

RESEARCH ARTICLE | OCTOBER 06 2026

Physics-informed wavelet–Fourier representation for multiscale fluid dynamics

Chao Wang ; Shilong Li ; Yunpeng Wang; Tianbai Xiao ; Zelong Yuan  ; Chenyue Xie ; Chunyu Guo 



Physics of Fluids 38, 105126 (2026)

<https://doi.org/10.1063/5.0351931>



Articles You May Be Interested In

AVS Quantum Science publishes original research and more in 2021

AVS Quantum Sci. (March 2021)

Seven nonstandard models coupling quantum matter and gravity

AVS Quantum Sci. (May 2022)



Physics of Fluids

Special Topics Open
for Submissions

[Learn More](#)



Physics-informed wavelet-Fourier representation for multiscale fluid dynamics

Cite as: Phys. Fluids **38**, 105126 (2026); doi: [10.1063/5.0351931](https://doi.org/10.1063/5.0351931)

Submitted: 10 July 2026 · Accepted: 15 September 2026 ·

Published Online: 6 October 2026



View Online



Export Citation



CrossMark

Chao Wang,^{1,2,3} Shilong Li,^{1,2} Yunpeng Wang,⁴ Tianbai Xiao,⁵ Zelong Yuan,^{1,2,3,6,a)} Chenyue Xie,^{7,b)} and Chunyu Guo^{1,2,3,6}

AFFILIATIONS

¹College of Shipbuilding Engineering, Harbin Engineering University, Harbin 150001, China

²Qingdao Innovation and Development Base, Harbin Engineering University, Qingdao 266000, China

³National Key Laboratory of Hydrodynamics, Harbin Engineering University, Harbin 150001, China

⁴Department of Mechanics and Aerospace Engineering, Southern University of Science and Technology, Shenzhen 518055, China

⁵Centre for Interdisciplinary Research in Fluids, Institute of Mechanics, Chinese Academy of Sciences, Beijing 100190, China

⁶Nanhai Institute of Harbin Engineering University, Harbin Engineering University, Sanya 572024, China

⁷Department of Modern Mechanics, University of Science and Technology of China, Hefei 230026, China

^{a)}Author to whom correspondence should be addressed: yuanzelong@hrbeu.edu.cn

^{b)}Electronic mail: cyxie@ustc.edu.cn

ABSTRACT

Multiscale fluid flows often contain localized flow structures, such as viscous shock layers, wet-dry fronts, steady viscous wakes, decaying vortical structures, and vortex-shedding patterns, whose accurate prediction requires the simultaneous preservation of global conservation trends and small-scale gradients. This study examines these flow-physics requirements through a physics-informed wavelet-Fourier (PIWF) representation for multiscale fluid dynamics. Instead of relying on a monolithic neural approximator, the formulation separates two complementary components of the flow field within a physics-informed neural representation: long-range coherent modes through a Fourier-basis branch and localized steep-gradient or vortical features through a compactly supported wavelet branch. The outputs are fused with a residual multilayer perceptron using channel attention, and the governing equations, initial conditions, and boundary conditions are imposed directly through the physics-informed loss. The model is assessed on five canonical fluid-dynamics problems: Burgers' equation, the shallow water equations, Kovasznay flow, Taylor-Green vortex flow, and two-dimensional cylinder wake. The results show that PIWF improves the resolution of shock-like gradients, wet-dry interfaces, steady wake fields, decaying vortical structures, vorticity extrema, and broadband wake spectra relative to standard physics-informed neural networks and physics-informed Kolmogorov-Arnold networks. These findings indicate that a wavelet-Fourier physics-informed representation can provide a useful route for analyzing multiscale flow phenomena when high-fidelity interior reference data are limited or unavailable.

Published under an exclusive license by AIP Publishing. <https://doi.org/10.1063/5.0351931>

I. INTRODUCTION

Multiscale fluid flows are organized by interacting structures across widely separated length and time scales, including shock-like gradients, wet-dry fronts, coherent vortices, and broadband wake motions. Partial differential equations (PDEs) provide the fundamental mathematical framework for describing a broad class of physical phenomena and engineering applications,¹ including fluid dynamics,² wave propagation,³ reactive transport,^{4,5} and multiphysics coupling.⁶ In the absence of analytical solutions for most practical problems, classical numerical methods such as finite element methods,⁷ finite

difference methods,⁸ and finite volume methods⁹ have been widely employed and have achieved remarkable success.¹⁰ However, these classical numerical methods face fundamental limitations in real-world PDE problems.^{11,12} Multiscale dynamics demand extremely fine resolutions to capture sharp gradients, discontinuities, and singularities, leading to the curse of dimensionality and prohibitively high computational cost.¹³ Complex geometries require manual and time-consuming mesh generation, while strongly coupled nonlinear processes introduce stiffness, stability issues, and conservation difficulties.¹⁴ Consequently, the development of robust, flexible, and high-fidelity numerical solvers

remains critical for advancing scientific discovery and engineering applications.¹⁵ For fluid-dynamics applications, this need is closely tied to the faithful representation of the flow structures that control transport, separation, front propagation, and vortex dynamics.

In this context, physics-informed and data-augmented approaches have emerged as promising alternatives that effectively balance predictive accuracy with physical consistency. These methods facilitate flexible exploration of parameter spaces, inverse problems, and multi-fidelity modeling.¹⁶ In particular, physics-informed neural networks (PINNs) have gained considerable attention by embedding the governing PDEs directly into the training loss via automatic differentiation. This approach yields mesh-free and differentiable solvers that can incorporate sparse or noisy data while strictly enforcing physical laws.^{17–20} For fluid-dynamics problems, recent physics-informed studies have emphasized that predictive accuracy should be assessed together with the consistency of the recovered multiscale flow structures, rather than only through aggregate error metrics.^{21–23} PINNs have been successfully applied to a wide range of forward and inverse problems.²⁴ In parameter estimation tasks, they enable the simultaneous inference of the solution to the governing PDEs and the identification of unknown coefficients or parameters directly from sparse and noisy observations.¹⁶ In topology optimization, PINNs facilitate mesh-free, gradient-based design of material distributions by embedding physical constraints into the loss function.²⁵ For multiphysics couplings, they provide a unified framework for handling strongly coupled systems (like microfluidic problems) without requiring separate solvers or explicit interface conditions.²⁶ These applications often provide substantial practical advantages over conventional mesh-based solvers in terms of flexibility, ease of implementation, and handling of complex geometries.^{27–29}

However, standard PINN architectures struggle to simultaneously capture global structures and small-scale local features, especially in problems involving sharp interfaces, vortical structures, or highly intermittent dynamics.^{21,30–32} A key challenge is that neural networks tend to capture dominant global modes more readily than localized phenomena, reflecting their spectral biases in fixed-activation networks, which might lead to slow convergence and suboptimal accuracy in multiscale regimes.³³ To circumvent these limitations, various strategies have been proposed. These include adaptive sampling techniques that dynamically refine collocation points near discontinuities, the explicit incorporation of Rankine–Hugoniot jump conditions for conservation laws, domain decomposition methods, and the use of multiscale activation functions or positional encodings. While these approaches can alleviate spectral bias to some extent, each comes with its own tradeoffs, such as increased implementation complexity, problem-specific tuning, or additional computational overhead. To circumvent these limitations, recent research has shifted toward architectures that depart from the traditional fixed-node paradigm to more effectively extract multiscale information without prohibitive computational overhead. Another promising direction is the Kolmogorov–Arnold network (KAN) introduced by Liu *et al.*,³⁴ which represents a fundamental shift in neural-network topology by replacing fixed activation functions on nodes with learnable, spline-based functions on the network edges. This structural innovation allows for more flexible nonlinear mapping and has demonstrated superior Pareto frontiers in both data-fitting and PDE benchmarks. Building upon this framework to further

enhance numerical stability and spectral accuracy, Guo *et al.*³⁵ integrated orthogonal Chebyshev polynomials into the KAN architecture. Bozorgasl *et al.* proposed the wavelet Kolmogorov–Arnold networks,³⁶ which leverages wavelet transforms to efficiently capture both high- and low-frequency components of data, significantly improving predictive accuracy, training efficiency, and inference robustness compared to KAN and MLPs. Aghaei *et al.* proposed the rational Kolmogorov–Arnold networks,³⁷ which introduces rational basis functions based on the Padé approximation and rational Jacobi functions, demonstrating superior performance in function approximation across deep learning and physics-informed tasks.

Although KAN-based architectures enhance the network's ability to capture multiscale features, the underlying B-spline basis functions are inherently ill-suited for approximating discontinuities and abrupt changes, limiting their effectiveness in representing extremely localized multiscale phenomena. In such cases, wavelet analysis has been recognized as a powerful tool for representing time series and signals,^{38,39} which excels at capturing localized features and nonlinearities.^{40,41} In the context of PDE solving, Harnish *et al.*⁴² developed an adaptive wavelet method for coupled nonlinear partial differential equations. By exploiting the multi-resolution properties of wavelet bases, their approach dynamically adapts the computational grid, maintains high accuracy, and achieves substantial data compression, with successful applications in dynamic damage modeling. However, classical wavelet-based methods are typically restricted to low-dimensional problems, as the explicit construction of wavelet bases becomes computationally prohibitive in high-dimensional spaces.^{43,44}

In contrast, neural networks can approximate arbitrary nonlinear functions without requiring strong prior assumptions about the underlying dynamics.⁴⁵ Nevertheless, traditional neural architectures may encounter optimization difficulties, including vanishing or poorly scaled gradients, when representing multiscale solution fields with sharp transitions. As a result, their training can become slow, initialization-sensitive, and prone to inaccurate reconstruction of localized high-frequency or high-gradient features.⁴⁶ These limitations have motivated the integration of wavelet techniques into neural architectures. Alexandridis *et al.*⁴⁷ established a comprehensive statistical framework for wavelet networks, addressing network structure, training algorithms, initialization, variable selection, and model validation. Navaneeth *et al.*⁴⁸ proposed the physics-informed wavelet neural operator for learning solution operators for parametric PDEs without requiring high-fidelity training data. Gu *et al.*⁴⁹ introduced the physics-informed multiscale attention wavelet neural operator as a surrogate model for linear parabolized stability equations, demonstrating strong performance on canonical instability modes in hyper-sonic boundary layers.

Despite these promising advances, each approach carries its own limitations when applied to physics-informed modeling of multiscale phenomena. KANs, despite their improved PDE accuracy, rely on B-spline basis functions that are naturally suited for approximating smooth functions. They become less effective in resolving sharp local discontinuities or abrupt transitions, as spline representations tend to smooth out high-frequency oscillations. This architecture makes it difficult to balance the extraction of global structures and localized features simultaneously, especially when confronted with shocks or highly intermittent dynamics. Existing approaches still struggle to simultaneously capture global large-scale structures and localized

high-frequency features within a single physics-informed framework, especially when high-fidelity reference data are scarce or unavailable. This persistent trade-off between global consistency and local detail, inherited from the spectral bias of standard PINNs and the dimensionality limitations of classical wavelets, limits their robustness for practical multiscale PDE problems. To address the aforementioned limitations, we propose the physics-informed wavelet–Fourier (PIWF) model, a scale-aware representation for multiscale fluid dynamics that synergistically integrates multiscale wavelet representations with physics-informed learning. The core innovation lies in effectively bridging the persistent trade-off between global coherence and local fidelity that constrains conventional PINNs. Specifically, PIWF incorporates a dedicated wavelet layer that embeds wavelet series expansions directly into the network, enabling accurate capture of localized high-frequency features such as sharp interfaces and coherent vortical structures. This is complemented by two parallel branches: a Fourier-basis layer⁵⁰ for efficient extraction of global features and a conventional multi-layer perceptron (MLP) for residual compensation and information integration. Channel attention⁵¹ is then employed to adaptively fuse the outputs from these branches, enabling a data-driven balance between global consistency and local fidelity. By embedding the governing physical laws into the loss function, PIWF can be trained with little or no high-fidelity interior reference data. The emphasis is therefore on resolving physically meaningful multiscale flow structures.

The remainder of this paper is organized as follows: Sec. II presents the proposed PIWF architecture, including the formulation of the wavelet layer, the integration of the Fourier-basis layer and MLP branches, the channel attention fusion mechanism, and the incorporation of physics-informed constraints. Section III assesses PIWF through numerical experiments on several canonical fluid dynamics problems, including Burgers' equation, the shallow-water equations, Kovasznay flow, Taylor–Green vortex flow, and the two-dimensional cylinder wake. Finally, conclusions and outlooks are discussed in Sec. IV.

II. METHODOLOGY

The PIWF formulation is constructed to reflect a common structure of multiscale fluid solutions: large-scale coherent motion coexists with localized fronts, shear layers, shocks, or vortex cores. A discrete wavelet expansion provides localized multiresolution support for the latter features, whereas Fourier bases efficiently represent spatially extended and smoothly varying modes. We therefore reformulate the conventional fully connected representation into three parallel components: a wavelet layer for local high-gradient content, a Fourier-basis layer⁵⁰ for global coherent content, and a residual MLP branch for nonlinear compensation. Their outputs are fused through channel attention so that the relative contribution of each scale-aware representation can vary during training. The resulting physics-informed wavelet–Fourier model is designed for flow fields governed by conservation laws while retaining interpretable links between the numerical representation and the physical structures being resolved.

A. Physics-informed neural network (PINN)

The core idea of physics-informed neural networks is to incorporate physical governing equations into neural network models.

This allows PINNs to overcome key limitations of purely data-driven methods by enforcing physical consistency in the model outputs. Consider a general form of nonlinear PDE

$$\frac{\partial \mathbf{u}}{\partial t} = R(\mathbf{u}; \mathbf{x}, t), \quad \text{in } \Omega \times [T_0, T], \quad (1)$$

$$\mathcal{B}(\mathbf{u}) = \mathbf{u}_g(\mathbf{x}, t), \quad \text{on } \partial\Omega \times [T_0, T], \quad (2)$$

$$\mathcal{I}(\mathbf{u}) = \mathbf{u}_a(\mathbf{x}, t), \quad \text{in } \Omega \times \{T_0\}, \quad (3)$$

where Ω is the spatial domain and $[T_0, T]$ is the time interval, with T_0 and T denoting the initial and terminal time, respectively. The operators $\mathcal{B}(\mathbf{u})$ and $\mathcal{I}(\mathbf{u})$ represent the boundary and initial conditions, respectively. The parameters of the neural network are optimized by minimizing the composite loss function

$$\begin{aligned} \mathcal{L}(\theta) = & \underbrace{\frac{\lambda_1}{N_{ic}} \sum_{i=1}^{N_{ic}} \|\mathcal{I}[\mathbf{u}_\theta(\mathbf{x}^i, T_0)] - \mathbf{u}_a(\mathbf{x}^i, T_0)\|^2}_{\mathcal{L}_{ic}(\theta)} \\ & + \underbrace{\frac{\lambda_2}{N_{bc}} \sum_{i=1}^{N_{bc}} \|\mathcal{B}[\mathbf{u}_\theta(t^i, \mathbf{x}_{bc}^i)] - \mathbf{u}_g(\mathbf{x}_{bc}^i, t^i)\|^2}_{\mathcal{L}_{bc}(\theta)} \\ & + \underbrace{\frac{\lambda_3}{N_{pde}} \sum_{i=1}^{N_{pde}} \|\mathcal{L}_{pde}(\mathbf{u}_\theta, \mathbf{x}_r^i, t_r^i)\|^2}_{\mathcal{L}_{pde}(\theta)}, \end{aligned} \quad (4)$$

where $\mathbf{x}^i \in \Omega$, $\mathbf{x}_{bc}^i \in \partial\Omega$, $t^i \in [T_0, T]$, and $\mathbf{u}_\theta$ represents the outputs of neural networks. $\mathbf{u}_g$ and $\mathbf{u}_a$ denote the boundary conditions and initial conditions, respectively. N_{ic} , N_{bc} , and N_{pde} are the numbers of sampling points corresponding to initial conditions, boundary conditions, and collocation points, respectively. λ_1 , λ_2 , and λ_3 are the correspondingly given weights of each loss. By substituting the input pair of collocation points for PDE ($\mathbf{x}_{pde}$, t_{pde}) and model prediction $\mathbf{u}_\theta(\mathbf{x}_{pde}, t_{pde})$ into Eq. (1), the equation residual can be obtained through automatic differentiation, namely,

$$\mathcal{L}_{pde}(\mathbf{u}_\theta; \mathbf{x}_{pde}, t_{pde}) = \frac{\partial \mathbf{u}_\theta}{\partial t}(\mathbf{x}_{pde}, t_{pde}) - R(\mathbf{u}_\theta; \mathbf{x}_{pde}, t_{pde}). \quad (5)$$

The standard PINN employs a fully connected multi-layer perceptron (MLP) as the backbone to approximate the solution $\mathbf{u}(\mathbf{x}, t)$. In this work, the MLP consists of an input layer, L hidden layers, and an output layer. Let $\mathbf{a} = (\mathbf{x}, t)$ denote the input coordinate vector. The neural network is defined recursively by

$$\begin{aligned} \mathbf{y}^{(0)} &= \mathbf{a}, \\ \mathbf{y}^{(l)} &= \sigma(\mathbf{W}^{(l)} \mathbf{y}^{(l-1)} + \mathbf{b}^{(l)}), \quad l = 1, 2, \dots, L-1, \\ \mathbf{u}_\theta(\mathbf{x}, t) &= \mathbf{W}^{(L)} \mathbf{y}^{(L-1)} + \mathbf{b}^{(L)}, \end{aligned} \quad (6)$$

where $\mathbf{W}^{(l)} \in \mathbb{R}^{N_l \times N_{l-1}}$ and $\mathbf{b}^{(l)} \in \mathbb{R}^{N_l}$ are the weight matrix and bias vector of the l th layer, respectively, N_l is the number of neurons in the l th layer, and $\sigma(\cdot)$ is the activation function. The trainable parameters $\theta = \{\mathbf{W}^{(l)}, \mathbf{b}^{(l)}\}_{l=1}^L$ are optimized by minimizing the composite loss in Eq. (4) using the Adam optimizer,⁵² with gradients evaluated by automatic differentiation.

B. Wavelet layers

The implementation detail relevant to the physics-informed residual is stated here explicitly. The reported calculations use the exact Haar scaling and wavelet functions. Their intervals are selected by Boolean comparisons and evaluated with `torch.where`; no smoothed wavelet surrogate is introduced. Let ξ denote the coordinate vector and let $\mathbf{h}(\xi)$ be the input to a PIWF layer. In the implementation, the wavelet-designated branch contains an exact Haar subvector together with a differentiable GELU subvector. Let $\mathbf{r}_H$ collect the exact Haar-derived subvectors and let $\mathbf{s}$ collect the differentiable feature components, including the Fourier sine and cosine terms and the GELU components of the three branches. The prediction can then be written as

$$\mathbf{u}_\theta(\xi) = \mathcal{G}_\theta(\mathbf{s}(\xi), \mathbf{r}_H(\xi)), \quad \text{and} \quad \mathbf{r}_H = \Gamma_H[\mathbf{W}_H \Psi_H(\mathbf{h}) + \mathbf{b}_H], \quad (7)$$

where Ψ_H contains the retained Haar scaling and wavelet values and Γ_H denotes their learned componentwise scaling. Define $\mathcal{S}_H$ as the union of the preimages of the retained Haar interval boundaries at all wavelet layers. For $\xi \notin \mathcal{S}_H$, the masked Haar values are locally constant and automatic differentiation gives

$$D_\xi \mathbf{r}_H = \mathbf{0}, \quad \text{and} \quad D_\xi^2 \mathbf{r}_H = \mathbf{0}. \quad (8)$$

At such a collocation point, the chain rule therefore yields

$$D_\xi \mathbf{u}_\theta = \mathcal{G}_s D_\xi \mathbf{s}, \quad (9)$$

$$D_\xi^2 \mathbf{u}_\theta = \mathcal{G}_s D_\xi^2 \mathbf{s} + \mathcal{G}_{ss} [D_\xi \mathbf{s}, D_\xi \mathbf{s}]. \quad (10)$$

Here, the derivatives of $\mathcal{G}_\theta$ are evaluated at the complete state $(\mathbf{s}, \mathbf{r}_H)$. Although $\mathbf{r}_H$ is treated as fixed when the output is differentiated with respect to ξ , its value still affects the fused representation, the attention weights, the downstream Jacobians and Hessians, the predicted solution, and the optimized residual. The coordinate-dependent terms in Eqs. (9) and (10) are supplied by the Fourier and GELU components.

At a coordinate in $\mathcal{S}_H$, the classical derivative of the corresponding Haar function is undefined. In the implemented computational graph, the Boolean comparison selects one of the adjacent intervals, and `torch.where` returns the constant Haar value assigned to that interval. PyTorch then differentiates the selected computational branch; it does not introduce the distributional derivative associated with the jump. In the reported calculations, the collocation points are sampled randomly from continuous coordinate ranges, so exact coincidence with a prescribed Haar jump location is unlikely. If such a coincidence occurs on a deterministic grid, the same interval-selection rule is applied. The resulting residual is the strong-form residual obtained by automatic differentiation at the sampled points. No differentiable approximation of the Haar functions or reconstruction of their distributional derivatives is used.

Conventional fully connected layers in multi-layer perceptrons are expressed as $\mathbf{y} = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b})$, where σ denotes a nonlinear activation function. Although effective for general function approximation, this structure treats the input as a flattened vector, assigning equal importance to all data points and largely neglecting underlying spatial, temporal, or topological structures. As a result, it often fails to effectively capture localized features with sharp gradients.

To address this limitation, we introduce a dedicated wavelet layer that emulates the expansion of a wavelet series within the neural-network mapping. In the present fluid-dynamics context, the purpose of this layer is to provide localized basis support for abrupt spatial changes, including shock-like gradients, wet-dry interfaces, and compact vortical features. The core idea is to utilize a neural network to emulate the wavelet series in Eq. (A9) of Appendix A, thereby achieving function approximation at the layer level. Specifically, given input-output pairs, there exists a unique mapping $f(\mathbf{x}) : \mathbb{R}^{d_x} \mapsto \mathbb{R}^{d_y}$, where d_x and d_y denote the dimensions of the input and output spaces, respectively. The role of the neural network is to act as a surrogate model for this unique mapping, such that $\mathbf{y}_i = f(\mathbf{x}_i) \approx f_\theta(\mathbf{x}_i)$, where $f_\theta(\mathbf{x}_i)$ is the neural network function with parameters θ to be optimized.

As detailed in Appendix A, the wavelet-series representation is employed to approximate the target input-output mapping $\mathbf{x}_i \mapsto \mathbf{y}_i$. This formulation enables localized multi-resolution approximation and provides the basis for the proposed wavelet layer. Our objective is to construct the neural network layer to emulate the mathematical formulation of this wavelet series, namely,

$$f_\theta(\mathbf{x}_i) = \sum_k c_{J,k} \phi_{J,k}(\mathbf{x}_i) + \sum_{j=J}^{\infty} \sum_k d_{j,k} \psi_{j,k}(\mathbf{x}_i). \quad (11)$$

To alleviate computational redundancy and enhance efficiency, the wavelet series is further simplified as follows:

i. Regarding the scaling function, the decomposition level J is chosen such that the dilated width of the scaling function $\phi(2^{-J}\mathbf{x} - \mathbf{k})$ is greater than or equal to the entire computational domain. Assuming the mother scaling function is compactly supported on the interval $[0, L]$, it follows from Eq. (A10) in Appendix A that the support of the dilated mother scaling function is given by

$$0 \leq 2^{-J} \cdot \mathbf{x} - \mathbf{k} < L. \quad (12)$$

Solving this inequality for $\mathbf{x}$ gives

$$\mathbf{k} \cdot 2^J \leq \mathbf{x} < (\mathbf{k} + L) \cdot 2^J. \quad (13)$$

Therefore, the support interval after dilation is $[\mathbf{k} \cdot 2^J, (\mathbf{k} + L) \cdot 2^J]$, where the width of the dilated wavelet is

$$(\mathbf{k} + L) \cdot 2^J - \mathbf{k} \cdot 2^J = L \cdot 2^J. \quad (14)$$

Assuming the width of the computational domain is T , it requires that $T \leq L \cdot 2^J$. This configuration ensures that the computation of the scaling function involves only a single term corresponding to $\mathbf{k} = 0$. Furthermore, after dilation at the coarsest scale J , the wavelet spans the entire computational domain and therefore cannot be further translated. This design forces the network layer to concentrate its resources on learning the global information of the field. Without this simplification, the network would suffer not only from increased computational redundancy, but also from a greater tendency to over-fit local details.

ii. For the detail space, we restrict $\mathbf{k} \in (-2, 2)$ and arrange the indices j in ascending order, applying a truncation. This arrangement enables the wavelet basis functions to be learned sequentially, from coarse to fine details. An excessive number of detail space wavelets would cause the neural network to over-emphasize the learning of

local details at the expense of global information. Concurrently, this requires the coarsest scale level J to be sufficiently large, otherwise, for specific problems, the fixed range of k and an insufficiently small j would result in wavelets that can only be dilated and translated within the central region of the computational domain, failing to cover its boundaries. Unless otherwise specified, we set $J = 3$ throughout this paper. Here, J is the upper scale index, and $J = 3$ defines the available levels $j = 0, 1, 2, 3$. Candidate functions are generated from the finest available level toward the coarser levels, and only the first five functions are retained. Because $\psi_{j,k}(x) = 2^{j/2}\psi(2^jx - k)$, changing J alters the spatial resolution and localization of the retained Haar features while leaving their number unchanged. We examined this choice in the cylinder wake case using $J = 2, J = 3$, and $J = 4$. The corresponding mean full-field relative errors were 9.510×10^{-4} , 7.007×10^{-4} , and 1.180×10^{-3} , respectively. The intermediate value $J = 3$ gave the lowest error, indicating a suitable balance between resolving localized wake structures and maintaining stable joint optimization with the Fourier and MLP branches. We therefore use $J = 3$ as a common setting for all cases rather than tuning it separately for each benchmark. With these simplifications, the resulting streamlined network layer can be written as follows:

$$f_\theta(\mathbf{x}_i) = c_{J,0}\phi_{J,0}(\mathbf{x}_i) + \sum_{j=0}^J \sum_k d_{j,k}\psi_{j,k}(\mathbf{x}_i). \quad (15)$$

Due to the applied truncation with a cutoff number of N , we reformulate the network layer into a matrix form, yielding

$$\begin{aligned} f_\theta(\mathbf{x}) &= \mathbf{C}_{J,0}\phi_{J,0}(\mathbf{x}) + \sum_{n=0}^{N-1} \mathbf{D}_n\psi_n(\mathbf{x}) \\ &= [\mathbf{C}_{J,0}, \mathbf{D}_0, \mathbf{D}_1, \dots, \mathbf{D}_{N-1}] [\phi_{J,0}(\mathbf{x}), \psi_0(\mathbf{x}), \psi_1(\mathbf{x}), \dots, \psi_{N-1}(\mathbf{x})]^T \\ &= \mathbf{W}_1 [\phi_{J,0}(\mathbf{x}), \psi_0(\mathbf{x}), \psi_1(\mathbf{x}), \dots, \psi_{N-1}(\mathbf{x})]^T, \end{aligned} \quad (16)$$

where $\mathbf{C}_{J,0} \in \mathbb{R}^{d_\theta \times d_x}$, $\mathbf{D}_n \in \mathbb{R}^{d_\theta \times d_x}$, $\mathbf{W}_1 \in \mathbb{R}^{d_\theta \times (N+1) \cdot d_x}$, $[\dots]^T$ and $[\dots]$ denote the concatenations along the first and second dimensions, respectively, d_x is the dimension of input $\mathbf{x}$, and d_θ represents the number of nodes in the network layer.

To compensate for approximation errors introduced by basis functions, we combine the wavelet branch with a parallel fully-connected residual branch. The complete output of the wavelet layer is then

$$\mathbf{y}_w(\mathbf{x}) \triangleq \begin{bmatrix} \mathbf{W}_1 [\phi_{J,0}(\mathbf{x}), \psi_0(\mathbf{x}), \psi_1(\mathbf{x}), \dots, \psi_{N-1}(\mathbf{x})] \\ \sigma(\mathbf{B} + \mathbf{W}_2\mathbf{x}) \end{bmatrix}, \quad (17)$$

where $\mathbf{W}_1 \in \mathbb{R}^{d_\theta \times (N+1) \cdot d_x}$ and $\mathbf{W}_2 \in \mathbb{R}^{(d_\theta - d_q) \times (N+1) \cdot d_x}$. The first block corresponds to the wavelet contribution and the second to the residual MLP branch. The computation of the wavelet basis functions is implemented over the nodes of the network layer. Therefore, for each layer, a hyperparameter r_w is introduced to define the ratio between the number of nodes allocated for computing the wavelet basis functions and the number of nodes in the standard fully-connected component, namely, $r_w \cdot d_\theta = d_q$.

In this work, the Haar wavelet function is adopted as the mother wavelet because its compact support, orthogonality, symmetry, and computational efficiency make it well suited for steep and localized flow features.⁵³ In contrast to smooth global bases, Haar functions

can represent sudden changes without spreading the correction over the entire domain,⁵³ which is desirable for shocks, wet-dry fronts, and localized extrema in derivative-based flow diagnostics. The scaling and wavelet functions of the Haar wavelet are defined by

$$\phi(t) = \begin{cases} 1, & 0 \leq t < 1, \\ 0, & \text{otherwise,} \end{cases} \quad (18)$$

$$\psi(t) = \begin{cases} 1, & 0 \leq t < \frac{1}{2}, \\ -1, & \frac{1}{2} \leq t < 1, \\ 0, & \text{otherwise.} \end{cases} \quad (19)$$

The Haar wavelet function enables precise localization of singularities while maintaining sparsity and avoiding phase distortion, making it particularly suitable for capturing sharp interfaces and steep-gradient regions in fluid dynamics problems.⁵⁴

C. Fourier-basis layer

Given input-output pairs $\{\mathbf{x}_i, \mathbf{y}_i\}$, a neural network aims to learn a mapping $f_\theta(\mathbf{x}) : \mathbb{R}^{d_x} \mapsto \mathbb{R}^{d_y}$, such that $\mathbf{y}_i = f(\mathbf{x}_i) \approx f_\theta(\mathbf{x}_i)$. In the Fourier-basis layer, this mapping is approximated by a truncated Fourier series, namely,

$$f(\mathbf{x}_i) \approx A_0 + \sum_{n=1}^N \left(A_n \cos\left(\frac{2\pi n \mathbf{x}_i}{L_\Omega}\right) + B_n \sin\left(\frac{2\pi n \mathbf{x}_i}{L_\Omega}\right) \right) \approx f_\theta(\mathbf{x}_i), \quad (20)$$

where N is the order of the Fourier series. L_Ω is the domain length. The Fourier coefficients (A_0 , A_n , and B_n) are given by

$$\begin{aligned} A_n &= \frac{1}{L_\Omega} \int_0^{L_\Omega} f(\mathbf{x}) \cos\left(\frac{2\pi n \mathbf{x}}{L_\Omega}\right) d\mathbf{x}, \\ \text{and } B_n &= \frac{1}{L_\Omega} \int_0^{L_\Omega} f(\mathbf{x}) \sin\left(\frac{2\pi n \mathbf{x}}{L_\Omega}\right) d\mathbf{x}. \end{aligned} \quad (21)$$

To implement Eq. (20) within a neural network layer, the coefficients a_n^l and b_n^l respectively represent the Fourier coefficients of the l th ($l \in [0, L]$) layer of Fourier basis neural network and a weight matrix $\mathbf{W}^l = [w_1^l, w_2^l, w_3^l, \dots, w_N^l]^T$ represents the angular frequencies $2\pi n/L_\Omega$ in the Fourier series

$$\begin{aligned} f_\theta^l(\mathbf{x}_i) &\approx A_0 + \sum_{n=1}^N \left(a_n^l \cos(w_n^l \mathbf{x}_i) + b_n^l \sin(w_n^l \mathbf{x}_i) \right) \\ &= A_0 + [a_1^l, a_2^l, \dots, a_N^l] \cos\left([w_1^l, w_2^l, \dots, w_N^l]^T \mathbf{x}_i\right) \\ &\quad + [b_1^l, b_2^l, \dots, b_N^l] \sin\left([w_1^l, w_2^l, \dots, w_N^l]^T \mathbf{x}_i\right) \\ &= A_0 + \mathbf{A}^l \cos(\mathbf{W}^l \mathbf{x}_i) + \mathbf{B}^l \sin(\mathbf{W}^l \mathbf{x}_i), \end{aligned} \quad (22)$$

where $[\dots]^T$ and $[\dots]$ denote the concatenation along the first and second dimensions, respectively. To enrich the fitting possibilities of the neural network, the neural network embedded with Fourier series is extended and uses different angular frequency weight matrices

$\mathbf{W}_a^l = [w_{1a}^l, w_{2a}^l, w_{3a}^l, \dots, w_{Na}^l]^T$, and $\mathbf{W}_b^l = [w_{1b}^l, w_{2b}^l, w_{3b}^l, \dots, w_{Nb}^l]^T$, namely,

$$\begin{aligned} f_\theta^l(\mathbf{x}_i) &\approx A_0 + \mathbf{A}^l \cos(\mathbf{W}_a^l \mathbf{x}_i) + \mathbf{B}^l \sin(\mathbf{W}_b^l \mathbf{x}_i) \\ &= A_0 + \Theta^l [\cos(\mathbf{W}_a^l \mathbf{x}_i), \sin(\mathbf{W}_b^l \mathbf{x}_i)]^T \\ &= \mathcal{F}^l \circ (\mathbf{x}_i), \end{aligned} \quad (23)$$

where $\Theta^l = [\mathbf{A}^l, \mathbf{B}^l]$ and $\mathcal{F}^l$ denotes the functional mapping of the l th Fourier-basis layer. Inspired by Fourier analysis networks (FANs),⁵⁵ the stacked Fourier layers and the Fourier-basis mapping can be expressed as

$$\begin{aligned} f_\theta(\mathbf{x}) &= \mathcal{F}^L(\mathcal{F}^{L-1} \circ \mathcal{F}^{L-2} \circ \dots \circ \mathcal{F}^1 \circ \mathbf{x}) \\ &= \mathbf{A}^L + \Theta^L [\cos(\mathbf{W}_a^L(\mathcal{F}^{1:L-1} \circ \mathbf{x})) \parallel \sin(\mathbf{W}_b^L(\mathcal{F}^{1:L-1} \circ \mathbf{x}))], \end{aligned} \quad (24)$$

where $\mathbf{x} = [x_1, x_2, \dots, x_i, \dots, x_{d_x}]^T$ and d_x is the dimension of $\mathbf{x}$. This results in very poor learning of the coefficient matrix Θ^L , and it would undergo more refined learning as the depth of the neural network increases. Therefore, in each layer, we follow the strategy of FAN⁵⁵ and parallelize the coefficient matrix Θ^l and angular frequency matrix $\mathbf{W}_a^l \in \mathbb{R}^{d_p \times d_x}$ and $\mathbf{W}_b^l \in \mathbb{R}^{d_p \times d_x}$ in the same layer of neural network for learning, namely,

$$\phi^l(\mathbf{x}) \triangleq [\cos(\mathbf{W}_a^l \mathbf{x}), \sin(\mathbf{W}_b^l \mathbf{x}), \sigma(\mathbf{A}_c^l + \Theta_c^l \mathbf{x})]^T, \quad (25)$$

where $\mathbf{A}_c^l \in \mathbb{R}^{d_0}$ and $\Theta_c^l \in \mathbb{R}^{d_0 \times d_x}$ are learnable parameters. Here, d_0 denotes the first dimension of Θ_c . We refer trigonometric nodes that perform cosine and sine operations as Fourier nodes. The fitting capability of a neural network is closely related to the diversity of basis functions at each node, therefore, two learnable bias matrices are

added to the trigonometric operations of the Fourier nodes, namely, $\mathbf{B}_a \in \mathbb{R}^{d_p}$, $\mathbf{B}_b \in \mathbb{R}^{d_p}$, where d_p is half the number of Fourier neural nodes. Here, we divide the cosine and sine calculations equally. According to the trigonometric addition formulas, it can be concluded that the trigonometric function with added bias provides fitting possibilities for Fourier nodes such as $\cos^2$, $\sin^2$, and $\cos \cdot \sin$. Additional learnable coefficients C_a and $C_b \in \mathbb{R}$ are added at the Fourier nodes to increase the construction complexity of the neural network layers. Therefore, our final Fourier-basis layer is

$$\mathbf{y}_f^l(\mathbf{x}) = [C_a^l \cos(\mathbf{W}_a^l \mathbf{x} + \mathbf{B}_a^l), C_b^l \sin(\mathbf{W}_b^l \mathbf{x} + \mathbf{B}_b^l), \sigma(\mathbf{A}_c^l + \Theta_c^l \mathbf{x})]^T, \quad \text{if } l < L. \quad (26)$$

D. Physics-informed wavelet-Fourier model

Fourier series are effective for representing the global modes that organize smooth advection, pressure variation, and wake-scale coherence, whereas wavelet series are naturally suited to localized high-frequency details such as sharp gradients, discontinuities, and compact vortical structures. Motivated by these complementary physical roles, the proposed physics-informed wavelet-Fourier model combines a Fourier-basis layer for global flow content with a wavelet layer for multiscale local flow content. These two layers are used in parallel with a conventional fully connected residual layer to form a three-branch representation.

As shown in Fig. 1, to adaptively fuse the heterogeneous features from the three parallel branches, we employ a channel attention mechanism.⁵¹ Treating the outputs of the wavelet, Fourier bases, and MLP layers as three distinct channels, we compute attention weights using a lightweight two-layer MLP and employ a channel attention mechanism to fuse the information from these three parallel

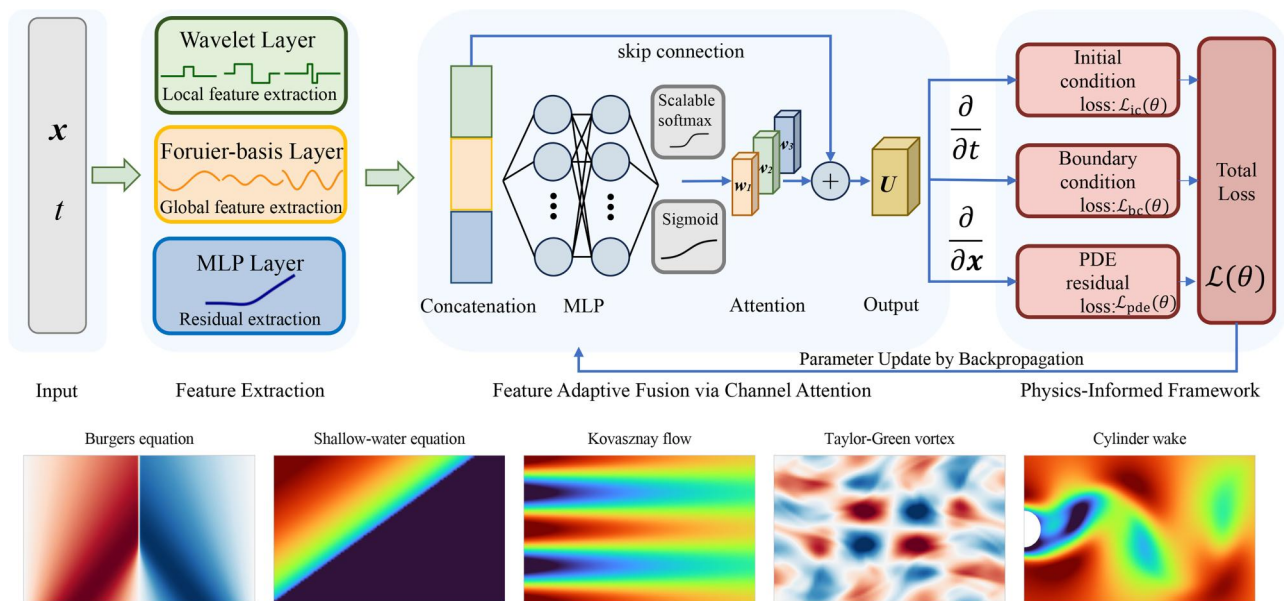


FIG. 1. The physics-informed wavelet-Fourier (PIWF) representation for multiscale fluid dynamics.

branches. Given the concatenated features $\mathbf{Y}^l = [\mathbf{y}_w^l, \mathbf{y}_f^l, \mathbf{y}_m^l] \in \mathbb{R}^{3 \times d}$, the attention score matrix $\mathbf{S}$ is obtained as

$$\mathbf{S} = \delta(\mathbf{W}_{\text{att},2} \sigma(\mathbf{W}_{\text{att},1} \mathbf{Y})), \quad (27)$$

where σ denotes the GELU activation function,⁵⁶ $\mathbf{W}_{\text{att},1}$ and $\mathbf{W}_{\text{att},2}$ are learnable weight matrices. δ is a composite activation function, which will be presented in the following text.

Standard softmax normalization tends to produce overly sharp distributions and can suffer from numerical instability for large channel dimensions. To mitigate these issues, we adopt a scalable softmax followed by a sigmoid function defined by

$$\delta(\cdot) = \sigma_s(\sigma_{\text{ssm}}(\cdot)), \quad (28)$$

where $\sigma_s(\mathbf{x}) = \frac{1}{1+e^{-\mathbf{x}}}$ is the sigmoid function. The scalable softmax function is defined as

$$\sigma_{\text{ssm}} : z_i \mapsto \frac{n^{sz_i}}{\sum_{j=1}^n n^{sz_j}} = \frac{e^{(s \log n) z_i}}{\sum_{j=1}^n e^{(s \log n) z_j}}, \quad (29)$$

where s is a scaling hyperparameter and $n = 3$ denotes the number of branches. This combination yields a smoother and more balanced attention distribution that prevents any single branch from dominating while still emphasizing the most informative features.

Inspired by residual learning, the attention mechanism is applied in a residual form, namely,

$$\mathbf{y}^l = \mathbf{S} + \mathbf{Y}^l = \delta(\mathbf{W}_{\text{att},2} \sigma(\mathbf{W}_{\text{att},1} \mathbf{Y}^l)) + \mathbf{Y}^l. \quad (30)$$

With the global modeling capability now provided by the Fourier-basis layer, the wavelet layer can focus exclusively on detail spaces. Accordingly, the truncation strategy for the wavelet decomposition levels is adjusted so that j starts from larger values, allowing the wavelet branch to concentrate on high-frequency localized features.

Finally, physical constraints are incorporated by embedding the governing partial differential equations, boundary conditions, and initial conditions directly into the loss function. The complete implementation strategy of the PIWF model is summarized in [Algorithm 1](#).

The PIWF framework is implemented as a modular physics-informed learning package in Python. PyTorch⁵⁷ is employed to construct the neural-network architecture and perform automatic differentiation, whereas DeepXDE⁵⁸ is used to manage the physics-informed training procedure, including the construction of residual losses and the enforcement of initial and boundary conditions. The main components of PIWF, including the wavelet-series feature layer, Fourier-basis feature layer, residual neural network, and channel-attention module, are encapsulated as standard `nn.Module` classes. This modular implementation facilitates code reuse, ablation studies, and extensions to different partial differential equations. For a given set of spatiotemporal coordinates, the wavelet branch extracts localized multiscale representations, and the Fourier branch provides global spectral features. These features are combined with the residual-network output and subsequently processed by a channel-attention module to obtain the final prediction of the physical variables. The governing-equation residuals are evaluated by differentiating the complete network output at the collocation points. The Fourier and GELU components provide the coordinate-dependent paths, while the exact Haar subvector enters the fused state as described in Eqs. (7)–(10). The complete loss function, consisting of equation

ALGORITHM 1. Optimization of the PIWF model.

-
- Require:** Initial data $\mathcal{D}_{\text{ic}}$, boundary data $\mathcal{D}_{\text{bc}}$, PDE data $\mathcal{D}_{\text{pde}}$, loss weights $(\lambda_1, \lambda_2, \lambda_3)$, maximum iterations N_{iter} .
- Ensure:** Trained PIWF parameters θ^* .
- 1: Initialize all trainable parameters θ .
 - 2: **for** $n = 1, \dots, N_{\text{iter}}$ **do**
 - 3: Set the network input as $\mathbf{y}^0 = (\mathbf{x}, t)$.
 - 4: **for** $l = 1, \dots, L - 1$ **do**
 - 5: Evaluate the three parallel branches at the l th layer:

Waveletbranch: $\mathbf{y}_w^l = \begin{bmatrix} \mathbf{W}_1^l \Phi_w^l(\mathbf{y}^{l-1}) \\ \sigma(\mathbf{B}^l + \mathbf{W}_2^l \mathbf{y}^{l-1}) \end{bmatrix}$,

where $\Phi_w^l(\mathbf{y}^{l-1}) = [\phi_{j,0}(\mathbf{y}^{l-1}), \psi_0(\mathbf{y}^{l-1}), \dots, \psi_{N-1}(\mathbf{y}^{l-1})]^T$,

Fourier-basisbranch: $\mathbf{y}_f^l = \begin{bmatrix} C_a^l \cos(\mathbf{W}_a^l \mathbf{y}^{l-1} + \mathbf{B}_a^l), \\ C_b^l \sin(\mathbf{W}_b^l \mathbf{y}^{l-1} + \mathbf{B}_b^l), \sigma(\mathbf{A}_c^l + \mathbf{O}_c^l \mathbf{y}^{l-1}) \end{bmatrix}^T$,

MLPbranch: $\mathbf{y}_m^l = \sigma(\mathbf{W}_m^l \mathbf{y}^{l-1} + \mathbf{b}_m^l)$.
 - 6: Fuse the three branches through residual channel attention:

$\mathbf{Y}^l = [\mathbf{y}_w^l, \mathbf{y}_f^l, \mathbf{y}_m^l] \in \mathbb{R}^{3 \times d}$, and $\mathbf{y}^l = \mathbf{Y}^l + \delta(\mathbf{W}_{\text{att},2} \sigma(\mathbf{W}_{\text{att},1} \mathbf{Y}^l))$.
 - 7: **end for**
 - 8: Obtain the PIWF prediction from the final-layer representation: $\mathbf{u}_\theta(\mathbf{x}, t) = \mathbf{W}_L \mathbf{y}^L + \mathbf{b}_L$.
 - 9: Compute the PDE residual by automatic differentiation: $\mathcal{L}_{\text{pde}} = \partial_t \mathbf{u}_\theta - R(\mathbf{u}_\theta; \mathbf{x}, t)$.
 - 10: Evaluate $\mathcal{L}_{\text{ic}}$, $\mathcal{L}_{\text{bc}}$, and $\mathcal{L}_{\text{pde}}$ on $\mathcal{D}_{\text{ic}}$, $\mathcal{D}_{\text{bc}}$, and $\mathcal{D}_{\text{pde}}$.
 - 11: Update θ by minimizing $\mathcal{L}(\theta) = \mathcal{L}_{\text{ic}}(\theta) + \mathcal{L}_{\text{bc}}(\theta) + \mathcal{L}_{\text{pde}}(\theta)$.
 - 12: **end for**
 - 13: **return** θ^* .
-

residuals together with initial and boundary condition penalties, is then minimized in an end-to-end training process. The implementation is fully compatible with GPU acceleration through the CUDA backend. All major tensor operations, basis evaluations, feature transformations, and loss evaluations are performed within the PyTorch computational graph, thereby reducing unnecessary host-device data transfer and ensuring stable gradient propagation. This software design provides an efficient and reproducible implementation of the proposed PIWF method for solving multiscale nonlinear partial differential equations.

III. RESULTS

This section evaluates PIWF on five fluid-dynamics problems chosen to isolate different physical mechanisms: viscous shock formation in Burgers' equation, wet-dry-front propagation in the shallow-water equations, steady viscous wake recovery in Kovasznay flow, decaying vortical motion in Taylor–Green vortex flow, and coherent vortex shedding in the wake of a circular cylinder. The proposed model is compared with a high-resolution finite-difference (FD) solver where appropriate, a standard physics-informed neural

network (PINN), and a physics-informed Kolmogorov–Arnold network (PIKAN; see Appendix B). All neural-network models are trained without high-fidelity interior reference data, using boundary and initial conditions when present together with the governing-equation residual. High-fidelity reference fields are used only for quantitative assessment. This design allows the comparison to focus on whether the learned solution preserves the relevant flow physics under sparse constraints. Unless otherwise stated, the Fourier node ratio is $p_{\text{ratio}} = 0.15$ and the wavelet node ratio is $q_{\text{ratio}} = 0.3$. The channel-attention module uses a two-layer MLP with 256 neurons per layer and GELU activation, and all models are trained with the Adam optimizer at a fixed learning rate of 0.001 on an RTX 4090 GPU.

A. Burgers' equation

The Burgers' equation is given by

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2}, \quad \text{where } x \in [-1, 1], \quad \text{and } t \in [0, 1]. \quad (31)$$

Here, $u(x, t)$ represents the fluid velocity, $\nu = 1/\text{Re}$ represents viscosity coefficient and Re is the Reynolds number. The equation is supplemented with the Dirichlet boundary conditions $u(-1, t) = u(1, t) = 0$ and the initial condition $u(x, 0) = -\sin(\pi x)$.

To examine different levels of nonlinear steepening, we consider two representative Reynolds numbers, $\text{Re} = 100\pi$ and 1000. In the lower-Reynolds-number case, viscous diffusion still regularizes the convectively generated gradient. In the higher-Reynolds-number case, the velocity field develops a much thinner shock-like layer whose position and amplitude are sensitive indicators of the balance between nonlinear convection and viscous dissipation. This case is particularly demanding for neural solvers because an accurate prediction must localize the steep transition without introducing spurious oscillations or excessive numerical smoothing.

The ground-truth solution for quantitative comparison is obtained via the Cole-Hopf transformation,⁵⁹ which converts the nonlinear Burgers' equation into the linear heat equation. Specifically, the transformation $u = -2\nu \partial(\ln \phi)/\partial x$ converts the Burgers' equation into the heat equation $\partial \phi / \partial t = \nu \partial^2 \phi / \partial x^2$. The analytical solution of the heat equation can be expressed as the ratio of two infinite integrals. In the numerical implementation, we employ the Sobol low-discrepancy sequence to generate uniformly distributed sampling points within the integration interval and use the quasi-Monte Carlo method⁶⁰ to compute the two integrals efficiently, thereby obtaining the exact solution at any spatiotemporal point (x, t) . This approach provides higher accuracy than conventional finite difference methods and is therefore suitable as a benchmark solution for validating numerical methods.

A high-resolution finite difference solver is employed as a classical numerical baseline. The solver adopts a second-order central difference scheme for spatial discretization and the forward Euler method for time marching. To ensure stability and satisfy the CFL condition, a significantly denser grid than that used for the neural network collocation points is employed. Specifically, for the low- and high-Reynolds-number cases, the spatial domain $x \in [-1, 1]$ is discretized into 1000 and 40 000 grid points, respectively, and the temporal interval $t \in [0, 1]$ is divided into 2000 and 3000 time steps, respectively. The same initial condition $u(x, 0) = -\sin(\pi x)$ and Dirichlet

boundary conditions $u(-1, t) = u(1, t) = 0$ are imposed. At each time step, the first and second spatial derivatives u_x and u_{xx} are computed using the second-order central difference scheme at interior points, and the solution is advanced explicitly via the forward Euler method, namely $u^{n+1} = u^n + \Delta t(\nu u_{xx} - u^n u_x)$.

For the low-Reynolds-number case ($\nu = 0.01/\pi$), the PINN, PIKAN, and PIWF models are trained using 50 collocation points randomly sampled via Latin hypercube sampling⁶¹ on the boundaries for the Dirichlet conditions and 50 points at $t = 0$ for the initial condition. The weight of the PDE residual term in Eq. (4) is set to $\lambda_3 = 10^{-4}$. For the high-Reynolds-number case ($\nu = 0.001$), larger PDE residuals arise in the sharp shock region, potentially overwhelming the contributions from the initial and boundary conditions. To maintain training stability, the residual weight is reduced by one order of magnitude to $\lambda_3 = 10^{-5}$, while the numbers of boundary and initial condition points are both increased to 100.

Within the computational domain, 1000 high-fidelity reference data points are randomly sampled for performance evaluation and are strictly excluded from the training process. To enforce the PDE residual $L_{\text{pde}}(\theta)$, 10 000 collocation points are uniformly distributed across the spatiotemporal domain. All neural networks (PINN, PIKAN, and PIWF) are designed to reconstruct the entire flow field by enforcing physical constraints through the residual loss, using only the given initial and boundary conditions. The baseline PINN employs a fully connected MLP with 6 hidden layers and 50 neurons per hidden layer. The PIKAN architecture replaces the MLP backbone with a KAN of the same depth and hidden dimension, i.e., 6 hidden layers with a hidden dimension of 50. In contrast to MLPs, where nonlinear activations are applied at neurons, KAN parameterizes learnable univariate functions on the edges. For PIWF, each of the three parallel branches adopts the same configuration as the baseline PINN, with 6 hidden layers and 50 neurons per layer, while no direct connections are introduced among different branches. The outputs of these branches are finally fused by a lightweight channel-attention module, implemented using a two-layer MLP with 256 neurons per hidden layer, which adaptively assigns channel-wise weights to balance the contributions of different feature representations. The hyperbolic tangent activation function is employed in the hidden layers. The networks map the spatiotemporal coordinates (x, t) to the velocity field u , resulting in input and output layers with two and one neurons, respectively. All models are trained for 200 000 epochs using the Adam optimizer to ensure adequate convergence of the loss function.

Figure 2 compares the predicted velocity profiles from the FD, PINN, PIKAN, and PIWF models at representative time instants. The comparison highlights the ability of each method to preserve the physically localized shock layer. In the lower-Reynolds-number case, all models reproduce the overall velocity evolution, but differences appear near the steepest part of the profile. In the higher-Reynolds-number case, these differences become more pronounced: PIWF follows the reference shock front most closely, whereas the standard PINN produces larger deviations both at the discontinuity and in its immediate neighborhood.

The DNS markers in Fig. 2 are the reference profiles for the numerical predictions. For $\nu = 0.01/\pi$, the FD, PINN, PIKAN, and PIWF predictions remain close to DNS over most of the domain, with their main differences concentrated near the steepening layer at $x = 0$. For $\nu = 0.001$, the DNS profiles contain a substantially thinner

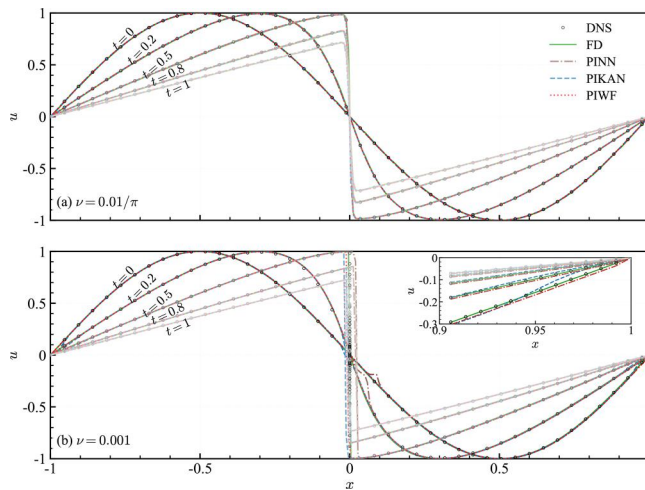


FIG. 2. Comparison of velocity profiles predicted by the FD, PINN, PIKAN, and PIWF models for the Burgers' equation at two different viscosities. (a) low-Reynolds-number case with $\nu = 0.01/\pi$; (b) high-Reynolds-number case with $\nu = 0.001$.

transition. The inset resolves the differences near the right boundary and shows that agreement away from the shock does not establish accurate shock localization. PIWF follows the DNS transition more closely than PINN and PIKAN at the plotted times, while the high-resolution finite-difference solution also remains close to the reference.

To quantitatively examine the predictive performance of each model, the temporal evolution of the relative error is illustrated in Fig. 3. The root mean square relative error is defined by

$$Er_{rms} = \frac{\sqrt{\frac{1}{N} \sum_{i=1}^N (u_{true,i} - u_{pred,i})^2}}{\sqrt{\frac{1}{N} \sum_{i=1}^N (u_{true,i})^2}}. \quad (32)$$

In both low- and high-Reynolds-number cases, the finite difference method exhibits non-negligible errors despite using a mesh that is more than two orders of magnitude denser than the collocation points employed by the neural-network models.

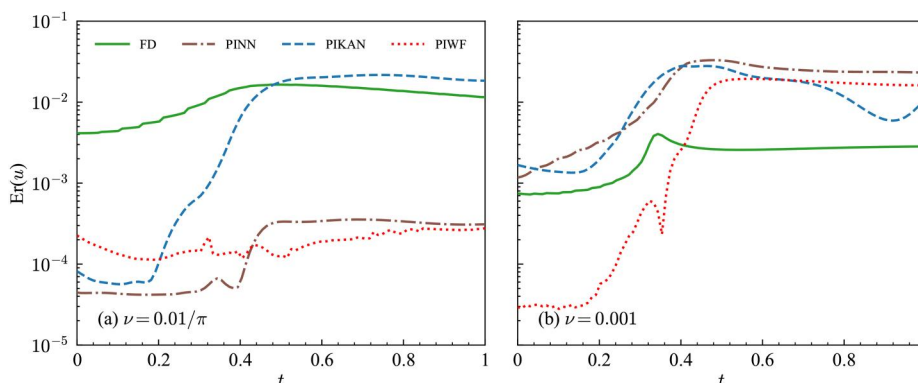


FIG. 3. Relative error evolution along time t for the Burgers' equation predicted by the FD, PINN, PIKAN, and PIWF models. (a) low-Reynolds-number case with $\nu = 0.01/\pi$; (b) high-Reynolds-number case with $\nu = 0.001$.

In the low-Reynolds-number case, both PINN and PIKAN models exhibit a pronounced localized increase in error near the discontinuity. The PIKAN model demonstrates the poorest performance, with its error in the high-gradient region even exceeding that of the finite difference method. In contrast, the proposed PIWF effectively suppresses error amplification in the steep-gradient zone and achieves the lowest error magnitude near the discontinuity. However, its accuracy in the smoother, low-gradient regions is slightly lower than that of PINN and PIKAN.

This behavior reflects the role of the wavelet basis in concentrating representational capacity around the physical region where the velocity gradient is largest. Under the tested architecture and training schedule, PIWF preferentially resolves the shock layer, where small phase or amplitude errors have a disproportionate influence on the solution. Because the optimizer minimizes a global loss, improved shock localization can be accompanied by a slight redistribution of error in smoother low-gradient regions.

A related tendency is observed for PIKAN in the high-Reynolds-number regime. As the velocity gradient steepens, the model devotes more capacity to the discontinuity, but the resulting representation is less accurate at maintaining the surrounding smooth field. By contrast, PIWF balances global Fourier content and localized wavelet corrections, thereby preserving the shock while keeping the low-gradient regions relatively stable.

Figure 4 illustrates the relative error contours for the Burgers' equation predicted by the FD, PINN, PIKAN, and PIWF models. In the low-Reynolds-number case, the standard PINN achieves overall satisfactory performance. Lacking embedded basis functions, it behaves similarly to a global approximator: the network layers learn the solution through spatially averaged representation governed by the universal approximation theorem. Although PIKAN and PIWF share the idea of embedding basis-function representations into hidden-layer nodal operations, they differ in the choice of bases: PIKAN relies on B-splines, whereas PIWF combines wavelet and Fourier series to enhance multiscale representation. While this design allows the neural networks to allocate more representational capacity to challenging shock regions, the global nature of the optimization often forces a rebalancing of previously converged smooth regions, introducing additional errors in low-gradient areas.

Although both PIWF and PIKAN are designed to enhance the representation of localized features, PIWF achieves significantly better shock resolution than the standard PINN, whereas PIKAN still

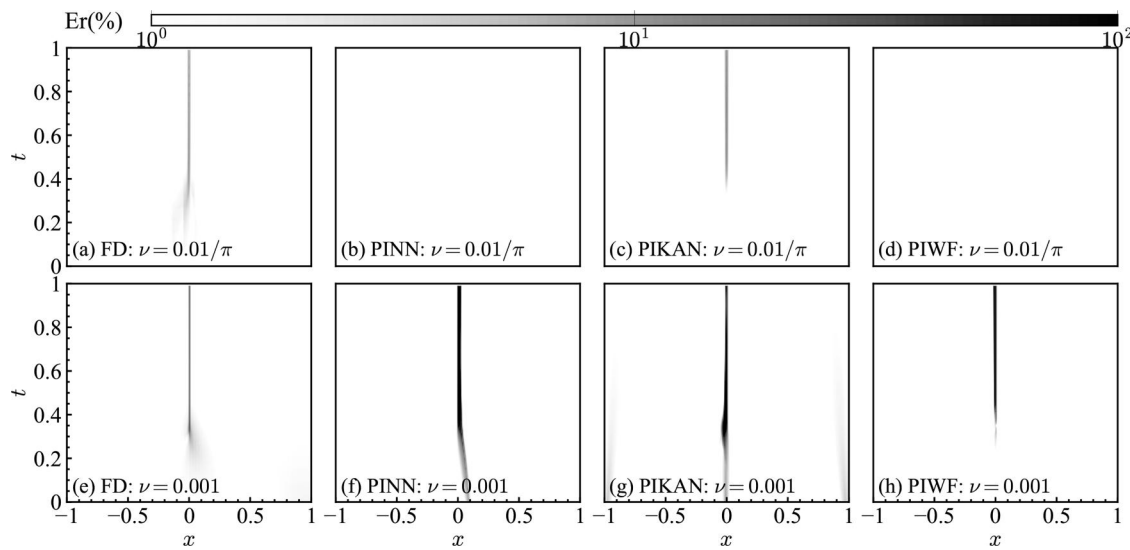


FIG. 4. Relative error contours for the Burgers' equation predicted by the FD, PINN, PIKAN, and PIWF models at two different viscosities. The top row corresponds to the low-Reynolds-number case with $\nu = 0.01/\pi$, and the bottom row corresponds to the high-Reynolds-number case with $\nu = 0.001$.

exhibits substantially larger errors near the discontinuity. This performance gap arises primarily from the inherent limitations of B-splines in representing discontinuous solutions. B-spline functions are optimized for smooth and continuous manifolds. When faced with sharp gradients or abrupt transitions, they tend to enforce global regularity at the expense of local fidelity, resulting in artificial smoothing of the shock and loss of high-frequency detail.

Even with sufficient computational resources, the PIKAN model still faces difficulty in concentrating its learning capacity on localized complex regions when regions of substantially different complexity have to be resolved simultaneously. The large PDE residuals generated by steep velocity gradients tend to dominate the composite loss function, causing the optimizer to overemphasize the shock region. This often leads to marginalization of smoother zones and, in some cases, compromises the satisfaction of initial and boundary conditions.

This issue is particularly acute for architectures that embed basis functions directly into the network nodes, as their primary goal is to enhance localized feature extraction. In such models, balancing contributions across regions is difficult since all nodal weights are coupled under the universal approximation theorem. In contrast, the proposed PIWF effectively modulates the relative importance of global and local features through its channel attention mechanism, thereby avoiding the pronounced degradation in boundary and initial condition accuracy observed in other basis-embedded frameworks.

B. Shallow water equations

The shallow water equations (SWEs) are a system of nonlinear, time-dependent, hyperbolic partial differential equations that describe the water depth and horizontal momentum of free-surface flows. They can be expressed as

$$\frac{\partial h}{\partial t} + \frac{\partial(hu)}{\partial x} = 0, \quad (33)$$

$$\frac{\partial(hu)}{\partial t} + \frac{\partial(hu^2 + gh^2/2)}{\partial x} = gh(s_{ox} - s_{fx}). \quad (34)$$

Here, $h(x, t)$ is the water depth, $u(x, t)$ is the depth-averaged velocity, and g is the gravitational acceleration. The term $s_{ox} = -\partial z/\partial x$ denotes the bed slopes, with $z(x)$ being the time-invariant topographic elevation. The bed friction slope s_{fx} is modeled using the Manning–Strickler formula,⁶² namely,

$$s_{fx} = n^2 u \sqrt{u^2 + v^2} h^{-4/3}, \quad (35)$$

where n is the Manning coefficient. In this study, the solution is obtained by solving the SWEs in a fully physics-informed manner without high-fidelity reference data.

This test case considers wave propagation over a horizontal bed driven by a time-dependent boundary condition and can be solved analytically⁶³

$$h(x, t) = \left[\frac{7}{3} (n^2 u^2 (x - ut)) \right]^{3/7}, \quad (36)$$

where the constant velocity is $u = 0.29$ m/s and the Manning coefficient is $n = 0.03$ s m^{-1/3}. The inflow boundary condition at $x = 0$ is given by Eq. (36), while the domain is initialized by $h(x, 0) = 0$.

The shallow water equations are particularly challenging for standard neural networks because the wet–dry front is a moving free-boundary-like structure where the water depth changes abruptly and the discharge must remain physically consistent with the nearly dry state. In this study, this problem is used to evaluate whether PIWF can preserve both the front location and the coupled depth–momentum response without relying on high-fidelity interior reference data.

The computational domain for this numerical experiment is defined as $x \in [0, 1200]$ m and $t \in [0, 3600]$ s. The reference solution is given by the analytical expression in Eq. (36). For training, the initial condition at $t = 0$ is enforced using 121 uniformly spaced points

with $\Delta x = 10$ m. The time-dependent inflow boundary condition at $x = 0$ is imposed with a temporal resolution of $\Delta t = 180$ s. Model performance is evaluated on a test set consisting of a 121×61 grid. The PDE residual is enforced at 2541 collocation points arranged on a 121×21 grid.

All neural architectures (PINN, PIKAN, and PIWF) consist of 6 hidden layers with 50 neurons per layer. The input vector X comprises the spatiotemporal coordinates (x, t) , the topographic slope, and the Manning friction coefficient, resulting in four input variables. The output vector U contains the water depth h and discharge hu , yielding two output variables. All models are trained for 50 000 epochs using the Adam optimizer.

For comparison, a high-resolution finite-difference solver is employed. The convective term is discretized using the first-order upwind scheme. The spatial domain is discretized with $\Delta x = 1$ m (1201 grid points), and the temporal domain uses $\Delta t = 1$ s (3601 time steps) to strictly satisfy the Courant–Friedrichs–Lewy (CFL) condition.

Figure 5 illustrates the predicted water depth h and discharge hu profiles along the x -direction at $t = 900$, 1800, and 2700 s. The dominant physical feature is the advancing wet–dry front. PIWF provides the most robust reconstruction in the immediate vicinity of this front, maintaining consistency for both water depth and discharge even when the transition becomes extremely steep. The finite-difference solution is affected by the numerical dissipation of the first-order upwind scheme, which introduces artificial viscosity and smooths the ideally sharp front. PIKAN shows a modest improvement over the standard PINN near the front at $t = 900$ s, indicating improved local representation, but its accuracy deteriorates more noticeably away from the front.

Table I summarizes the global relative L^2 errors for both h and hu . Although PIKAN exhibits higher local accuracy near the wet-dry front, its overall error for water depth remains slightly higher than that of PINN. This indicates that PIKAN exhibits larger errors in the smoother and computationally simpler regions, particularly in the stabilized wet and dry zones. Among all tested frameworks, the PIWF model consistently delivers the lowest errors across both fields and all evaluation metrics.

Figure 6 presents the relative L^2 error profiles along the x -direction for the water depth h and discharge hu . The relative error exhibits noticeable oscillatory behavior for all models. This behavior can be primarily attributed to the presence of the wet-dry front, where the solution magnitude becomes very small, causing even minor prediction deviations to be significantly amplified in the relative-error measure. The finite-difference solution generally exhibits lower relative errors than PINN and PIKAN, particularly in the dry region. This indicates that the FD scheme maintains values closer to the near-zero reference state in these regions, although this behavior is partly associated with its dissipative smoothing of the wet-dry transition.

Among all tested methods, PIWF achieves overall the lowest relative errors for both h and hu . This improvement is also reflected in the relative error contours shown in Fig. 7, where the error concentration around the wet-dry interface is weaker and more localized. Physically, this indicates that the model captures not only the front position but also the coupled momentum response across the transition. The compact support and orthogonality of the Haar wavelet basis allow the correction associated with the front to remain localized, while the Fourier branch preserves the smoother background propagation. This combination is particularly useful for discontinuity-dominated shallow-water dynamics.

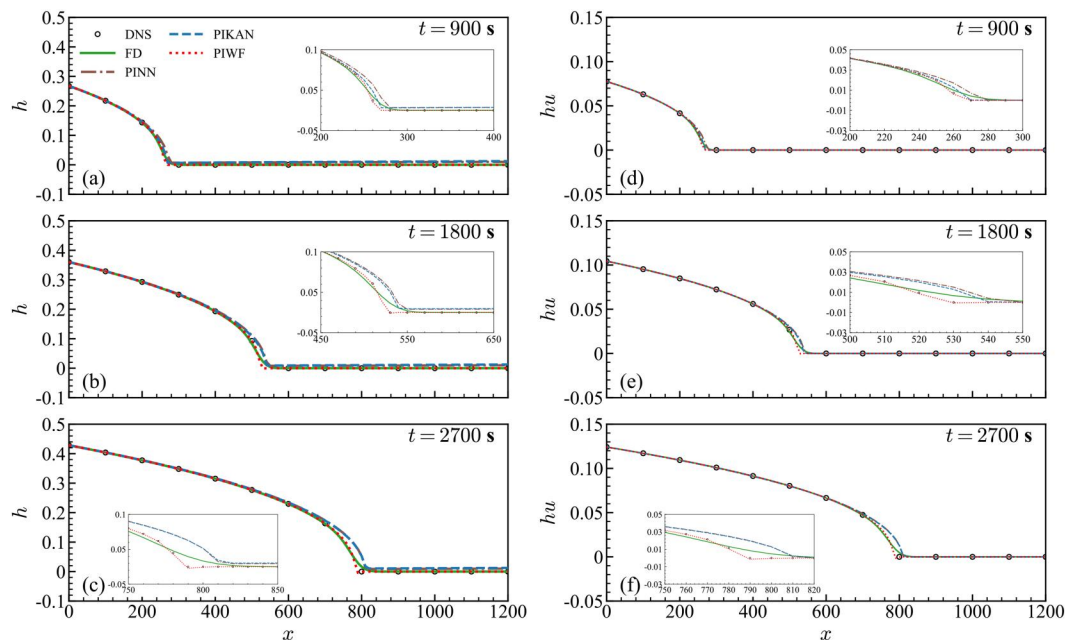
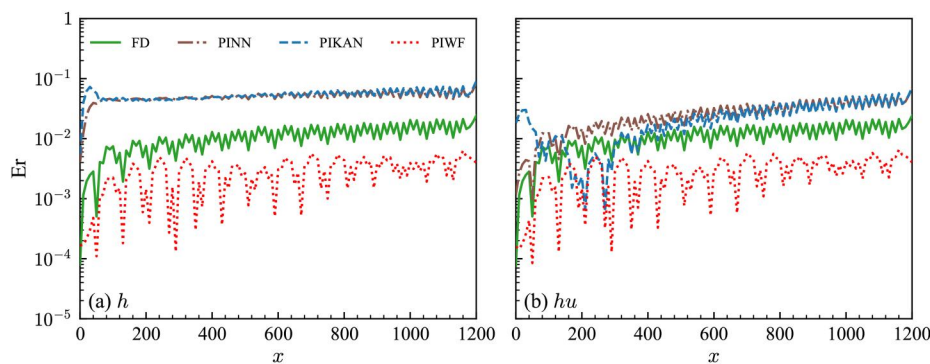


FIG. 5. Comparisons of water depth h and discharge hu profiles predicted by the FD, PINN, PIKAN, and PIWF models for the shallow water equations at $t = 900$, 1800, and 2700 s. Panels (a)–(c) and (d)–(f) show the spatial distributions of h and hu , respectively. Magnified views near the wet-dry fronts are provided in the insets to highlight resolution quality.

TABLE I. Relative L^2 errors for the shallow water equations at different spatial locations.

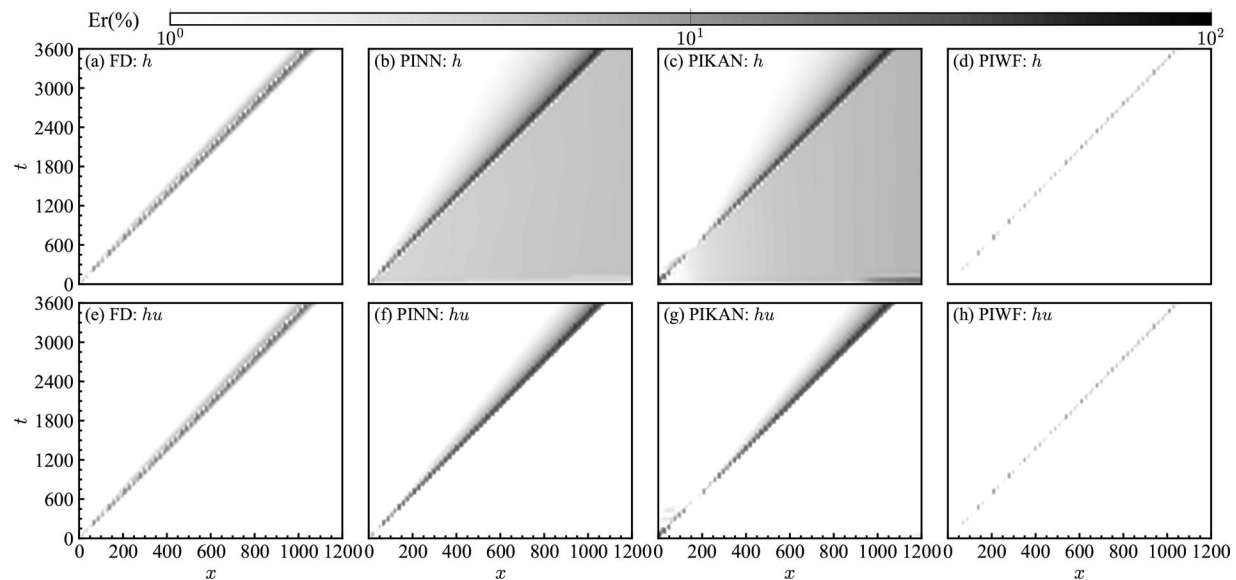
Model/position	$x = 0$	200	400	600	800	1000	1200	Average
h (FD)	5.066×10^{-6}	3.039×10^{-3}	6.538×10^{-3}	7.282×10^{-3}	9.660×10^{-3}	1.300×10^{-2}	1.265×10^{-2}	3.866×10^{-3}
h (PINN)	4.981×10^{-5}	4.068×10^{-2}	4.517×10^{-2}	4.829×10^{-2}	5.347×10^{-2}	5.989×10^{-2}	6.344×10^{-2}	3.685×10^{-2}
h (PIKAN)	2.768×10^{-4}	3.412×10^{-2}	4.191×10^{-2}	4.781×10^{-2}	5.510×10^{-2}	6.334×10^{-2}	6.888×10^{-2}	3.935×10^{-2}
h (PIWF)	8.499×10^{-5}	5.009×10^{-4}	1.494×10^{-3}	8.833×10^{-4}	1.603×10^{-3}	1.947×10^{-3}	2.357×10^{-3}	8.779×10^{-4}
hu (FD)	8.304×10^{-8}	3.039×10^{-3}	6.538×10^{-3}	7.282×10^{-3}	9.660×10^{-3}	1.300×10^{-2}	1.265×10^{-2}	3.866×10^{-3}
hu (PINN)	1.953×10^{-6}	5.378×10^{-3}	1.074×10^{-2}	1.512×10^{-2}	2.185×10^{-2}	3.021×10^{-2}	3.599×10^{-2}	9.899×10^{-3}
hu (PIKAN)	1.024×10^{-4}	1.184×10^{-3}	6.158×10^{-3}	1.147×10^{-2}	1.974×10^{-2}	2.968×10^{-2}	3.733×10^{-2}	8.405×10^{-3}
hu (PIWF)	8.652×10^{-5}	4.557×10^{-4}	1.415×10^{-3}	6.686×10^{-4}	1.373×10^{-3}	1.721×10^{-3}	2.155×10^{-3}	8.742×10^{-4}

**FIG. 6.** Relative L^2 error distributions along the x -direction for water depth h and discharge hu predicted by the FD, PINN, PIKAN, and PIWF models for the shallow water equations. (a) Water depth h ; (b) discharge hu .

C. Kovasznyai flow

We next consider Kovasznyai flow, an exact steady solution of the two-dimensional incompressible Navier–Stokes equations. This problem complements the preceding unsteady and discontinuity-dominated

examples because it contains a smooth but slowly decaying viscous wake with coupled velocity and pressure fields. It therefore provides a controlled benchmark for examining whether a physics-informed model can recover a two-dimensional wake structure

**FIG. 7.** Relative error contours of water depth h and discharge hu predicted by the FD, PINN, PIKAN, and PIWF models for the shallow water equations.

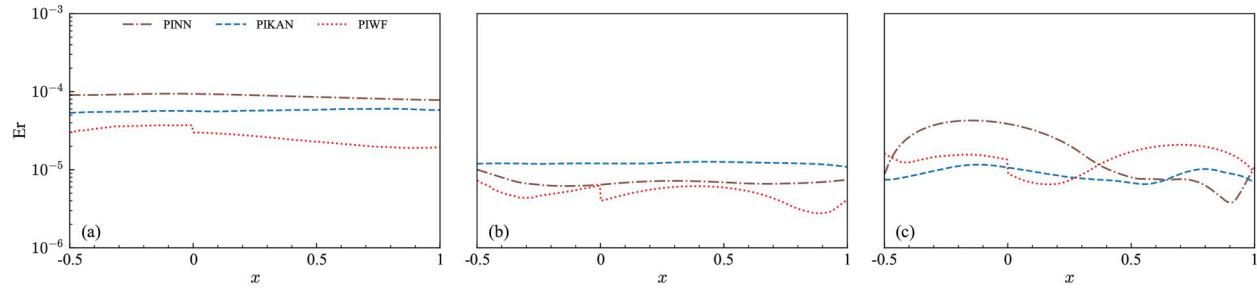


FIG. 8. Learning curves of the PINN, PIKAN, and PIWF models for Kovaszny flow at $Re = 1000$: (a) training loss; (b) testing loss.

without using high-fidelity interior reference data. The governing equations are

$$\begin{aligned} \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} &= 0, \\ u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} &= -\frac{\partial p}{\partial x} + \frac{1}{Re} \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right), \\ u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} &= -\frac{\partial p}{\partial y} + \frac{1}{Re} \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right). \end{aligned} \quad (37)$$

The analytical solution is written as

$$\begin{aligned} u(x, y) &= 1 - \exp(\lambda x) \cos(2\pi y), \\ v(x, y) &= \frac{\lambda}{2\pi} \exp(\lambda x) \sin(2\pi y), \\ p(x, y) &= \frac{1}{2} [1 - \exp(2\lambda x)], \\ \lambda &= \frac{Re}{2} - \sqrt{\frac{Re^2}{4} + 4\pi^2}. \end{aligned} \quad (38)$$

$$\lambda = \frac{Re}{2} - \sqrt{\frac{Re^2}{4} + 4\pi^2}. \quad (39)$$

The Reynolds number is set to $Re = 1000$, and the computational domain is $(x, y) \in [-0.5, 40] \times [-0.5, 1.5]$. The extended streamwise length is used because, at this Reynolds number, the exponential decay rate is small and the wake structure persists over a long distance. Dirichlet boundary values are prescribed from Eq. (38) on all four sides of the domain. The training set contains 2601 interior collocation points and 101 points on each boundary segment, while an independent set of 400 analytical points is used only for testing. For

this steady two-dimensional case, the input vector is (x, y) and the model output is (u, v, p) . All three models use four hidden layers. PINN and PIKAN use 128 neurons per hidden layer, whereas PIWF uses 50. The narrower PIWF configuration was selected for the extended $x = 40$ domain to limit model size and computational burden while preserving the accuracy of the scale-aware representation. All models are trained for 400 000 epochs.

The learning curves are shown in Fig. 8. The three models all reduce the physics-informed loss during training, but their testing behavior differs markedly. PIKAN reaches a nearly stagnant testing loss after the early stage, indicating that the spline-based representation has difficulty improving the long downstream wake once the leading-order field has been fitted. PINN continues to reduce the testing error, whereas PIWF obtains the lowest final testing loss. This behavior is consistent with the structure of the exact solution: the Fourier branch represents the smooth periodic dependence in the transverse direction, while the wavelet branch provides localized corrections where the wake amplitude and pressure gradient change most rapidly.

The relative error profiles in Fig. 9 and the mean relative errors in Table II further quantify the advantage of PIWF. For the streamwise velocity, PIWF reduces the mean relative error from 0.20% for PINN and 3.17% for PIKAN to 0.03%. The same trend is observed for the transverse velocity and pressure, although their absolute ranges are much smaller. This is important because the transverse velocity in Kovaszny flow is approximately two orders of magnitude smaller than the dominant streamwise component; a model that mainly fits the largest-amplitude variable can therefore achieve a visually reasonable u field while still missing the weaker coupled response. PIWF reduces this imbalance and gives a more consistent reconstruction of all three variables.

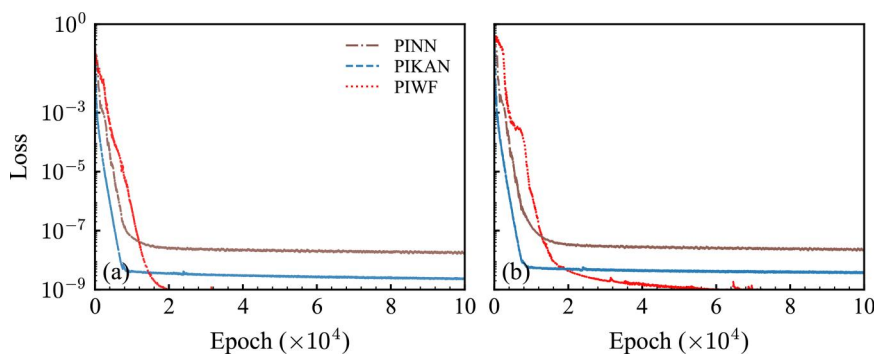


FIG. 9. Relative error profiles for Kovaszny flow at $Re = 1000$: (a) Streamwise velocity u , (b) transverse velocity v , and (c) pressure p . The profiles are averaged in the transverse direction.

TABLE II. Mean relative errors for Kovaszny flow at $Re = 1000$.

Model	u	v	p
PINN	0.20%	0.0055%	0.0018%
PIKAN	3.17%	0.073%	0.114%
PIWF	0.03%	0.0026%	0.0012%

The contour comparison of primary variables (u , v , and p) in Fig. 10 confirms that the error reduction is spatially coherent rather than confined to a small set of sample points. PINN produces larger errors in the near-wake region and along portions of the downstream decay, while PIKAN shows broader error bands across all three variables. PIWF gives the weakest and most localized relative-error contours, particularly in the streamwise velocity and pressure fields, where the exponential decay controls the downstream wake structure. These results show that the wavelet–Fourier representation is also effective for smooth viscous wakes, not only for sharp fronts or unsteady vortex shedding.

D. Taylor–Green vortex flow

The Taylor–Green vortex is then considered to examine the reconstruction of unsteady vortical motion in a periodic domain. In

contrast to the Kovaszny benchmark, the flow evolves in time and contains interacting vortical structures whose amplitudes decay under viscosity. This case is used to evaluate whether the model can recover a two-dimensional incompressible Navier–Stokes solution from initial and boundary data while using only residual points in the interior. The governing equations are

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0,$$

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = -\frac{\partial p}{\partial x} + \frac{1}{Re} \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right), \quad (40)$$

$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} = -\frac{\partial p}{\partial y} + \frac{1}{Re} \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right).$$

The reference solution is generated in vorticity form,

$$\frac{\partial \omega}{\partial t} + u \frac{\partial \omega}{\partial x} + v \frac{\partial \omega}{\partial y} = \frac{1}{Re} \left(\frac{\partial^2 \omega}{\partial x^2} + \frac{\partial^2 \omega}{\partial y^2} \right), \quad \omega = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}. \quad (41)$$

The Reynolds number is set to $Re = 1000$, and the periodic computational domain is $(x, y) \in [0, 2\pi) \times [0, 2\pi)$ over $t \in [0, 30]$. The initial vorticity is prescribed as

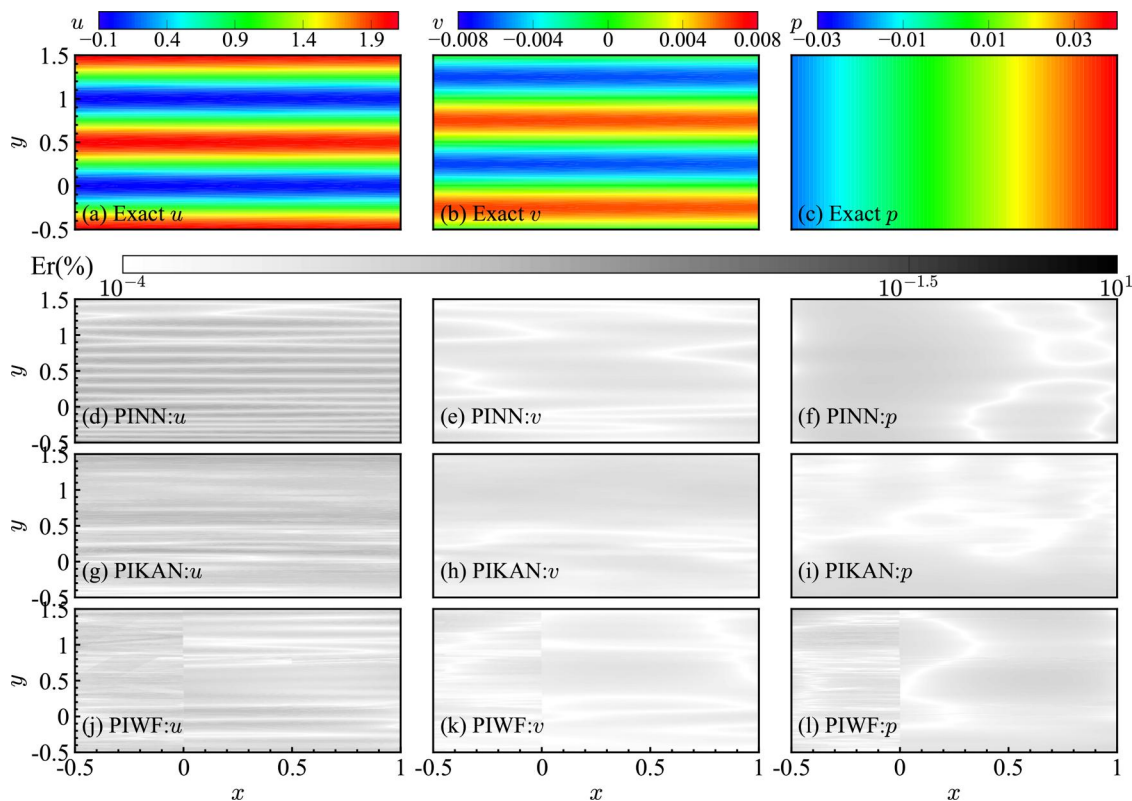


FIG. 10. Relative errors of velocities and pressure using PINN, PIKAN, and PIWF for Kovaszny flow at $Re = 1000$. From top to bottom: analytical solution and relative errors of PINN, PIKAN, and PIWF. From left to right: u , v , and p .

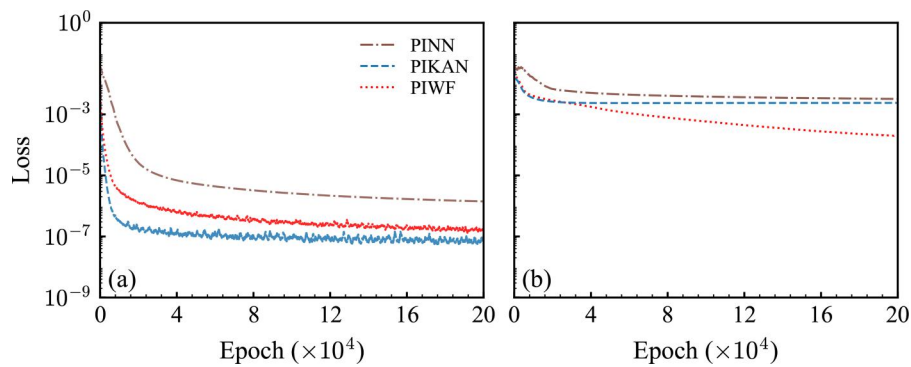


FIG. 11. Learning curves of the PINN, PIKAN, and PIWF models for the Taylor–Green vortex flow at $Re = 1000$: (a) training loss; (b) testing loss.

$$\begin{aligned} x_\theta &= x \cos \theta - y \sin \theta, \\ y_\theta &= x \sin \theta + y \cos \theta, \\ \omega(x, y, 0) &= 2 \cos(2x_\theta) \cos(2y_\theta) + \kappa, \end{aligned} \quad (42)$$

where θ is uniformly sampled from $(-10^\circ, 10^\circ)$ and κ is sampled from a zero-mean normal distribution with variance 10^{-3} . The reference field is first computed on a 256×256 grid and then downsampled by a factor of four to obtain the 64×64 fields used for evaluation. During training, high-fidelity reference data are supplied only on the initial plane and on the periodic boundaries; the remaining interior samples enter the loss only through the residual of Eq. (40). The PINN, PIKAN, and PIWF models use the same input vector (x, y, t) , the same output vector (u, v, p) , and the same depth and width as in the cylinder wake case, namely, four hidden layers with 128 neurons per layer. All models are trained for 200 000 epochs with the Adam optimizer at a learning rate of 0.001.

The learning curves in Fig. 11 show that all three models reduce the physics-informed objective, but their testing errors separate once the vortical field becomes more difficult to reconstruct from sparse constraints. PIWF maintains the lowest testing loss over the later training stage, indicating that the combined wavelet and Fourier-basis representation gives a more accurate reconstruction for this periodic vortical flow than the two baseline architectures considered here.

Figure 12 compares the relative L^2 errors of the three predicted fields over the full time interval. The error curves show that PIWF gives lower errors for both velocity components and pressure over most of the interval. This comparison is particularly

relevant for the pressure field because pressure is constrained indirectly through its gradients in the momentum equations, and pressure errors can affect the phase and amplitude of the predicted velocity field.

The vorticity contours shown in Figs. 13 and 14 provide a stricter spatial diagnostic than the primary velocity fields because vorticity depends on velocity gradients. At early times, all models recover the large-scale vortex arrangement, but the error fields already show stronger localization for PIWF. As the flow evolves, PINN and PIKAN develop broader error regions around the vortex cores and inter-vortex shear layers. PIWF gives smaller and more spatially confined errors over the selected time interval, which indicates improved preservation of both the periodic large-scale pattern and the local gradient information.

E. Two-dimensional cylinder wake flow

Flow past a circular cylinder is a classical benchmark problem in fluid mechanics because it contains boundary-layer separation, shear-layer roll-up, and periodic vortex shedding. It therefore provides a stringent test of whether a physics-informed model can preserve coherent wake dynamics and derivative-based flow diagnostics, rather than merely approximate the primary velocity and pressure fields. In this study, two-dimensional cylinder wake flow is used to assess the capability of neural-network-based models in capturing complex, time-dependent flow physics. The cylinder wake flow is governed by the incompressible Navier–Stokes equations, namely,

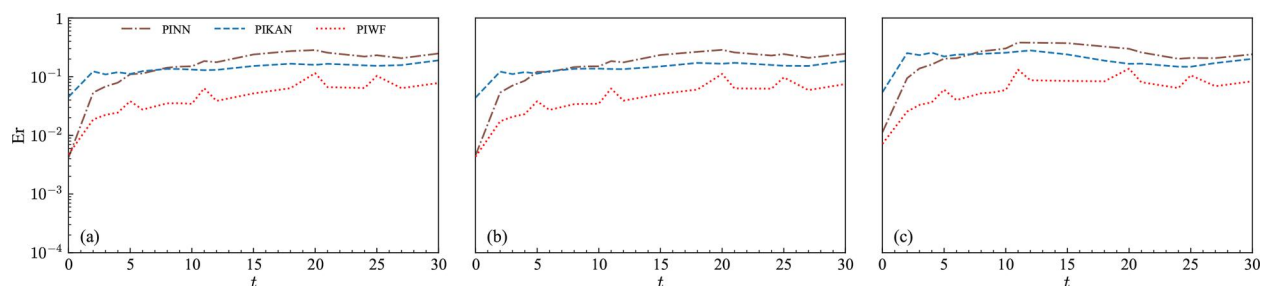


FIG. 12. Temporal evolution of the relative L^2 errors for the Taylor–Green vortex flow at $Re = 1000$: (a) streamwise velocity u , (b) transverse velocity v , and (c) pressure p .

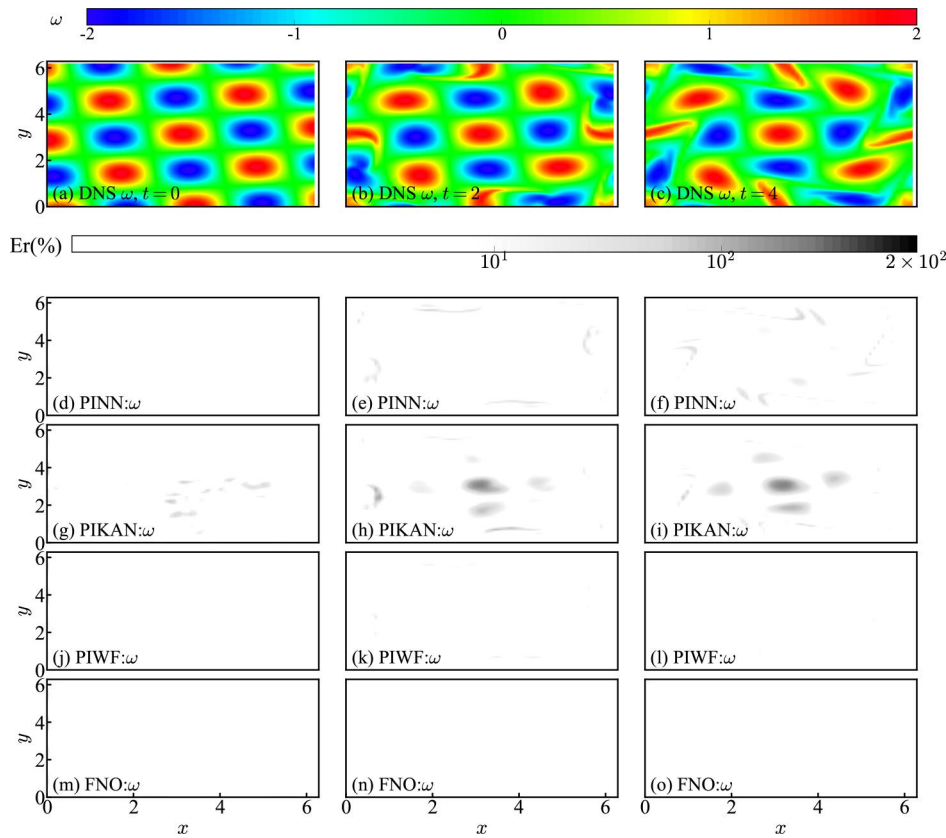


FIG. 13. Relative errors of spanwise vorticity using PINN, PIKAN, and PIWF for the Taylor–Green vortex flow at $t = 0, 2,$ and 4 . The reference spanwise vorticity is shown in the top row.

$$\begin{aligned} \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} &= 0, \\ \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} &= -\frac{\partial p}{\partial x} + \frac{1}{\text{Re}} \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right), \\ \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} &= -\frac{\partial p}{\partial y} + \frac{1}{\text{Re}} \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right). \end{aligned} \quad (43)$$

Here, u and v denote the velocity components in the x - and y - directions, respectively, and p is the modified pressure divided by the constant density. The Reynolds number is set to $\text{Re} = 100$. The reference solution is generated using the spectral/ hp element solver Nektar.¹⁶ The boundary conditions are as follows: a uniform free-stream velocity is imposed at the inlet ($x = -15$), a zero-pressure outflow condition is applied at the outlet ($x = 25$), and periodic boundary conditions are enforced along the transverse boundaries. The global computational domain is $(x, y) \in [-15, 25] \times [-8, 8]$. In the present study, we focus the analysis on the wake region, specifically the subdomain $[1, 8] \times [-2, 2]$ over the time interval $t \in [0, 7]$.

Training data are generated as follows: 100 points are sampled along the transverse boundaries and 50 points along the inlet and outlet boundaries using Latin hypercube sampling (LHS) strategy to enforce the Dirichlet conditions. Within the spatiotemporal domain, 140 000 collocation points are uniformly randomly sampled to evaluate the physics-informed residual in Eq. (5). An additional 8000 points are reserved as a test set for quantitative evaluation.

All neural-network models (PINN, PIKAN, and PIWF) consist of 4 hidden layers with 128 neurons per layer. The input variables comprise the spatiotemporal coordinates (x, y, t) , while the output vector consists of the velocity components (u, v) and pressure p . Thus, both the input and output layers contain three neurons. All models are trained for 100 000 epochs to ensure that the loss function converges to a stable minimum.

Figure 15 illustrates the learning curves of the PINN, PIKAN, and PIWF models for the two-dimensional cylinder wake flow at $\text{Re} = 100$. During the later training stages, all three models reach similar training loss values, indicating that the imposed boundary and residual constraints are comparably satisfied. The testing loss, however, reveals clear differences in the recovered wake physics. PIWF consistently achieves the lowest testing error, suggesting that the hybrid wavelet–Fourier representation generalizes better to interior wake structures that are not directly prescribed by the boundary data.

The relative error contours in Figs. 16 and 17 further show that PIWF yields lower and more spatially localized errors than the baseline models. The improvement is most apparent in regions associated with separated shear layers and wake vortices, where the gradients of velocity and pressure are large. Accurately resolving these regions is essential because errors in the transverse velocity v and pressure p directly affect the phase, strength, and induced pressure field of the vortex street. PIWF reduces the disparity among the errors of u , v , and p , whereas PINN and PIKAN exhibit stronger localized deviations in the v and p fields.

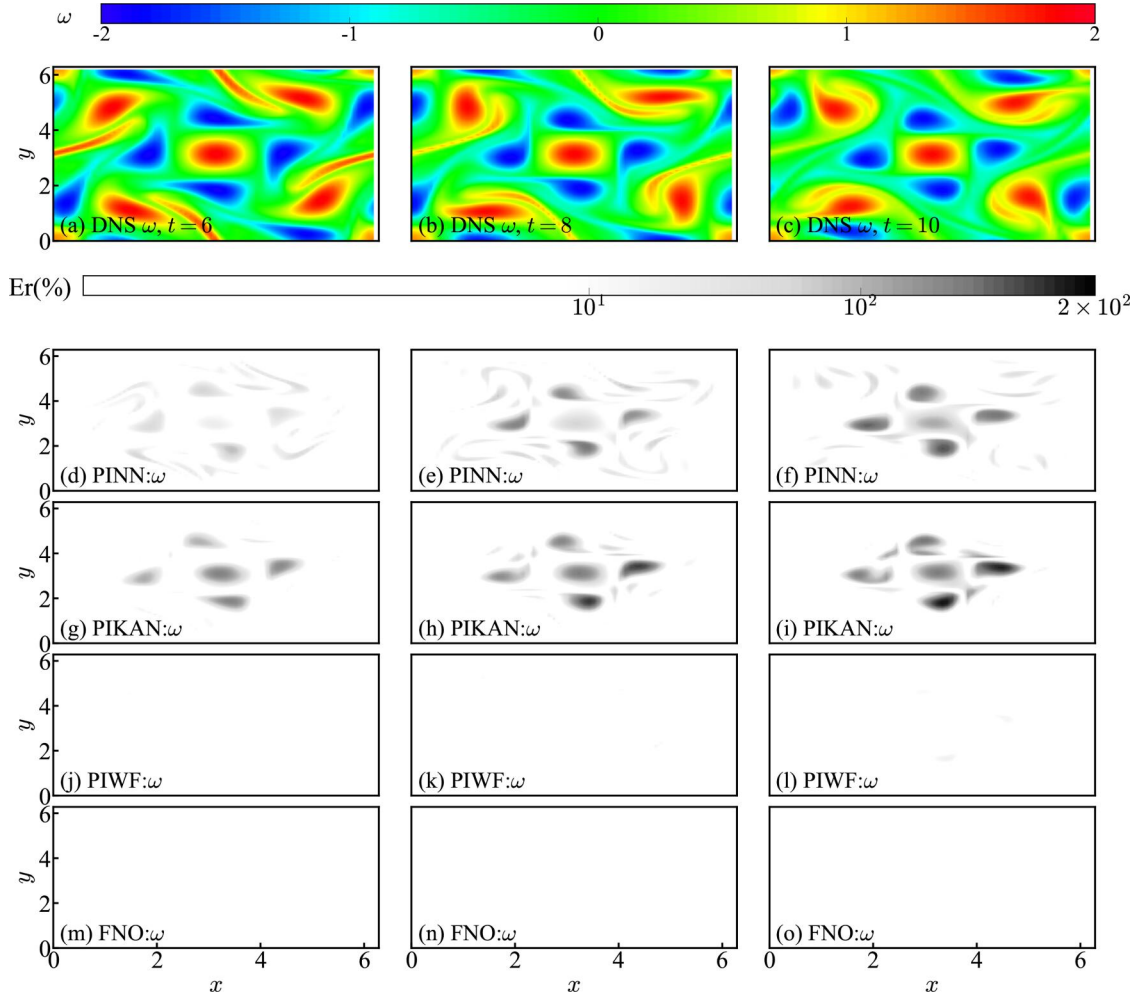


FIG. 14. Relative errors of spanwise vorticity using PINN, PIKAN, and PIWF for the Taylor–Green vortex flow at $t = 6, 8$, and 10 . The reference spanwise vorticity is shown in the top row.

The temporal evolution of the relative L^2 errors, displayed in Fig. 18, provides a clear quantitative comparison of the predictive performance across these three models. A consistent hierarchy is observed: PIKAN outperforms the standard PINN, while the proposed PIWF achieves the lowest error profiles throughout the temporal domain. These results are further demonstrated in Table III, which reports the relative L^2 errors for the velocity components and pressure at discrete instants as well as the time-averaged values. Notably, PIWF demonstrates a sustained and substantial advantage across all investigated temporal instances.

To further evaluate the models' capability in resolving fine-scale vortical structures, the vorticity ω_z and the Q -criterion are computed from the predicted velocity fields using second-order central difference scheme. These quantities are defined by

$$\omega_z = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \quad \text{and} \quad Q = \frac{1}{2} (||\mathbf{\Omega}||^2 - ||\mathbf{S}||^2), \quad (44)$$

where $\mathbf{\Omega}$ is the rotation-rate tensor, which is the antisymmetric part of the velocity gradient tensor. $||\mathbf{\Omega}||^2$ is defined by

$$||\mathbf{\Omega}||^2 = \frac{1}{2} \left(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \right)^2. \quad (45)$$

The tensor $\mathbf{S}$ is the strain-rate tensor, which is the symmetric part of the velocity gradient tensor. $||\mathbf{S}||^2$ is defined by

$$||\mathbf{S}||^2 = \left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial y} \right)^2 + \frac{1}{2} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right)^2. \quad (46)$$

The resulting contours and profiles of spanwise vorticity (ω_z) and Q are displayed in Figs. 19 and 20, respectively. Compared with the primary velocity and pressure fields, these derivative-based quantities provide a more stringent assessment of whether the predicted wake contains the correct rotational and strain-dominated structures. Both PINN and PIKAN deviate near local extrema in vorticity and Q , with

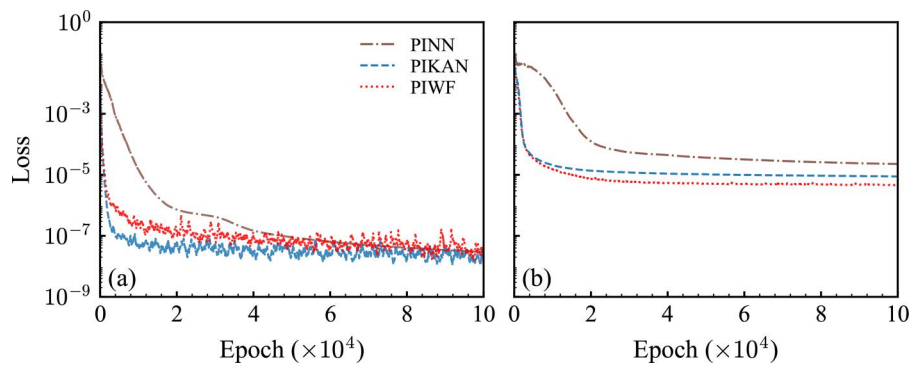


FIG. 15. Learning curves of the PINN, PIKAN, and PIWF models for the two-dimensional cylinder wake flow at $Re = 100$: (a) training loss; (b) testing loss.

the largest deviations observed for PINN. Table IV summarizes the absolute errors in spanwise vorticity and relative errors in Q at selected temporal instants, together with the time-averaged errors. The relative error of Q for PINN reaches order unity at some instants, indicating a

substantial distortion of the balance between rotation and strain. PIWF yields derivative-based diagnostics that are more consistent with the reference fields, demonstrating that the multiresolution wavelet component helps preserve fine-scale vortical features in the wake.

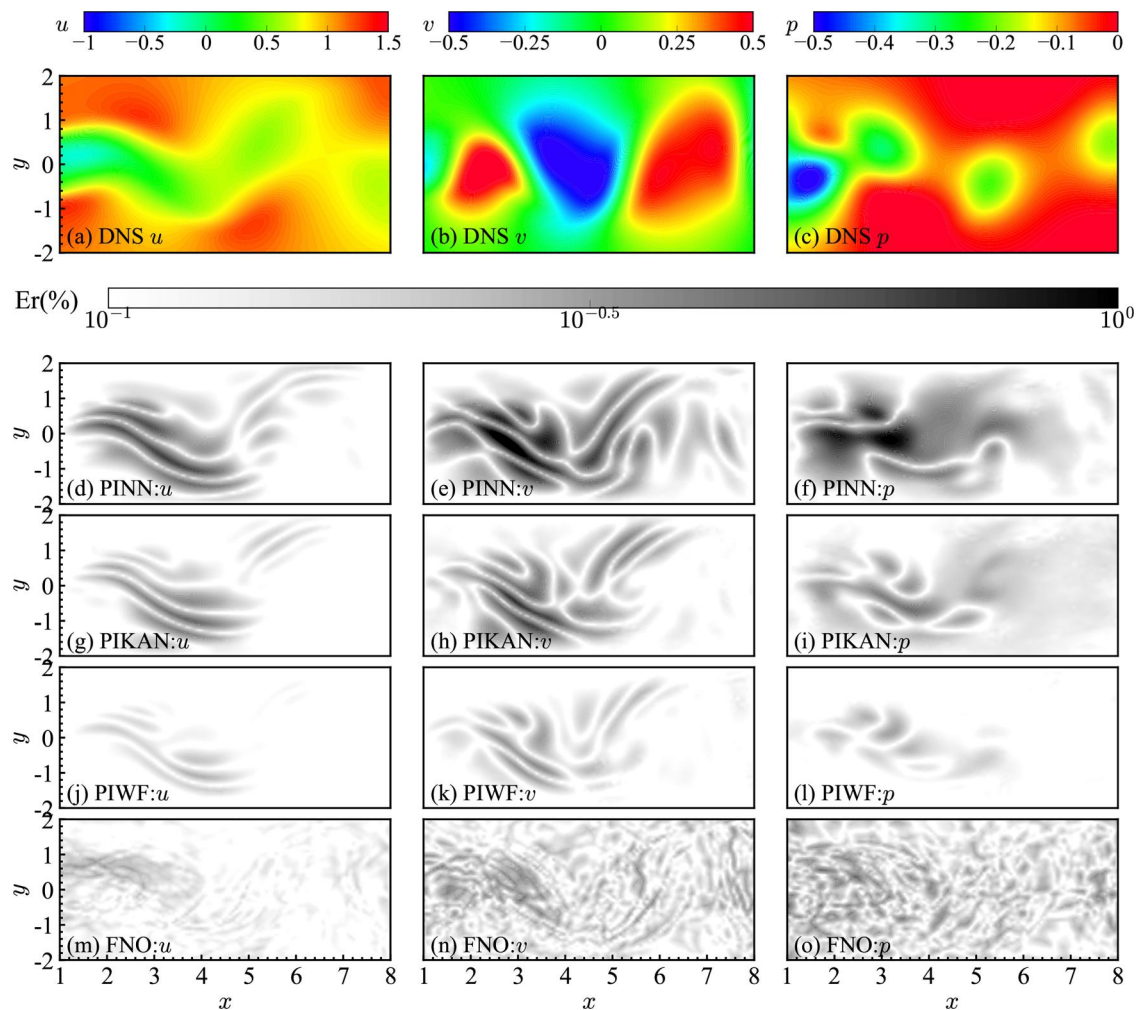


FIG. 16. Relative error contours of the velocity components (u, v) and pressure (p) for the cylinder wake flow at $t = 1$. From top to bottom: DNS, PINN, PIKAN, and PIWF results. From left to right: u, v , and p .

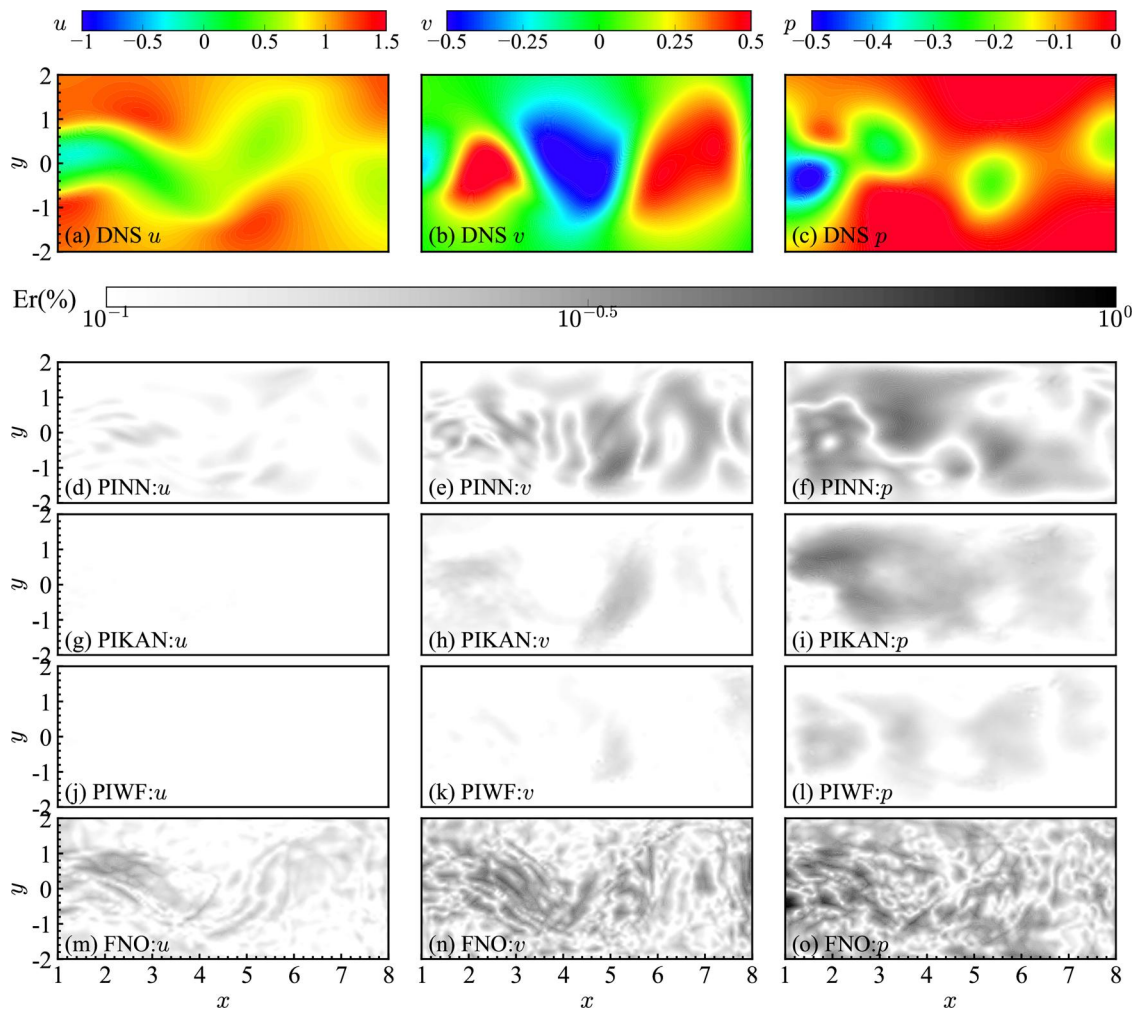


FIG. 17. Relative error contours of the velocity components (u, v) and pressure (p) for the cylinder wake flow at $t = 7$. From top to bottom: DNS, PINN, PIKAN, and PIWF results. From left to right: u, v , and p .

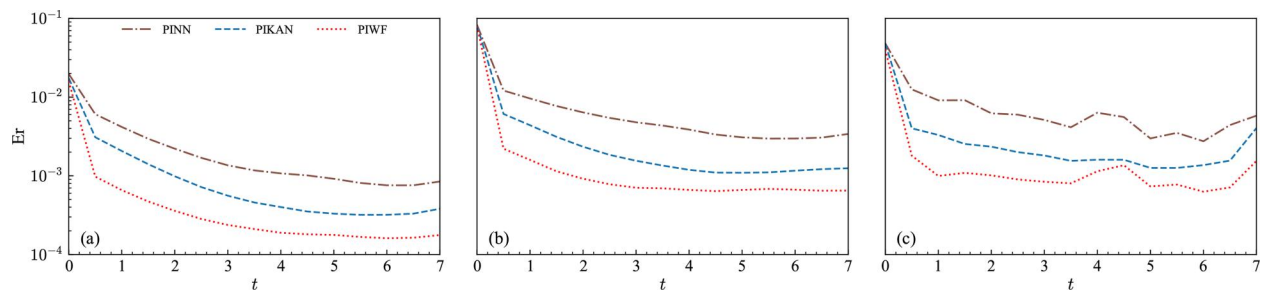


FIG. 18. Temporal evolution of the relative L^2 errors for the cylinder wake flow at $Re = 100$: (a) streamwise velocity u , (b) transverse velocity v , and (c) pressure p .

To further evaluate the models' ability to preserve the broadband energy distribution in the cylinder wake, we perform a two-dimensional fast Fourier transform (FFT) analysis in wavenumber space (k_x, k_y). The resulting spectra for the streamwise velocity u ,

transverse velocity v , and pressure p at representative time instants $t = 1$ and 7 are shown in Figs. 21 and 22. The top row presents the reference DNS spectra, while the lower rows display the relative FFT error (in percent, logarithmic scale) for the PINN, PIKAN, and

TABLE III. Relative L^2 errors for the two-dimensional cylinder wake flow at $Re = 100$.

Model/time	1	2	3	4	5	6	7	Average
u (PINN)	6.074×10^{-3}	4.178×10^{-3}	2.956×10^{-3}	2.212×10^{-3}	1.693×10^{-3}	1.360×10^{-3}	1.172×10^{-3}	2.806×10^{-3}
u (PIKAN)	3.095×10^{-3}	2.069×10^{-3}	1.406×10^{-3}	9.868×10^{-4}	7.194×10^{-4}	5.591×10^{-4}	4.583×10^{-4}	1.328×10^{-3}
u (PIWF)	9.769×10^{-4}	6.584×10^{-4}	4.716×10^{-4}	3.589×10^{-4}	2.830×10^{-4}	2.373×10^{-4}	2.106×10^{-4}	4.567×10^{-4}
v (PINN)	1.212×10^{-2}	9.613×10^{-3}	7.710×10^{-3}	6.396×10^{-3}	5.447×10^{-3}	4.790×10^{-3}	4.354×10^{-3}	7.204×10^{-3}
v (PIKAN)	6.117×10^{-3}	4.395×10^{-3}	3.127×10^{-3}	2.342×10^{-3}	1.846×10^{-3}	1.554×10^{-3}	1.350×10^{-3}	2.962×10^{-3}
v (PIWF)	2.212×10^{-3}	1.593×10^{-3}	1.141×10^{-3}	9.184×10^{-4}	7.809×10^{-4}	7.054×10^{-4}	6.929×10^{-4}	1.149×10^{-3}
p (PINN)	1.254×10^{-2}	9.112×10^{-3}	9.144×10^{-3}	6.225×10^{-3}	5.983×10^{-3}	5.134×10^{-3}	4.141×10^{-3}	7.469×10^{-3}
p (PIKAN)	4.014×10^{-3}	3.305×10^{-3}	2.541×10^{-3}	2.346×10^{-3}	2.000×10^{-3}	1.812×10^{-3}	1.550×10^{-3}	2.510×10^{-3}
p (PIWF)	1.802×10^{-3}	9.964×10^{-4}	1.090×10^{-3}	1.013×10^{-3}	8.990×10^{-4}	8.393×10^{-4}	8.027×10^{-4}	1.063×10^{-3}

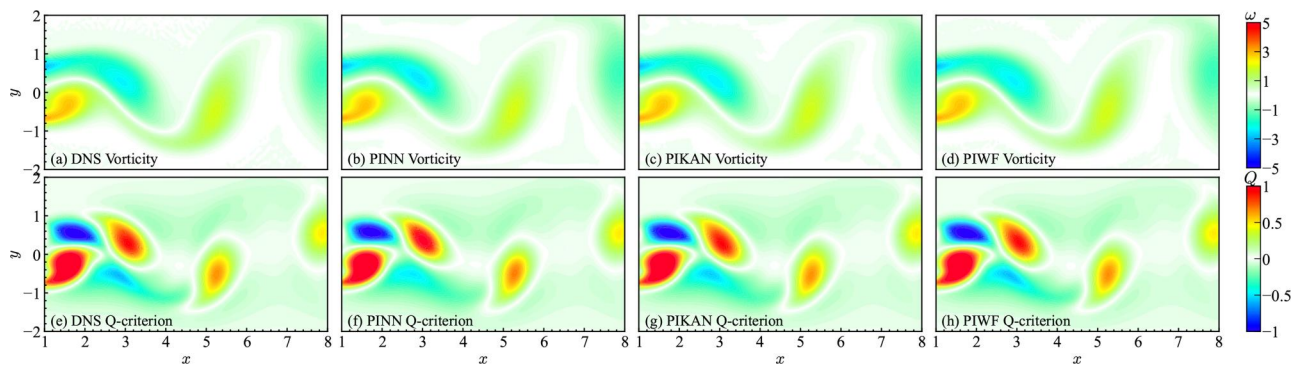
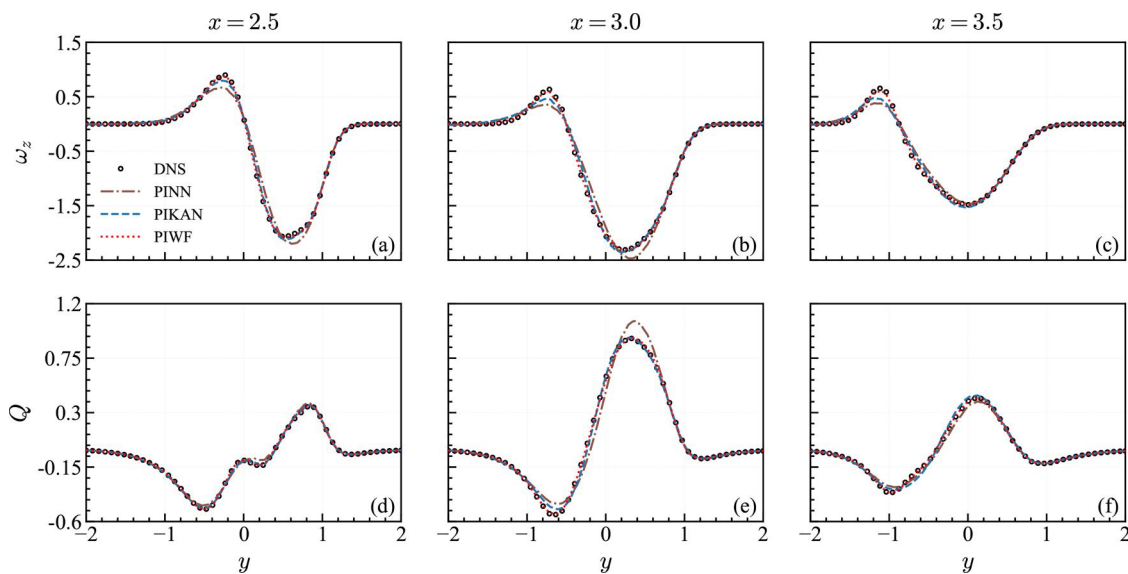
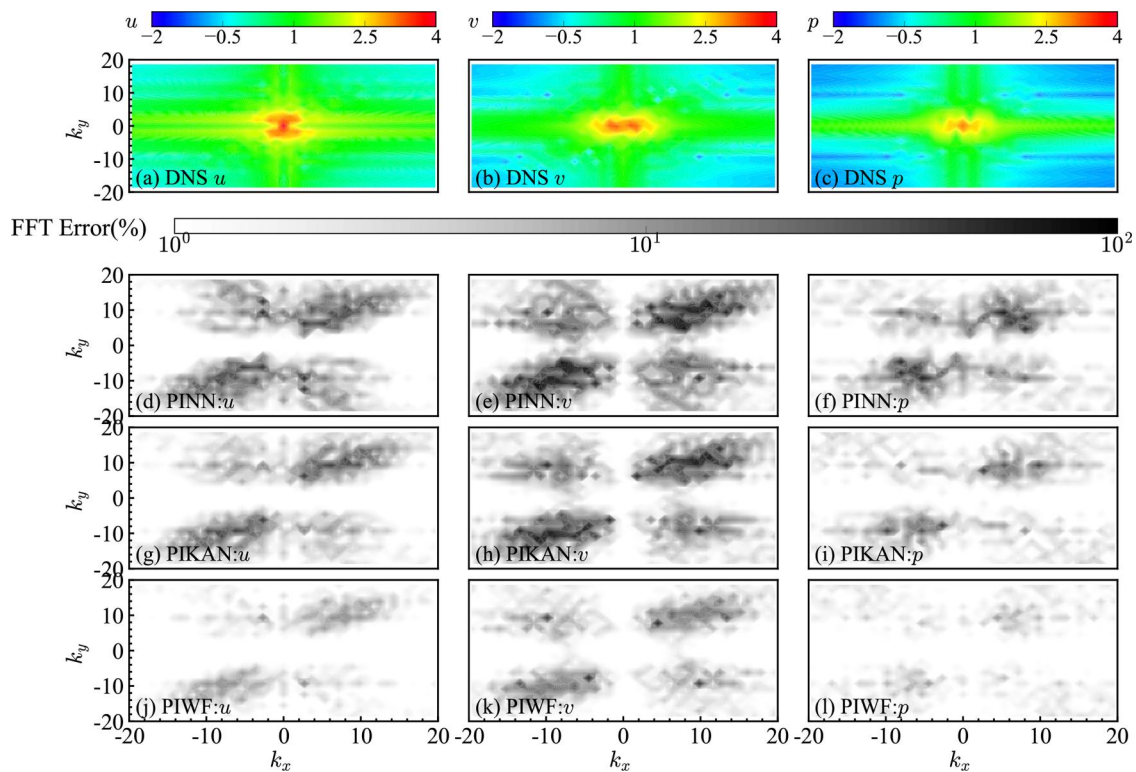
FIG. 19. Contours of vorticity (ω_z) and the Q-criterion for the cylinder wake flow predicted by the DNS, PINN, PIKAN, and PIWF models. Top row: vorticity; bottom row: Q. From left to right: DNS, PINN, PIKAN, and PIWF.FIG. 20. Profiles of spanwise vorticity (ω_z) and the Q-criterion along the lines $x = 2.5, 3.0$, and 3.5 at $t = 1$ predicted by the PINN, PIKAN, and PIWF models. Top row: spanwise vorticity; bottom row: Q.

TABLE IV. Relative errors for the Q-criterion and absolute errors for spanwise vorticity (ω_z) in the two-dimensional cylinder wake flow at $Re = 100$.

Model/time	1	2	3	4	5	6	7	Average
ω_z (PINN)	2.929×10^{-2}	1.654×10^{-2}	9.834×10^{-3}	6.539×10^{-3}	5.277×10^{-3}	4.855×10^{-3}	5.632×10^{-3}	1.114×10^{-2}
ω_z (PIKAN)	1.507×10^{-2}	7.293×10^{-3}	3.632×10^{-3}	2.074×10^{-3}	1.579×10^{-3}	1.381×10^{-3}	1.447×10^{-3}	4.639×10^{-3}
ω_z (PIWF)	5.237×10^{-3}	2.602×10^{-3}	1.510×10^{-3}	1.143×10^{-3}	1.053×10^{-3}	1.095×10^{-3}	1.113×10^{-3}	1.965×10^{-3}
Q (PINN)	1.506×10^{-1}	1.037×10^0	8.984×10^{-2}	8.134×10^{-2}	7.616×10^{-2}	7.524×10^{-2}	6.221×10^{-2}	2.247×10^{-1}
Q (PIKAN)	7.248×10^{-2}	1.081×10^{-1}	3.280×10^{-2}	3.417×10^{-2}	1.488×10^{-2}	1.761×10^{-2}	1.607×10^{-2}	4.230×10^{-2}
Q (PIWF)	3.066×10^{-2}	1.475×10^{-1}	2.356×10^{-2}	2.368×10^{-2}	1.049×10^{-2}	1.445×10^{-2}	1.123×10^{-2}	3.737×10^{-2}

**FIG. 21.** Relative error contours in wavenumber space for the cylinder wake flow at $t = 1$. From top to bottom: DNS, PINN, PIKAN, and PIWF results. From left to right: streamwise velocity u , transverse velocity v , and pressure p .

PIWF predictions. In wavenumber space, both PINN and PIKAN exhibit increased spectral errors at moderate-to-high wavenumbers. These errors indicate reduced fidelity in capturing the fine-scale turbulent structures and the correct energy cascade, with energy leakage and spurious high-frequency components.

In contrast, PIWF gives relatively low FFT errors across the wavenumber spectrum and remains close to the DNS reference even at the smallest resolved scales. This indicates that PIWF accurately captures the coherent structures in the cylinder wake while introducing less spectral distortion than the baseline models. The wavelet layer improves the localized high-wavenumber content without introducing strong Gibbs-like oscillations, whereas the Fourier branch preserves the low-wavenumber coherent modes.

Channel attention then adaptively weights these complementary representations, helping retain the broadband energy distribution of the wake.

Collectively, the spatial error distributions, derivative-based diagnostics, and spectral analyses show that PIWF improves pointwise reconstruction accuracy while better retaining the multiscale structures of the wake. This capability is particularly relevant for unsteady nonlinear wake dynamics, where accurate prediction requires simultaneous preservation of vortex position, vorticity magnitude, pressure response, and broadband spectral content. These results support the use of PIWF as a physics-informed representation for multiscale fluid problems. An ablation study and parameter sensitivity analysis are provided in [Appendix C](#).

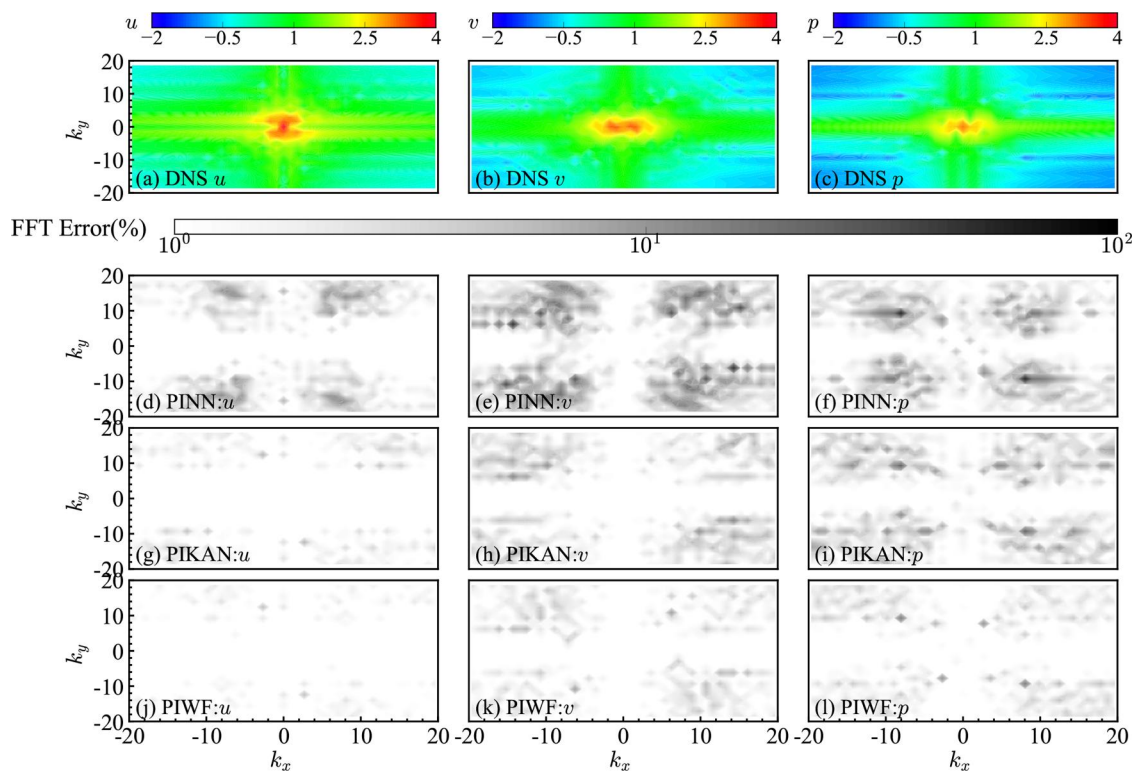


FIG. 22. Relative error contours in wavenumber space for the cylinder wake flow at $t = 7$. From top to bottom: DNS, PINN, PIKAN, and PIWF results. From left to right: streamwise velocity u , transverse velocity v , and pressure p .

IV. SUMMARY

In this work, we propose a physics-informed wavelet-Fourier (PIWF) representation for resolving multiscale structures in fluid-dynamics problems. The central element is a dedicated wavelet layer that reformulates each network mapping as a wavelet-series approximation, enabling multilevel extraction of both global structures and localized sharp features such as discontinuities, steep gradients, and singularities. The PIWF representation comprises three parallel branches operating at complementary scales: a Fourier-basis layer for capturing global low-frequency components, the proposed wavelet layer for targeted extraction of high-frequency localized content, and a MLP layer for residual compensation. These heterogeneous outputs are dynamically fused through a channel attention mechanism that adaptively balances global spectral consistency with local feature resolution. By embedding the governing physical laws into the loss function, PIWF can compute forward solutions without high-fidelity interior reference data.

The numerical results show that this scale-aware representation improves the recovery of several physically important flow features. For Burgers' equation, PIWF more accurately localizes the shock-like velocity layer, especially at higher Reynolds number. For the shallow-water equations, it better preserves the wet-dry-front position and the coupled response of water depth and discharge. For Kovasznay flow at $Re = 1000$, PIWF gives a more accurate reconstruction of the long viscous wake and reduces the relative errors of the coupled velocity and

pressure fields. For the Taylor-Green vortex at $Re = 1000$, PIWF better preserves the decaying vorticity field and reduces the physical-space errors. For the cylinder wake at $Re = 100$, PIWF improves the reconstruction of velocity, pressure, vorticity, and the Q -criterion, while also reducing spectral distortion across the wavenumber plane. These observations indicate that the advantage of PIWF is not limited to a lower global error metric; it lies in the improved preservation of the flow structures that control discontinuity propagation, free-surface-front dynamics, steady wake decay, vortex decay, and vortex shedding.

Overall, PIWF provides a scale-aware physics-informed representation for multiscale fluid problems involving sharp interfaces, vortex structures, and broadband energy distributions. Nevertheless, the current PIWF still has certain limitations. Its performance advantage is most pronounced in problems with relatively simple boundary conditions. For cases involving highly complex or irregular boundaries, the fixed global computational domain and uniform collocation strategy may lead to reduced efficiency and accuracy. Future work will focus on developing region-adaptive and block-wise training strategies that allow the wavelet layer to concentrate on challenging local regions while the Fourier-basis layer handles smoother areas, exploring boundary-aware collocation schemes for complex geometries, extending the formulation to three-dimensional transitional and turbulent flows, and incorporating adaptive loss weighting to further stabilize training. These developments would broaden the use of PIWF for physics-informed analysis of multiscale flow phenomena.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (Grant Nos. 52425111, 52306193, 12388101, and 12502261), the Natural Science Foundation of Hainan Province (No. 425QN376), the Natural Science Foundation of Shandong Province (No. ZR2024QA050), and the Basic Product Innovation Research Project (No. KY10100230067).

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Chao Wang: Funding acquisition (equal); Resources (equal); Supervision (equal); Writing – original draft (equal); Writing – review & editing (equal). **Shilong Li:** Conceptualization (equal); Formal analysis (equal); Methodology (equal); Validation (equal); Writing – original draft (equal). **Yunpeng Wang:** Conceptualization (equal); Supervision (equal); Writing – review & editing (equal). **Tianbai Xiao:** Conceptualization (equal); Supervision (equal); Writing – review & editing (equal). **Zelong Yuan:** Conceptualization (equal); Formal analysis (equal); Funding acquisition (equal); Methodology (equal); Writing – review & editing (equal). **Chenyue Xie:** Conceptualization (equal); Formal analysis (equal); Methodology (equal); Visualization (equal); Writing – review & editing (equal). **Chunyu Guo:** Funding acquisition (equal); Supervision (equal); Writing – review & editing (equal).

DATA AVAILABILITY

The data that support the findings of this study are openly available in GitHub at <https://github.com/OneMoreSpeed/PIWF>, Ref. 66.

APPENDIX A: WAVELET TRANSFORM AND WAVELET SERIES

The wavelet transform is introduced to overcome the fixed window limitation of the short-time Fourier transform (STFT) by employing variable window functions that enable decomposition of a signal into components at different frequencies and scales. These window functions are generated from a mother wavelet $[\psi(t)]$ through scaling and translation operations. The mother wavelet should satisfy the admissibility and unit-energy conditions

$$\int_{-\infty}^{\infty} \psi(t) dt = 0 \quad \text{and} \quad \int_{-\infty}^{\infty} |\psi(t)|^2 dt = 1. \quad (\text{A1})$$

Through scaling and shifting, a family of wavelet functions is then obtained by

$$\psi_{a,b}(t) = \frac{1}{\sqrt{|a|}} \psi\left(\frac{t-b}{a}\right), \quad (\text{A2})$$

where a represents the scaling factor and b denotes the translation factor. The continuous wavelet transform of a signal $f(t)$ is performed by computing the inner product between the signal and this family of wavelet functions, namely,

$$W_f(a, b) = \langle f, \psi_{a,b} \rangle = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} f(t) \psi^*\left(\frac{t-b}{a}\right) dt, \quad (\text{A3})$$

where the asterisk (*) denotes the complex conjugate. In practical computation, the continuous scaling and shifting factors (a, b) lead to substantial redundant calculations. To address this problem, discrete scaling and shifting factors are employed to generate a series of discrete functions. Typically, the scaling factor varies in powers of 2, namely,

$$a = 2^j, \quad j \in \mathbb{Z}, \quad (\text{A4})$$

where j denotes the scale series or number of layers. Larger values of j correspond to wider low-frequency wavelets, while smaller values yield narrower high-frequency wavelets. The translation factor b is set to $2^j \times k$ with integer k , ensuring that coarser scales use larger sampling steps and finer scales use smaller steps. This choice aligns with the principle of multi-resolution analysis and yields a non-redundant, complete computation. The resulting discrete wavelet functions are

$$\psi_{j,k}(t) = \frac{1}{\sqrt{2^j}} \psi\left(\frac{t-k \cdot 2^j}{2^j}\right) = 2^{-j/2} \psi(2^{-j}t - k). \quad (\text{A5})$$

The discrete wavelet series expansion of a signal $f(t)$ then takes the form

$$f(t) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} d_{j,k} \psi_{j,k}(t), \quad (\text{A6})$$

where the wavelet coefficients are given by

$$d_{j,k} = \langle f, \psi_{j,k} \rangle = \int_{-\infty}^{\infty} f(t) \psi_{j,k}(t) dt. \quad (\text{A7})$$

In practice, the lowest-frequency components of a signal are bounded, and an infinite number of basis functions is computationally infeasible. These issues are addressed within the multi-resolution analysis framework, which decomposes the function space into a nested sequence of approximation subspaces

$$V_{j-1} = V_j \oplus W_j. \quad (\text{A8})$$

Here, V_j denotes the approximation subspace, and W_j represents the detail subspace. The approximation subspace is spanned by the scaling functions $\phi(t)$ and the detail subspace by the wavelet functions $\psi(t)$. Any signal $f(t)$ can be expressed up to a coarsest scale J as

$$f(t) = \sum_k c_{J,k} \phi_{J,k}(t) + \sum_{j=J}^{\infty} \sum_k d_{j,k} \psi_{j,k}(t), \quad (\text{A9})$$

where the scaling functions $\phi(t)$ are constructed analogously, namely,

$$\phi_{j,k}(t) = 2^{-j/2} \phi(2^{-j}t - k). \quad (\text{A10})$$

APPENDIX B: PHYSICS-INFORMED KOLMOGOROV-ARNOLD NETWORK (PIKAN)

Kolmogorov–Arnold network (KAN) is inspired by the Kolmogorov–Arnold representation theorem,⁶⁴ which states that

TABLE V. Relative L^2 errors for the shallow water equations at different spatial locations.

Model/position	$x=0$	200	400	600	800	1000	1200	Average
h (PIWF)	8.499×10^{-5}	5.009×10^{-4}	1.494×10^{-3}	8.833×10^{-4}	1.603×10^{-3}	1.947×10^{-3}	2.357×10^{-3}	8.779×10^{-4}
h (PIWF-nonF)	8.834×10^{-1}	8.093×10^{-1}	7.136×10^{-1}	6.415×10^{-1}	5.925×10^{-1}	5.632×10^{-1}	5.469×10^{-1}	7.745×10^{-1}
h (PIWF-nonW)	1.154×10^0	3.147×10^0	5.365×10^0	7.739×10^0	1.096×10^1	1.311×10^1	1.548×10^1	8.213×10^0
h (PIWF-nonL)	1.982×10^{-3}	4.513×10^{-3}	5.254×10^{-3}	8.217×10^{-3}	9.662×10^{-3}	1.101×10^{-2}	1.580×10^{-2}	4.759×10^{-3}
h (PINN)	4.981×10^{-5}	4.068×10^{-2}	4.517×10^{-2}	4.829×10^{-2}	5.347×10^{-2}	5.989×10^{-2}	6.344×10^{-2}	3.685×10^{-2}
h (PINN-tri)	3.611×10^{-4}	6.945×10^{-3}	1.128×10^{-2}	1.295×10^{-2}	1.331×10^{-2}	1.026×10^{-2}	8.380×10^{-3}	6.040×10^{-3}
h (PINN-six)	5.946×10^{-3}	5.580×10^{-2}	8.111×10^{-2}	1.145×10^{-1}	1.380×10^{-1}	1.528×10^{-1}	1.592×10^{-1}	6.033×10^{-2}
hu (PIWF)	8.652×10^{-5}	4.557×10^{-4}	1.415×10^{-3}	6.686×10^{-4}	1.373×10^{-3}	1.721×10^{-3}	2.155×10^{-3}	8.742×10^{-4}
hu (PIWF-nonF)	3.668×10^{-1}	3.493×10^{-1}	3.772×10^{-1}	4.299×10^{-1}	5.068×10^{-1}	6.026×10^{-1}	7.116×10^{-1}	6.692×10^{-1}
hu (PIWF-nonW)	8.518×10^{-1}	9.952×10^{-1}	5.743×10^0	8.773×10^0	7.893×10^0	1.556×10^1	2.875×10^1	8.746×10^0
hu (PIWF-nonL)	1.987×10^{-3}	3.855×10^{-3}	4.019×10^{-3}	6.108×10^{-3}	7.076×10^{-3}	7.916×10^{-3}	1.210×10^{-2}	4.347×10^{-3}
hu (PINN)	1.953×10^{-6}	5.378×10^{-3}	1.074×10^{-2}	1.512×10^{-2}	2.185×10^{-2}	3.021×10^{-2}	3.599×10^{-2}	9.899×10^{-3}
hu (PINN-tri)	3.746×10^{-4}	7.128×10^{-3}	1.157×10^{-2}	1.327×10^{-2}	1.356×10^{-2}	1.038×10^{-2}	8.380×10^{-3}	6.093×10^{-3}
hu (PINN-six)	5.744×10^{-3}	5.406×10^{-2}	7.908×10^{-2}	1.131×10^{-1}	1.388×10^{-1}	1.541×10^{-1}	1.609×10^{-1}	6.114×10^{-2}

any multivariate continuous function defined on a compact domain can be represented by finite compositions of continuous univariate functions. Specifically, for a smooth function $f: [0, 1]^n \rightarrow \mathbb{R}$, the theorem gives

$$f(x) = f(x_1, \dots, x_n) = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^n \phi_{q,p}(x_p) \right), \quad (\text{B1})$$

where $\phi_{q,p}: [0, 1] \mapsto \mathbb{R}$ and $\Phi_q: \mathbb{R} \mapsto \mathbb{R}$ are continuous univariate functions. Unlike conventional multilayer perceptrons, which place nonlinear activation functions on neurons, KAN introduces learnable univariate functions on network edges. This provides a flexible representation in which both the nonlinear transformation and the corresponding functional basis are learned from data.

TABLE VI. Relative L^2 errors for the two-dimensional cylinder wake flow at $\text{Re} = 100$.

Model/time	1	2	3	4	5	6	7	Average
u (PIWF)	9.769×10^{-4}	6.584×10^{-4}	4.716×10^{-4}	3.589×10^{-4}	2.830×10^{-4}	2.373×10^{-4}	2.106×10^{-4}	4.567×10^{-4}
u (PIWF-nonF)	2.571×10^{-3}	1.637×10^{-3}	1.084×10^{-3}	7.605×10^{-4}	5.776×10^{-4}	4.943×10^{-4}	4.961×10^{-4}	1.089×10^{-3}
u (PIWF-nonW)	3.857×10^{-4}	3.434×10^{-4}	3.266×10^{-4}	3.269×10^{-4}	3.373×10^{-4}	3.595×10^{-4}	3.758×10^{-4}	3.507×10^{-4}
u (PIWF-nonL)	6.551×10^{-4}	5.629×10^{-4}	5.499×10^{-4}	5.276×10^{-4}	5.023×10^{-4}	4.929×10^{-4}	4.870×10^{-4}	5.397×10^{-4}
u (PINN)	6.074×10^{-3}	4.178×10^{-3}	2.956×10^{-3}	2.212×10^{-3}	1.693×10^{-3}	1.360×10^{-3}	1.172×10^{-3}	2.806×10^{-3}
u (PINN-tri)	4.176×10^{-3}	2.832×10^{-3}	1.972×10^{-3}	1.412×10^{-3}	1.029×10^{-3}	7.812×10^{-4}	6.237×10^{-4}	1.832×10^{-3}
u (PINN-six)	5.495×10^{-3}	2.792×10^{-3}	1.607×10^{-3}	1.133×10^{-3}	9.631×10^{-4}	7.932×10^{-4}	9.545×10^{-4}	1.963×10^{-3}
v (PIWF)	2.212×10^{-3}	1.593×10^{-3}	1.141×10^{-3}	9.184×10^{-4}	7.809×10^{-4}	7.054×10^{-4}	6.929×10^{-4}	1.149×10^{-3}
v (PIWF-nonF)	5.658×10^{-3}	4.019×10^{-3}	2.854×10^{-3}	2.220×10^{-3}	1.812×10^{-3}	1.559×10^{-3}	1.373×10^{-3}	2.785×10^{-3}
v (PIWF-nonW)	1.115×10^{-3}	1.032×10^{-3}	9.195×10^{-4}	9.040×10^{-4}	9.141×10^{-4}	9.055×10^{-4}	8.754×10^{-4}	9.522×10^{-4}
v (PIWF-nonL)	1.349×10^{-3}	1.122×10^{-3}	1.090×10^{-3}	1.042×10^{-3}	9.585×10^{-4}	9.019×10^{-4}	9.082×10^{-4}	1.053×10^{-3}
v (PINN)	1.212×10^{-2}	9.613×10^{-3}	7.710×10^{-3}	6.396×10^{-3}	5.447×10^{-3}	4.790×10^{-3}	4.354×10^{-3}	7.204×10^{-3}
v (PINN-tri)	7.568×10^{-3}	5.785×10^{-3}	4.323×10^{-3}	3.419×10^{-3}	2.817×10^{-3}	2.485×10^{-3}	2.213×10^{-3}	4.087×10^{-3}
v (PINN-six)	1.373×10^{-2}	8.271×10^{-3}	5.899×10^{-3}	4.458×10^{-3}	3.894×10^{-3}	3.967×10^{-3}	4.294×10^{-3}	6.359×10^{-3}
p (PIWF)	1.802×10^{-3}	9.964×10^{-4}	1.090×10^{-3}	1.013×10^{-3}	8.990×10^{-4}	8.393×10^{-4}	8.027×10^{-4}	1.063×10^{-3}
p (PIWF-nonF)	3.570×10^{-3}	2.260×10^{-3}	2.088×10^{-3}	1.869×10^{-3}	2.125×10^{-3}	2.076×10^{-3}	1.784×10^{-3}	2.253×10^{-3}
p (PIWF-nonW)	9.982×10^{-4}	1.114×10^{-3}	1.442×10^{-3}	1.059×10^{-3}	1.050×10^{-3}	1.141×10^{-3}	9.876×10^{-4}	1.113×10^{-3}
p (PIWF-nonL)	2.335×10^{-3}	1.421×10^{-3}	1.434×10^{-3}	1.509×10^{-3}	1.504×10^{-3}	1.307×10^{-3}	1.276×10^{-3}	1.541×10^{-3}
p (PINN)	1.254×10^{-2}	9.112×10^{-3}	9.144×10^{-3}	6.225×10^{-3}	5.983×10^{-3}	5.134×10^{-3}	4.141×10^{-3}	7.469×10^{-3}
p (PINN-tri)	6.494×10^{-3}	4.796×10^{-3}	5.878×10^{-3}	3.434×10^{-3}	3.024×10^{-3}	2.224×10^{-3}	2.678×10^{-3}	4.075×10^{-3}
p (PINN-six)	1.050×10^{-2}	8.310×10^{-3}	4.515×10^{-3}	8.416×10^{-3}	3.474×10^{-3}	5.343×10^{-3}	9.175×10^{-3}	7.104×10^{-3}

TABLE VII. Trainable-parameter counts and training costs. Entries are reported as trainable parameters/GPU hours when both values are available; otherwise, only the trainable-parameter count is listed.

Model	Burgers' equation ($\nu = 0.01/\pi$)	Burgers' equation ($\nu = 0.001$)	Shallow water	Kovaszny flow	Taylor–Green vortex	Cylinder wake
FD	0/-	0/-	0/-	...	...	...
PINN	12 951/0.45	12 951/0.45	13 402/3.20	50 819	50 435	50 435/3.52
PINN-tri	...	...	115 202/5.21	...	...	446 211/21.90
PINN-six	...	...	455 402/5.52	...	...	1 777 155/29.70
PIKAN	126 500/5.81	126 500/5.53	128 000/4.26	497 920	499 200	499 200/48.06
PIWF	269 223/5.30	359 188/5.30	269 374/5.63	102 322	794 299	794 299/28.8
FNO	549 633	549 633	...	2 368 227	2 368 355	2 368 355
PIWF-nonF	...	...	240 778	...	...	671 771
PIWF-nonW	...	...	229 684	...	...	584 227
PIWF-nonL	...	...	239 074	...	...	662 715

TABLE VIII. Mean relative errors of the cylinder wake averaged over $t = 1, \dots, 7$ for three independent runs. The reported run is reproduced from Table VI.

Model	Variable	Reported run	Repeat 2	Repeat 3
PIWF	u	4.567×10^{-4}	5.403×10^{-4}	3.850×10^{-4}
	v	1.149×10^{-3}	1.010×10^{-3}	1.408×10^{-3}
	p	1.063×10^{-3}	1.286×10^{-3}	8.590×10^{-4}
PIWF-nonF	u	1.089×10^{-3}	1.186×10^{-3}	8.210×10^{-4}
	v	2.785×10^{-3}	3.181×10^{-3}	2.289×10^{-3}
	p	2.253×10^{-3}	1.791×10^{-3}	2.787×10^{-3}
PIWF-nonW	u	3.507×10^{-4}	3.911×10^{-4}	3.044×10^{-4}
	v	9.522×10^{-4}	7.627×10^{-4}	1.110×10^{-3}
	p	1.113×10^{-3}	1.381×10^{-3}	9.420×10^{-4}
PIWF-nonL	u	5.397×10^{-4}	4.458×10^{-4}	6.466×10^{-4}
	v	1.053×10^{-3}	1.155×10^{-3}	8.180×10^{-4}
	p	1.541×10^{-3}	1.331×10^{-3}	1.864×10^{-3}

TABLE IX. Relative L^2 errors of the FNO predictions for the Taylor–Green vortex. The average is taken over all 31 snapshots from $t = 0$ to 30.

Variable/Time	0	5	10	15	20	25	30	Average
u	1.0384×10^{-2}	1.0850×10^{-2}	1.0000×10^{-2}	8.7145×10^{-3}	1.0715×10^{-2}	1.1331×10^{-2}	1.2604×10^{-2}	1.0773×10^{-2}
v	1.4275×10^{-2}	1.3983×10^{-2}	1.3353×10^{-2}	1.2703×10^{-2}	1.4770×10^{-2}	1.4881×10^{-2}	1.6363×10^{-2}	1.4386×10^{-2}
p	3.2454×10^{-2}	2.4910×10^{-2}	2.5206×10^{-2}	3.3880×10^{-2}	3.1312×10^{-2}	2.4164×10^{-2}	2.8892×10^{-2}	2.8585×10^{-2}

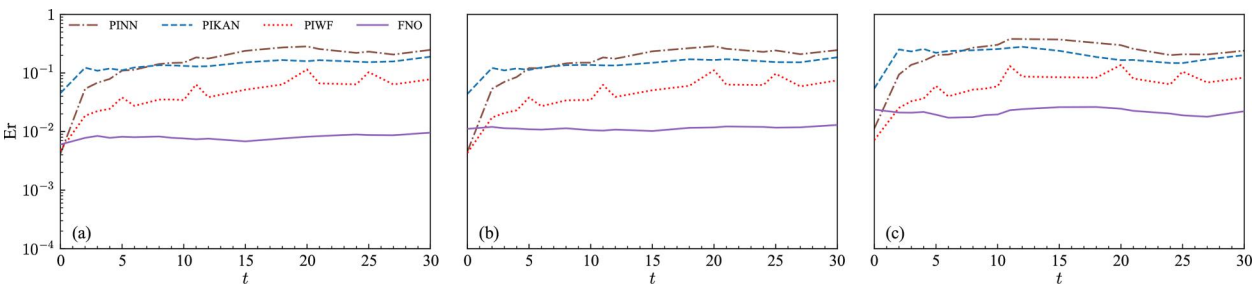


FIG. 23. Temporal evolution of the relative L^2 errors for the Taylor–Green vortex predicted by PINN, PIKAN, PIWF, and FNO: (a) u , (b) v , and (c) p .

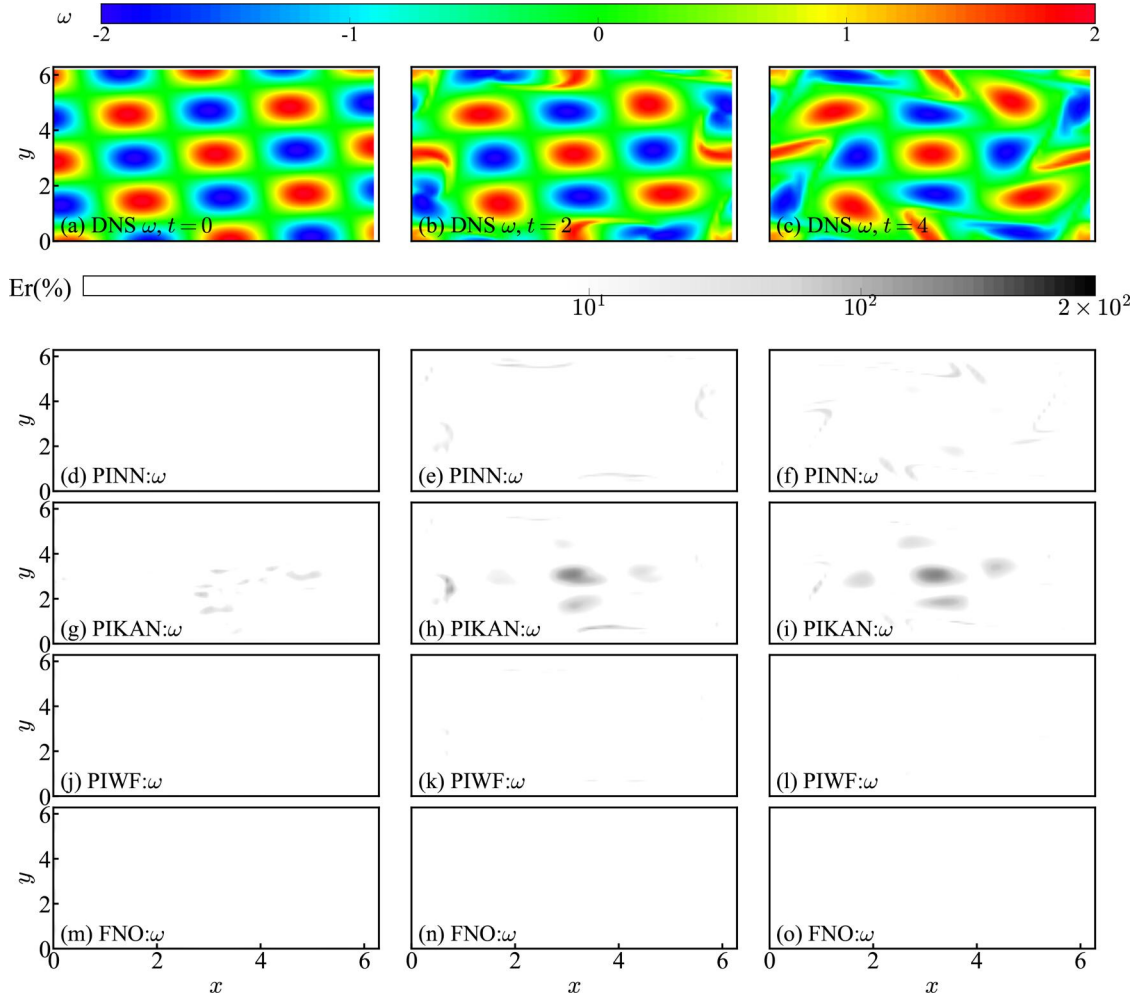


FIG. 24. Relative errors of spanwise vorticity predicted by PINN, PIKAN, PIWF, and FNO for the Taylor–Green vortex at $t = 0, 2$, and 4 . The reference spanwise vorticity is shown in the top row.

For the l th KAN layer, let $\mathbf{x}_l \in \mathbb{R}^{N_l}$ and $\mathbf{x}_{l+1} \in \mathbb{R}^{N_{l+1}}$ denote the input and output feature vectors, respectively. The layer mapping can be written as

$$\mathbf{x}_{l+1,j} = \sum_{i=1}^{N_l} \phi_{j,i}^{(l)}(\mathbf{x}_{l,i}), \quad j = 1, 2, \dots, N_{l+1}, \quad (\text{B2})$$

or equivalently,

$$\mathbf{x}_{l+1} = \Phi_l(\mathbf{x}_l) = \begin{bmatrix} \phi_{1,1}^{(l)}(\cdot) & \phi_{1,2}^{(l)}(\cdot) & \cdots & \phi_{1,N_l}^{(l)}(\cdot) \\ \phi_{2,1}^{(l)}(\cdot) & \phi_{2,2}^{(l)}(\cdot) & \cdots & \phi_{2,N_l}^{(l)}(\cdot) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{N_{l+1},1}^{(l)}(\cdot) & \phi_{N_{l+1},2}^{(l)}(\cdot) & \cdots & \phi_{N_{l+1},N_l}^{(l)}(\cdot) \end{bmatrix} \mathbf{x}_l, \quad (\text{B3})$$

where Φ_l denotes the collection of learnable edge functions in the l th KAN layer. Each univariate function $\phi_{j,i}^{(l)}(\cdot)$ is parameterized by a residual activation component and a spline component as

$$\phi_{j,i}^{(l)}(z) = w_{b,ji}^{(l)} b(z) + w_{s,ji}^{(l)} \text{spline}_{ji}^{(l)}(z), \quad (\text{B4})$$

where z is a scalar input, $w_{b,ji}^{(l)}$ and $w_{s,ji}^{(l)}$ are trainable scaling coefficients, and the spline term is expressed as a linear combination of B-spline basis functions,

$$\text{spline}_{ji}^{(l)}(z) = \sum_{m=1}^M c_{jim}^{(l)} B_m(z). \quad (\text{B5})$$

Here, $B_m(z)$ denotes the m th B-spline basis function, and $c_{jim}^{(l)}$ are learnable spline coefficients. The residual activation is chosen as the SiLU function,

$$b(z) = \text{SiLU}(z) = \frac{z}{1 + \exp(-z)}. \quad (\text{B6})$$

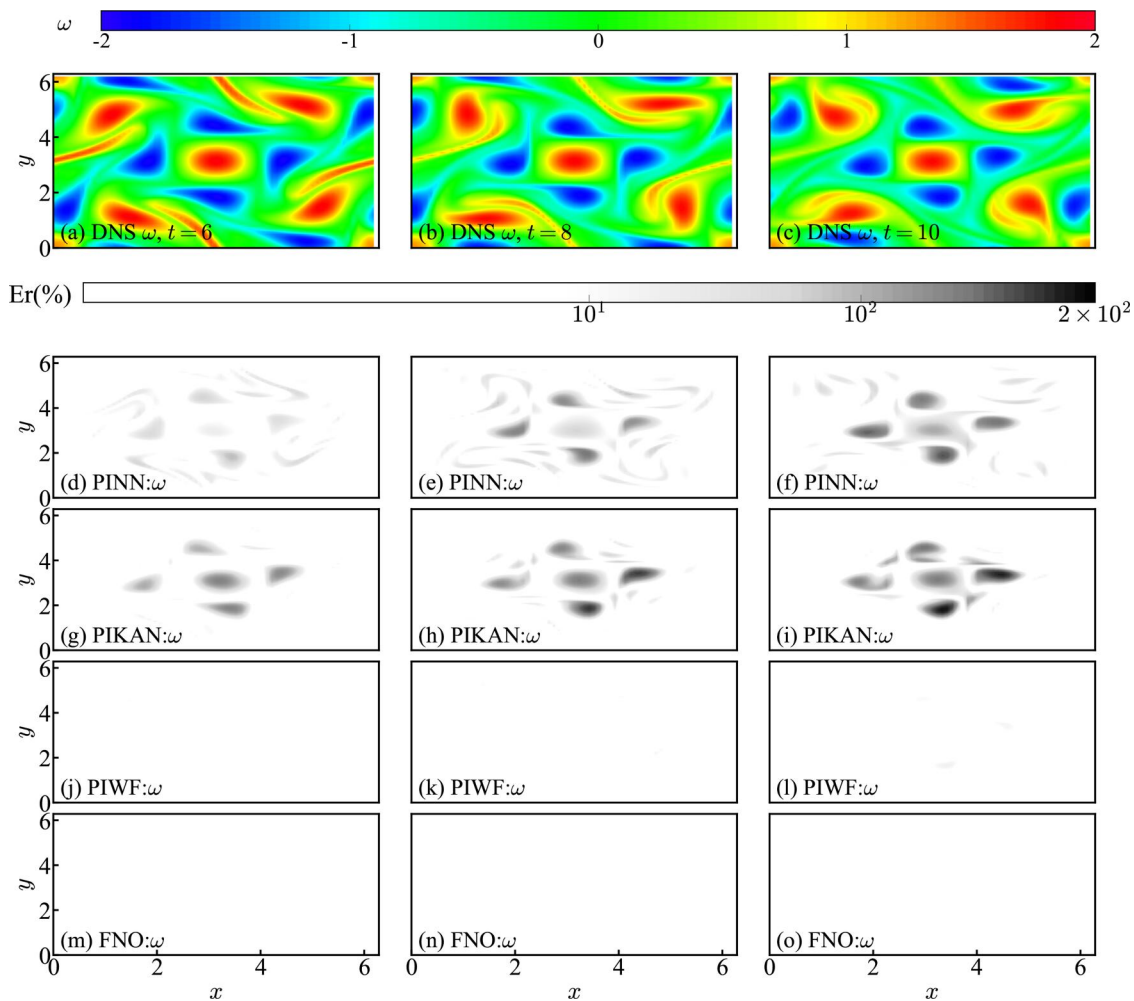


FIG. 25. Relative errors of spanwise vorticity predicted by PINN, PIKAN, PIWF, and FNO for the Taylor–Green vortex at $t = 6, 8$, and 10 . The reference spanwise vorticity is shown in the top row.

TABLE X. Relative L^2 errors of the FNO predictions for the cylinder wake. The average is taken over $t = 1, \dots, 7$.

Vaiable/ Time	1	2	3	4	5	6	7	Average
u	2.3697×10^{-3}	2.3950×10^{-3}	2.2244×10^{-3}	2.1941×10^{-3}	2.1868×10^{-3}	2.2574×10^{-3}	3.1738×10^{-3}	2.4002×10^{-3}
v	5.5015×10^{-3}	5.0785×10^{-3}	5.3941×10^{-3}	5.8382×10^{-3}	5.5211×10^{-3}	5.6137×10^{-3}	8.1996×10^{-3}	5.8781×10^{-3}
p	6.6438×10^{-3}	7.2330×10^{-3}	7.2977×10^{-3}	6.8685×10^{-3}	7.0540×10^{-3}	7.1544×10^{-3}	1.1741×10^{-2}	7.7131×10^{-3}

In the physics-informed Kolmogorov–Arnold network (PIKAN) considered in this work, the KAN serves as the surrogate model for the solution field

$$\mathbf{u}_\theta^{\text{KAN}}(\mathbf{x}, t) = \text{KAN}_\theta(\mathbf{x}, t), \quad (\text{B7})$$

where θ collects all trainable parameters, including the spline coefficients and scaling factors in the KAN layers. The physical constraints are imposed in the same manner as in the standard PINN

formulation described in Sec. II A. Specifically, the initial-condition, boundary-condition, and governing-equation residual losses are combined into the composite objective function defined in Eq. (4). The PDE residual is evaluated by substituting $\mathbf{u}_\theta^{\text{KAN}}(\mathbf{x}, t)$ into the governing equation, namely,

$$\mathcal{L}_{\text{pde}}(\mathbf{u}_\theta^{\text{KAN}}; \mathbf{x}_{\text{pde}}, t_{\text{pde}}) = \frac{\partial \mathbf{u}_\theta^{\text{KAN}}}{\partial t}(\mathbf{x}_{\text{pde}}, t_{\text{pde}}) - R(\mathbf{u}_\theta^{\text{KAN}}; \mathbf{x}_{\text{pde}}, t_{\text{pde}}), \quad (\text{B8})$$

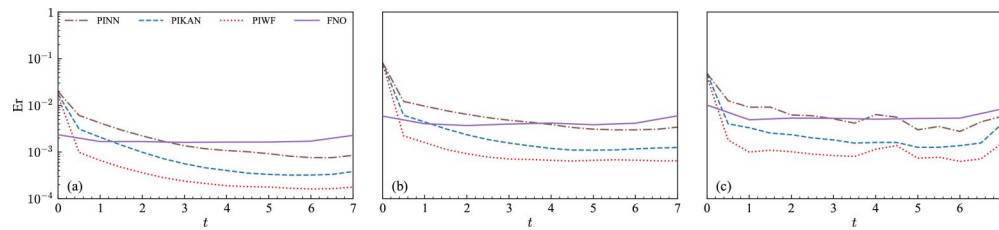


FIG. 26. Temporal evolution of the relative L^2 errors for the cylinder wake predicted by PINN, PIKAN, PIWF, and FNO: (a) u , (b) v , and (c) p .

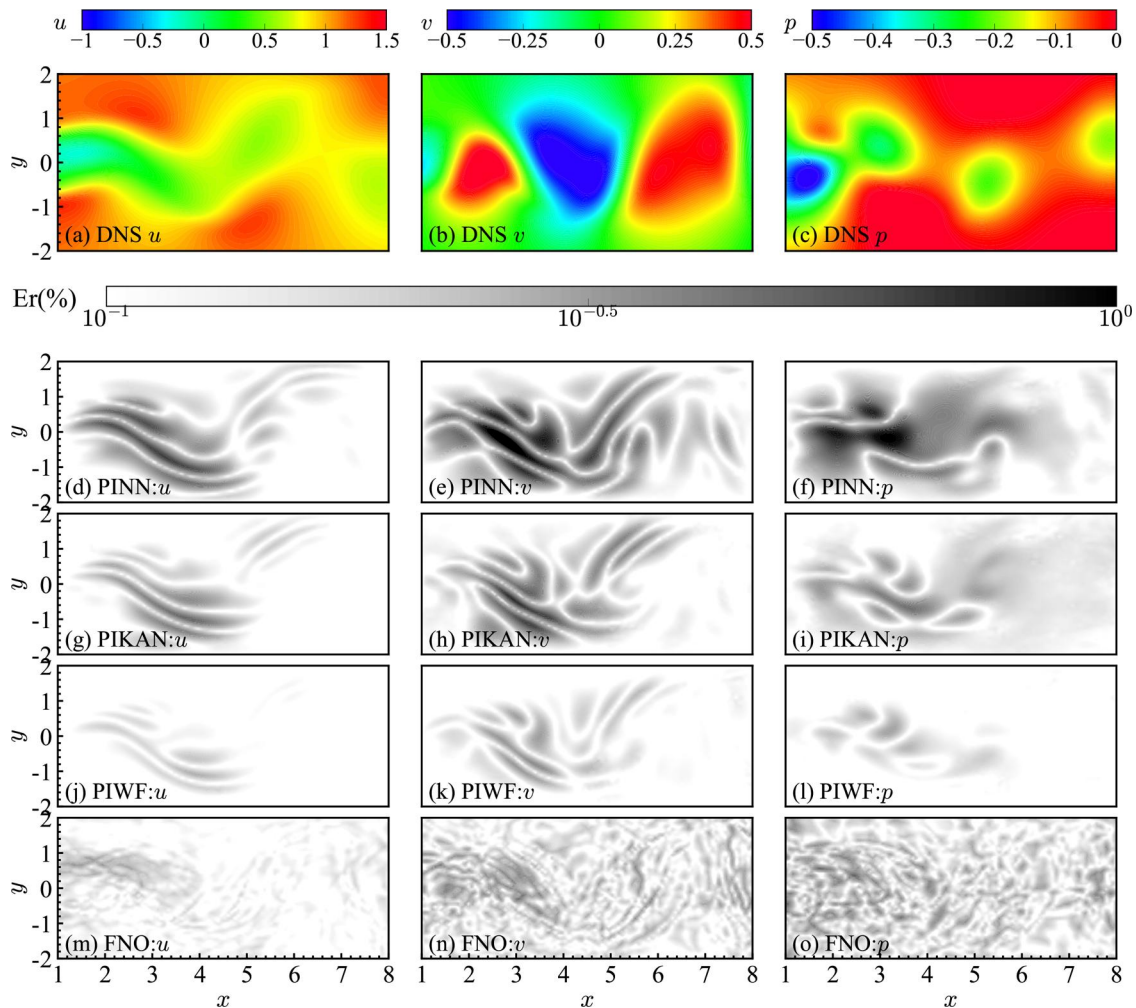


FIG. 27. Relative error contours of u , v , and p for the cylinder wake at $t = 1$. From top to bottom: DNS, PINN, PIKAN, PIWF, and FNO results.

with the required derivatives computed by automatic differentiation. Therefore, PIKAN differs from the standard PINN mainly in the choice of the neural architecture: the fully connected MLP backbone in PINN is replaced by a KAN backbone, while the physics-informed training objective remains consistent with the formulation in Sec. II A.

APPENDIX C: ABLATION EXPERIMENT

We conduct an ablation experiment and a parameter sensitivity analysis of the PIWF model in this section. Three PIWF-based control models are considered: PIWF without the wavelet layer (PIWF-nonW), PIWF without the Fourier-basis layer (PIWF-nonF), and

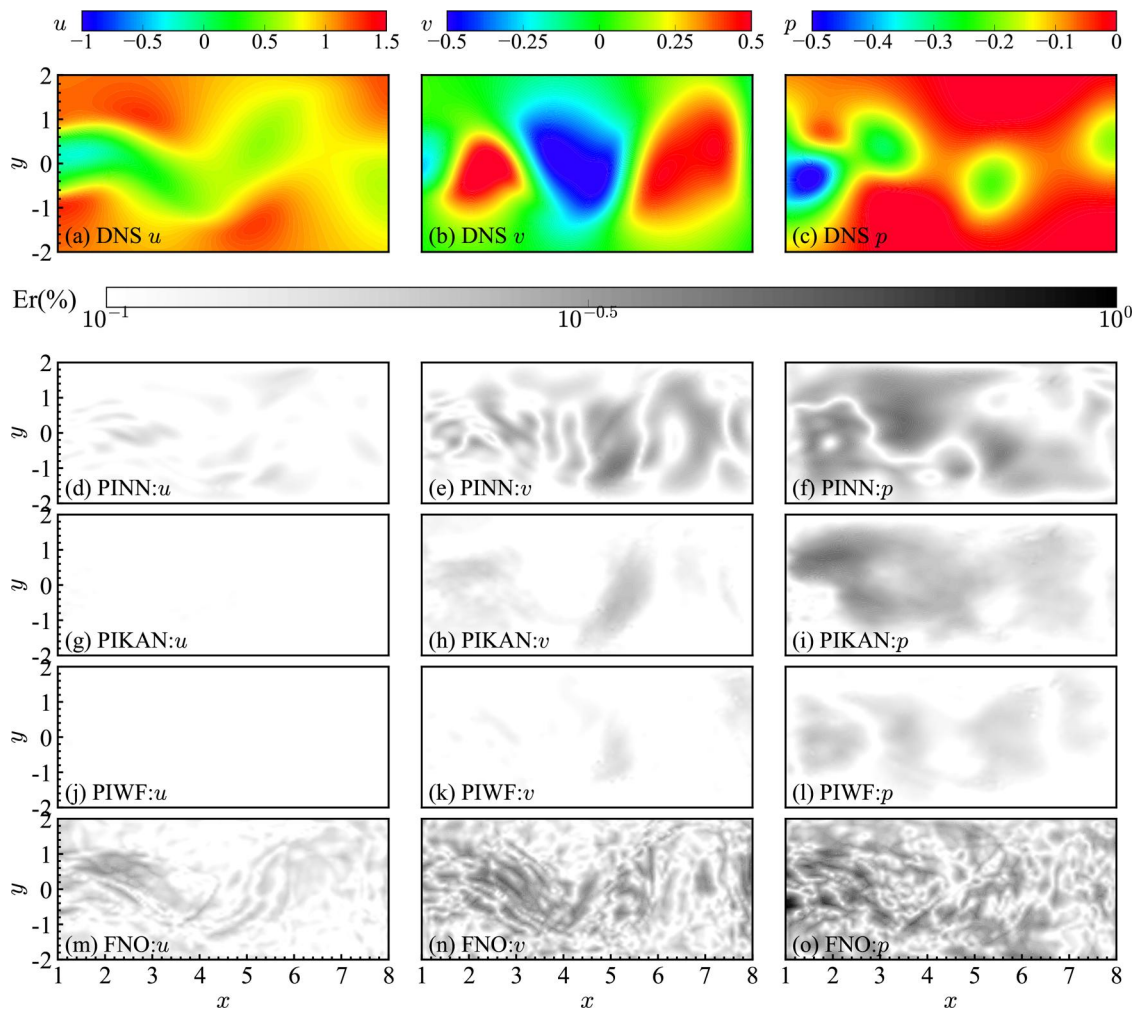


FIG. 28. Relative error contours of u , v , and p for the cylinder wake at $t = 7$. From top to bottom: DNS, PINN, PIKAN, PIWF, and FNO results.

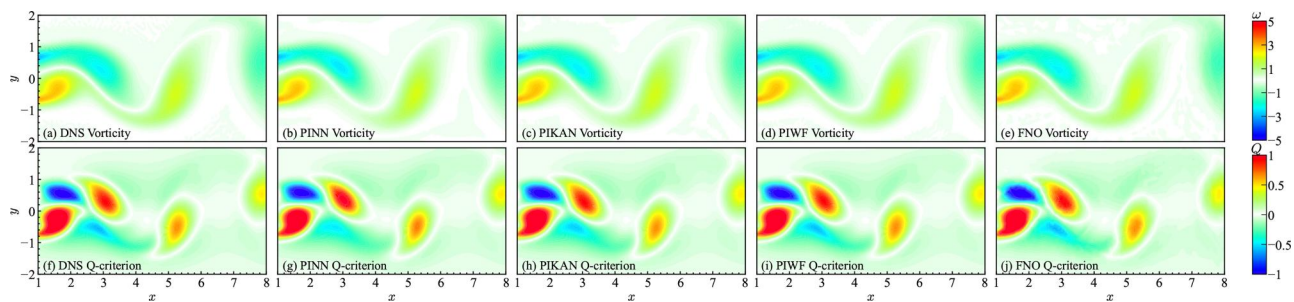


FIG. 29. Vorticity and Q-criterion fields for the cylinder wake at $t = 1$ obtained from DNS, PINN, PIKAN, PIWF, and FNO.

PIWF without the MLP layer (PIWF-nonL). Since each layer of PIWF contains three parallel feature extraction sublayers, the model appears to have three times as many neurons as a conventional neural network. Although the neurons among the three feature extraction

sublayers of our PIWF model do not connect to each other, we still include a PINN model with three times the width (PINN-tri) as a control for rigor. We additionally include PINN-six, whose hidden-layer width is six times that of the standard PINN, to provide a capacity

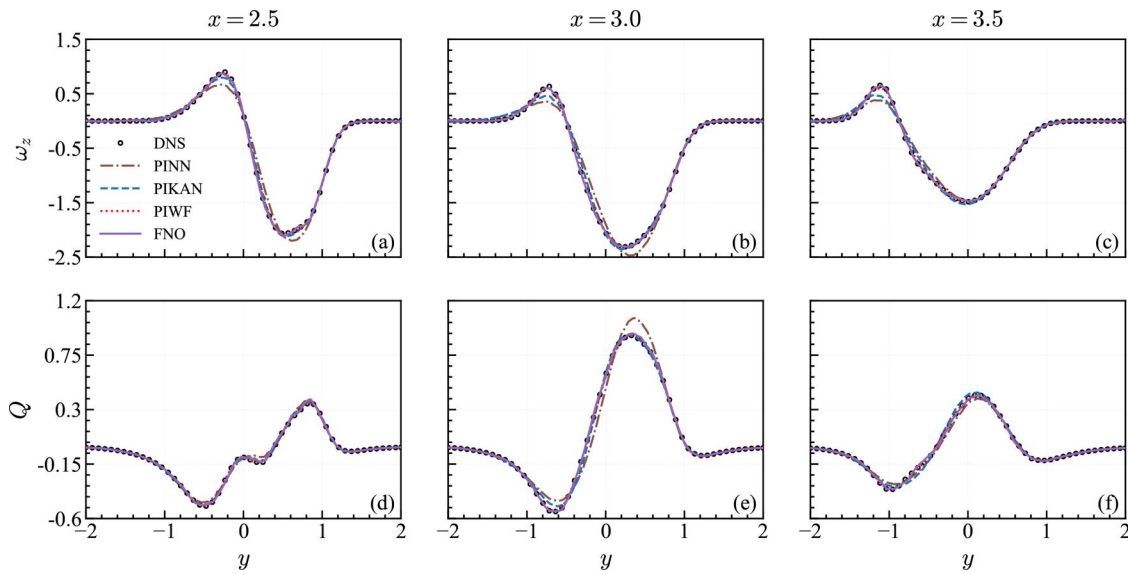


FIG. 30. Profiles of spanwise vorticity and the Q-criterion at $x = 2.5, 3.0$, and 3.5 for the cylinder wake at $t = 1$ predicted by PINN, PIKAN, PIWF, and FNO.

TABLE XI. Relative L^2 errors of the FNO predictions for Burgers' equation and extended-domain Kovasznay flow.

Case	u	v	p
Burgers' equation, $\nu = 0.01/\pi$	8.3571×10^{-4}	n/a	n/a
Burgers' equation, $\nu = 0.001$	7.8122×10^{-4}	n/a	n/a
Kovasznay flow, $Re = 1000$, $x \in [0, 40]$	2.3636×10^{-1}	7.1988×10^{-1}	1.2249×10^{-1}

control with a parameter count comparable to or greater than that of PIWF. Within each case, all physics-informed models use the same governing equations, initial and boundary conditions, training-point specifications, optimizer, learning rate, number of training iterations,

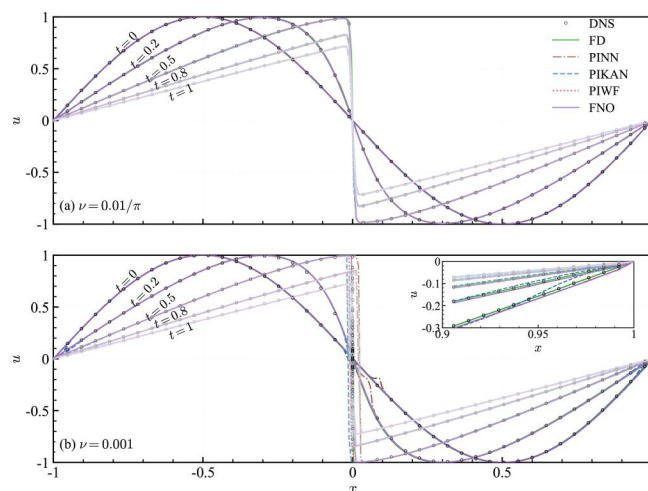


FIG. 31. Velocity profiles for Burgers' equation predicted by FD, PINN, PIKAN, PIWF, and FNO: (a) $\nu = 0.01/\pi$ and (b) $\nu = 0.001$.

and hardware. The trainable-parameter counts of the implemented architectures are reported in Table VII. Table V reports the relative errors for the shallow water equation. The ablated models all exhibit an increase in error to varying degrees compared with the original model, which demonstrates the contribution of the wavelet layer, the Fourier-basis layer, and the MLP layer in this case. The PIWF-nonW model shows the most significant error increase, followed by PIWF-nonF and then PIWF-nonL. For the shallow water equation with a discontinuous solution, the wavelet layer plays a key role in capturing local features; therefore, PIWF-nonW exhibits a larger error than PIWF-nonL. The results also indicate that the absence of either Fourier or wavelet feature extraction causes the model to overemphasize either global or local features, leading to training imbalance and a substantial increase in the final error. The PIWF-nonL model shows a relatively small error increase since the Fourier and wavelet feature extractions capture most of the important features, leaving the residual feature extraction to account for a minor portion. Notably, the three-times-width PINN (PINN-tri) shows a clear advantage over the standard PINN. Despite the absence of information exchange among the different feature extraction sublayers, PIWF still maintains a significant advantage over PINN-tri, which demonstrates that the superiority of PIWF does not arise simply from stacking more neurons but from a more efficient training architecture compared with conventional neural networks. PINN-six is also less accurate than PIWF, with average errors of 6.033×10^{-2} for h and 6.114×10^{-2} for hu .

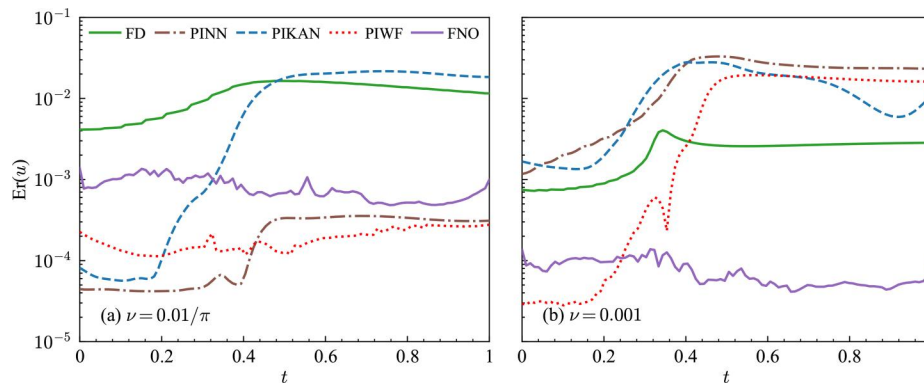


FIG. 32. Temporal evolution of the relative L^2 errors for Burgers' equation predicted by FD, PINN, PIKAN, PIWF, and FNO in two cases with different viscosities: (a) $\nu = 0.01/\pi$ and (b) $\nu = 0.001$.

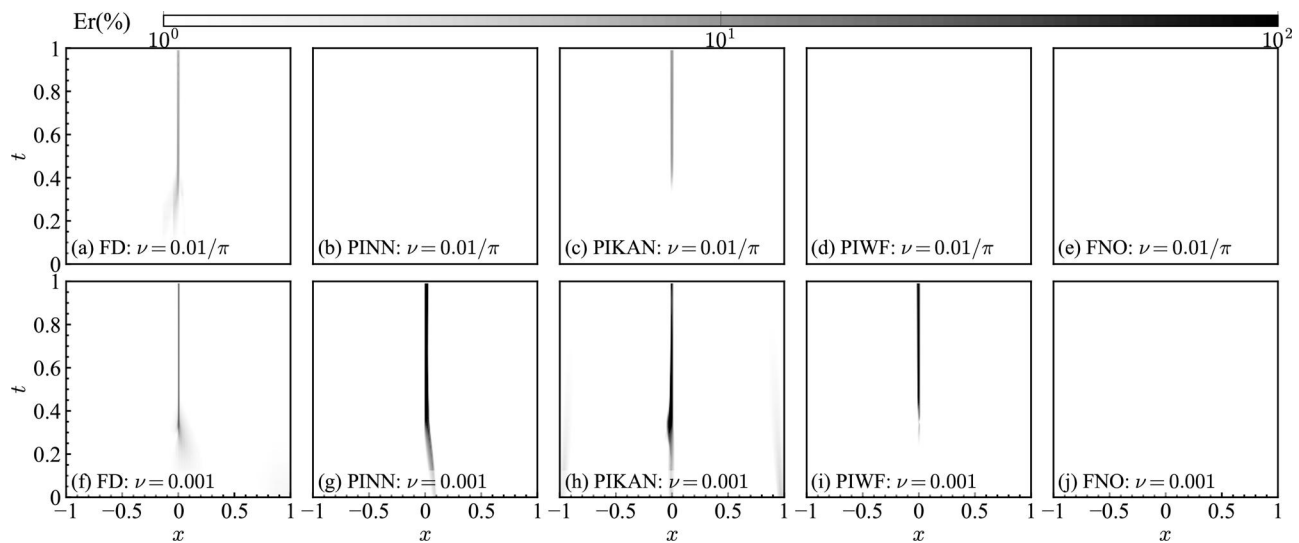


FIG. 33. Relative error contours for Burgers' equation predicted by FD, PINN, PIKAN, PIWF, and FNO. The top and bottom rows correspond to $\nu = 0.01/\pi$ and $\nu = 0.001$, respectively.

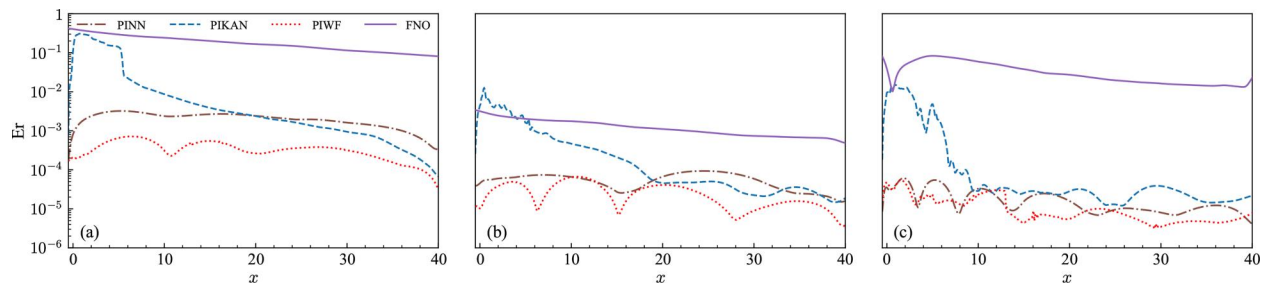


FIG. 34. Relative error profiles for the extended-domain Kovasznay flow predicted by PINN, PIKAN, PIWF, and FNO: (a) u , (b) v , and (c) p .

Table VI reports the relative errors for the cylinder wake flow. Compared with the shallow water equation case that contains a discontinuous solution, the cylinder wake flow favors the capture of global features. Consequently, in this example, PIWF-nonF shows a more pronounced error increase than PIWF-nonW. The

component-wise results also show that PIWF-nonW gives slightly lower mean errors for u and v but a higher error for p , whereas PIWF-nonL gives a slightly lower error for v but higher errors for u and p . Thus, an individual branch does not necessarily reduce the error of every output variable. The Fourier, wavelet, and MLP

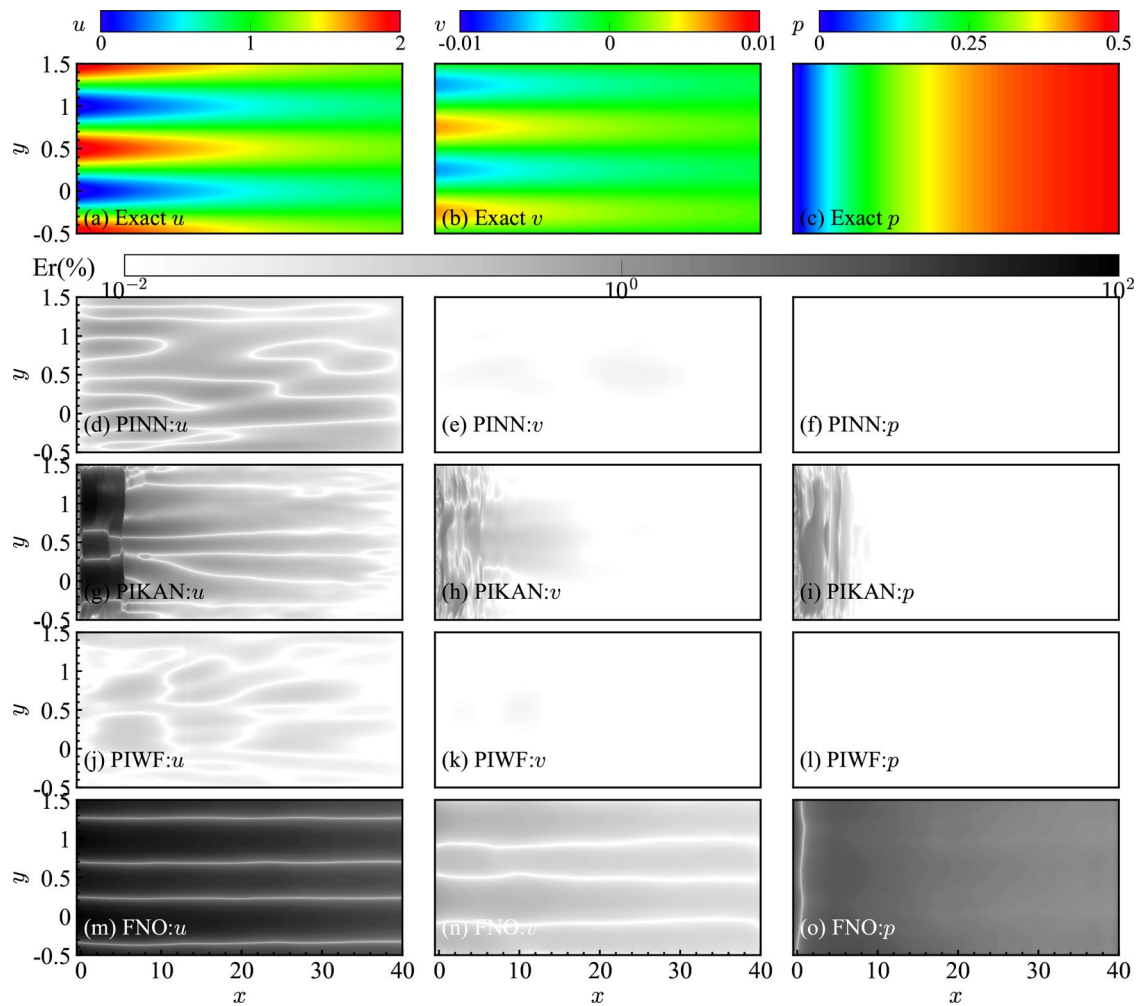


FIG. 35. Reference fields and relative error contours for the extended-domain Kovasznay flow. From top to bottom: reference, PINN, PIKAN, PIWF, and FNO results; from left to right: u , v , and p .

features are fused into a shared representation and optimized jointly for the coupled outputs and physics losses. For a relatively smooth velocity component, localized Haar features may be less useful than that for a wet-dry front or a sharply varying quantity, and their inclusion can redistribute the finite fitting capacity and channel-attention weights. PINN-tri improves upon the standard PINN but remains less accurate than PIWF for all three variables. PINN-six also remains less accurate than PIWF, although it contains more trainable parameters.

The parameter and timing data in Table VII further show that trainable-parameter count does not determine wall-clock cost. In the cylinder wake case, PIKAN contains 499 200 parameters, fewer than PIWF's 794 299, but requires 48.06 GPU hours, compared with 28.8 GPU hours for PIWF on the same RTX 4090. In the extended-domain Kovasznay case, PIWF uses a hidden-layer width of 50, whereas PINN and PIKAN use a width of 128; the corresponding parameter counts are 102 322, 50 819, and 497 920. This narrower PIWF configuration limits capacity while retaining the

lowest reported errors. PINN-tri provides an additional width control: it improves upon PINN but remains less accurate than PIWF for both shallow-water variables and all three cylinder wake variables. We have also added PINN-six, whose parameter count is 455 402 in the shallow-water case and 1 777 155 in the cylinder wake case. Although PINN-six contains more trainable parameters than PIWF in both cases, its relative L^2 errors remain higher: its shallow-water average errors are 6.033×10^{-2} for h and 6.114×10^{-2} for hu , and its cylinder wake average errors are 1.963×10^{-3} , 6.359×10^{-3} , and 7.104×10^{-3} for u , v , and p , respectively. Thus, increasing the width of a conventional PINN alone does not account for the accuracy of PIWF.

The repeated runs in Table VIII remain within the same order of magnitude as the originally reported run. Removing the Fourier branch consistently gives the largest errors for all three output variables. PIWF-nonW gives slightly lower errors for the smooth velocity components but a higher pressure error than the complete PIWF model. The existing results support a more limited

conclusion about attention: the module is optimized jointly over the fused channels, outputs, and physics losses. It can reduce the influence of a less useful branch, but it is not an output-wise gate and is not expected to minimize every component independently. The repeated runs should therefore be used to assess the stability of the error trends; any claim about branch suppression would additionally require the learned attention weights to be recorded and analyzed by case.

APPENDIX D: COMPARISONS WITH FOURIER NEURAL OPERATOR

The Fourier neural operator (FNO) is a widely-used operator-learning method, and its Fourier-space convolution makes it a reasonable reference for the present study.⁶⁵ The comparison is not strictly one-to-one because FNO is trained here as a discrete field-to-field rollout model, whereas PIWF represents the continuous coordinate-to-solution mapping and is trained with the governing-equation residual together with the prescribed initial and boundary data. Because the two methods address different learning settings, the FNO results are presented separately from the physics-informed comparisons in Sec. III.

The FNO predictions are evaluated with the same error definitions and post-processing procedures used for the corresponding comparisons in Sec. III. The parameter counts of the implemented FNO models are included in Table VII. The errors of the Taylor–Green vortex case are reported in Table IX and Figs. 23–25, the errors of the cylinder wake case in Table X and Figs. 26–30, and the results for Burgers’ equation and extended-domain Kovasznay flow in Table XI and Figs. 31–35.

The FNO gives accurate predictions for the Taylor–Green vortex and retains the periodic vortical pattern over the reported interval. This behavior is consistent with the suitability of Fourier-space operators for smooth periodic fields, while the different training setting should still be kept in mind.

For the cylinder wake, FNO reconstructs the principal wake pattern and gives low primary-field errors over most of the interval. PIWF gives lower reported errors for this strongly unsteady case while using the same physics-informed constraints as the comparisons in Sec. III.

For Burgers’ equation, FNO gives overall relative L^2 errors of 8.3571×10^{-4} for $\nu = 0.01/\pi$ and 7.8122×10^{-4} for $\nu = 0.001$. The solution profiles follow the reference solutions closely, and the error contours show that the remaining differences are concentrated near the steep velocity layer. For extended-domain Kovasznay flow, FNO reproduces the overall downstream variation but has larger errors in the transverse velocity, with relative L^2 errors of 2.3636×10^{-1} , 7.1988×10^{-1} , and 1.2249×10^{-1} for u , v , and p , respectively. The comparatively large Kovasznay errors should be interpreted in light of the field-to-field rollout setting and the long spatial interval. Under the tested settings, PIWF gives lower errors for all three variables in this case.

REFERENCES

- ¹W. A. Strauss, *Partial Differential Equations: An Introduction* (John Wiley & Sons, 2007).
- ²C. E. Pearson, “A computational method for viscous flow problems,” *J. Fluid Mech.* **21**, 611 (1965).
- ³G. Griffiths and W. E. Schiesser, *Traveling Wave Analysis of Partial Differential Equations: Numerical and Analytical Methods with MATLAB and Maple* (Academic Press, 2010).
- ⁴P. D. Christofides and J. Chow, “Nonlinear and robust control of PDE systems: Methods and applications to transport-reaction processes,” *Appl. Mech. Rev.* **55**, B29 (2002).
- ⁵C. I. Steefel and K. T. B. MacQuarrie, “Approaches to modeling of reactive transport in porous media,” in *Reactive Transport in Porous Media*, edited by P. C. Lichtner, C. I. Steefel, and E. H. Oelkers (De Gruyter, Berlin, Boston, 1996), Chap. II, pp. 83–130.
- ⁶M. Errera, A. Dugeai, Ph. Girodroux-Lavigne, J.-D. Garaud, M. Poinot, S. Cerqueira, and G. Chaineray, “Multi-physics coupling approaches for aerospace numerical simulations,” *AerospaceLab* 1–16 (2011).
- ⁷F. Kang and S. Zhong-Ci, *Mathematical Theory of Elastic Structures* (Springer, 1996), pp. 289–385.
- ⁸M. N. Özisik, H. R. Orlande, M. J. Colaço, and R. M. Cotta, *Finite Difference Methods in Heat Transfer* (CRC Press, 2017).
- ⁹F. Moukalled, L. Mangani, and M. Darwish, *The Finite Volume Method in Computational Fluid Dynamics: An Advanced Introduction with OpenFOAM® and Matlab* (Springer, 2015), pp. 103–135.
- ¹⁰P. Moin and K. Mahesh, “Direct numerical simulation: A tool in turbulence research,” *Annu. Rev. Fluid Mech.* **30**, 539 (1998).
- ¹¹X.-Y. Liu, M. Zhu, L. Lu, H. Sun, and J.-X. Wang, “Multi-resolution partial differential equations preserved learning framework for spatiotemporal dynamics,” *Commun. Phys.* **7**, 31 (2024).
- ¹²M. Bauerheim, “Routes towards an effective AI in CFD: An epistemological and technical perspective,” *J. Fluid Mech.* **1031**, P1 (2026).
- ¹³M. Jebahi, F. Dau, J.-L. Charles, and I. Iordanoff, “Multiscale modeling of complex dynamic problems: An overview and recent developments,” *Arch. Comput. Methods Eng.* **23**, 101 (2016).
- ¹⁴X.-W. Zhao, X.-M. Tong, F.-W. Ning, M.-L. Cai, F. Han, and H.-G. Li, “Review of empowering computer-aided engineering with artificial intelligence,” *Adv. Manuf.* **14**, 103 (2026).
- ¹⁵G. P. Guruswamy, “A review of numerical fluids/structures interface methods for computations using high-fidelity equations,” *Comput. Struct.* **80**, 31 (2002).
- ¹⁶M. Raissi, P. Perdikaris, and G. E. Karniadakis, “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations,” *J. Comput. Phys.* **378**, 686 (2019).
- ¹⁷S. Zhao, Z. Li, B. Fan, Y. Wang, H. Yang, and J. Wang, “LESnets (large-eddy simulation nets): Physics-informed neural operator for large-eddy simulation of turbulence,” *J. Comput. Phys.* **537**, 114125 (2025).
- ¹⁸J.-X. Wang, J.-L. Wu, and H. Xiao, “Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data,” *Phys. Rev. Fluids* **2**, 034603 (2017).
- ¹⁹X. Jin, S. Cai, H. Li, and G. E. Karniadakis, “NSFnets (Navier–Stokes flow nets): Physics-informed neural networks for the incompressible Navier–Stokes equations,” *J. Comput. Phys.* **426**, 109951 (2021).
- ²⁰A. Arzani, J.-X. Wang, and R. M. D’Souza, “Uncovering near-wall blood flow from sparse data with physics-informed neural networks,” *Phys. Fluids* **33**, 071905 (2021).
- ²¹C. Zhao, F. Zhang, W. Lou, X. Wang, and J. Yang, “A comprehensive review of advances in physics-informed neural networks and their applications in complex fluid dynamics,” *Phys. Fluids* **36**, 101301 (2024).
- ²²W. Zhou, S. Miwa, and K. Okamoto, “Advancing fluid dynamics simulations: A comprehensive approach to optimizing physics-informed neural networks,” *Phys. Fluids* **36**, 013615 (2024).
- ²³J. Wu, Y. Wu, X. Li, and G. Zhang, “Physics-informed neural networks for Kelvin–Helmholtz instability with spatiotemporal and magnitude multiscale,” *Phys. Fluids* **37**, 034118 (2025).
- ²⁴L. Lu, R. Pestourie, W. Yao, Z. Wang, F. Verdugo, and S. G. Johnson, “Physics-informed neural networks with hard constraints for inverse design,” *SIAM J. Sci. Comput.* **43**, B1105 (2021).
- ²⁵H. Jeong, J. Bai, C. P. Batuwatta-Gamage, C. Rathnayaka, Y. Zhou, and Y. Gu, “A physics-informed neural network-based topology optimization (PINNTO) framework for structural optimization,” *Eng. Struct.* **278**, 115484 (2023).

- ²⁶R. Sun, H. Jeong, J. Zhao, Y. Gou, E. Sauret, Z. Li, and Y. Gu, "A physics-informed neural network framework for multi-physics coupling microfluidic problems," *Comput. Fluids* **284**, 106421 (2024).
- ²⁷K. Luo, J. Zhao, Y. Wang, J. Li, J. Wen, J. Liang, H. Soekmadji, and S. Liao, "Physics-informed neural networks for PDE problems: A comprehensive review," *Artif. Intell. Rev.* **58**, 323 (2025).
- ²⁸H. Wang, Y. Cao, Z. Huang, Y. Liu, P. Hu, X. Luo, Z. Song, W. Zhao, J. Liu, J. Sun *et al.*, [arXiv:2408.12171](https://arxiv.org/abs/2408.12171) (2024).
- ²⁹Z. Hao, S. Liu, Y. Zhang, C. Ying, Y. Feng, H. Su, and J. Zhu, [arXiv:2211.08064](https://arxiv.org/abs/2211.08064) (2022).
- ³⁰R. Lu, J. Jia, Y. J. Lee, Z. Lu, and C.-S. Zhang, "R-PINN: Recovery-type a-posteriori estimator enhanced adaptive PINN," *J. Comput. Phys.* **551**, 114684 (2026).
- ³¹X. Zou, Z. Li, Y. Wang, H. Yang, and J. Wang, "Uncertainty quantification and stability of neural operators for prediction of three-dimensional turbulence," *J. Comput. Phys.* **549**, 114640 (2026).
- ³²M. Kang, X. Lei, and J. Zhao, "Asymptotic-preserving PINNs for relaxation systems of conservation laws: Applications to the Euler equations," *J. Comput. Phys.* **553**, 114712 (2026).
- ³³P. Ren, C. Rao, Y. Liu, Z. Ma, Q. Wang, J.-X. Wang, and H. Sun, "PhySR: Physics-informed deep super-resolution for spatiotemporal data," *J. Comput. Phys.* **492**, 112438 (2023).
- ³⁴Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljavic'c, T. Y. Hou, and M. Tegmark, [arXiv:2404.19756](https://arxiv.org/abs/2404.19756) (2024).
- ³⁵C. Guo, L. Sun, S. Li, Z. Yuan, and C. Wang, "Physics-informed Kolmogorov–Arnold network with Chebyshev polynomials for fluid mechanics," *Phys. Fluids* **37**, 095120 (2025).
- ³⁶Z. Bozorgasl and H. Chen, [arXiv:2405.12832](https://arxiv.org/abs/2405.12832) (2024).
- ³⁷A. A. Aghaei, [arXiv:2406.14495](https://arxiv.org/abs/2406.14495) (2024).
- ³⁸H. Zhou, R. Douvenot, and A. Chabory, "Modeling the long-range wave propagation by a split-step wavelet method," *J. Comput. Phys.* **402**, 109042 (2020).
- ³⁹S. G. Chang, B. Yu, and M. Vetterli, "Adaptive wavelet thresholding for image denoising and compression," *IEEE Trans. Image Process.* **9**, 1532 (2000).
- ⁴⁰L. Cao, Y. Hong, H. Fang, and G. He, "Predicting chaotic time series with wavelet networks," *Physica D* **85**, 225 (1995).
- ⁴¹Y. Fang and T. W. Chow, in *International Symposium on Neural Networks* (Springer, 2006), pp. 80–85.
- ⁴²C. Harnish, L. Dalessandro, K. Matouš, and D. Livescu, "An adaptive wavelet method for nonlinear partial differential equations with applications to dynamic damage modeling," *J. Comput. Phys.* **479**, 112002 (2023).
- ⁴³X. Liu, G. Liu, J. Wang, and Y. Zhou, "A wavelet multiresolution interpolation Galerkin method for targeted local solution enrichment," *Comput. Mech.* **64**, 989 (2019).
- ⁴⁴B. Yang, J. Wang, X. Liu, and Y. Zhou, "High-order adaptive multiresolution wavelet upwind schemes for hyperbolic conservation laws," *Comput. Fluids* **269**, 106111 (2024).
- ⁴⁵P. Frasconi, M. Gori, and G. Soda, "Recurrent neural networks and prior knowledge for sequence processing: A constrained nondeterministic approach," *Knowl.-Based Syst.* **8**, 313 (1995).
- ⁴⁶T. Szandala, *Bio-Inspired Neurocomputing* (Springer, 2020), pp. 203–224.
- ⁴⁷A. K. Alexandridis and A. D. Zaprani, "Wavelet neural networks: A practical guide," *Neural Networks* **42**, 1–27 (2013).
- ⁴⁸N. Navaneeth, T. Tripura, and S. Chakraborty, "Physics informed WNO," *Comput. Methods Appl. Mech. Eng.* **418**, 116546 (2024).
- ⁴⁹Y. Gu, X. Qian, H. Dong, Y. Jiang, M. Gong, L. Yang, M. Song, and Z. Xu, "Physics-informed multiscale attention wavelet neural operator: A novel framework for instability-wave prediction in hypersonic boundary layers," *Phys. Fluids* **38**, 014112 (2026).
- ⁵⁰C. Wang, S. Li, Z. Yuan, and C. Guo, [arXiv:2508.02166](https://arxiv.org/abs/2508.02166) (2025).
- ⁵¹Z. Qin, P. Zhang, F. Wu, and X. Li, in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (IEEE, 2021), pp. 783–792.
- ⁵²A. G. Vigneshwara Raj, P. Murugesan, and M. Nanmaran, "Tuning the hyper-parameter in ADAM optimizer," *Appl. Intell.* **56**, 185 (2026).
- ⁵³A. Haar, "Zur Theorie der orthogonalen Funktionensysteme," *Math. Ann.* **69**, 331 (1910).
- ⁵⁴T. Guo, T. Zhang, E. Lim, M. Lopez-Benitez, F. Ma, and L. Yu, "A review of wavelet analysis and its applications: Challenges and opportunities," *IEEE Access* **10**, 58869 (2022).
- ⁵⁵Y. Dong, G. Li, Y. Tao, X. Jiang, K. Zhang, J. Li, J. Deng, J. Su, J. Zhang, and J. Xu, [arXiv:2410.02675](https://arxiv.org/abs/2410.02675) (2024).
- ⁵⁶M. Lee, "Mathematical analysis and performance evaluation of the GELU activation function in deep learning," *J. Math.* **2023**, 4229924.
- ⁵⁷A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, in *Advances in Neural Information Processing Systems* **32**, 2019.
- ⁵⁸L. Lu, X. Meng, Z. Mao, and G. E. Karniadakis, "DeepXDE: A deep learning library for solving differential equations," *SIAM Rev.* **63**, 208 (2021).
- ⁵⁹Y. Chen, X. Liu, Z. Chai, and B. Shi, "A Cole–Hopf transformation based fourth-order multiple-relaxation-time lattice Boltzmann model for the coupled Burgers' equations," *J. Sci. Comput.* **103**, 25 (2025).
- ⁶⁰F. J. Hickernell, N. Kirk, and A. G. Sorokin, "Quasi-Monte Carlo methods: What, why, and how?," [arXiv:2502.03644](https://arxiv.org/abs/2502.03644) (2025).
- ⁶¹B. Huang, G. Yang, J. Lei, and X. Wang, "A partitioned conditioned Latin hypercube sampling method considering spatial heterogeneity in digital soil mapping," *Sci. Rep.* **15**, 12851 (2025).
- ⁶²B. C. Yen, "Dimensionally homogeneous Manning's formula," *J. Hydraul. Eng.* **118**, 1326 (1992).
- ⁶³N. M. Hunter, M. S. Horritt, P. D. Bates, M. D. Wilson, and M. G. Werner, "An adaptive time step solution for raster-based storage cell modelling of floodplain inundation," *Adv. Water Resour.* **28**, 975 (2005).
- ⁶⁴J. Schmidt-Hieber, "The Kolmogorov–Arnold representation theorem revisited," *Neural Networks* **137**, 119 (2021).
- ⁶⁵Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, [arXiv:2010.08895](https://arxiv.org/abs/2010.08895) (2020).
- ⁶⁶S. Li (2026). "Physics-informed wavelet–Fourier model for multiscale nonlinear PDEs," GitHub. <https://github.com/OneMoreSpeed/PIWF>