entire temporal domain. These results demonstrate that error-oriented sampling is more effective in minimizing prediction error over time. The error-oriented sampling strategy can be applied in practical scenarios if the areas with significant errors can be identified or predicted in advance.

5 Conclusion

In this paper, an innovative FV-PIUnet framework was developed by integrating the FVM with U-Net. After comparing the prediction performance of the proposed framework with that of the existing models such as PINN and PhyCRNet, it becomes evident that the proposed model exhibits higher accuracy, regardless of the presence of shocks. This outcome suggests the superiority of

incorporating physical laws through the FVM instead of automatic differentiation. The sampling strategy and number of internal samples affect the predictions. The results from different ratios of *I*/*BC* samples indicate that FV-PIUnet can achieve accurate global solutions with incomplete *I*/*BC* conditions, which cannot be obtained from traditional numerical simulations.

While FV-PIUnet has successfully predicted the velocity field within a square region using the structured grid, its application to irregular geometries has not yet been explored. There is significant potential to utilize the unstructured mesh in the future, given that the FVM is well-suited for such applications. Moreover, the current prediction is limited to a single variable. A multi-headed decoder could be implemented when extending the model to solve other fluid mechanics equations, like Euler or N–S equations. This would allow for simultaneous predictions of more flow-related variables such as density, pressure, and temperature.

Acknowledgment

This work was supported in part by the University Grants Committee of Hong Kong (No. RMGS24EG04) and HKUST-Bright Dream Robotics Joint Research Institute (No. OKT24EG05).

Conflict of Interest

There are no conflicts of interest.

Data Availability Statement

The datasets generated and supporting the findings of this article are obtainable from the corresponding author upon reasonable request.

References

[1] Sun, Q. B., Shi, Z. W., Zhang, W. L., Sun, Z. K., and Chen, Y. L., 2022, "Numerical Simulation of Ground Effect on Circulation Control Airfoil," Int. J. Aerosp. Eng., **2022**, p. 20.

[2] Wang, H., Uhlmann, K., Vedula, V., Balzani, D., and Varnik, F., 2022, "Fluid-Structure Interaction Simulation of Tissue Degradation and its Effects on Intra-Aneurysm Hemodynamics," Biomech. Model. Mechanobiol., **21**(2), pp. 671–683.

[3] Zhu, T., Xia, A., Lin, K., Huang, Y., Zhu, X., Zhu, X., and Liao, Q., 2022, "Numerical Investigation of Bio-Inspired Mixing Enhancement for Enzymatic Hydrolysis," Chem. Eng. Sci., **260**, p. 117950.

[4] Fu, L., 2019, "A Very-High-Order TENO Scheme for All-Speed Gas Dynamics and Turbulence," Comput. Phys. Commun., **244**, pp. 117–131.

[5] Liu, Y. Q., Yin, X. L., Feng, L. B., and Sun, H. G., 2018, "Finite Difference Scheme for Simulating a Generalized Two-Dimensional Multi-Term Time Fractional Non-Newtonian Fluid Model," Adv. Differ. Eq., **2018**, p. 442.

[6] Chaparian, E., and Tammisola, O., 2019, "An Adaptive Finite Element Method for Elastoviscoplastic Fluid Flows," J. Non-Newtonian Fluid Mech., **271**, p. 104148.

[7] Ünverdi, M., 2022, "Prediction of Heat Transfer Coefficient and Friction Factor of Mini Channel Shell and Tube Heat Exchanger Using Numerical Analysis and Experimental Validation," Int. J. Therm. Sci., **171**, p. 107182.

[8] Lu, Y., and Wang, Y., 2020, "Physics-Based Compressive Sensing Approach to Monitor Turbulent Flow," AIAA J., **58**(8), pp. 3299–3307.

[9] Lu, Y., and Wang, Y., 2021, "Physics Based Compressive Sensing to Monitor Temperature and Melt Flow in Laser Powder Bed Fusion," Addit. Manuf., **47**, p. 102304.

[10] Li, H., Zhang, Z., Li, T., and Si, X., 2024, "A Review on Physics-Informed Data-Driven Remaining Useful Life Prediction: Challenges and Opportunities," Mech. Syst. Signal Process., **209**, p. 111120.

[11] Oommen, V., and Srinivasan, B., 2022, "Solving Inverse Heat Transfer Problems Without Surrogate Models: A Fast, Data-Sparse, Physics Informed Neural Network Approach," ASME J. Comput. Inf. Sci. Eng., **22**(4), p. 041012.

[12] Chiu, P.-H., Wong, J. C., Ooi, C., Dao, M. H., and Ong, Y.-S., 2022, "CAN-PINN: A Fast Physics-Informed Neural Network Based on Coupled-Automatic–Numerical Differentiation Method," Comput. Methods Appl. Mech. Eng., **395**, p. 114909.

[13] Gao, H., Zahr, M. J., and Wang, J.-X., 2022, "Physics-Informed Graph Neural Galerkin Networks: A Unified Framework for Solving PDE-Governed Forward and Inverse Problems," Comput. Methods Appl. Mech. Eng., **390**, p. 114502.

[14] Ranade, R., Hill, C., and Pathak, J., 2021, "DiscretizationNet: A Machine-Learning Based Solver for Navier–Stokes Equations Using Finite Volume Discretization," Comput. Methods Appl. Mech. Eng., **378**, p. 113722.

[15] Yadav, S., and Ganesan, S., 2024, "Artificial Neural Network-Augmented Stabilized Finite Element Method," J. Comput. Phys., **499**, p. 112702.

[16] Zhu, Y., Zabaras, N., Koutsourelakis, P.-S., and Perdikaris, P., 2019, "Physics-Constrained Deep Learning for High-Dimensional Surrogate Modeling and Uncertainty Quantification Without Labeled Data," J. Comput. Phys., **394**, pp. 56–81.

[17] Yuan, B., Wang, H., Heitor, A., and Chen, X., 2024, "f-PICNN: A Physics-Informed Convolutional Neural Network for Partial Differential Equations With Space-Time Domain," J. Comput. Phys., **515**, p. 113284.

[18] Praditia, T., Karlbauer, M., Otte, S., Oladyshkin, S., Butz, M. V., and Nowak, W., 2022, "Learning Groundwater Contaminant Diffusion-Sorption Processes With a Finite Volume Neural Network," Water Resour. Res., **58**(12), p. 033149.

[19] Ramírez, L., Nogueira, X., Ouro, P., Navarrina, F., Khelladi, S., and Colominas, I., 2018, "A Higher-Order Chimera Method for Finite Volume Schemes," Arch. Comput. Methods Eng., **25**(3), pp. 691–706.

[20] Dumbser, M., and Balsara, D. S., 2016, "A New Efficient Formulation of the HLLEM Riemann Solver for General Conservative and Non-Conservative Hyperbolic Systems," J. Comput. Phys., **304**, pp. 275–319.

[21] Wei, Y.-X., Zhang, F., Liu, J., Su, H., and Xu, C., 2023, "A Constrained Boundary Gradient Reconstruction Method for Unstructured Finite Volume Discretization of the Euler Equations," Comput. Fluids, **252**, p. 105774.

[22] Zhan, N., Chen, R., and You, Y., 2023, "Three-Dimensional High-Order Finite-Volume Method Based on Compact WENO Reconstruction With Hybrid Unstructured Grids," J. Comput. Phys., **490**, p. 112300.

[23] Ge, L., Libin, L., Qingquan, L., Chun, F., and Xiaoliang, W., 2023, "A Unified Consistent Source Term Computational Algorithm for the γ-Based Compressible Multi-Fluid Flow Model," Comput. Fluids, **259**, p. 105899.

[24] Musa, O., Huang, G., Yu, Z., and Li, Q., 2020, "An Improved Roe Solver for High Order Reconstruction Schemes," Comput. Fluids, **207**, p. 104591.

[25] Zhang, H., Zhang, F., and Xu, C., 2019, "Towards Optimal High-Order Compact Schemes for Simulating Compressible Flows," Appl. Math. Comput., **355**, pp. 221–237.

[26] Schneider, K. A., Gallardo, J. M., Balsara, D. S., Nkonga, B., and Parés, C., 2021, "Multidimensional Approximate Riemann Solvers for Hyperbolic Nonconservative Systems. Applications to Shallow Water Systems," J. Comput. Phys., **444**, p. 110547.

[27] Yang, L. M., Shu, C., Wu, J., Liu, Y. Y., and Shen, X., 2022, "An Efficient Discrete Velocity Method With Inner Iteration for Steady Flows in All Flow Regimes," Phys. Fluids, **34**(2), p. 027110.

[28] Traore, B. B., Kamsu-Foguem, B., and Tangara, F., 2018, "Deep Convolution Neural Network for Image Recognition," Ecol. Inf., **48**, pp. 257–268.

[29] Yu, H., Yang, Z., Tan, L., Wang, Y., Sun, W., Sun, M., and Tang, Y., 2018, "Methods and Datasets on Semantic Segmentation: A Review," Neurocomputing, **304**, pp. 82–103.

[30] Goldberg, Y., 2016, "A Primer on Neural Network Models for Natural Language Processing," J. Artif. Intell. Res., **57**, pp. 345–420.

[31] Dourado, A., and Viana, F. A. C., 2020, "Physics-Informed Neural Networks for Missing Physics Estimation in Cumulative Damage Models: A Case Study in Corrosion Fatigue," ASME J. Comput. Inf. Sci. Eng., **20**(6), p. 061007.

[32] Faroughi, S. A., Pawar, N. M., Fernandes, C., Raissi, M., Das, S., Kalantari, N. K., and Mahjour, S. K., 2024, "Physics-Guided, Physics-Informed, and Physics-Encoded Neural Networks and Operators in Scientific Computing: Fluid and Solid Mechanics," ASME J. Comput. Inf. Sci. Eng., **24**(4), p. 040802.

[33] Islam, M., Chen, G., and Jin, S., 2019, "An Overview of Neural Network," Am. J. Neural Netw. Appl., **5**(1), pp. 7–11.

[34] Janssen, J. A., Haikal, G., DeCarlo, E. C., Hartnett, M. J., and Kirby, M. L., 2024, "A Physics-Informed General Convolutional Network for the Computational Modeling of Materials With Damage," ASME J. Comput. Inf. Sci. Eng., **24**(11), p. 111002.

[35] Bai, Y., Chaolu, T., and Bilige, S., 2021, "Solving Huxley Equation Using an Improved PINN Method," Nonlinear Dyn., **105**(4), pp. 3439–3450.

[36] Liu, D., Pusarla, P., and Wang, Y., 2023, "Multifidelity Physics-Constrained Neural Networks With Minimax Architecture," ASME J. Comput. Inf. Sci. Eng., **23**(3), p. 031008.

[37] Malashkhia, L., Liu, D., Lu, Y., and Wang, Y., 2022, "Physics-Constrained Bayesian Neural Network for Bias and Variance Reduction," ASME J. Comput. Inf. Sci. Eng., **23**(1), p. 011012.

[38] Rao, C., Sun, H., and Liu, Y., 2020, "Physics-Informed Deep Learning for Incompressible Laminar Flows," Theoret. Appl. Mech. Lett., **10**(3), pp. 207–212.

[39] Arzani, A., Wang, J.-X., and D'Souza, R. M., 2021, "Uncovering Near-Wall Blood Flow From Sparse Data With Physics-Informed Neural Networks," Phys. Fluids, **33**(7), p. 071905.

[40] Wassing, S., Langer, S., and Bekemeyer, P., 2024, "Physics-Informed Neural Networks for Parametric Compressible Euler Equations," Comput. Fluids, **270**, p. 106164.

[41] Jagtap, A. D., Mao, Z., Adams, N., and Karniadakis, G. E., 2022, "Physics-Informed Neural Networks for Inverse Problems in Supersonic Flows," J. Comput. Phys., **466**, p. 111402.

[42] Dwivedi, V., and Srinivasan, B., 2020, "Solution of Biharmonic Equation in Complicated Geometries With Physics Informed Extreme Learning Machine," ASME J. Comput. Inf. Sci. Eng., **20**(6).

[43] Li, M., and McComb, C., 2022, "Using Physics-Informed Generative Adversarial Networks to Perform Super-Resolution for Multiphase Fluid Simulations," ASME J. Comput. Inf. Sci. Eng., **22**(4), p. 044501.

[44] Peng, J.-Z., Hua, Y., Li, Y.-B., Chen, Z.-H., Wu, W.-T., and Aubry, N., 2023, "Physics-Informed Graph Convolutional Neural Network for Modeling Fluid Flow and Heat Convection," Phys. Fluids, **35**(8), p. 087117.
[45] Tong, Y., Xiong, S., He, X., Yang, S., Wang, Z., Tao, R., Liu, R., and Zhu, B., 2024, "RoeNet: Predicting Discontinuity of Hyperbolic Systems From Continuous Data," Int. J. Numer. Methods Eng., **125**(6), p. 7406.
[46] Ren, P., Rao, C., Liu, Y., Wang, J.-X., and Sun, H., 2022, "PhyCRNet: Physics-Informed Convolutional-Recurrent Network for Solving Spatiotemporal PDEs," Comput. Methods Appl. Mech. Eng., **389**, p. 114399.
[47] Siddique, N., Paheding, S., Elkin, C. P., and Devabhaktuni, V., 2021, "U-net and Its Variants for Medical Image Segmentation: A Review of Theory and Applications," IEEE Access, **9**, pp. 82031–82057.
[48] Olimov, B., Sanjar, K., Din, S., Ahmad, A., Paul, A., and Kim, J., 2021, "FU-Net: Fast Biomedical Image Segmentation Model Based on Bottleneck Convolution Layers," Multimedia Syst., **27**(4), pp. 637–650.
[49] Zhu, T., Zheng, Q., and Lu, Y., 2024, "Physics-Informed Fully Convolutional Networks for Forward Prediction of Temperature Field and Inverse Estimation of Thermal Diffusivity," ASME J. Comput. Inf. Sci. Eng., **24**(11), p. 111004.