Journal of System Simulation

Volume 35 | Issue 3 Article 1

3-22-2023

Learning-Based High-Performance Algorithm for Long-Term Motion Prediction of Fluid Flows

Jingyuan Zhu
1.Department of Electronic Engineering, Tsinghua University, Beijing 100084, China;, jy-zhu20@mails.tsinghua.edu.cn

Huimin Ma
2.School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China;, mhmpub@ustb.edu.cn

Jian Yuan
1.Department of Electronic Engineering, Tsinghua University, Beijing 100084, China;

Learning-Based High-Performance Algorithm for Long-Term Motion Prediction of Fluid Flows

Abstract

Abstract: Simulating the dynamics of fluid flows accurately and efficiently remains a challenging task nowadays, and traditional fluid simulation methods consume large computational resources to obtain accurate results. Deep learning methods have developed rapidly, which makes data-based fluid simulation and generation possible. In this paper, *a motion prediction algorithm for long-term fluid simulation is proposed*, *which is based on a density field with a single frame and a previous velocity field of a sequence*. The model focuses on matching the velocity and density fields predicted by the neural network with the simulated data based on the Navier-Stokes equation at a macroscopic level. With the help of fully convolutional U-Net-based autoencoders and LSTM-based time series prediction subnetworks, the model better maintains the visual macroscopic similarity during temporal evolutions and significantly improves computation speed. As a result, *the proposed method achieves accurate and rapid long-term motion prediction for the macroscopic distributions of flow field evolution*. In addition, the paper demonstrates the effectiveness and efficiency of the proposed algorithm on a series of benchmark tests based on two-dimensional (2D) and three-dimension (3D) simulation data.

Keywords

motion prediction, fluid flow, long-term, high-performance, learning algorithm

Recommended Citation

Jingyuan Zhu, Huimin Ma, Jian Yuan. Learning-Based High-Performance Algorithm for Long-Term Motion Prediction of Fluid Flows[J]. Journal of System Simulation, 2023, 35(3): 435-453.

第 35 卷第 3 期
2023 年 3 月

系统仿真学报©
Journal of System Simulation

Vol. 35 No. 3
Mar. 2023

Learning-Based High-Performance Algorithm for Long-Term Motion Prediction of Fluid Flows

Zhu Jingyuan[1], *Ma Huimin*[2], *Yuan Jian*[1]

(1. Department of Electronic Engineering, Tsinghua University, Beijing 100084, China;
2. School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China)

Abstract: Simulating the dynamics of fluid flows accurately and efficiently remains a challenging task nowadays, and traditional fluid simulation methods consume large computational resources to obtain accurate results. Deep learning methods have developed rapidly, which makes data-based fluid simulation and generation possible. In this paper, *a motion prediction algorithm for long-term fluid simulation is proposed*, *which is based on a density field with a single frame and a previous velocity field of a sequence*. The model focuses on matching the velocity and density fields predicted by the neural network with the simulated data based on the Navier-Stokes equation at a macroscopic level. With the help of fully convolutional U-Net-based autoencoders and LSTM-based time series prediction subnetworks, the model better maintains the visual macroscopic similarity during temporal evolutions and significantly improves computation speed. As a result, *the proposed method achieves accurate and rapid long-term motion prediction for the macroscopic distributions of flow field evolution*. In addition, the paper demonstrates the effectiveness and efficiency of the proposed algorithm on a series of benchmark tests based on two-dimensional (2D) and three-dimension (3D) simulation data.

Keywords: motion prediction; fluid flow; long-term; high-performance; learning algorithm

学习流体仿真中的高效长时序运动预测

朱静远[1]，马惠敏[2]，袁坚[1]

(1. 清华大学 电子工程系，北京 100084；2. 北京科技大学 计算机与通信工程学院，北京 100083)

摘要：快速准确的进行流体仿真是一个具有挑战性的任务，基于传统方法的流体仿真需要消耗大量的计算资源以获得准确的结果。深度学习方法快速发展，为基于数据的流体仿真和生成提供了可能。*提出一种基于单帧的浓度场和一个序列的先验速度场的长时序流体仿真运动预测算法*。这一模型专注于将通过神经网络预测的速度和密度场基于纳维–斯托克斯方程获得的仿真数据在宏观尺度上进行匹配。通过使用基于全卷积U型网络的自动编码器和基于LSTM网络的时序预测子网络，本文模型在时间演化过程中更好地保持了视觉宏观相似性，并实现了显著的计算速度提升。*本文方法实现了对流体场演化的宏观分布的准确和快速的长时序运动预测*。在一系列二维和三维的仿真数据的基准测试上证明了本文算法的有效性和高效性。

关键词：运动预测；流体；长时序；高效；学习算法

中图分类号：TP391.9　　文献标志码：A　　文章编号：1004-731X(2023)03-0435-19

Received date：2022-12-05　　Revised date：2022-12-26

Foundation：National Natural Science Foundation of China(U20B2062)

First author：Zhu Jingyuan(1999-), male, Manchu, doctoral student, research area：computer graphics and computer vision.
E-mail：jy-zhu20@mails.tsinghua.edu.cn
Corresponding author：Ma Huimin(1972-), female, doctor, professor, research area：computer vision. E-mail：mhmpub@ustb.edu.cn

DOI: 10.16182/j.issn1004731x.joss.22-1507

Reference format: Zhu Jingyuan, Ma Huimin, Yuan Jian. Learning-Based High-Performance Algorithm for Long-Term Motion Prediction of Fluid Flows[J]. Journal of System Simulation, 2023, 35(3): 435-453.

0 Introduction

Fluid flow is one of the most common natural phenomena in our daily life. During the past decades, fluid simulation methods based on the Navier-Stokes equation have achieved great success. Related research plays significant roles in fields including aerospace engineering, physical animation, and game production. However, simulating the dynamics of fluid flows accurately and efficiently remains a challenging issue. The computational cost of traditional fluid simulation algorithms based on the Naiver-Stokes equation is extremely expensive to obtain stable and reliable results with a constrained time step size and grid resolution.

Nowadays, deep learning methods have been widely applied to solve physically-based tasks. As a replacement for traditional analytical methods, deep learning methods are capable of extracting information and regular patterns from large-scale datasets. Therefore, it is reasonable for learning-based models to realize fluid simulation with high precision and computing efficiency. Deep learning methods have been proven to be competitive alternatives against traditional methods in tasks including fluid generation [1-2] and super-resolution flows [3-4].

Predicting the temporal evolution of physical evolutions using deep learning methods has raised much interest in recent years. Neural networks (NNs) are employed for learning temporally coherent features in point clouds [5]. Latent space physics (LSP) [6] proposes models containing convolutional neural networks (CNNs) and long short-term memory (LSTM) prediction network for learning the temporal evolution of fluid flows in the compressed latent space to reduce computational cost. Latent space subdivision (LSS) [7] further proposes an end-to-end NN architecture, so as to predict the dynamics of fluid flows robustly with high temporal stability by subdividing the latent space according to physical meanings including density, velocity, and inflow.

Although these works can produce realistic fluid motions with impressive visual effects, it remains a challenging problem to ensure the similarity between the predicted flow shape and the simulated ones by using physically-based simulators. On the other hand, long-term matching of the flow shape will benefit the prediction of interactive behaviors between fluids and surrounding objects. In this paper, we provide a learning-based high-performance algorithm for long-term motion prediction of physical flow field evolution, including density and velocity fields. We focus on maintaining the long-term similarity of the macroscopic distributions in fluid flows since they dominate the interactive behaviors between fluids and environment objects.

Our model mainly consists of two functional sub-networks to realize long-term prediction of velocity and density fields. The long-term velocity prediction sub-network first employs fully convolutional U-Net-based autoencoders to compress input velocity field sequences into relatively smaller latent codes and then carries out the temporal evolution of velocity fields with LSTM-based networks in latent space. In addition, we make use of loss functions containing terms that have physical meanings [8-9] to improve the temporal evolution reality of velocity fields. With predicted velocity fields, another long-term density prediction sub-network is proposed to predict long-term density field evolution with a single frame of density field and a sequence of previous velocity fields as inputs. Input density and velocity fields are encoded to latent codes with fully convolutional autoencoders. In comparison with density advection in traditional fluid simulation, our density prediction sub-network implements long-term density advection through a single step of calculation in latent space, which ensures the reality of macroscopic distributions in density fields with significant speed-ups. The encoding-decoding network architecture also helps the long-term motion prediction model compress high-dimensional information in physical fields and significantly improves computing efficiency. Finally, we obtain a high-performance long-term motion prediction model concentrating on macroscopic distributions for physical fields of fluid flows. We will demonstrate our model's accuracy and efficiency with several 2D and 3D gas simulation datasets. The major contributions of our work could be concluded as follows:

(1) A learning-based algorithm taking advantage of prior physical information to realize long-term prediction of the macroscopic distributions of velocity fields;

(2) A long-term density prediction approach to predict the macroscopic distributions of density fields based on predicted macroscopic distributions of prior velocity fields;

(3) End-to-end high-performance network architectures to achieve significant speed-ups for the motion prediction of fluid flows.

1 Related Work

In this paper, our models are trained with incompressible fluid flow datasets simulated with the Navier-Stokes equation:

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} = \boldsymbol{f} - \frac{1}{\rho}\nabla p + \frac{\mu}{\rho}\Delta\boldsymbol{u}$$

$$\nabla\cdot\boldsymbol{u} = 0$$

where $\boldsymbol{u}$, ρ, μ and $\boldsymbol{f}$ denote flow velocity, pressure, flow density, kinematic viscosity, and external forces, respectively.

Traditional fluid simulation methods are mainly based on the Navier-Stoke equation. Related research mainly focuses on two aspects: improvement of visual effects and computing efficiency. All kinds of numerical methods like BFECC [10-11] are proposed to improve the visual effects of stable fluid simulation through large amounts of iterations. Fedkiw et al. [12] added extra parameters to realize vorticity confinement of the same scale in the whole fluid simulation area. Other vorticity confinement methods were also proposed to further improve detail reconstruction by considering concrete distributions of simulation areas [13-14]. To deal with the huge computational cost and difficult parallelization caused by traditional stable fluid simulation, the lattice Boltzmann method (LBM) was proposed to improve simulation efficiency with parallel computing by using discretized distribution function [15-16]. Wen et al. [17] proposed vorticity confinement methods for LBM algorithms. They achieved real-time simulation with high grid resolution by using the LBM model's parallelism and powerful computing capability provided by GPU. In addition to LBM methods, high-performance fluid solvers based on particle level set [18]and the fluid implicit particle (FLIP) method [19] have become mainstream tools for detailed liquid simulations. Except for those grid-based Eulerian methods, Lagrangian methods represented by smoothed particle hydrodynamics (SPH)[20-21] are other competitive alternatives for fluid simulation. Other methods focus on topics like the large-scale simulation of the complex dynamics of ferrofluids[22] and interesting behaviors of various mixtures[23].

Besides, data-driven machine learning methods[24], especially those related to deep learning methods, are widely applied in many fields. As for computer graphics, physics-based deep learning methods combine physical modeling and deep learning techniques and produce impressive results in many related topics. Ladicky et al [25]. took physics-based fluid simulation as a regression problem and achieved real-time calculation of systems with up to 2 million particles. Thuerey et al. [26] investigated the accuracy of deep learning models for Reynolds-Averaged Navier-Stokes solutions and obtained a mean relative pressure and a velocity error of less than 3% across a range of previously unseen airfoil shapes, which showed the potential capability of deep learning methods in solving physical systems. Related research has proven that deep learning methods can be competitive alternatives compared with traditional methods in solving physical problems. Other data-driven models perform well in tasks such as generating desired implicit surfaces [2] and droplet formation [27]. Hennigh et al. [28] compressed both computation time and memory usage of lattice Boltzmann flow simulation with the proposed Lat-Net based on deep neural networks. CNNs are applied for extracting features and generating descriptors from fluid data to track deformable fluid regions[29]. Another data-driven approach that leverages the approximation power of deep learning with the precision of standard solvers is proposed to obtain fast and highly realistic simulations based on incompressible Euler equations [8]. Deep residual recurrent neural networks are employed to learn the dynamical systems of subsurface multi-phase flows [30]. Graph networks also perform well in physics-based learning tasks representing the state of particles as nodes in graphs [2, 31]. Kim et al. demonstrated that complex parameterizations of fluid flow could be handled in reduced spaces to significantly improve simulation speed [1]. Apart from simulation in the form of Eulerian representation, deep learning techniques could be used to learn stable and temporally coherent feature spaces from data in the form of Lagrangian representation [32]. For example, temporally coherent features in point clouds could be learned by neural networks [5]. Moreover, deep learning methods can extract motion information from rendered image sequences [33]. Further applications in PBDL involve large-scale problems like predictions of global weather conditions with data-driven methods [34].

When it comes to the temporal evolution of fluid flow discussed in this paper, Wiewel et al. [6] demonstrated for the first time that the space and time function of 3D could be predicted within reduced latent spaces through neural networks. Then another end-to-end trained neural network architecture was proposed to predict the complex dynamics of fluid flow systems robustly with high temporal stability [7]. When their temporal evolution models perform fare well with pressure fields, the prediction of velocity fields significantly differs from ground truth data. Eivazi et al.[35] applied the autoencoder-LSTM method to predict fluid flow evolution in unsteady fluid systems. In the past several years, generative adversarial networks (GAN) have become highly successful in image generation tasks [36-37]. GAN models are proven to be effective in enhancing the details of fluid systems. Xie et al. proposed a temporally coherent generative model, namely tempoGAN, which could infer realistic high-resolution details to the super-

DOI: 10.16182/j.issn1004731x.joss.22-1507

resolution problems for fluid flows [3]. TecoGAN proposed by Chu et al. provided a generative adversarial model for video super-resolution tasks [38]. Temporally coherent solutions could be obtained without sacrificing spatial details through the TecoGAN model. The multi-pass GAN model deals with 3D generative problems by decomposing functions on the Cartesian field into multiple smaller sub-problems to learn more efficiently [4]. In addition, subsequent works further realize GAN-based coherent super-resolution generation for fluid flows [39].

Although these works successfully produced realistic fluid motions, few of them aimed to ensure the similarity between the predicted flow shape and the simulated ones using physically-based simulators. The model proposed in this paper concentrates on matching the network-predicted velocity and density fields with the simulated data in macroscopic distributions accurately in long-time ranges by taking advantage of physical laws. Besides, most previous physics-based learning approaches [6, 40] aim to accelerate physical simulation by replacing several steps in traditional simulation methods. Our proposed model only makes use of velocity and density fields and successfully predicts their macroscopic distributions with a speed significantly improved. In the following sections, we will demonstrate more details and performance evaluations of our models.

2 Method

Our model aims to realize high-performance long-term motion prediction for physical fields including the velocity field v and the density field x of fluid flows. Compared with previous works, our long-term motion prediction model focuses on matching the predicted physical fields with the simulated data, especially in macroscopic distributions over long periods. The structure diagram of the prediction model for the long-term motion of fluid flows is shown in Fig. 1. One sub-network for velocity field prediction is shown in the top half, and the other sub-network for density field prediction is shown in the bottom half.

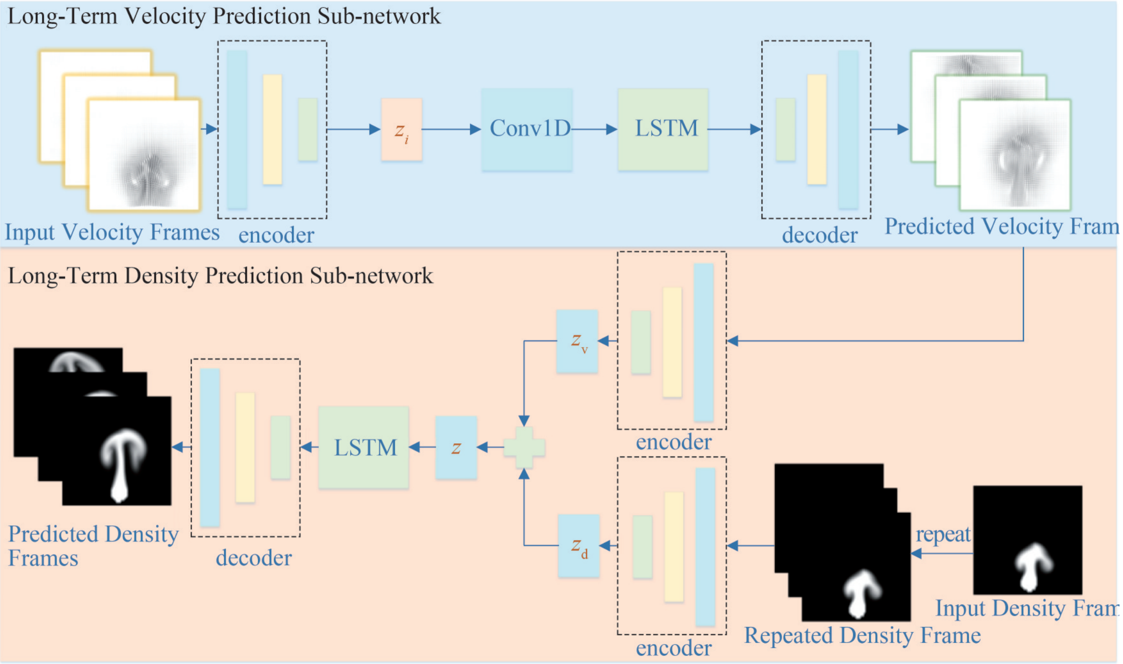


Fig. 1 Pipeline of the proposed model for long-term motion prediction of fluid flows

Our model consists of two functional sub-networks, so as to realize long-term velocity and density field prediction, respectively. It is worth noting that the long-term density prediction model requires prior velocity frames as inputs for density field prediction. Therefore, long-term density field prediction should be based on velocity field prediction in practical applications. According to these two functional sub-networks, our model is capable of predicting the long-term temporal evolution of physical fields for fluid flows. Given f as the representation of our motion prediction model, the whole prediction process can be demonstrated by the following equation which assumes that $\boldsymbol{v}_{t-n+1}$, $\boldsymbol{v}_{t-n+2}$,$\cdots$, $\boldsymbol{v}_t$ and $\boldsymbol{x}_t$ are known:

$$(\boldsymbol{v}_{t+1}, \cdots, \boldsymbol{v}_{t+o}, \boldsymbol{x}_{t+1}, \cdots, \boldsymbol{x}_{t+o+1})=f(\boldsymbol{v}_{t-n+1}, \cdots, \boldsymbol{v}_t, \boldsymbol{x}_t),$$

where n denotes the input length, and o denotes the predicted length of velocity fields.

Since our long-term motion prediction model employs fully convolutional autoencoders to compress physical fields with high complexity to relatively smaller latent codes, some details in physical fields will be ignored to improve prediction efficiency. In order to prevent the long-term density field prediction model from being affected by details lost in velocity fields, the two sub-networks are trained independently and combined for physical field prediction. Details of the sub-networks for velocity and density field prediction will be described in Sections 2.1 and 2.2, respectively. In addition, the combination of sub-networks for physical fields' motion prediction will be illustrated in Section 2.3.

2.1 Long-Term Velocity Field Prediction

In this section, we propose a sub-network for long-term velocity field prediction with a sequence of previous velocity fields as inputs. By taking f_{lv} as the representation of the long-term velocity field prediction function, the sub-network can be expressed as

$$(\boldsymbol{v}_{t+1}, \boldsymbol{v}_{t+2}, \cdots, \boldsymbol{v}_{t+o})=f_{\text{lv}}(\boldsymbol{v}_{t-n+1}, \boldsymbol{v}_{t-n+2}, \cdots, \boldsymbol{v}_t).$$

The network architecture of the long-term velocity prediction sub-network is shown in the top half of Fig. 1. The U-Net-based encoder-decoder network first compresses the velocity field to a latent code. Then LSTM-based prediction network performs temporal evolution in the latent space. Our model differs from the prediction networks for predicting pressure field in Latent Space Physics [6] in the following three aspects:

(1) 3D fully convolutional U-Net-based networks are used to encode and decode velocity fields temporally and spatially;

(2) 1D convolutional layer is employed for transforming input length to output length instead of repeating latent codes the same times as the predicted length;

(3) temporal evolution in latent spaces is realized with a simplified single-layer LSTM network.

The U-Net-based autoencoders composed of 3D fully convolutional networks can simultaneously encode and decode the temporal and spatial information of velocity fields. Details of the encoding-decoding network architecture can be found in Table 1 where autoencoders composed of five encode layers are shown. Specifically, $f_{\text{e}i}$ and $f_{\text{d}i}$ denote layers in the encoder and decoder stack, $\boldsymbol{r}$ denotes the resolution of data, and d denotes the dimensionality of velocity fields. t_{i} and t_{o} denote the input and output length. The 1D convolutional network can retain more information for long-term temporal evolution than the repeated method used in Latent Space Physics [6]. Simplified LSTM-based prediction networks composed of a single layer of LSTM and following batch normalization layers reduce the complexity of our prediction model significantly. These modifications improve our model's computing efficiency and make it more appropriate for long-term velocity field prediction.

Table 1 Parameters of U-Net-based autoencoder layers

Layer	Kernel	Stride	Activation	Output	Feature
f_{e1}	4	2	Linear	$\boldsymbol{r}/2$	$16t_i$
f_{e2}	2	2	LeakyReLU	$\boldsymbol{r}/4$	$32t_i$
f_{e3}	2	2	LeakyReLU	$\boldsymbol{r}/8$	$64t_i$
f_{e4}	2	2	LeakyReLU	$\boldsymbol{r}/16$	$128t_i$
f_{e5}	2	2	LeakyReLU	$\boldsymbol{r}/32$	$256t_i$
f_{d1}	2	2	LeakyReLU	$\boldsymbol{r}/16$	$128t_o$
f_{d2}	2	2	LeakyReLU	$\boldsymbol{r}/8$	$64t_o$
f_{d3}	2	2	LeakyReLU	$\boldsymbol{r}/4$	$32t_o$
f_{d4}	2	2	LeakyReLU	$\boldsymbol{r}/2$	$16t_o$
f_{d5}	4	2	LeakyReLU	$\boldsymbol{r}$	dt_o

To make the velocity fields predicted by our long-term velocity prediction sub-network consistent with physical laws, we employ a law-based loss function containing four parts as its minimization target. Apart from the pixel-wise L1 norm loss, the weighted loss of velocity fields guides our model to center on the macroscopic distributions of velocity fields. In addition, the other two terms are added to guide the sub-network to learn boundary conditions and fluid incompressibility. The law-based loss function is the weighted sum of these four parts:

$$L_{\text{vel}}=\sum_{f=t+1}^{f=t+o}\alpha L_{\text{norm}}(\tilde{\boldsymbol{v}}_f,\ \boldsymbol{v}_f)+\beta L_{\text{weight}}(\tilde{\boldsymbol{v}}_f,\ \boldsymbol{v}_f)+ \\ \gamma L_{\text{div}}(\tilde{\boldsymbol{v}}_f)+(1-\alpha-\beta-\gamma)L_{\text{bound}}(\tilde{\boldsymbol{v}}_f) \qquad (1)$$

where α, β, and γ are trade-off parameters set empirically. Here

$$L_{\text{norm}}(\tilde{\boldsymbol{v}}_f,\ \boldsymbol{v}_f)=|\tilde{\boldsymbol{v}}_f-\boldsymbol{v}_f|_1;$$

$$L_{\text{weight}}(\tilde{\boldsymbol{v}}_f,\ \boldsymbol{v}_f)=|(\tilde{\boldsymbol{v}}_f-\boldsymbol{v}_f)\boldsymbol{v}_f|_1;$$

$$L_{\text{div}}(\tilde{\boldsymbol{v}}_f)=|\nabla\cdot\tilde{\boldsymbol{v}}_f|;$$

$$L_{\text{bound}}(\tilde{\boldsymbol{v}}_f)=|\tilde{\boldsymbol{v}}_{f\perp}|.$$

denote the L1 norm loss between the predicted velocity fields $\tilde{\boldsymbol{v}}$ and corresponding ground truth $\boldsymbol{v}$, the velocity-weighted loss weighted by the absolute value of velocity fields’ ground truth, the absolute value of the predicted velocity fields’ divergence (in Literature [8] and [9]), and the absolute value of velocity perpendicular to rigid boundaries, respectively.

2.2 Long-Term Density Field Prediction

In fluid flow simulation algorithms, density fields are normally updated with velocity fields according to the advection function $\boldsymbol{x}_{t+1}=\boldsymbol{x}_t-\boldsymbol{v}_t\mathrm{d}t$. Our long-term density prediction sub-network is proposed to realize density field prediction with a single frame of density field $\boldsymbol{x}_t$ and a sequence of predicted velocity fields $\boldsymbol{v}_t,\ \boldsymbol{v}_{t+1},\cdots,\boldsymbol{v}_{t+o}$. Given f_{ld} as the representation of the long-term density field prediction function, this sub-network can be expressed as

$$(\boldsymbol{x}_{t+1},\ \boldsymbol{x}_{t+2},\cdots,\ \boldsymbol{x}_{t+o+1})=f_{\text{ld}}(\boldsymbol{x}_t,\ \boldsymbol{v}_t,\ \boldsymbol{v}_{t+1},\cdots,\boldsymbol{v}_{t+o}).$$

As shown in the bottom half of Fig. 1, the sub-network maintains the U-Net-based encoding-decoding architecture to accelerate the temporal evolution of density fields. Different from traditional iteration methods for temporal evolution, the long-term density prediction sub-network first repeats the input density field as many times as the frame number of the input velocity fields. The input velocity fields and repeated density fields are encoded with fully convolutional networks. Then the velocity and repeated density latent code is added to density-velocity latent code for further temporal evolution with a simplified single-layer LSTM network similar to the network used in velocity field prediction. Different from the concatenation of density and velocity latent code used in LSS [7], the sum of the density and velocity latent code is applied to our model as a replacement. For one thing, it is reasonable to utilize the sum of density and velocity latent code since the density field is updated as the weighted sum of the original density and velocity field in the advection function, namely $\boldsymbol{x}_{t+1}= \boldsymbol{x}_t - \boldsymbol{v}_t\mathrm{d}t$. As a result, our model actually carries out the advection of density fields in the latent space. For another, the sum of latent code reduces the size of density-velocity latent code by half compared with the concatenation method, which reduces the complexity of our model in the sacrifice of some details in physical fields. Compared with the density advection method, which should be calculated sequentially, our model could obtain continuous multiple frames of density fields in one calculation, so as to effectively improve the computation speed. In conclusion, the long-term density prediction sub-network realizes the prediction of macroscopic distributions of density fields in long-time ranges with the help of prior velocity fields and improves computing efficiency with advection in latent spaces.

The minimization problem solved by the long-term density prediction sub-network contains two parts. The first part focuses on L1 norm loss between the predicted density fields$\tilde{\boldsymbol{x}}$and corresponding ground truth $\boldsymbol{x}$, while the second part focuses on density-weighted loss. The whole loss function is the weighted sum of those two parts

$$L_{\mathrm{den}}=\sum_{f=t+1}^{f=t+o+1}\lambda|\tilde{\boldsymbol{x}}_f-\boldsymbol{x}_f|_1+(1-\lambda)|(\tilde{\boldsymbol{x}}_f-\boldsymbol{x}_f)\boldsymbol{x}_f|_1=$$

$$\sum_{f=t+1}^{f=t+o+1}\lambda L_{\mathrm{norm}}(\tilde{\boldsymbol{x}}_f,\ \boldsymbol{x}_f)+(1-\lambda)L_{\mathrm{weight}}(\tilde{\boldsymbol{x}}_f,\ \boldsymbol{x}_f)\ (2)$$

where λ is a trade-off parameter set empirically.

2.3 Long-Term Motion Prediction

With the two long-term prediction sub-networks proposed in previous sections, we can predict both velocity fields and density fields. With a sequence of velocity fields and density fields known, we can first predict velocity fields by the long-term velocity prediction sub-network. Then the density field can be predicted through the long-term density prediction sub-network with a single frame of density field and the predicted velocity fields as inputs.

As shown in Fig. 1, the whole prediction process can be demonstrated by the following equations, which assume that $\boldsymbol{v}_{t-n+1}$, $\boldsymbol{v}_{t-n+2}$, $\cdots$, $\boldsymbol{v}_t$, and $\boldsymbol{x}_t$ are known

$$(\boldsymbol{v}_{t+1},\ \boldsymbol{v}_{t+2},\cdots,\boldsymbol{v}_{t+o})=f_{\mathrm{lv}}(\boldsymbol{v}_{t-n+1},\ \boldsymbol{v}_{t-n+2},\cdots,\boldsymbol{v}_t),$$

$$(\boldsymbol{x}_{t+1},\ \boldsymbol{x}_{t+2},\cdots,\boldsymbol{x}_{t+o+1})=f_{\mathrm{ld}}(\boldsymbol{x}_t,\boldsymbol{v}_t,\boldsymbol{v}_{t+1},\cdots,\boldsymbol{v}_{t+o}).$$

Specifically, f_{lv} and f_{ld} denote the long-term velocity and density prediction sub-network respectively.

3 Training and Evaluation

In this section, the temporal evolution models proposed above will be evaluated by simulation datasets. Peak signal-to-noise ratio (PSNR) is used as the basic metric for similarity evaluation between the prediction and ground truth. Different from the PSNR calculated for RGB images, PSNR used in our experiments is calculated according to the following equation for physical fields:

$$PSNR=10\lg(\frac{\boldsymbol{x}_{\max}^2}{MSE}),$$

where $\boldsymbol{x}_{\max}$represents the maximum absolute value in the ground truth physical fields, and mean squared error (MSE) represents the mean squared error between the prediction and ground truth. In addition, cosine similarity is added to evaluate the similarity of the macroscopic distributions in physical fields between our temporal evolution results and the corresponding ground truth. In addition, MSE is appended for analysis. All of our datasets are divided

DOI: 10.16182/j.issn1004731x.joss.22-1507

into three parts for training, validation, and testing. In model training, 80% of our datasets are used for model training, 10% for validation, and 10% for testing. Our models are trained for 40 epochs with an Adam optimizer of 0.001 learning rate.

3.1 Training Datasets

The fluid flow datasets used for our model training are simulated by an open-source framework, namely Mantaflow. Mantaflow is a Navier-Stokes equation solver, which simulates fluid flows with discrete points in space and time. Since our model pays attention to the temporal evolution of density and velocity fields, the datasets mainly contain randomized smoke flows in different scenes. Furthermore, we employ several typical scenes for dataset generation including single source smoke, single source smoke interacting with obstacles, and smoke in rotating and moving cups. Example simulation sequences of our datasets are displayed in Fig. 2, which shows smoke in rotating and moving cups (top row), single source smoke (center row), and single source smoke with obstacles (bottom row). The smoke density is rendered as white with a black background, and the obstacle is rendered as blue.

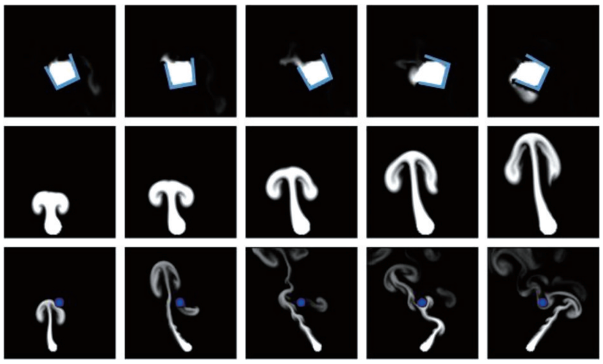


Fig. 2 Example simulation sequences visualization of our 2D datasets

As for the single source smoke and single source smoke interacting with obstacles scenes, the smoke source is randomly distributed at the bottom of the simulation area, which helps generate various data. In the rising smoke scene, the initial distribution is randomly generated for data diversity. Single source smoke scenes in our datasets are simulated with open boundaries, while the rising smoke scenes are simulated within the space surrounded by rigid boundaries. Datasets of different resolutions are generated for training and evaluating our model. For comparison, we also add rotating cup (2D) and rotating and moving cup (2D) datasets used in LSS[7]. The rotating cup scene simulates the motion of cold smoke in a rotating cup-shaped obstacle, while the rotating and moving cup scene additionally applies a translation. In addition, two simple 3D datasets are employed for training our model. More detailed information of our datasets is listed in Table 2.

Table 2 Statistics of simulation datasets used in this paper

Scene Type	Resolution	Scene	Frames
Single source smoke (3D)	64^3	600	100
Single source smoke with obstacles (3D)	64^3	600	200
Single source smoke (2D)	64^2	600	200
Single source smoke (2D)	128^2	600	200
Single source smoke with obstacles (2D)	64^2	600	200
Single source smoke with obstacles (2D)	128^2	600	200
Rising smoke (2D)	64^2	1 000	200
Rising smoke(2D)	128^2	1 000	200
Rotating cup (2D)	64^2	600	200
Rotating and moving cup (2D)	64^2	300	300

3.2 Long-Term Prediction Sub-networks

Our long-term prediction sub-networks including the density and velocity prediction sub-networks are evaluated by rotating cup datasets (2D) and single source smoke (2D & 3D) datasets listed in Table 2. The long-term density prediction sub-network takes a single frame of density field and a sequence of prior velocity fields as inputs and outputs density fields sequence over long periods. Our model is trained to realize 2D density field prediction in 150 time steps. The long-term velocity prediction sub-network realizes velocity field prediction in a long-time range with a sequence of previous velocity fields as inputs. In our experiments, the sub-network predicts the following 50 frames of 2D velocity fields with the previous 30 frames as inputs. The latent space size of the long-term motion prediction model is set to be 1 024 and 2 048 for the data resolution of 64×64 and 128×128, respectively. As for 3D datasets, our sub-networks are trained to predict the previous 30 frames of density fields and velocity fields with the latent space size of 1 024.

(1) Long-term velocity prediction sub-network

The trade-off hyper-parameters in the long-term velocity prediction sub-network (Eq. (1)) are set as $\alpha=0.5$, $\beta=0.2$, and $\gamma=0.2$. The performance of the long-term velocity prediction sub-network on the single source smoke datasets is given in Table 3. As for the typical simulation scene of single source smoke (2D & 3D), our model can reach a PSNR higher than 23 and a cosine similarity value higher than 0.96. It can be seen that our model can predict the macroscopic distributions of velocity field evolution over long periods in the sacrifice of some detailed distributions. Since randomly moving obstacles are applied in the rotating cup datasets (2D), it is reasonable for our model to show relatively poor performance on velocity prediction.

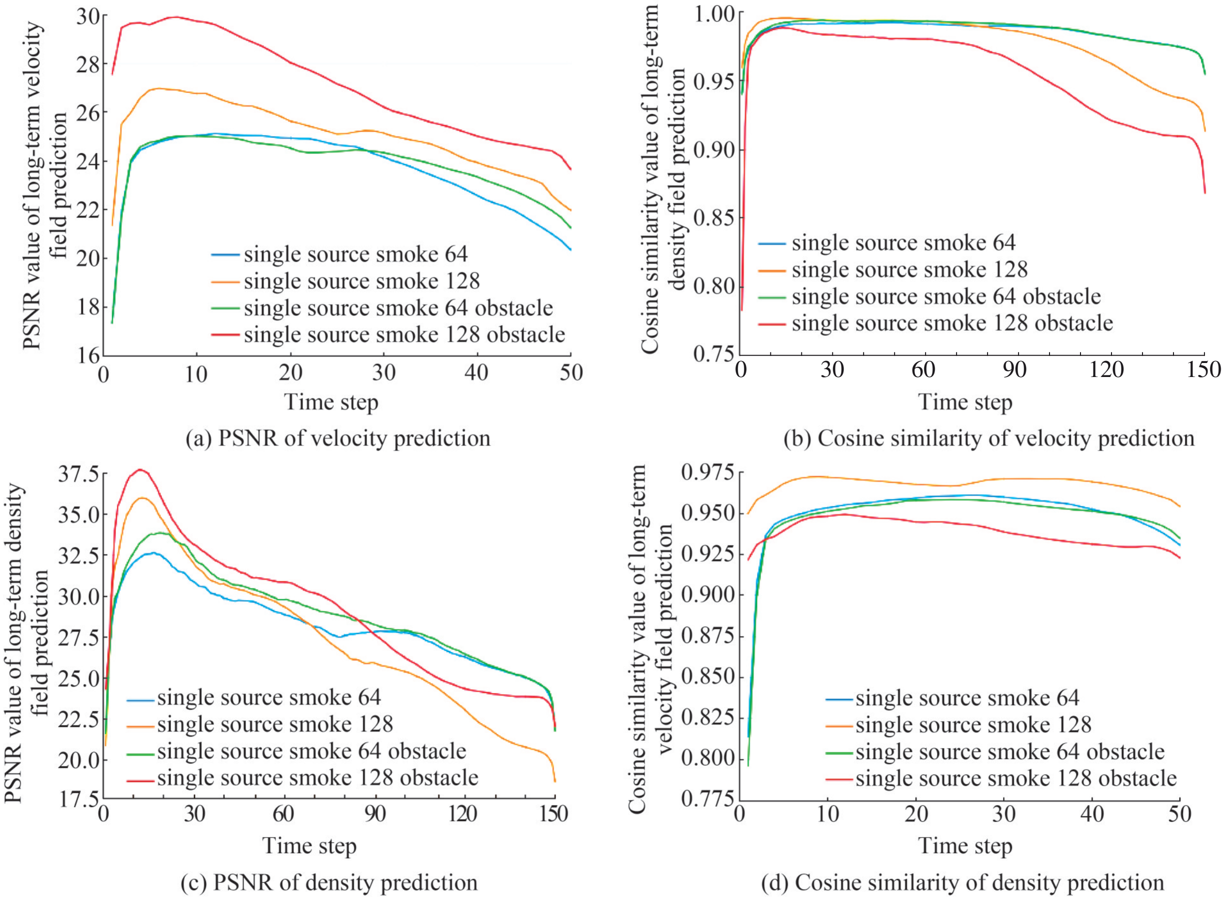
(2) Long-term density prediction sub-network

The trade-off hyper-parameter in the long-term density prediction sub-network (Eq. (2)) is set to $\alpha=$ 0.5. In Table 3, evaluation metrics of the long-term density prediction sub-network are listed. In addition, higher PSNR and cosine similarity are better. Our model accurately predicts velocity and density fields with high cosine similarity on a series of 2D and 3D benchmarks. Our model shows outstanding performance with PSNR higher than 23 and cosine similarity value higher than 0.88 on the simulation datasets (2D & 3D). Furthermore, our model can accurately predict density fields over long periods with the information of velocity fields.

Table 3 Statistics of long-term prediction sub-networks

Datasets	Resolution	Long-term velocity			Long-term density		
		PSNR	Cos similarity	*MSE*	*PSNR*	Cos similarity	*MSE*
Single source smoke (3D)	64^3	41.123	0.997	0.002 2	26.279	0.889	0.002 6
Single source smoke with obstacles (3D)	64^3	32.289	0.993	0.006 6	35.717	0.993	0.000 2
Rotating cup (2D)	64^2	20.430	0.835	0.008 0	23.244	0.970	0.002 9
Rotating and moving cup (2D)	64^2	20.279	0.863	0.030 8	25.579	0.972	0.003 8
Single source smoke (2D)	64^2	23.627	0.969	0.084 0	28.212	0.985	0.003 8
Single source smoke (2D)	128^2	25.024	0.976	0.264 8	27.377	0.978	0.006 4
Single source smoke with obstacles (2D)	64^2	23.810	0.966	0.070 8	28.866	0.986	0.003 4
Single source smoke with obstacles (2D)	128^2	27.098	0.960	0.095 4	29.092	0.954	0.003 8

The statistics of time-varying PSNR and cosine similarity value are shown in Fig. 3. As shown in Fig. 3(a) and 3(b), although PSNR gradually decreases with time steps, the long-term velocity prediction sub-network can maintain high cosine similarity in a longer time range. A similar time-varying phenomenon can also be found in Fig. 3(c) and 3(d) for long-term density field prediction. Therefore, our long-term prediction sub-networks can reconstruct macroscopic distributions of physical fields with a high degree of resemblance over long periods, and our model can achieve high accuracy for long-term density field prediction.



(a) PSNR of velocity prediction

(b) Cosine similarity of velocity prediction

(c) PSNR of density prediction

(d) Cosine similarity of density prediction

Fig. 3 Time-varying PSNR and cosine similarity of long-term prediction sub-networks

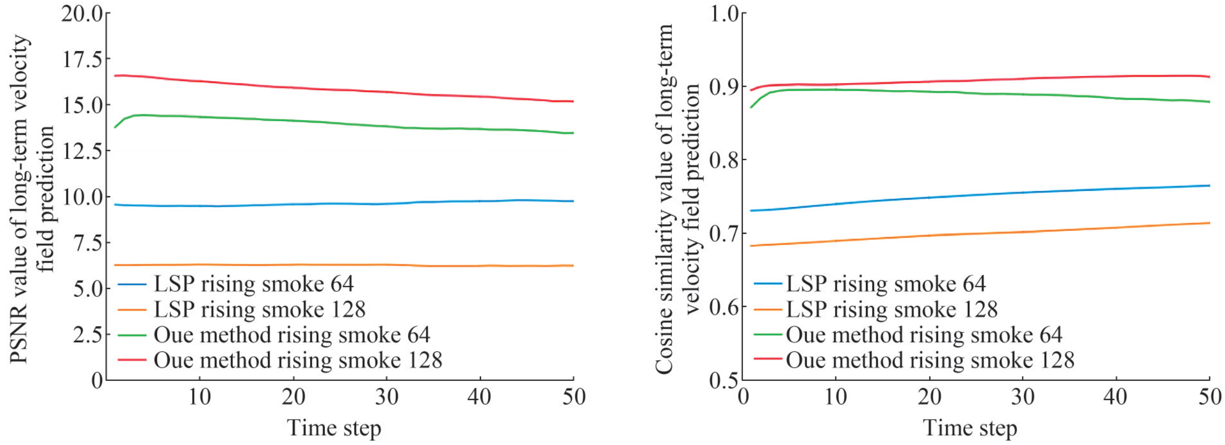
(3) Long-term velocity prediction sub-network v.s. LSP model

LSP [6] proposed a learning-based method for the temporal evolution of fluid flows, which shared a similar target of predicting physical fields with our long-term velocity prediction sub-network. To evaluate the improvement of our model for long-term velocity field prediction, we compare the model with the LSP model realizing velocity field prediction of 50 time steps with the previous 30 time steps as inputs on rising smoke datasets. The size of latent space is set to be 1 024 for both models. Our method is compared with the LSP method quantitatively in Table 4. In addition, higher PSNR and cosine similarity are better. However, lower MSE is welcomed. As can be seen from Table 4, the proposed long-term velocity prediction sub-network outperforms the LSP model significantly in long-term velocity field prediction.

Our method outperforms the LSP method with higher PSNR and cosine similarity on rising smoke datasets through network architecture modification and guidance of loss functions that have physical meanings. In addition, our model shows more remarkable accuracy improvements on high-resolution datasets. Time-varying PSNR and cosine similarity of the LSP model and our long-term velocity prediction sub-network are given in Fig. 4. It can be seen that our model outperforms the LSP model with higher PSNR and cosine similarity on long-term velocity field prediction.

Table 4 Comparison between our long-term velocity prediction sub-network and LSP model on rising smoke datasets

Datasets	Resolution	LSP model			Our model		
		PSNR	Cos similarity	*MSE*	*PSNR*	Cos similarity	*MSE*
Rising smoke (2D)	64^2	9.588	0.750	0.816	**13.929**	**0.888**	**0.264**
Rising smoke (2D)	128^2	6.138	0.697	2.523	**15.802**	**0.907**	**0.279**



(a) PSNR of velocity prediction (b) Cosine similarity of velocity prediction

Fig. 4 Time-varying PSNR and cosine similarity of long-term velocity prediction sub-network and LSP model

In Fig. 5, we visualize the prediction results of our long-term velocity prediction sub-network, LSP model, and corresponding ground truth on single source smoke datasets (2D). The prediction results of our model are shown in the top row, those of the LSP model are shown in the middle row, and the corresponding ground truth is shown in the bottom row. It can be seen that our sub-network achieves significant improvements in the capability of predicting the trend of velocity field evolution over long-time ranges. The LSP model obtains a mean PSNR of 12.938 and a cosine similarity of 0.627. In contrast, our model can reach a mean PSNR of 23.627 and cosine similarity of 0.969 averaged over test datasets. In addition, our model can predict the macroscopic distributions of velocity field evolution, while the LSP model seems to predict the average distributions of prior velocity fields.

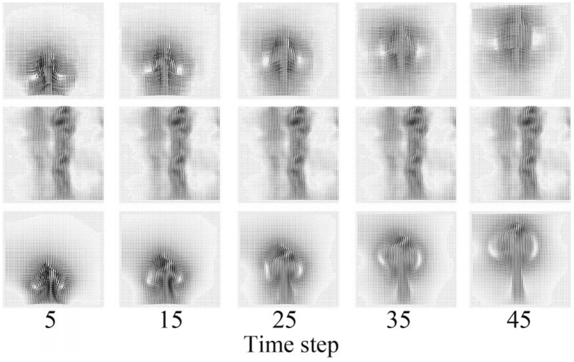


Fig. 5 Visualization Result comparison between our long-term velocity prediction sub-network and LSP model

DOI: 10.16182/j.issn1004731x.joss.22-1507

3.3 Long-Term Motion Prediction

As illustrated in Section 2.3, our long-term prediction sub-networks can realize long-term predictions for both velocity and density fields. Single source smoke (2D & 3D) and rising smoke (2D) datasets are employed for evaluating the long-term motion prediction model. We first train the long-term density and velocity prediction sub-networks separately. Then the predicted velocity field output from the long-term velocity prediction sub-network is used as input for the long-term density prediction sub-network to obtain prediction results. Finally, our long-term motion prediction model is trained to predict the previous 50 frames for 2D datasets and 30 frames for 3D datasets.

According to the previous analysis, our long-term velocity prediction sub-network mainly focuses on macroscopic distributions, but it neglects parts of detailed distributions. In Table 5, we provide the statistics of the long-term density prediction of original and predicted velocity fields, with original and predicted velocity fields from the long-term velocity prediction sub-network as inputs. Moreover, higher PSNR and cosine similarity are better, but lower MSE is welcomed. The long-term density prediction sub-network will suffer from the lack of detailed distributions in predicted velocity fields with lower PSNR and cosine similarity and higher MSE compared with density prediction based on original velocity fields. However, the PSNR higher than 17 and the cosine similarity higher than 0.8 show that our long-term motion prediction model can still realize macroscopic distributions of density field prediction with considerable accuracy. More visualization results in Section 4 will further prove our conclusion.

Table 5 Statistics of long-term density prediction sub-network

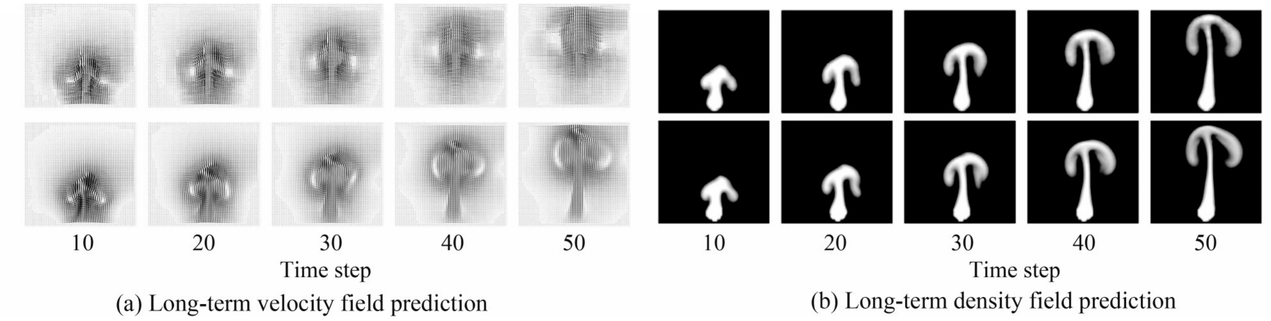
Datasets	Resolution	Prediction with original velocity fields			Prediction with predicted velocity fields		
		PSNR	Cos similarity	*MSE*	*PSNR*	Cos similarity	*MSE*
Single source smoke (3D)	64^3	26.279	0.889	0.003	26.738	0.865	0.004
Single source smoke with obstacles (3D)	64^3	35.717	0.993	0.000 2	35.340	0.989	0.000 02
Single source smoke (2D)	64^2	28.621	0.990	0.002	17.904	0.842	0.023
Single source smoke (2D)	128^2	25.886	0.984	0.003	18.418	0.886	0.019
Rising smoke (2D)	64^2	18.465	0.895	0.009	17.760	0.875	0.010
Rising smoke (2D)	128^2	19.483	0.897	0.019	17.717	0.837	0.029

4 Results

The effectiveness of our temporal evolution models is verified through visualization results and performance measurement with a series of tests.

4.1 Visualization Results

Typical outputs of long-term prediction sub-networks can be seen in Figs. 6 and 7. In Fig. 6, the density field and corresponding velocity field prediction of a testing example from the single source smoke datasets (resolution of 64×64) are shown. Model outputs are shown in the top row, and the corresponding ground truth is shown in the bottom row. Compared with the ground truth shown in the bottom row, our approach successfully predicts the macroscopic distributions of velocity fields with previous frames as inputs. In addition, the long-term density prediction sub-network also predicts the motion of single source smoke over long periods accurately. Another testing example from single source smoke with obstacles datasets (resolution of 128×128) is displayed in Fig. 7. Model outputs are shown in the top row, and the corresponding ground truth is shown in the bottom row. It proves that our model not only achieves high accuracy on high-resolution datasets but also learns interactions between smoke and obstacles without additional input information about obstacles. Our approach could learn the interactions between smoke and obstacles based on velocity fields.



(a) Long-term velocity field prediction

(b) Long-term density field prediction

Fig. 6 Long-term density and velocity field prediction of single source smoke datasets (resolution of 64×64)

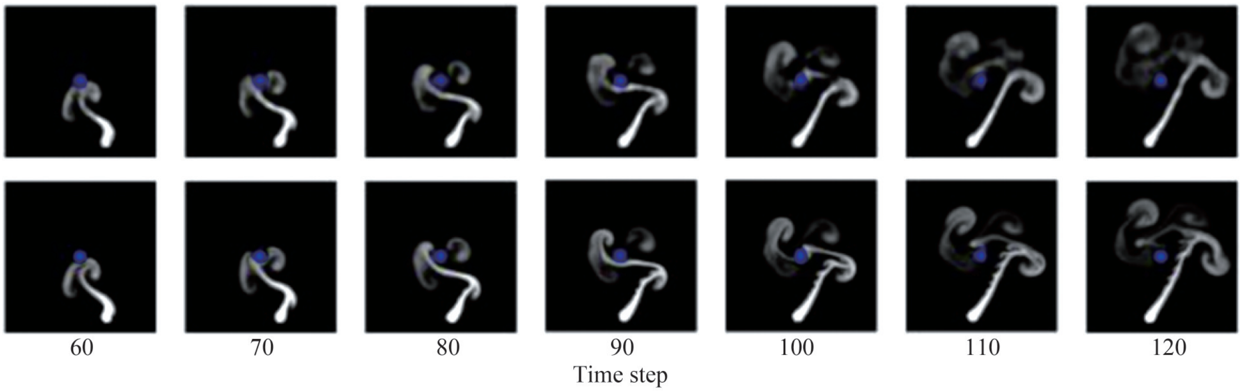


Fig. 7 Long-term density field prediction of single source smoke with obstacles datasets (resolution of 128×128)

As shown in Fig. 8, our long-term density prediction sub-network can also be applied to other simulation scenes like cold smoke in a rotating cup. Specifically, model outputs are shown in the top row, and the corresponding ground truth is shown in the bottom row. In addition, prediction examples of rotating and moving cup datasets (2D) with a translation are shown in Fig. 9. Specifically, model outputs are shown in the top row, and the corresponding ground truth is shown in the bottom row. It can be seen that our model can predict the macroscopic distributions of density fields with random moving obstacles.

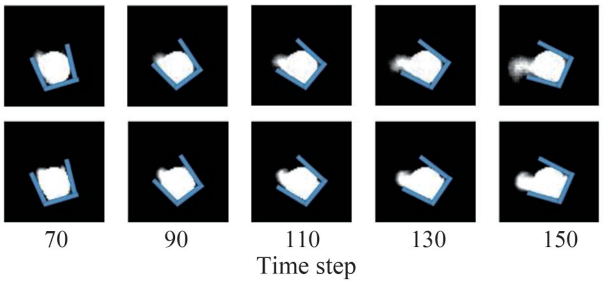


Fig. 8 Long-term density field prediction of rotating cup datasets (resolution of 64×64)

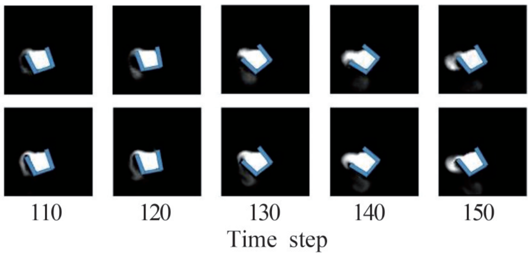


Fig. 9 Long-term density field prediction of rotating and moving cup datasets (resolution of 64×64)

In order to predict density and velocity fields by the long-term motion prediction model, the long-term density prediction sub-network and the velocity prediction sub-network are trained and tested with datasets in the first place. Then output velocity fields from the long-term velocity prediction sub-network are used as inputs for predicting density fields. Outputs of the long-term density prediction sub-network with predicted and original velocity fields as inputs and corresponding ground truth are listed together for comparison in Fig. 10, where density fields predicted with predicted and original velocity fields are shown in the top and middle row, and corresponding ground truth is shown in the bottom row. Therefore, our approach can predict the macroscopic distributions of density fields similar to ground truth with predicted velocity fields. It can be found that the output density fields of the long-term motion prediction model are less accurate than those predicted with original velocity fields by the long-term density prediction sub-network. Low accuracy of density field prediction is caused by parts of detailed distribution loss in velocity field prediction. However, the motion tendency of outputs from the long-term motion prediction model is still consistent with the ground truth. In other words, the macroscopic distributions of physical fields can be successfully predicted by the proposed model. Prediction results of the 3D single source smoke scenes in Figs. 11 and 12 demonstrate the capabilities of our long-term motion prediction model to achieve satisfying temporal evolution results on simple 3D datasets with encoders composed of deeper convolutional networks. In Fig. 11, density fields predicted with predicted and original velocity fields are shown in the top and middle row, and corresponding ground truth is shown in the bottom row. Therefore, our approach can accurately predict the macroscopic distributions of physical fields in simple 3D scenes. In Fig. 12, density fields predicted with predicted and original velocity fields are shown in the top and middle row, and corresponding ground truth is shown in the bottom row.

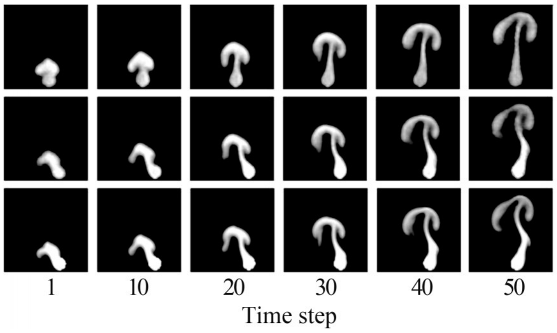


Fig. 10 Long-term density field prediction of single source smoke datasets (resolution of 64×64)

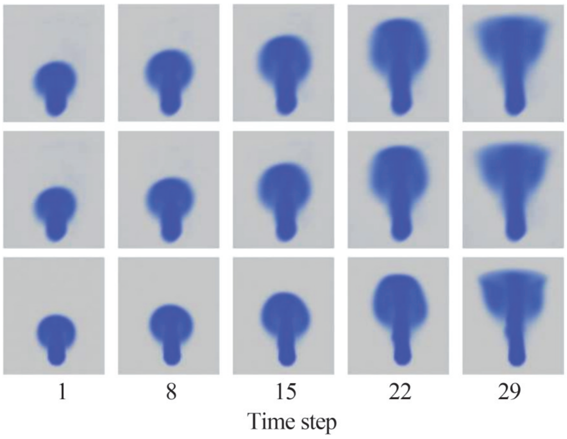


Fig. 11 Long-term density field prediction of 3D single source smoke datasets (resolution of 64×64×64)

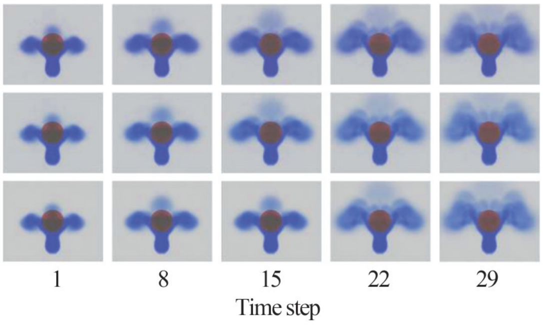


Fig. 12 Long-term density field prediction of 3D single source smoke with obstacles datasets (resolution of 64×64×64)

4.2 Performance Measurement

Different from traditional fluid simulation methods updating physical fields frame by frame, the U-Net-based encoders extracting information from sequences of input physical fields and the advection in latent space enable our model to predict the physical fields of many frames in one calculation. We further compare the long-term velocity prediction sub-network with the LSP and LSS models (velocity prediction only). As shown in the method performance measurement in Table 6, our model can increase speed by up to three times for rising smoke (resolution of 64×64) datasets and about two times for the rising smoke datasets (higher resolution of 128×128) with the same size of latent space. As for density prediction, our model implements rapid long-term density advection through one calculation in latent space instead of iteration frame by frame. It can be found that 3D fully convolutional encoding-decoding networks and advection carried out in latent space help our model improve both accuracy and performance. Concrete mean inference time statistics of our long-term prediction sub-networks on different datasets can be found in Table 7. All performance measurements are created with the time module provided by python on Intel(R) Xeon(R) E5-2690 v4 (2.60 GHz) and NVIDIA GTX TITAN X (Pascal).

Table 6 Timing of a simulation step computed via long-term velocity prediction sub-network, LSP model, and LSS model averaged over testing data (50 time steps prediction) Our model achieves further speed-ups than the LSP and LSS models ms

Datasets	Rising smoke 64×64	Rising smoke 128×128
LSP model	1.50	1.83
LSS model	17.31	24.20
Our model	**0.56**	**0.92**

Table 7 Timing of a simulation step computed via long-term prediction sub-networks averaged over testing data (50 time steps for velocity prediction and 150 time steps for density prediction)

Datasets	Resolution	Velocity prediction/ ms	Density prediction/ ms
Single source smoke (3D)	64^3	25.80	24.14
Single source smoke with obstacles (3D)	64^3	25.43	23.66
Rotating cup (2D)	64^2	0.59	0.41
Rotating and moving cup (2D)	64^2	0.84	0.52
Rising smoke (2D)	64^2	0.56	0.66
Rising smoke (2D)	128^2	0.92	1.02
Single source smoke (2D)	64^2	0.66	0.43
Single source smoke (2D)	128^2	1.06	0.89
Single source smoke with obstacles (2D)	64^2	0.65	0.43
Single source smoke with obstacles (2D)	128^2	1.03	0.81

4.3 Limitations

While our model shows outstanding performance on long-term motion prediction of physical field evolution, there are still several aspects that can be improved by follow-up works. Firstly, our model concentrates on the prediction of macroscopic distributions in physical fields in the sacrifice of parts of details. As a result, detail loss becomes more serious and leads to blurred visual effects as the data

https://dc-china-simulation.researchcommons.org/journal/vol35/iss3/1
DOI: 10.16182/j.issn1004731x.joss.22-1507

resolution increases. Therefore, it will be meaningful to combine our model with some mature detail enhancement methods to achieve better visualization results. Secondly, we find it difficult for the proposed model to realize motion prediction for complex 3D datasets in our experiments. We believe that with appropriate modifications, our model can reach better performance on more complex 3D datasets.

5 Conclusion

In this paper, we propose a learning-based algorithm focusing on matching macroscopic distributions of the network-predicted physical fields with the simulated data based on the Navier-Stokes equation. Instead of replacing specific steps in traditional fluid simulation, our algorithm predicts density and velocity fields based on an initial frame of density fields and a sequence of previous velocity fields. Our model consists of two functional sub-networks that can predict density fields and velocity fields separately. The whole model reasonably combines the two fields to achieve long-term velocity and density field prediction. The proposed end-to-end high-performance network architectures can yield significant simulation performance increments. Through detailed evaluations with a series of 2D and 3D simulation datasets, we demonstrate remarkable improvements in predicting velocity and density fields provided by our approach compared with previous works.

In conclusion, we obtain an end-to-end high-performance model that significantly improves accuracy and efficiency for long-term motion prediction of physical flow field evolution. Our work can be applied to rapidly generate virtual scenes and animations. Besides, it is also applicable to generate simulation data in addition to limited real data for data-efficient tasks.

As for future work, it will be reasonable to take advantage of information from pressure fields to control and improve temporal evolution reality since our model is only based on density and velocity fields of fluid flows. Moreover, transfer learning methods could be applied to our model to increase its extensibility on all kinds of simulation scenes. Our approach can be applied to accelerate existing simulation methods as well.

References:

[1] Kim B, Azevedo V C, Thuerey N, et al. Deep Fluids: A Generative Network for Parameterized Fluid Simulations [J]. Computer Graphics Forum (S0167-7055), 2019, 38 (2): 59-70.

[2] Pfaff T, Fortunato M, Sanchez-Gonzalez A, et al. Learning Mesh-Based Simulation with Graph Networks[C]// International Conference on Learning Representations. Vienna, Austria: arXiv: 2010.03409, 2021.

[3] Xie Y, Franz E, Chu M, et al. TempoGAN: A Temporally Coherent, Volumetric GAN for Super-Resolution Fluid Flow[J]. ACM Transactions on Graphics (S1557-7368), 2018, 37(4CD): 95:1-95:15.

[4] Werhahn M, Xie Y, Chu M, et al. A Multi-Pass GAN for Fluid Flow Super-Resolution[J]. Proceedings of ACM on Computer Graphics and Interactive Techniques (S2577-6193), 2019, 2(2): 10:1-10:21.

[5] Prantl L, Chentanez N, Jeschke S, et al. Tranquil Clouds: Neural Networks for Learning Temporally Coherent Features in Point Clouds[C]// International Conference on Learning Representations. Addis Ababa, Ethiopia: arXiv: 1907.05279, 2020.

[6] Wiewel S, Becher M, Thuerey N, et al. Latent Space Physics: Towards Learning the Temporal Evolution of Fluid Flow[J]. Computer Graphics Forum (S0167-7055), 2019, 38(2): 71-82.

[7] Wiewel S, Kim B, Azevedo V C, et al. Latent Space Subdivision: Stable and Controllable Time Predictions for Fluid Flow[J]. Computer Graphics Forum (S0167-7055), 2020, 39(8): 15-25.

[8] Tompson J, Schlachter K, Sprechmann P, et al. Accelerating Eulerian Fluid Simulation with Convolutional Networks[C]// 34th International

Conference on Machine Learning. Sydney, Australia: PMLR, 2017: 3424-3433.

[9] Zhu Y, Zabaras N, Koutsourelakis P S, et al. Physics-Constrained Deep Learning for High-Dimensional Surrogate Modeling and Uncertainty Quantification without Labeled Data[J]. Journal of Computational Physics (S0021-9991), 2019, 394: 56-81.

[10] Li Xiaosheng, Liu Le, Wu Wen, et al. Dynamic BFECC Characteristic Mapping Method for Fluid Simulations[J]. Visual Computer (S1432-2315), 2014, 30: 787-796.

[11] Wang Xin, Liu Yingjie. Back and Forth Error Compensation and Correction Method for Linear Hyperbolic Systems with Application to the Maxwell's Equations[J]. Journal of Computational Physics: X, (S0021-9991), 2019, 1(4): 100014.

[12] Ronald Fedkiw, Jos Stam, Henrik Wann Jensen. Visual Simulation of Smoke[C]// SIGGRAPH 2001: Computer Graphics Conference Proceedings. Los Angeles, California, USA: ACM, 2001: 15-22.

[13] Ding W, Jia L, Jiao H, et al. A New Method of Smoke Simulation[C]// The 2010 International Conference on Educational and Network Technology (ICENT). Qinhuangdao, China: IEEE, 2010: 267-270.

[14] Xu Y, Liu S, Wu L. Smoke Simulation with Two-Scale Vorticity Confinement[C]// International Conference on Multimedia and Signal Processing. Shanghai, China: Springer, 2012: 467-474.

[15] He X, Luo L S. Theory of the Lattice Boltzmann Method: From the Boltzmann Equation to the Lattice Boltzmann Equation[J]. Physical Review E (S1539-3755), 1997, 56 (6): 6811-6817.

[16] Li W Z, Zhou X, Dong B, et al. A Thermal LBM Model for Studying Complex Flow and Heat Transfer Problems in Body-Fitted Coordinates[J]. International Journal of Thermal Sciences (S1290-0729). 2015, 98: 266-276.

[17] J Wen, H Ma. Real-Time Smoke Simulation Based on Vorticity Preserving Lattice Boltzmann Method[J]. Visual Computer (S1432-2315), 2019, 35: 1279-1292.

[18] Foster N, Fedkiw R. Practical Animation of Liquids[C]// SIGGRAPH 2001. Los Angeles, CA, USA: ACM, 2001: 23-30.

[19] Zhu Y, Bridson R. Animating Sand as a Fluid[J]. ACM Transactions on Graphics (TOG) (S1557-7368), 2005, 24 (3): 965-972.

[20] Macklin M, Mueller M, Chentanez N, et al. Unified Particle Physics for Real-Time Applications[J]. ACM Transactions on Graphics (TOG) (S1557-7368), 2014, 33 (4CD): 1-12.

[21] Gissler C, Peer A, Band S, et al. Interlinked SPH Pressure Solvers for Strong Fluid-Rigid Coupling[J]. ACM Transactions on Graphics (TOG) (S1557-7368), 2019, 38(1): 1-13.

[22] Huang L, Hdrich T, Michels D L. On the Accurate Large-Scale Simulation of Ferrofluids[J]. ACM Transactions on Graphics (TOG) (S1557-7368), 2019, 38(4): 1-15.

[23] Nagasawa K, Suzuki T, Seto R, et al. Mixing Sauces: A Viscosity Blending Model for Shear Thinning Fluids[J]. ACM Transactions on Graphics (TOG) (S1557-7368), 2019, 38(4): 1-17.

[24] Rumelhart D E, Hinton G E, Williams R J. Learning Representations by Back Propagating Errors[J]. Nature (S1476-4687). 1986, 323(6088): 533-536.

[25] Ladick L, Jeong S H, Solenthaler B, et al. Data-Driven Fluid Simulations Using Regression Forests[J]. ACM Transactions on Graphics (TOG) (S1557-7368), 2015, 34 (6): 1-9.

[26] Thuerey N, Weienow K, Prantl L, et al. Deep Learning Methods for Reynolds-Averaged Navier-Stokes Simulations of Airfoil Flows[J]. AIAA Journal (S1533-385X), 2020, 58(4): 1-12.

[27] Um K, Hu X, Thuerey N. Liquid Splash Modeling with Neural Networks[J]. Computer Graphics Forum (S0167-7055), 2018, 37(8): 171-182.

[28] Hennigh O. Lat-Net: Compressing Lattice Boltzmann Flow Simulations Using Deep Neural Networks[J]. 10.48550/arXiv: 1705.09036, 2017.

[29] Chu M, Thuerey N. Data-Driven Synthesis of Smoke Flows with CNN-Based Feature Descriptors[J]. ACM Transactions on Graphics (TOG) (S1557-7368), 2017, 36 (4): 69:1-69:14.

[30] Kani J N, Elsheikh A H. Reduced Order Modeling of Subsurface Multiphase Flow Models Using Deep Residual Recurrent Neural Networks[J]. Transport in Porous Media (S1573-1634), 2019, 126: 713-74.

[31] Sanchez-Gonzalez A, Godwin J, Pfaff T, et al. Learning to Simulate Complex Physics with Graph Networks[C]// 37th International Conference on Machine Learning. Vienna, Austria: JMLR, 2020: 8459-8468.

[32] Ummenhofer B, Prantl L, Thürey N, et al. Lagrangian Fluid Simulation with Continuous Convolutions[C]// International Conference on Learning Representations. Addis Ababa, Ethiopia: OpenReview.net, 2020.

[33] Zhang M, Wang J, Tlhomole J, et al. Learning to Estimate and Refine Fluid Motion with Physical Dynamics[C]// 39th International Conference on Machine Learning. Baltimore, Maryland, USA: arXiv: 2206.10480, 2022.

[34] Rasp P, Düben P D, Scher S, et al. WeatherBench: A

https://dc-china-simulation.researchcommons.org/journal/vol35/iss3/1
DOI: 10.16182/j.issn1004731x.joss.22-1507

Benchmark Data Set for Data-Driven Weather Forecasting[J]. Journal of Advances in Modeling Earth Systems (S1942-2466), 2020, 12(11): e2020MS002203.

[35] Eivazi H, Veisi H, Naderi M H, et al. Deep Neural Networks for Nonlinear Model Order Reduction of Unsteady Flows[J]. Physics of Fluids (S1070-6631), 2020, 32(10): 105104: 1-105104:21.

[36] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative Adversarial Nets[C]// Neural Information Processing Systems. Montreal, Quebec, Canada: MIT Press, arXiv: 1406.2661, 2014.

[37] Reed S, Akata Z, Yan X, et al. Generative Adversarial Text to Image Synthesis[C]//. International Conference on Machine Learning. New York City, NY, USA: Curran Associates, Inc., 2016: 1060-1069.

[38] Chu M, Xie Y, Mayer J, et al. Learning Temporal Coherence Via Self-Supervision for GAN-Based Video Generation[J]. ACM Transactions on Graphics (TOG) (S1557-7368), 2020, 39(4): 75:1-75:13.

[39] Wei W, Liu S. Interpolating Frames for Super-Resolution Smoke Simulation with GANs[C]// International Conference on Computer Animation and Social Agents (CASA). Bournemouth, UK: Springer, 2020: 14-21.

[40] Kochkov D, Smith J A, Alieva A, et al. Machine Learning Accelerated Computational Fluid Dynamics[J]. Proceedings of the National Academy of Sciences (S0027-8424), 2021, 118(21): e2101784118.