Computers and Fluids 258 (2023) 105811

Contents lists available at ScienceDirect

Computers and Fluids

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

Neural vortex method: From finite Lagrangian particles to infinite dimensional Eulerian dynamics

Shiying Xiong [a,b,*], Xingzhe He [b], Yunjin Tong [b], Yitong Deng [b], Bo Zhu [b]

[a] Department of Engineering Mechanics, School of Aeronautics and Astronautics, Zhejiang University, Hangzhou 310027, China
[b] Department of Computer Science, Dartmouth College, Hanover 03755, NH, USA

ARTICLE INFO

Keywords:
Vortex method
Neural network
Lagrangian dynamics
Eulerian dynamics

ABSTRACT

In fluid analysis, there has been a long-standing problem: lacking a rigorous mathematical tool to map from a continuous flow field to finite discrete particles, hurdling the Lagrangian particles from inheriting the high resolution of a large-scale Eulerian solver. To tackle this challenge, we propose a novel learning-based framework, the neural vortex method (NVM). NVM builds a neural-network description of the Lagrangian vortex structures and their interaction dynamics to reconstruct the high-resolution Eulerian flow field in a physically-precise manner. The key components of our infrastructure consist of two networks: a vortex detection network to identify the Lagrangian vortices from a grid-based velocity field and a vortex dynamics network to learn the underlying governing interactions of these finite structures. By embedding these two networks with a vorticity-to-velocity Poisson solver and training its parameters using the fluid data obtained from grid-based numerical simulation, we can predict the accurate fluid dynamics on a precision level that was infeasible for all the previous conventional vortex methods. We demonstrate the efficacy of our method in generating highly accurate prediction results with low computational cost by predicting the evolution of the leapfrogging vortex rings system, the turbulence system, and the systems governed by Navier–Stokes (NS) equations with different external forces. We compare the prediction results made by NVM and the Lagrangian vortex method (LVM) for solving the NS equation in the periodic box and find that the relative error of the predicted velocity using NVM is more than 10 times lower than that of the LVM. Moreover, our method only requires data collected from a very short training window, more than 100 times smaller than the prediction period, which potentially facilitates data acquisition in real systems.

1. Introduction

Due to the high number of degrees of freedom in the motion space, the complex nonlinear coupling between particles, and the susceptibility to numerical dissipation, reproducing the behavior of fluids with such features in a physically-accurate manner presents a challenge. Two general approaches are developed to solve the fluid equations, one is grid-based, or Eulerian, and the other is particle-based, or Lagrangian. Conventionally, many pieces of research are done on grid-based methods mainly to achieve high resolution, for example on channel flows [1,2], boundary layers [3–5], isotropic turbulence [6–8], and pipe flows [9]. However, the main drawback of grid-based methods is their high computational cost that is unaffordable under many conditions [10]. On the contrary, the particle-based approach is efficient in computation but suffers from inaccuracy in results [11,12].

Lagrangian vortex methods (LVMs) are well-known particle-based methods which use vortices as computational elements, mimicking the physical structures in turbulence. In LVMs, the Navier–Stokes (NS) equations are formulated in terms of vorticity in contrast to the conventional formulations with velocity and pressure and are solved using a Lagrangian approach instead of the Eulerian formulation [13–16]. Discretizing the NS equations with finite particles results in a set of ordinary differential equations (ODEs) for the particle strengths and positions [17]. The Lagrangian method has some advantages, such as its automatic adaptivity of the computational elements, the numerical conservation of physically conserved quantities, the ability to simulate phenomena covering many orders of magnitude, and the rigorous treatment of boundary conditions at infinity, in the numerical simulation of fluids in unsteady flows where the vortex structures play a leading role. For many typical flows, such as wall-bounded flows [18,19], wake flows [20–22], and jet flows [23], their vorticity distributions are usually concentrated in certain regions, which enables us to distribute the density of the vortex elements precisely. Besides, with Kelvin's

* Corresponding author.
E-mail address: shiying.xiong@zju.edu.cn (S. Xiong).

https://doi.org/10.1016/j.compfluid.2023.105811
Received 18 July 2022; Received in revised form 20 November 2022; Accepted 30 January 2023
Available online 1 February 2023

circulation theorem directly embedded into the Lagrangian dynamics of the vortex elements [24], the conservativeness during the numerical calculation can be reasonably guaranteed. Finally, we can obtain the incompressible velocity field through the discrete Biot–Savart (BS) law using vortex elements, showing a weak coupling between evolutionary quantity and the solenoidal condition of velocity, which can be beneficial to maintain the computational stability in fluid simulations in the face of physical discontinuities. [25,26].

However, the implementation of the LVM faces a major challenge which is to model the right-hand sides (RHSs) of the set of ordinary differential equations based on the NS equations. Firstly, the assumption that the vortices are point-like largely limits the use of the continuous BS law. Second, the drift velocity due to the external force cannot be obtained using the LVM without knowing the function of the external force. Even given the function, the LVM still fails to capture the drift velocity accurately in most cases [24]. Finally, when two particles are close enough, the singularity of the discrete BS law leads to a significant numerical error. The above problems make the LVM inaccurate and inapplicable in solving the underlying fluid dynamics under many situations [14].

While the development of traditional numerical methods facing obstacles, fortunately in the last decade, with the drastic advancement in computational power and data availability, we are presented with the new hope of approaching these previously elusive systems from a new angle: a data-driven one powered by machine learning [27–31]. Since brute-force machine learning with conventional toolkits such as deep neural networks typically suffers from the high dimensionality of input–output spaces, expensive cost of data, the tendency to yield physically implausible results, and the inability to robustly handle extrapolation, recent research interests have been focused on embedding physical priors in learning algorithms so that the networks approach the data not as wide-eyed infants but as physicists familiar with the fundamental rules of how our world operates [32–37]. Recently, many pieces of research are developed to efficiently learn fluid dynamics by incorporating physical priors into the learning framework, e.g., encoding the NS equations [38], embedding the notion of an incompressible fluid [39], and identifying a mapping between the dynamics of wave-based physical phenomena and the computation in a recurrent neural network [40].

In the realm of learning complex fluid dynamics, various works have been proposed along this line of thinking, seeking to engrave the structure of partial differential equations (PDEs) into the network architectures [41–46]. Ideally, the PDEs should learn to evolve the flow field without consulting the particular solver to obtain initial-condition invariance. Still, due to the high dimensionality and the lack of supervision, the machine-learning community has not solved such a task to date.

In this work, we propose a novel approach, the neural vortex method (NVM), to embed prior physical knowledge to elegantly achieve invariance to initial conditions while at the same time being efficient in data usage, easy to implement, fast to compute, adaptable to arbitrarily high precision, and suitable for handling complex, unsteady flows. Our method is inspired by the vortex method in fluid dynamics, which discretizes the NS equations with a set of Lagrangian particles — vortices, based on Helmholtz's theorems which states that the behavior of the fluid can be described by several vortex elements flowing with the fluid [47]. We remark that accurate simulation of complex vortex dynamics requires a large number of Lagrangian vortex particles, considering the spatially continuous distribution of vortex structures. However, significant theoretical and numerical errors may occur if we only advect the fluid with a small number of vortex particles without considering vortex deformation, external forces, and viscous dissipation. We aim to establish a method to reconstruct continuous vortex dynamics with a small number of vortex particles. We model the position and dynamics of the vortex particles in a data-driven manner, and errors can be significantly reduced compared with a direct formula-based model.

In the same spirit as the LVM, our method learns to describe a complex fluid dynamics system by first learning to associate a small number of Lagrangian vortex particles to the observation and simulate the forward dynamics of these vortices instead. This approach can be viewed as identifying a compact, physics-based latent space where simulation can be performed efficiently. To view it another way, instead of tailoring our networks to solve the governing PDEs, we design our networks to describe the underlying behavior of our system that such equations are proven to imply. We show the superiority of our vortex-based approach to the previous approaches with its ability to generalize to different initial conditions, adapt to arbitrary precision, learn with small data sizes, and simulate complex turbulent flows.

Our method brings a new approach to identifying fluid systems exhibiting complex vortical motions. Still, its idea can be enlightening to other fields as well. We map continuous flow fields to low-dimensional Lagrangian vortex dynamics to study fluid evolution over time, which can also be generalized to common physical systems. In general, we can map the physical space described by PDE to a low-dimensional discrete space through a neural network to predict the time evolution of the transformed system and recover to the original physical space through neural networks or physical prior. We remark that to better preserve the critical structures, the intrinsic geometric symmetry should be taken into account when transforming the physical system into a different space. As incorporating physical priors is an imminent and promising trend, with this schematically novel approach, our work can potentially open up a new horizon for future endeavors.

The structure of this paper is as follows. In Section 2, we will first introduce the NS equations and LVM that serve as the mathematical foundation of our methodology. Then, we give an overview of our method in Section 3. The following Section 4 describes the details of our dataset generation and training settings. In Section 5, we show the numerical results, which compare our methodology with the Lagrangian method. Lastly, conclusions are drawn in Section 6, with an emphasis on the significance of our method.

2. NS equations and LVM

2.1. NS equations

Given a fluid velocity field $\boldsymbol{u}(\boldsymbol{x}, t)$ with an incompressible constraint, its underlying dynamics can be described by the NS equations

$$\begin{cases} \frac{D\boldsymbol{u}}{Dt} = -\frac{1}{\rho}\boldsymbol{\nabla} p + \nu \nabla^2 \boldsymbol{u} + \boldsymbol{f}, \\ \boldsymbol{\nabla} \cdot \boldsymbol{u} = 0, \end{cases} \tag{1}$$

where t denotes the time, $D/Dt = \partial/\partial t + \boldsymbol{u} \cdot \boldsymbol{\nabla}$ is the material derivative, p is the pressure, ν is the kinematic viscosity, ρ is the density, and $\boldsymbol{f}$ is the body accelerations (per unit mass) acting on the continuum, for example, gravity, inertial accelerations, electric field acceleration, and so on.

The alternative form of the NS equations could be obtained by defining the vorticity field $\boldsymbol{\omega} = \boldsymbol{\nabla} \times \boldsymbol{u}$, which leads to the following vorticity dynamical equation

$$\begin{cases} \frac{D\boldsymbol{\omega}}{Dt} = (\boldsymbol{\omega} \cdot \boldsymbol{\nabla})\boldsymbol{u} + \nu \boldsymbol{\nabla}^2 \boldsymbol{\omega} + \boldsymbol{\nabla} \times \boldsymbol{f}, \\ \nabla^2 \boldsymbol{\Psi} = -\boldsymbol{\omega}, \quad \boldsymbol{u} = \boldsymbol{\nabla} \times \boldsymbol{\Psi}, \end{cases} \tag{2}$$

where $\boldsymbol{\Psi}$ is a vector potential whose curl is the velocity field. Although this form does not seem to bring any simplification, the key illumination of doing this transformation stems the Helmholtz's theorems,[48] which states that the dynamics of the vorticity field can be described by vortex surfaces/lines, which are Lagrangian surfaces/lines flowing with the velocity field in inviscid flows [49,50].

2.2. LVM

The LVM discretizes the vorticity dynamical Eq. (2) with N particles resulting in a set of ODEs for the particle strengths $\boldsymbol{\Gamma} = \{\boldsymbol{\Gamma}_i | i = 1, \dots, N\}$ and the particle positions $\boldsymbol{X} = \{\boldsymbol{X}_i | i = 1, \dots, N\}$ as

$$\begin{cases} \frac{\mathrm{d}\boldsymbol{\Gamma}_i}{\mathrm{d}t} = \boldsymbol{\gamma}_i, \\ \frac{\mathrm{d}\boldsymbol{X}_i}{\mathrm{d}t} = \boldsymbol{u}_i + \boldsymbol{v}_i. \end{cases} \tag{3}$$

Here, the particle strength $\boldsymbol{\Gamma}_i$ is the integral of $\boldsymbol{\omega}$ over the ith computational element, $\boldsymbol{u}_i$ is the induced velocity calculated by BS law

$$\boldsymbol{u}_i = \frac{1}{2(n_d - 1)\pi} \sum_{j \neq i}^{N} \frac{\boldsymbol{\Gamma}_j \times (\boldsymbol{X}_i - \boldsymbol{X}_j)}{|\boldsymbol{X}_i - \boldsymbol{X}_j|^{n_d} + \mathcal{R}^{n_d}}, \tag{4}$$

where n_d is the dimension of the flow field. In addition, $\boldsymbol{\gamma}_i$ and $\boldsymbol{v}_i$ are the change rate of the particle strength and the drift velocity [24], respectively. To avoid singularities in the BS law, we introduce the numerical regularization parameter $\mathcal{R}$ in the LVM as $\mathcal{R} = 0.1$. The effect of the regularization parameter on the dynamics of the flow evolution of the simulated vortex particles is rather small because of the large spacing between the vortex particles.

In a two-dimensional ideal fluid flow, i.e., a strictly inviscid barotropic flow with conservative body forces, the movements of Lagrangian particles with conserved vorticity strength are determined by the velocity field they create, thus allowing us to advance the simulation temporally [14]. However, in the real three-dimensional flow, under the action of vortex stretching, vortex distortion, viscous dissipation, external forces, etc., the Lagrangian advection of vortex particles and their strength need to be corrected by γ_i and $\boldsymbol{v}_i$ in (3).

We remark that the NS equations can be accurately modeled by the Lagrangian vortex method with a large number of computational elements and a reasonable discrete distribution. Optimizing the discrete distribution and dynamics modeling by neural networks deserves further research.

3. NVM

3.1. Methodology

To quantify the fluid dynamics accurately and efficiently, we propose a novel framework, the NVM, that extracts information from the Eulerian specification of the flow field (or the images of flow visualizations) and translates it into knowledge about the underlying fluid field through physics-informed neural networks. To build a fully automated toolchain that can extract a high-resolution Eulerian flow field from the Lagrangian inductive priors, we embed these two networks with a vorticity-to-velocity Poisson solver, as shown in Fig. 1. The reason behind this structure is that learning directly from high-dimensional observations, such as images, is unable to be achieved using traditional methods since extracting the velocity and pressure fields directly from the images is challenging.

We construct a vortex detection network in Section 3.2 to identify the positions and the vorticity of Lagrangian vortices from a grid-based velocity field, which from a mathematical perspective connects (1) with (3). In this way, we simplify the vorticity field into a field only constituted of the identified vortices. Given the detected vortices, we then use a vortex dynamics network in Section 3.3 to learn the underlying governing dynamics of these finite structures. Dynamics networks accurately model the RHSs of (3) under various conditions, resolving the long-standing problem in LVM.

We train the parameters using high-fidelity data from high-resolution direct numerical simulation. The model is trained only with information collected from the interaction of 2 to 6 vortices. The trained model can be applied to any arbitrary vorticity field with any number of vortices. The training process of NVM consists of 2 major steps: detection network training and dynamics network training. We utilize data collected from randomly generated vortices and the corresponding vorticity fields to train the detection network. Direct numerical simulation (DNS) is used to calculate the data of the evolutionary vorticity fields. We use the well-trained detection network to identify vortices' positions and vorticity strength from the initial and the evolved vorticity fields. The identified vortices are then used to train the dynamics network.

3.2. Detection network

The input of the detection network is a vorticity field of size $200 \times 200 \times 1$. As shown in Fig. 2, we first feed the vorticity field into a small one-stage detection network and get the feature map of size $25 \times 25 \times 512$ (we downsampled 3 times). The detection network consists of a Conv2d-BatchNorm-ReLU combo and a 6-layer-structured ResBlock chain whose size can be adjusted dynamically to the complexity of the problem. The primary reason for downsampling is to avoid extremely unbalanced data and multiple predictions for the same vortex. We then forward the feature map to 2 branches. In the first branch, we conduct a 1×1 convolution to generate a probability score $\hat{p}$ of the possibility that there exists a vortex. If $\hat{p} > 0.5$, we believe there exists a vortex within the corresponding cells of the original $200 \times 200 \times 1$ vorticity field. In the second branch, we predict the relative position to the left-up corner of the cell of the feature map if the cell contains a vortex. Afterward, we set a bounding box of 10×10 around these predicted vortices and use the weighted average of the positions of the cells of the original vorticity field to find the exact position of the vortex. Finally, the vortex particle strength is calculated as the sum of the value of the cells in the bounding box normalized by the cell area.

In the training process, we penalize the wrong position detection only if the cell containing a vortex in the ground truth given by DNS is not detected. This idea is similar to the real-time object detection in Redmon et al. [52]. We do not use the weighted average method to find the position in the training to ensure the detection network can produce detection results as accurately as possible. We use the focal loss [53] to further relieve the unbalanced classification problem.

We mainly use the detection network to generate training data for the dynamics network because we want to use the high-resolution data generated by the method mentioned in Section 4.1 instead of by the approximate particle method (BS law). Moreover, there are many situations where BS law is inapplicable, as discussed previously in Section 1. The detection network enables us to find the positions of the vortices accurately regardless of the situation.

The detection network is responsible for providing necessary information to the dynamics network. After the training, we use the well-trained detection network to detect the vortices in the initial vorticity fields and the evolved vorticity field, both generated by the method in Section 4.1. We then apply the nearest-neighbor method to pair the vortices detected in these two fields. Fig. 3 shows the case of two fields at $t = 0$ and $t = 0.2$. The idea of nearest-neighbor pairing can be perceived from Fig. 3(c). The sample, or these two fields, is dropped if different numbers of vortices are detected in the initial and evolved fields or if a large difference exists in the vorticity of paired vortices. The successfully detected vortices in the initial and evolved vorticity fields are passed together into the dynamics network for its training.

We remark that our method does not venture into the detection of three-dimensional vortex structure. It is more challenging to localize vortices in complex three-dimensional structures exhibiting tubular or sheet-like morphology by neural networks. A 3-D vortex identification model requires a large number of vortex particles, to reduce its computational cost, a promising direction is to treat vortex elements as line segments with length and direction as in Xiong et al. [17].

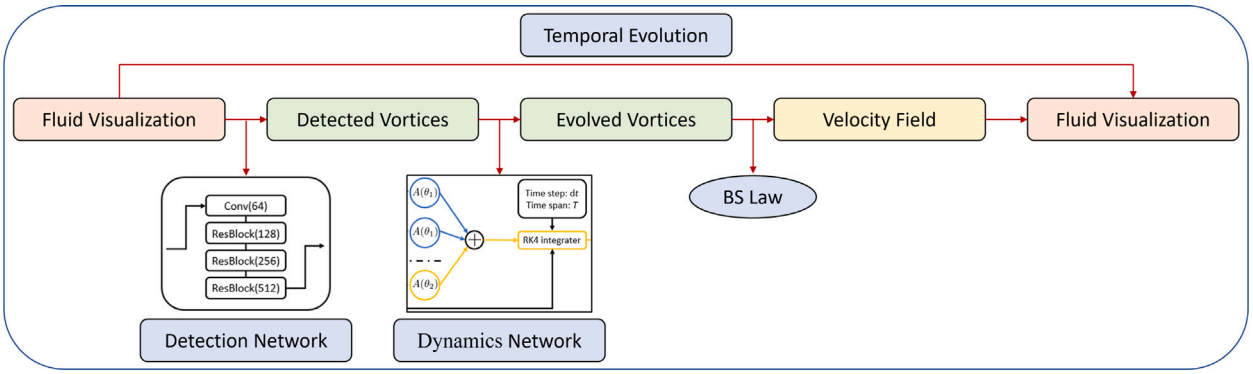

Fig. 1. Schematic diagram of the NVM. Our system is constituted of two networks, the detection network and the dynamics network, which are embedded with a vorticity-to-velocity Poisson solver.

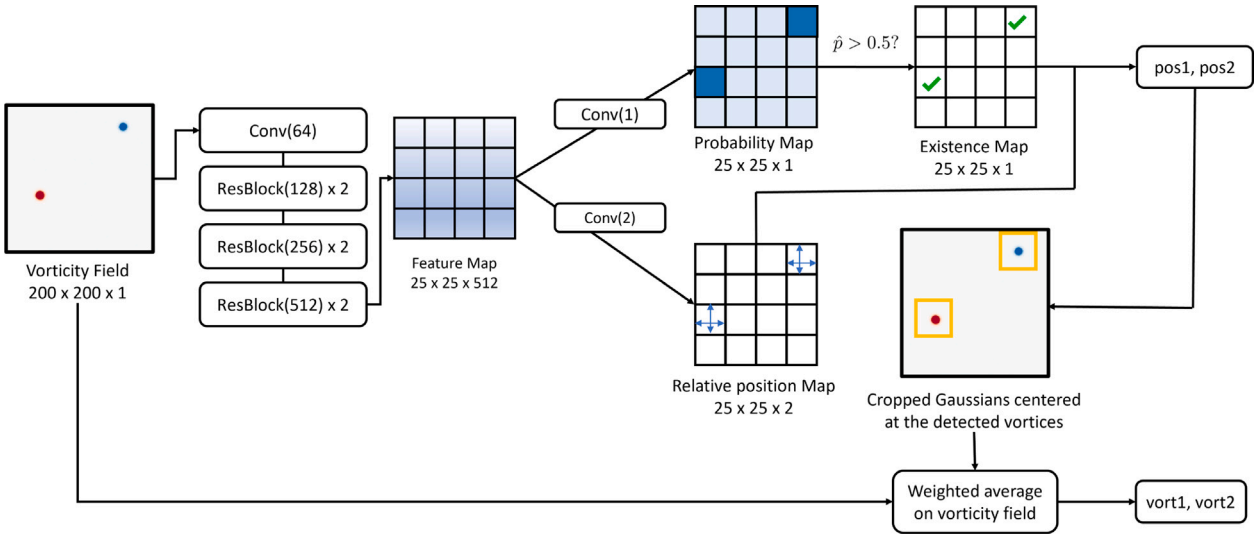

Fig. 2. The architecture of the detection network. It takes the vorticity field as input and outputs the position and vortex particle strength for each vortex detected. The *Conv* means the Conv2d-BatchNorm-ReLU combo, and the *ResBlock* is the same as in He et al. [51]. In each *ResBlock*, we use stride 2 to downsample the feature map. The Resblock chain is six-layer structured. The number in the parenthesis is the output dimension.

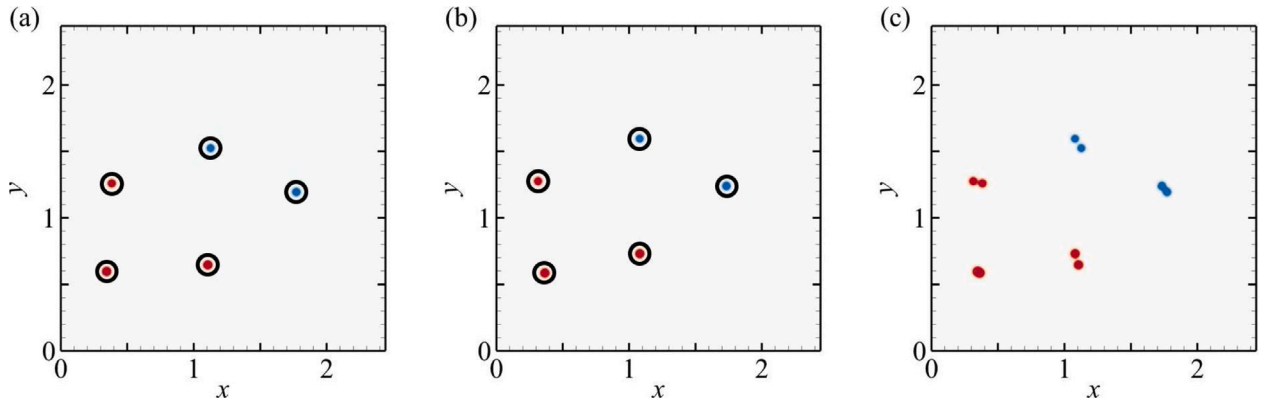

Fig. 3. A example of vorticity contour at (a) $t = 0$, (b) $t = 0.2$, and (c) superposition of $t = 0$ and $t = 0.2$. The black circles indicate the location recognized by the detection network. The evolution from (a) to (b) is calculated by DNS.

3.3. Dynamics network

To learn the underlying dynamics of the vortices, we build a graph neural network similar to Battaglia et al. [54]. We predict the velocity of one vortex due to influences exerted by the other vortices and the external force. Then we use the fourth-order Runge–Kutta integrator to calculate the position in the next timestamp. As shown in Fig. 4, for each vortex, we use a neural network $A(\theta_1)$ to predict the influences exerted by the other vortices and add them up. Specifically, for each ith vortex, we consider the vortex $j(j \neq i)$. The difference of their positions can be calculated by $\text{diff}_{ij} = \text{pos}_i - \text{pos}_j$, and their L2 distance is $\text{dist}_{ij} = \|\text{diff}_{ij}\|_2$. The input of the $A(\theta_1)$ is the vector $(\text{diff}_{ij}, \text{dist}_{ij}, \text{vort}_j)$ of length 4. Here, pos and vort are detected by the detection network. The output of $A(\theta_1)$ is a vector with the same dimension of the flow

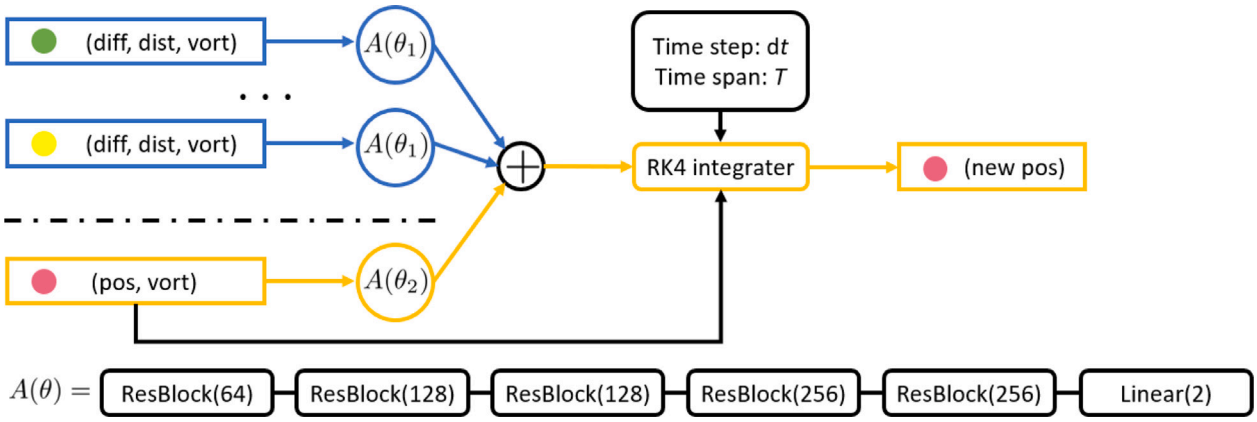

Fig. 4. The architecture of the dynamics network. It takes the particle's attribution as input and outputs each vortex's position. The *ResBlock* has the same architecture as in He et al. [51] with the convolution layers replaced by linear layers. The number in the parenthesis is the output dimension.

field, characterizing the induced velocity of the jth vortex to the ith vortex. In this way, we can calculate the induced velocity of each vortex j ($j \neq i$) on the vortex i. We sum up all the induced velocities on the vortex i and treat the result as the induced velocity exerted by the other vortices.

In addition, we use another neural network $A(\theta_2)$, to predict the influence caused by the external force, which is determined by the local vorticity and the position of the vortex. The input of $A(\theta_2)$ is a vector of length 3. The output is the influence exerted by the environment on the vortex i, i.e., the induced velocity of the external force to ith vortex.

The reason we separate the induced velocity into two parts, i.e., $A(\theta_1)$ and $A(\theta_2)$, is as follows. On the one hand, the induced velocities between vortex particles are global, and exhibit a certain symmetry, i.e., the vortex particles interact with each other following the same law. In contrast, the influence of external forces on vortex particles is usually local and direct; thus, we do not need to consider the interaction between particles. The effect of the vortex stretching term in three-dimensional vortex flows or diffusion term in viscous flows is also local and should be included in network $A(\theta_2)$. Note that both the outputs of $A(\theta_1)$ and $A(\theta_2)$ are a vector with the same dimension of the flow field. Thus, we can add the two kinds of influence together, whose result is defined as the velocity of the vortex i. We feed the velocity into the fourth-order Runge–Kutta integrator to obtain the predicted position of vortex i.

In addition, in predicting the evolution of the flow field, NVM replaces the discrete BS method with a dynamics network composed of ResBlocks. We chose a 5-layer ResBlocks to improve the expressiveness of the dynamics network so that we can learn dynamics of different complexity on the same network. Since the dynamics network with 5-layer ResBlocks is more complex than the discrete BS method, the computational cost of NVM is higher than that of the Lagrangian vortex method. We remark that although the computational cost of ResBlocks itself is relatively large in NVM, the number of vortex particles needed to predict the evolution of the flow field using NVM is much smaller. Therefore, the overall computational cost of NVM can be greatly reduced.

4. Dataset generation and training settings

4.1. Dataset generation and training settings

We randomly sample 2 to 6 vortices and create the initial vorticity field through convolution with a Gaussian kernel $\sim \mathcal{N}(0, 0.01)$. This process is repeated 2000 times to generate $N_s = 2000$ samples. DNS is performed to solve (1) in the periodic box using a standard pseudo-spectral method [55]. Aliasing errors are removed using the two-thirds truncation method with the maximum wavenumber $k_{\max} \approx N/3$. The Fourier coefficients of the velocity are advanced in time using a second-order Adams–Bashforth method. The time step is chosen to ensure that the Courant–Friedrichs–Lewy number is less than 0.5 for numerical stability and accuracy. To obtain accurate DNS data samples, we set the grid size as $N = 1024$. Regarding the kinematic viscosity, we set $\nu = 0$ for the cases in Section 5.4 and $\nu = 0.001$ in the other cases. The pseudo-spectral method used in this DNS is similar to that described in Xiong and Yang [50,56,57].

We use $N_{train} = 0.9N_s = 1600$ samples with the time span T_{train} for the training of the dynamics network. The DNS dataset is generated with random initial conditions independent of the predicted vortex evolution. The time step of vortex evolution is set as dt. For the leapfrog example, we set the parameters as $T_{train} = 1$ and d$t = 0.001$. For the turbulent flow example, we set the parameters as $T_{train} = 0.001$ and d$t = 0.001$. For other examples, the parameters are set as $T_{train} = 0.2$ and d$t = 0.1$. In general, the parameters are chosen within a wide range, indicating the robustness of the network. We use the trained network to predict the vortex dynamics at time $T_{predict}$. We show that the prediction time span $T_{prediction}$ is larger than the training time span T_{train} in the results section, in some cases up to tens of times of T_{train}.

For both the detection network and the dynamics network, we use Adam optimizer [58] with a learning rate of 1e−3. The learning rate decays every 20 epochs by a multiplicative factor of 0.8. For the detection network, we use a batch size of 32 and train it for 350 epochs. We use the cross entropy as the classification loss and L1 loss for position prediction. To relieve the unbalanced data problem in the detection network, we implement Focal loss [53] with $\alpha = 0.4$ and $\gamma = 2$. It takes 15 min to converge on a single Nvidia RTX 2080Ti GPU. For the dynamics network, we use a batch size of 64 and train it for 500 epochs. We use L1 loss for position prediction. It takes 25 min to converge on a single Nvidia RTX 2080Ti GPU.

5. Results

We demonstrate the efficacy of our method in generating highly accurate prediction results, with low computational cost, for the leapfrogging vortex rings system, the turbulence system, and the systems governed by NS equations with different external forces that are challenging to model for the LVM.

5.1. Comparison between NVM and LVM

To demonstrate that NVM is a better approach to capturing fluid dynamics than the traditional LVM, we compare the prediction results of the NVM and the LVM for solving NS equations in the periodic box. In the prediction, we initialize two vortex particles at $\boldsymbol{X}_1 = (\pi - 0.4, \pi - 0.6)$ and $\boldsymbol{X}_2 = (\pi + 0.4, \pi + 0.6)$, where the corresponding particle strength

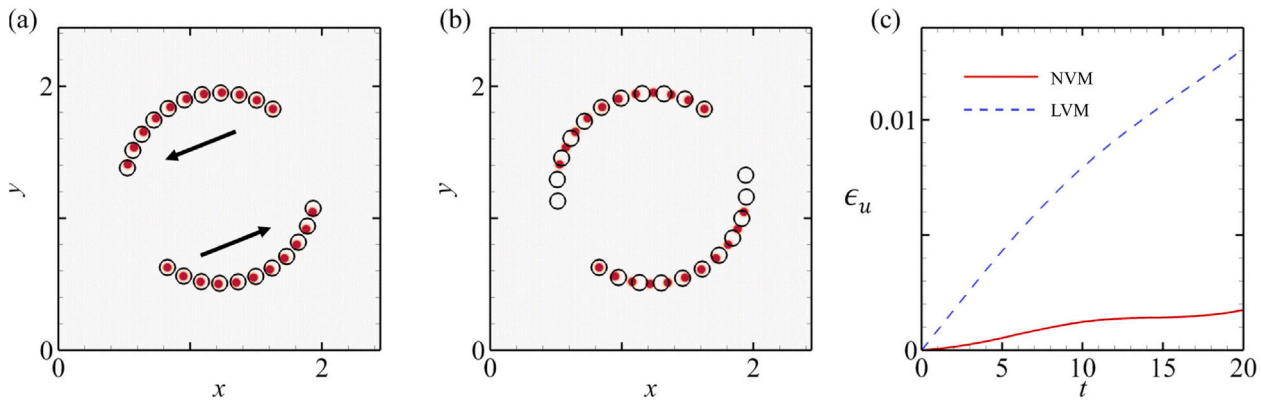

Fig. 5. Comparison of NVM and LVM for solving NS equations in the periodic box. (a) NVM, (b) LVM, and (c) The relative error of velocity in flow simulation. The red dots indicate the positions of 2 vortices at different time steps generated by DNS. The black circles in (a) and (b) are the prediction and simulation results of the NVM and LVMs, respectively. The black arrows indicate the directions of the motions of the 2 vortices.

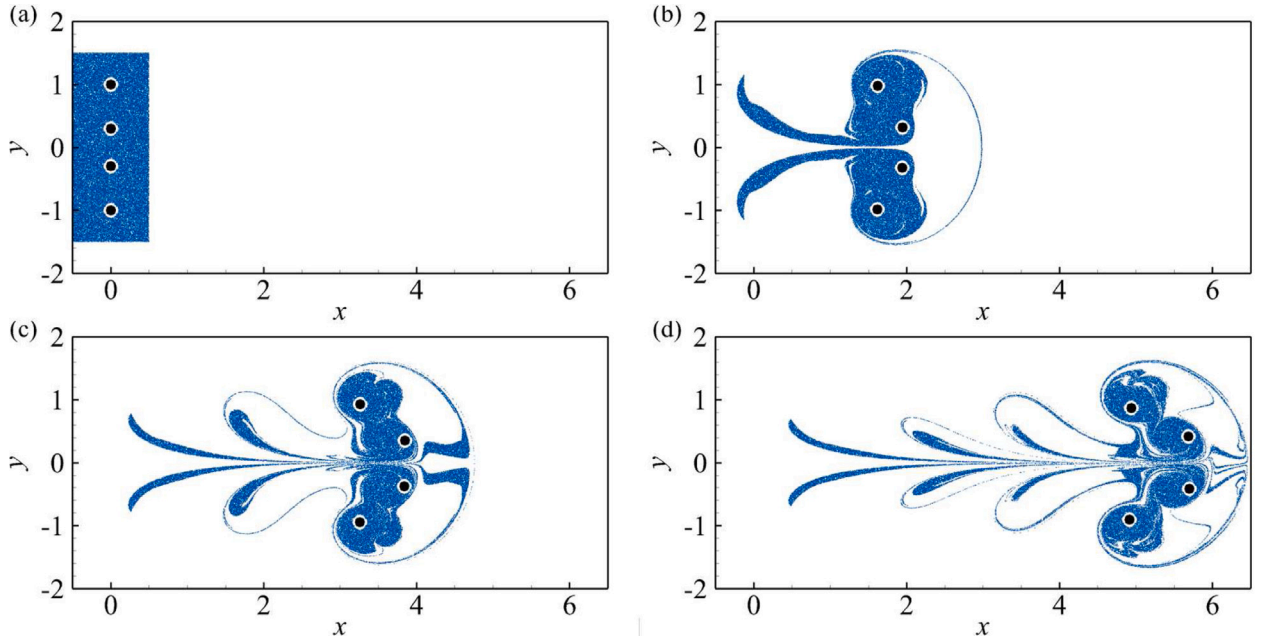

Fig. 6. Vorticity field predicted by 4 NVM vortices under the initial condition of leapfrogging vortex rings at $t = 0$, $t = 11$, $t = 22$, and $t = 33$. Vortices are indicated by the white–black circles. For better visualization, we use 80000 tracers.

are $\Gamma_1 = 0.75$ and $\Gamma_2 = 0.75$. We plot the results using the NVM and LVM and the relative error of velocity in the simulation in Fig. 5(a), (b), and (c), respectively. Here, the relative error of velocity is defined as

$$\epsilon_u = \frac{\|\boldsymbol{u}_{predict} - \boldsymbol{u}_{true}\|_{L^2}}{\|\boldsymbol{u}_{true}\|_{L^2}}, \tag{5}$$

where $\boldsymbol{u}_{predict}$ denotes the predicted or simulated solution and $\boldsymbol{u}_{true}$ denote the ground truth solution. It is quite obvious that in Fig. 5(a), the predictions made by NVM match the positions of vortices generated by DNS almost perfectly, while the predictions made by BS law in Figs. 5 (b) contain a large error. The divergence of the relative error of velocity is shown in Fig. 5(c), which shows that the NVM outperforms traditional methods by increasing amounts as the predicting period becomes longer.

5.2. Leapfrog vortex

A classic example of interesting vortex dynamics is the leapfrogging vortex rings, which is an axisymmetric laminar flow. This example demonstrates our method's ability to keep the fluid field's symmetric structure. Here, we use NVM to predict the motions of 4 vortices and add 80000 randomly initialized tracers for better visualization. Since the tracers do not affect the dynamics of the underlying vorticity field, we use BS law to calculate the motions of these tracers for faster visualization. In the prediction, we initialize four vortex particles at $\boldsymbol{X}_1 = (0, 1)$, $\boldsymbol{X}_2 = (0, -1)$, $\boldsymbol{X}_3 = (0, 0.3)$, and $\boldsymbol{X}_4 = (0, -0.3)$, respectively, where the corresponding particle strength are $\Gamma_1 = 0.8$, $\Gamma_2 = -0.8$, $\Gamma_3 = 0.8$, and $\Gamma_4 = -0.8$. Although the time span of the training samples is $T_{train} = 1$, we predict the evolution of vortex dynamics with a time span $T_{predict}$ of more than 30. Fig. 6 shows the evolution of the vorticity field predicted by NVM under the initial condition of leapfrogging vortex rings at $t = 0$, $t = 11$, $t = 22$, and $t = 33$. NVM accurately captures the symmetric structure of the leapfrogging vortex rings without losing such features as time evolves. In addition, the velocity field at the $t = 10$ is visualized in Fig. 7. Here we show the spatial distribution of the direction and magnitude of the velocity field. We can further see that the NVM maintains the intrinsic symmetry of the leapfrog vortex flow.

In the vortex evolution simulated by the DNS grid method, the spatially continuous vortices are distorted into spiral structures during the interaction, and thus the mutual induction velocity of leapfrog

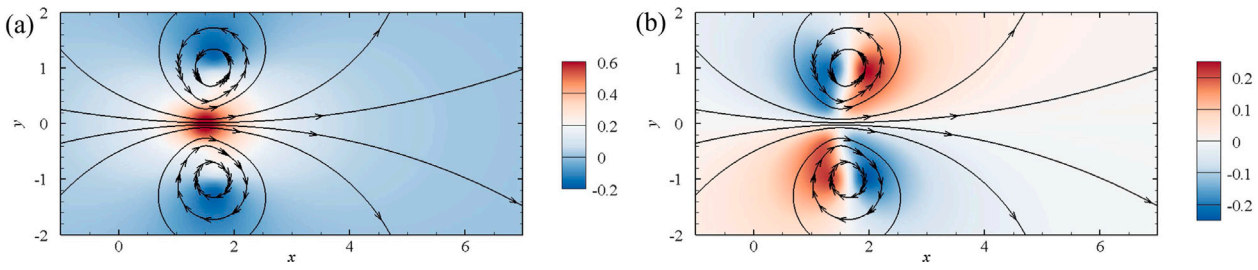

Fig. 7. Velocity field of the leapfrog vortex at $t = 10$. The black arrow lines are the fluid streamlines. The contours represent the velocity components in (a) x-direction and (b) y-direction, respectively.

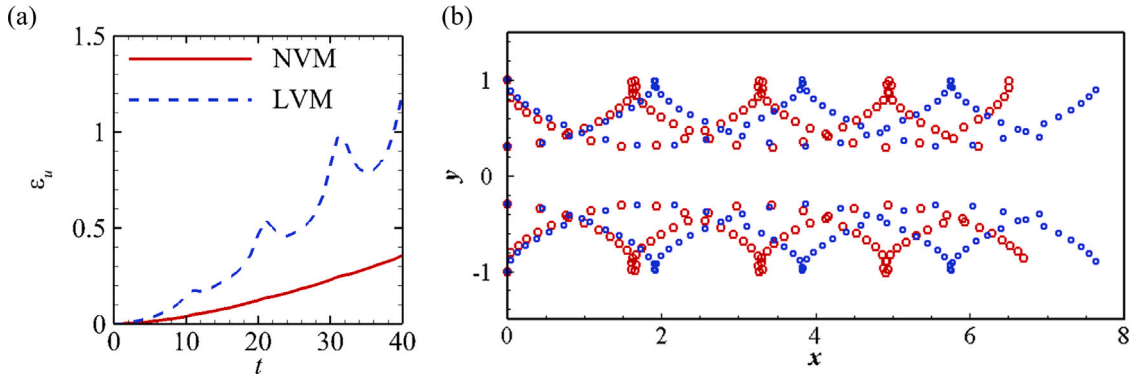

Fig. 8. Comparison of NVM and LVM for solving leapfrog vortex flow with same number of vortex particles. (a) The relative error of velocity in flow simulation. (b) The trajectories of four vortex particles at different times from $t = 0$ to 40.

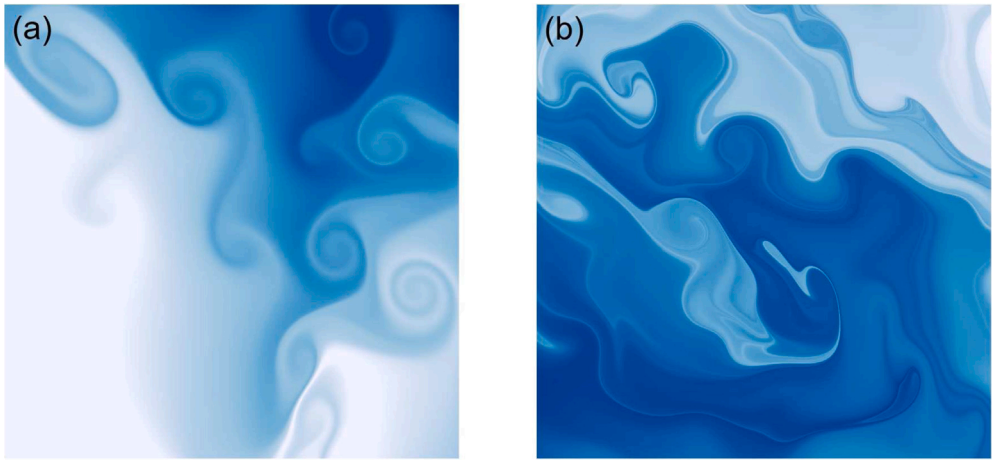

Fig. 9. Two-dimensional Lagrangian scalar fields at $t = 1$ with the initial condition $\phi = x$ and resolution 2000^2. The evolution of the Lagrangian scalar fields is induced by (a) $O(10)$ and (b) $O(100)$ random NVM vortex particles.

vortices is slowed down accordingly. The extent of this evolution cannot be achieved by modeling the mathematical equations of only a few vortex particles. In Fig. 8, we compare NVM and LVM to solve leapfrog vortex flow. LVMs require a large number of vortex particles to accurately simulate such vortex interactions. We use neural networks to learn the intrinsic dynamics to accurately reconstruct the complex vortex dynamics with a small number of vortex particles in a data-based manner.

5.3. Turbulent flows

Besides simple systems like leapfrogging vortex rings, NVM is capable of predicting complicated turbulence systems. This example's primary purpose is to illustrate our network's ability to handle more complex problems.

Fig. 9 depicts the two-dimensional Lagrangian scalar fields at $t = 1$ with the initial condition $\phi = x$ and resolution 2000^2. The governing equation of the Lagrangian scalar fields is

$$\frac{\partial \phi}{\partial t} + \boldsymbol{u} \cdot \boldsymbol{\nabla} \phi = 0. \tag{6}$$

The evolution of the Lagrangian scalar fields is induced by $O(10)$ and $O(100)$ NVM vortex particles at random positions $\backsim U(0, 4)$ with random strengths $\backsim U(0, 2)$. We remark that the same trained model is used for both cases. There is no correlation between the positions and vortex particle strengths of the two sets of vortex particles.

Based on the particle velocity field from the NVM, a backward-particle-tracking method is applied to solve (6). Then the iso-contour of the Lagrangian field can be extracted as material structures in the evolution [59–63]. In Fig. 9(a), the spiral structure [64,65] of individual NVM vortex particles can be observed clearly due to the

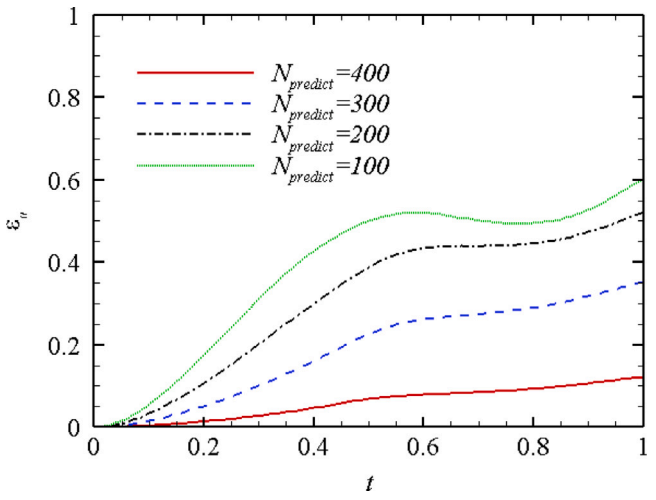

Fig. 10. The relative errors of velocity for predicting the problem in Fig. 9(b) with different particle sizes. Here $N_{predict}$ is the number of NVM particles used to predict the fluid evolution.

small number of NVM vortex particles. In Fig. 9(b), the underlying field exhibits turbulent behaviors since it is generated with a large number of NVM vortex particles.

Generally, the high-resolution results shown in Fig. 9 can only be achieved by supercomputation using grid-based methods [59], while NVM allows these to be generated on any laptop with GPU. We demonstrate that NVM is capable of generating an accurate depiction of complex turbulence systems with low computational costs.

We show in Fig. 10 the relative errors of velocity for predicting the problem in Fig. 9(b) with different particle sizes. It can be seen that the flow evolution predicted by the NVM converges to the exact solution simulated by the DNS grid method as the number of NVM particles increases.

5.4. NS equations with different external forces

In Fig. 11, we show NVM's ability to stably make accurate predictions of fluid dynamics governed by NS equations with different external forces, which are

$$\begin{cases} \boldsymbol{f} = \boldsymbol{0}, \\ \boldsymbol{f} = (0.05\omega, 0), \\ \boldsymbol{f} = 0.02\omega[\cos(x - x_c), -\sin(y - y_c)], \end{cases} \tag{7}$$

where ω represents the vorticity, and $(x_c, y_c) = (\pi, \pi)$ is the center of the computation domain. In the prediction, we initialize four vortex particles at $\boldsymbol{X}_1 = (\pi - 0.3, \pi - 0.5)$, $\boldsymbol{X}_2 = (\pi + 0.3, \pi - 0.4)$, $\boldsymbol{X}_3 = (\pi + 0.3, \pi + 0.5)$, and $\boldsymbol{X}_4 = (\pi - 0.3, \pi + 0.4)$, where the corresponding particle strength are $\Gamma_1 = 0.75$, $\Gamma_2 = -0.75$, $\Gamma_3 = 0.75$, and $\Gamma_4 = -0.75$. Here, we did not plot the results generated by LVM because LVM is incapable of solving the fluid dynamics with nontrivial external forces since it fails to capture the drift velocity $\boldsymbol{v}_i$ caused by the external force.

We compare the relative errors of velocity in Fig. 12. It shows that for the case in Fig. 12(a) without external force, NVM and LVM behave similarly, which is also related to the relatively short evolution time. Despite this, we can see that LMV shows a rapid error increase in Figs. 12(b) and 12(c).

6. Conclusion

We propose a novel learning-based framework, NVM, which builds a neural-network description of the Lagrangian vortex structures and their interaction dynamics to reconstruct the high-resolution Eulerian flow field in a physically-precise manner. We demonstrate the efficacy of our method in generating highly accurate prediction results, with low computational cost, for the leapfrogging vortex rings system, the turbulence system, and the systems governed by NS equations with different external forces. Our method is the first approach to utilizing the motions of finite particles to learn infinite dimensional dynamical systems. Featured by its unique ability to generate highly accurate prediction results with low computational cost, NVM marks a significant advancement in numerical fluid simulation.

Our method has the following limitations. We train the network with a vorticity field dataset rather than an experimentally measured velocity field dataset. While ideally the latter is an interesting and promising study worth exploring in the future, given that more extensive datasets and complex network structures are probably required to train the network using velocity field datasets, we have not ventured into this area yet. In addition, for the particle system in our neural network, the computational cost $O(N^2)$ of the particle interactions limits larger-scale numerical simulations.

In the future, we will explore a broader range of applications of NVM. Our method only considers two-dimensional flows. In fact, natural flows have high-dimensional complexity and pose a more significant challenge to the training of detection neural networks. Additionally, for flows with stationary or moving boundaries, the vortex structures are concentrated near a thin boundary layer, and we should design neural networks that take into account the boundaries in a general network framework. Moreover, since our detection network takes pictures as input, an interesting direction is to use high-resolution experimental flow fields as training datasets and use the trained model to predict fluid evolution.

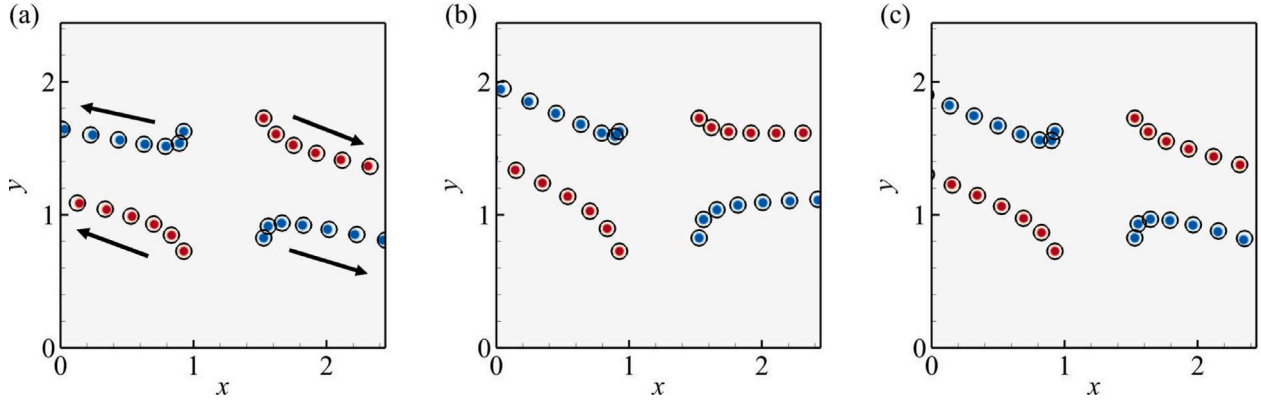

Fig. 11. Prediction results using NVM for three cases with (a) $\boldsymbol{f} = \boldsymbol{0}$, (b) $\boldsymbol{f} = (0.05\omega, 0)$, and (c) $\boldsymbol{f} = 0.02\omega[\cos(x - x_c), -\sin(y - y_c)]$. The black arrows indicate the directions of the motions of the 2 vortices. ω represents the vorticity, and (x_c, y_c) is the center of the computation domain.

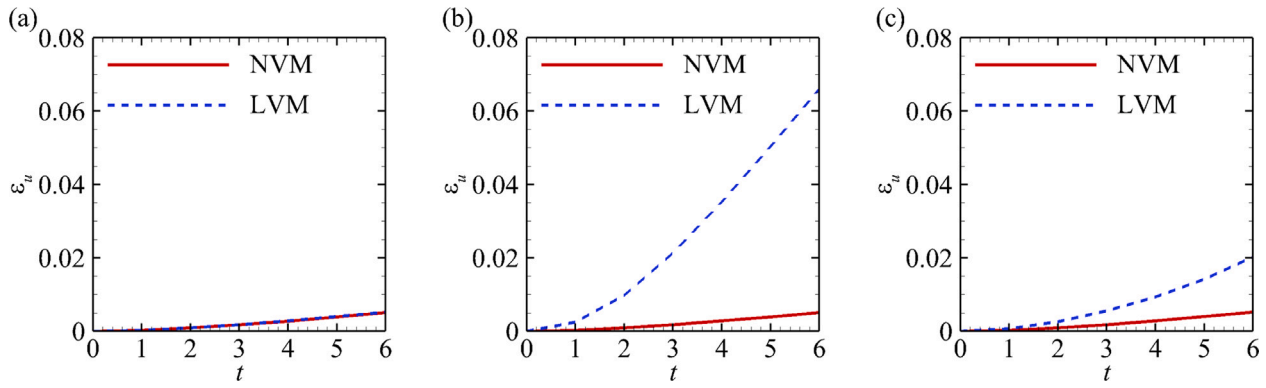

Fig. 12. The comparison of relative error of velocity in flow simulation using LVM and NVM for three cases with (a) $\boldsymbol{f} = \mathbf{0}$, (b) $\boldsymbol{f} = (0.05\omega, 0)$, and (c) $\boldsymbol{f} = 0.02\omega[\cos(x-x_c), -\sin(y-y_c)]$.

CRediT authorship contribution statement

Shiying Xiong: Methodology, Software, Validation, Writing – original draft, Writing – review & editing, Visualization. **Xingzhe He:** Software, Writing – original draft. **Yunjin Tong:** Writing – original draft, Visualization. **Yitong Deng:** Writing – original draft, Visualization. **Bo Zhu:** Supervision, Project administration, Funding acquisition, Writing – original draft.

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.

Data availability

Data will be made available on request.

Acknowledgments

We acknowledge NSF-1919647, 2106733, 2144806, and 2153560 for funding support.

References

[1] Kim J, Moin P, Moser RD. Turbulent statistics in fully developed channel flow at low Reynolds number. J Fluid Mech 1987;177:133–66.
[2] Zhou J, Adrian RJ, Balachandar S, Kendall TM. Mechanisms for generating coherent packets of hairpin vortices in channel flow. J Fluid Mech 1999;387:353–96.
[3] Lu P, Wang Z, Chen L, Liu C. Numerical study on U-shaped vortex formation in late boundary layer transition. Comput Fluids 2012;55:36–47.
[4] Wu X, Moin P, Wallace JM, Skarda J, Lozano-Durán A, Hickey J. Transitional–turbulent spots and turbulent–turbulent spots in boundary layers. Proc Natl Acad Sci USA 2017;114:E5292–9.
[5] Sayadi T, Hamman CW, Moin P. Direct numerical simulation of complete H-type and K-type transitions with implications for the dynamics of turbulent boundary layers. J Fluid Mech 2013;724:480–509.
[6] Ishihara T, Morishita K, Yokokawa M, Uno A, Kaneda Y. Energy spectrum in high-resolution direct numerical simulations of turbulence. Phys Rev Fluids 2016;1:082403.
[7] Jimenez J, Wray AA, Saffman PG, Rogallo RS. The structure of intense vorticity in isotropic turbulence. J Fluid Mech 1993;255:65–90.
[8] Jimenez J, Wray AA. The structure of intense vorticity in isotropic turbulence. J Fluid Mech 1993;373:255–85.
[9] Wu X, Moin P, Adrian RJ, Baltzer JR. Osborne Reynolds pipe flow: Direct simulation from laminar through gradual transition to fully developed turbulence. Proc Natl Acad Sci USA 2015;112:7920–4.
[10] Xiong S, Wang Z, Wang M, Zhu B. A Clebsch method for free-surface vortical flow simulation. ACM Trans Graph 2022;41(4).
[11] Lindsay K, Krasny R. A particle method and adaptive treecode for vortex sheet motion in three-dimensional flow. J Comput Phys 2001;172:879–907.
[12] Losasso F, Talton J, Kwatra N, Fedkiw R. Two-way coupled SPH and particle level set fluid simulation. IEEE Trans Vis Comput Graphics 2008;14:797–804.
[13] Leonard A. Vortex methods for flow simulation. J Comput Phys 1980;37:289–335.
[14] Cottet G, Koumoutsakos P. Vortex methods: theory and practice. Cambridge University Press; 2000.
[15] Anderson C, Greengard C. On vortex methods. SIAM J Numer Anal 1985;22:413–40.
[16] Selle A, Rasmussen N, Fedkiw R. A vortex particle method for smoke, water and explosions. ACM Trans Graph 2005;3:910–4.
[17] Xiong S, Tao R, Zhang Y, Feng F, Zhu B. Incompressible flow simulation on vortex segment clouds. ACM Trans Graph 2021;40:98.
[18] Zhao Y, Xiong S, Yang Y, Chen S. Sinuous distortion of vortex surfaces in the lateral growth of turbulent spots. Phys Rev Fluids 2018;3:074701.
[19] Renzo M, Urzay J. Direct numerical simulation of a hypersonic transitional boundary layer at suborbital enthalpies. J Fluid Mech 2021;912:A29.
[20] Tong W, Yang Y, Wang S. Characterizing three-dimensional features of vortex surfaces in the flow past a finite plate. Phys Fluids 2020;32:011903.
[21] Mimeau C, Marié S, Mortazavi I. A comparison of semi-Lagrangian vortex method and lattice Boltzmann method for incompressible flows. Comput Fluids 2021;224:104946.
[22] Karimi MS, Raisee M, Farhat M, Hendrick P, Nourbakhsh A. On the numerical simulation of a confined cavitating tip leakage vortex under geometrical and operational uncertainties. Comput Fluids 2021;220:104881.
[23] You J, Yang Y, Pope SB. Effects of molecular transport in LES/PDF of piloted turbulent dimethyl ether/air jet flames. Combust Flame 2017;176:451–61.
[24] Hao J, Xiong S, Yang Y. Tracking vortex surfaces frozen in the virtual velocity in non-ideal flows. J Fluid Mech 2019;863:513–44.
[25] Saye R. Interfacial gauge methods for incompressible fluid dynamics. Sci Adv 2016;2:e1501869.
[26] Yang S, Xiong S, Zhang Y, Feng F, Liu J, Zhu B. Clebsch gauge fluid. ACM Trans Graph 2021;40:99.
[27] Raissi M, Perdikaris P, Karniadakis GE. Multistep neural networks for data-driven discovery of nonlinear dynamical systems. 2018, arXiv preprint arXiv:1801.01236.
[28] González-García R, Rico-Martínez R, Kevrekidis IG. Identification of distributed parameter systems: A neural net based approach. Comput Chem Eng 1998;22:S965–8.
[29] Duraisamy K, Iaccarino G, Xiao H. Turbulence modeling in the age of data. Annu Rev Fluid Mech 2019;51:357–77.
[30] Xie Y, Franz E, Chu M, Thuerey N. Tempogan: A temporally coherent, volumetric gan for super-resolution fluid flow. ACM Trans Graph 2018;37(4):1–15.
[31] Chu M, Thuerey N. Data-driven synthesis of smoke flows with CNN-based feature descriptors. ACM Trans Graph 2017;36(4):1–14.
[32] Anderson J, Kevrekidis I, Rico-Martinez R. A comparison of recurrent training algorithms for time series analysis and system identification. Comput. Chem. Eng. 1996;20:S751–6.
[33] Crutchfield JP, McNamara BS. Equations of motion from a data series. Complex Systems 1987;1(417–452):121.
[34] Daniels BC, Nemenman I. Automated adaptive inference of phenomenological dynamical models. Nature Commun 2015;6(1):1–8.
[35] Wang J, Wu J, Xiao H. Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data. Phys Rev Fluids 2017;2(3):034603.
[36] Hammond J, Montomoli F, Pietropaoli M, Sandberg RD, M. V. Machine learning for the development of data-driven turbulence closures in coolant systems. J Turbomach 2022;144(8):081003.
[37] Xu X, Waschkowski F, Ooi AS, Sandberg RD. Towards robust and accurate Reynolds-averaged closures for natural convection via multi-objective CFD-driven machine learning. Int J Heat Mass Transfer 2022;187:122557.
[38] Raissi M, Karniadakis GE. Hidden physics models: Machine learning of nonlinear partial differential equations. J Comput Phys 2018;357:125–41.

[39] Mohan AT, Lubbers N, Livescu D, Chertkov M. Embedding hard physical constraints in convolutional neural networks for 3D turbulence. In: International conference on learning representations. 2020.
[40] Hughes TW, Williamson IAD, Minkov M, Fan S. Wave physics as an analog recurrent neural network. Sci Adv 2019;5:6946.
[41] Yang X, Zafar S, Wang J, Xiao H. Predictive large-eddy-simulation wall modeling via physics-informed neural networks. Phys Rev Fluids 2019;4:034602.
[42] Raissi M, Yazdani A, Karniadakis GE. Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. Science 2020;367(6481):1026–30.
[43] Belbute-Peres F, Economon T, Kolter Z. Combining differentiable PDE solvers and graph neural networks for fluid flow prediction. In: International Conference on Machine Learning. 2020, p. 2402–11.
[44] Lye K, Mishra S, Ray D. Deep learning observables in computational fluid dynamics. J Comput Phys 2020;410:109339.
[45] White C, Ushizima D, Farhat C. Neural networks predict fluid dynamics solutions from tiny datasets. 2019, arXiv preprint arXiv:1902.00091.
[46] Mohan AT, Lubbers N, Livescu D, Chertkov M. Embedding hard physical constraints in neural network coarse-graining of 3d turbulence. 2020, arXiv preprint arXiv:2002.00021.
[47] Barba LA. Vortex method for computing high-reynolds number flows: increased accuracy with a fully mesh-less formulation (Ph.D. thesis), California Institute of Technology; 2004.
[48] Helmholtz H. Uber integrale der hydrodynamischen Gleichungen welche den Wirbel-bewegungen ensprechen. J Reine Angew Math 1858;55:25–55.
[49] Yang Y, Pullin DI. On Lagrangian and vortex-surface fields for flows with Taylor–Green and Kida–Pelz initial conditions. J Fluid Mech 2010;661:446–81.
[50] Xiong S, Yang Y. The boundary-constraint method for constructing vortex-surface fields. J Comput Phys 2017;339:31–45.
[51] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on computer vision and pattern recognitionn. 2016, p. 770–8.
[52] Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on computer vision and pattern recognitionn. 2016, p. 779–88.
[53] Lin T, Goyal P, Girshick R, He K, Dollár P. Focal loss for dense object detection. IEEE Trans Vis Comput Graphics 2017;2980–8.
[54] Battaglia P, Pascanu R, Lai M, Rezende DJ. Interaction networks for learning about objects, relations and physics. In: Advances in neural information processing systems. 2016, p. 4502–10.
[55] Rogallo R. Numerical experiments in homogeneous turbulence. 1981.
[56] Xiong S, Yang Y. Construction of knotted vortex tubes with the writhe-dependent helicity. Phys. Fluids 2019;31:047101.
[57] Xiong S, Yang Y. Effects of twist on the evolution of knotted magnetic flux tubes. J Fluid Mech 2020;895:A28.
[58] Kingma DP, Ba J. Adam: A method for stochastic optimization. 2014, arXiv:1412.6980.
[59] Yang Y, Pullin DI, Bermejo-Moreno I. Multi-scale geometric analysis of Lagrangian structures in isotropic turbulence. J Fluid Mech 2010;654:233–70.
[60] Yang Y, Pullin DI. Geometric study of Lagrangian and Eulerian structures in turbulent channel flow. J Fluid Mech 2011;674:67–92.
[61] Zhao Y, Yang Y, Chen S. Evolution of material surfaces in the temporal transition in channel flow. J Fluid Mech 2016;793:840–76.
[62] Zheng W, Yang Y, Chen S. Evolutionary geometry of Lagrangian structures in a transitional boundary layer. Phys Fluids 2016;28:035110.
[63] Zheng W, Ruan S, Yang Y, He L, Chen S. Image-based modelling of the skin-friction coefficient in compressible boundary-layer transition. J Fluid Mech 2019;875:1175–203.
[64] Lundgren T. Strained spiral vortex model for turbulent fine structure. Phys Fluids 1982;25:2193–203.
[65] Lundgren T. A small-scale turbulence model. Phys Fluids A 1993;5:1472.