Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Fourier Double Gate Physics-Informed Neural Networks for Multiscale PDE Solutions and Spatiotemporal Field Modeling

Guangzheng Zhu
University of Shanghai for Science and Technology
Changgui Gu
Gu_changgui@163.com

University of Shanghai for Science and Technology
Hailing Wang
Yunnan Normal University

Research Article

Keywords: Physics-informed neural networks, Multiscale Fourier features, Double Gate network, Adaptive loss weighting, Mixed-frequency PDEs, Spatiotemporal temperature fields

Posted Date: September 4th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10648460/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Fourier Double Gate Physics-Informed Neural Networks for Multiscale PDE Solutions and Spatiotemporal Field Modeling

Guangzheng Zhu · **Changgui Gu** · **Hailing Wang** ·

Received: date / Accepted: date

Abstract Physics-Informed Neural Networks (PINNs) provide an effective framework for forward and inverse problems by jointly optimizing the residuals of partial differential equations together with initial and boundary conditions. However, standard multilayer perceptrons exhibit a pronounced spectral bias, and the different loss terms in PINNs often suffer from gradient and scale imbalance. As a result, PINNs may converge slowly or inadequately satisfy local constraints in mixed-frequency and multiscale problems. To address these issues, this paper combines multiscale Fourier feature encoding, a Double Gate network architecture, and adaptive loss weighting to construct a Fourier Double Gate PINN (FDG-PINN). Experiments on mixed-frequency partial differential equations (PDEs) demonstrate that FDG-PINN achieves lower relative errors under a unified frequency configuration. The FDG-PINN framework is subsequently applied to real-world temperature-field reconstruction and temporal extrapolation over northern South Asia and North China to evaluate its ability to reconstruct and extrapolate temperature fields characterized by complex nonlinear spatiotemporal dynamics.

G. Zhu
Business School, University of Shanghai for Science and Technology, Shanghai, 200093, China.
E-mail: 241170106@st.usst.edu.cn

C. Gu (✉)
Business School, University of Shanghai for Science and Technology, Shanghai, 200093, China.
E-mail: Gu_changgui@163.com

H. Wang
School of Mathematics, Yunnan Normal University, Kunming, 650500, China
E-mail: wanghl@ynnu.edu.cn

Keywords Physics-informed neural networks · Multiscale Fourier features · Double Gate network · Adaptive loss weighting · Mixed-frequency PDEs · Spatiotemporal temperature fields

1 Introduction

Physics-Informed Neural Networks (PINNs), systematically introduced by Raissi et al., represent the unknown solution with a neural network and use automatic differentiation to evaluate equation residuals, thereby incorporating the PDE, initial conditions (ICs), and boundary conditions (BCs) into a unified loss function [1]. The appeal of this framework lies in its direct formulation: once the governing equation is available and collocation points can be sampled, a continuous function approximating the solution can be trained. Compared with finite difference, finite element, or spectral methods, PINNs do not require a prescribed mesh and are therefore convenient for high-dimensional problems, irregularly sampled data, and settings where limited observations are combined with physical equations. Existing studies have demonstrated the use of PINNs in solving forward problems governed by the Burgers, Schrödinger, Poisson, and Navier–Stokes equations, as well as in parameter identification and inverse problems [1–3]. Subsequent developments have extended the framework to high-order differential operators and PDE-based reliability analysis [4, 5]. PINNs have also begun to find their way into practical engineering systems. For example, they have been incorporated into adaptive optimal control for automated surface vessels [6] and coupled with conventional heat-transfer models for real-time wellbore-temperature prediction in deep shale-gas drilling [7]. These developments have broadened the role of PINNs from equation solvers to a more general scientific machine-learning framework in which observational data, governing equations,

and prior constraints can be incorporated into a unified optimization problem [8].

A large body of work has attempted to improve PINNs, with different studies emphasizing different aspects of the method. Some approaches combine PINNs with temporal models such as LSTM and Echo State Networks to better represent time-evolving states and history dependence [9–11]; others introduce Bayesian neural networks, clustering ideas, neural architecture search, data-driven decomposition, or conservation constraints to address uncertainty, local heterogeneity, and structural selection [12–16]. Another line of research modifies the network backbone through residual connections, attention residual modules, or gating mechanisms to improve feature propagation [17, 18]. In recent years, frequency encodings have also been introduced into PINNs. Multiscale positional encoding in NeRF and Fourier feature networks show that suitable sine and cosine features can reduce the preference of ordinary MLPs for low-frequency functions; this property is particularly important for high-frequency and multiscale PDE solutions [19–22].

Nevertheless, these extensions do not remove the fundamental training difficulties of PINNs. One direct reason is that the PINN objective is inherently multi-objective: the PDE residual, initial-condition error, and boundary-condition error may differ substantially in numerical scale and gradient direction. Studies on gradient pathologies in PINNs have shown that such composite losses can lead to gradient imbalance and training stiffness, where some loss terms dominate the updates for long periods while others receive insufficient optimization [23]. From a frequency perspective, ordinary MLPs also tend to fit low-frequency components first, whereas high-frequency components usually converge more slowly [20, 22]. Therefore, for high-frequency propagation, multiscale diffusion, or strongly constrained boundary-value problems, a standard PINN may still converge to a solution with seemingly moderate residuals but inaccurate global phase or boundary behavior, even after long training. Existing studies have attempted to mitigate these issues through enhanced network architectures, gradient-based loss balancing, and uncertainty-based adaptive weighting [24–29].

Motivated by the limited ability of PINNs to represent mixed-frequency components, the slow convergence toward high-frequency features, and the imbalance among competing loss terms, this study develops a Fourier Double Gate Physics-Informed Neural Network (FDG-PINN). Specifically, deterministic multiscale Fourier encoding is introduced at the input to explicitly represent solution components at different frequencies; a Double Gate structure is incorporated into the network backbone to enhance nonlinear feature propagation; and adaptive loss weighting is employed to balance the contributions of the governing equation, initial conditions, and boundary conditions during training. Through the integration of these components, FDG-PINN aims to accurately capture coexisting mixed-frequency structures in both PDE solutions and spatiotemporal fields, while alleviating the problem that a seemingly small aggregate training loss may still correspond to an inaccurate solution because of severe imbalance among the individual loss terms.

2 Preliminaries

2.1 Physics-Informed Neural Networks

Physics-informed neural networks (PINNs) incorporate governing physical laws into the training of deep neural networks, providing a mesh-free approach for approximating solutions to partial differential equations (PDEs). In the standard PINN formulation, physical knowledge is introduced through the residuals of the governing equations and the associated initial and boundary conditions, which are jointly incorporated into the optimization objective. Consequently, the learned solution is encouraged to satisfy the prescribed physical constraints throughout the computational domain.

Consider a partial differential equation defined on a computational domain Ω:

$$\mathcal{N}[u](x) = f(x), \qquad x \in \Omega,$$

subject to

$$\mathcal{B}[u](x) = g(x), \qquad x \in \partial\Omega,$$

where $u(x)$ is the unknown solution, $\mathcal{N}[\cdot]$ denotes the differential operator associated with the governing equation, and $\mathcal{B}[\cdot]$ represents the prescribed conditions on the boundary of the computational domain. These conditions may be of Dirichlet, Neumann, Robin, or periodic type. For a time-dependent problem, the temporal coordinate is included as one component of x, and the initial condition can be treated as a constraint on the boundary of the resulting spatiotemporal domain.

In a PINN, the unknown solution is approximated by a neural network $u_\theta(x)$, where θ denotes the trainable weights and biases. The required derivatives of u_θ with respect to its inputs are evaluated through automatic differentiation. The residual of the governing equation is then written as

$$r_\theta(x) = \mathcal{N}[u_\theta](x) - f(x).$$

Let $\{x_r^i\}_{i=1}^{N_r}$ denote the collocation points sampled in the computational domain. The residual loss is defined by

$$\mathcal{L}_r(\theta) = \frac{1}{N_r}\sum_{i=1}^{N_r} \left| r_\theta\left(x_r^i\right)\right|^2.$$

Similarly, let $\{x_b^i\}_{i=1}^{N_b}$ denote the points at which the initial or boundary conditions are imposed. The corresponding loss is

$$\mathcal{L}_b(\theta) = \frac{1}{N_b}\sum_{i=1}^{N_b}\left|\mathcal{B}\left[u_\theta\right]\left(x_b^i\right) - g\left(x_b^i\right)\right|^2.$$

When additional observations of the solution are available, they can be included through a data loss

$$\mathcal{L}_d(\theta) = \frac{1}{N_d}\sum_{i=1}^{N_d}\left|u_\theta\left(x_d^i\right) - u_d^i\right|^2,$$

where $\{(x_d^i, u_d^i)\}_{i=1}^{N_d}$ denotes the available observations. This term is omitted when no additional solution data are used.

The network is trained by minimizing the composite objective

$$\mathcal{L}(\theta) = \lambda_r\mathcal{L}_r(\theta) + \lambda_b\mathcal{L}_b(\theta) + \lambda_d\mathcal{L}_d(\theta),$$

where λ_r, λ_b, and λ_d determine the relative contributions of the different loss terms. If no observational data are included, the last term is removed. Minimizing this objective encourages the network prediction to satisfy the governing equation together with the prescribed initial and boundary conditions over the computational domain.

2.2 Training strategy

This section presents two key components of the training strategy: the optimization procedure and the loss-weighting scheme.

Following common practice in PINN training, the proposed network is optimized using a two-stage procedure that combines Adam and L-BFGS. Adam [30] is based on stochastic gradient descent and adaptively adjusts the learning rate using first- and second-moment estimates. Its rapid convergence and relatively strong global search capability allow the model parameters to move quickly toward a favorable region during the early stage of training. L-BFGS [31], a quasi-Newton method, subsequently uses historical gradient information to approximate the Hessian matrix. It typically provides higher local optimization accuracy and faster local convergence, making it suitable for refining the model parameters and obtaining more accurate numerical solutions. Owing to the complementary characteristics of the two optimizers, Adam is used for rapid initial optimization, followed by L-BFGS for further refinement of the solution.

The weighting schemes used in PINN training can be broadly classified into fixed and adaptive approaches, and their selection can substantially affect training efficiency and generalization. Fixed loss weights are straightforward to implement; however, the PDE residual, initial-condition error, and boundary-condition error may differ considerably in magnitude across different problems. For problems with relatively simple governing equations and training objectives, fixed loss weights provide a computationally efficient choice. As the governing equations and application scenarios become more challenging, adaptive loss weighting is generally more effective in balancing competing loss terms and improving overall model performance, albeit at additional computational cost. Since manually tuned loss weights are typically problem-dependent, the development of adaptive weighting methods has become an important direction in PINN optimization. Existing approaches include loss balancing, adaptive physical constraints, and uncertainty-based weighting [25–29]. The method adopted here employs a learnable adaptive weighting scheme in which each active loss term is associated with a trainable parameter that is mapped to a positive weight through an exponential transformation. During the Adam stage, the network parameters and weight parameters are updated alternately. During the L-BFGS stage, the weight parameters are frozen, and only the network parameters are optimized.

3 Methodology

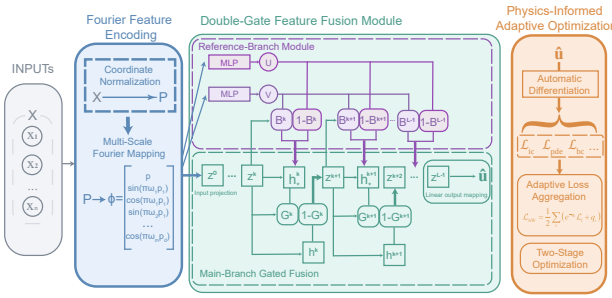


Fig. 1 Fourier Double Gate Physics-Informed Neural Network framework. The normalized input coordinates are first transformed using multiscale Fourier feature encoding. The resulting features are subsequently propagated through the Double Gate network to obtain frequency-enriched representations of the solution. During training, adaptive loss weighting is employed to balance the contributions of different loss terms.

Although Physics-Informed Neural Networks (PINNs) have been widely applied to forward and inverse PDE problems, their practical performance is still influenced by several challenges during network representation and optimization. On one hand, due to the inherent spectral bias of multilayer perceptrons, solution components with different frequency characteristics are often learned at inconsistent rates, resulting in slow convergence for PDEs containing rich multiscale information. On the other hand, the expressive capability of conventional fully connected networks may not be sufficiently flexible to accommodate the diverse characteristics of different PDEs and practical scientific computing tasks. Furthermore, the multiple physical constraints in-

volved in PINNs usually exhibit significantly different convergence behaviors during training, making the optimization process sensitive to manually selected loss weights.

To address the above issues from different perspectives, this paper proposes the Fourier Double Gate Physics-Informed Neural Network (FDG-PINN) framework. Rather than relying on a single architectural modification, FDG-PINN combines three complementary components to improve both representation capability and optimization efficiency. Inspired by the positional encoding strategy adopted in NeRF [19], multiscale Fourier feature encoding is introduced to enrich the spectral representation of normalized input coordinates, thereby improving the learning efficiency of solution components with different frequency characteristics. Subsequently, a lightweight Double Gate architecture is designed to dynamically regulate feature propagation and interaction within the network, enabling more effective feature extraction and stronger generalization across different PDE problems. Finally, adaptive loss weighting is employed during optimization to automatically balance multiple physical constraints with different convergence rates, resulting in more stable optimization and improved solution accuracy.

The overall architecture of FDG-PINN is illustrated in Fig. 1. The normalized spatiotemporal coordinates are first transformed into multiscale Fourier representations before being propagated through the proposed Double Gate network. The predicted solution is then constrained by the governing equations together with the corresponding initial and boundary conditions, while adaptive loss weighting dynamically adjusts the relative importance of different optimization objectives throughout the training process.

3.1 Fourier feature module

The Fourier features used in this paper are similar to the multiscale positional encoding in NeRF, but with several deliberate differences. The input coordinates are first normalized and then expanded using sine and cosine functions with a fixed set of frequencies. The normalized coordinates are concatenated with their Fourier expansions to form the new input. The motivation is straightforward: when the target solution contains oscillatory structures, the Fourier components provide explicit frequency information, reducing the need for the hidden layers to learn these patterns entirely from scratch. When the solution is dominated by smooth or non-oscillatory structures, retaining the normalized coordinates provides a direct representation of the original spatial and temporal variables, thereby avoiding unnecessary reliance on the Fourier-expanded features. Classical Fourier feature networks often use a random matrix B_{rand} to generate random features [20]. Although randomly sampled frequencies may offer greater flexibility, PDE solutions can be highly sensitive to the selected frequency scale, and random sampling introduces an additional source of variability that is difficult to control in ablation studies. Therefore, a deterministic frequency set is adopted to improve experimental stability and reproducibility, albeit at the cost of some flexibility.

Let the input coordinate be denoted by $x = (x_1, \ldots, x_d)$. Each coordinate is first normalized according to the lower and upper bounds of the computational domain:

$$p_j = 2\frac{x_j - a_j}{b_j - a_j} - 1, \qquad j = 1, \ldots, d,$$

where $[a_j, b_j]$ denotes the value range of the j-th input dimension and $p = (p_1, \ldots, p_d)$ is the normalized coordinate projection. Given the frequency set $\mathcal{W} = \{\omega_1, \ldots, \omega_m\}$, the Fourier encoding is defined as

$$\begin{aligned}\phi = \gamma(p) = \big[&p, \sin(\pi\omega_1 p_1), \cos(\pi\omega_1 p_1), \ldots, \\ &\sin(\pi\omega_m p_d), \cos(\pi\omega_m p_d)\big].\end{aligned} \tag{1}$$

In Eq. (1), this paper uses $\mathcal{W} = \{1, 2, 4, 8\}$ by default. For some examples with clearly known frequencies, the corresponding frequencies can be added without changing the main method. The output dimension of this encoding is $d(1 + 2m)$, and the encoded vector is then fed into the Double Gate backbone. Compared with random Fourier features, fixed-frequency encoding is easier to reproduce and is more suitable for controlled comparisons across multiple PDEs.

3.2 Double Gate architecture

The Double Gate architecture is inspired by residual learning and gated feature-transformation architectures [17, 23]. ResNet highlights the importance of effective feature-propagation pathways in deep-network optimization [17], while attention-based residual structures demonstrate that learnable channel-wise modulation can improve feature transmission [18]. Rather than merely increasing network width, the proposed backbone introduces two continuous, channel-wise gates in each Double Gate layer. The first gate dynamically interpolates between two reference feature branches generated from the Fourier-encoded input. The second gate then fuses the ordinary candidate update with this gated reference feature. Together, the two gates adaptively regulate the contribution of each feature pathway according to the encoded input and the current hidden state.

Let $\phi = \gamma(p) \in \mathbb{R}^{d_e}$ denote the Fourier-encoded input. Before the layer-wise gated updates, the network computes two reference feature branches using two independent feed-forward networks:

$$U = F_U(\phi; \theta_U), \qquad V = F_V(\phi; \theta_V),$$

where F_U and F_V have independent trainable parameters θ_U and θ_V. In the implementation, each branch is a three-layer multilayer perceptron: the first two linear layers are followed

by tanh, whereas the final linear layer has no activation. Both branches map the encoded input to the hidden width, so $U, V \in \mathbb{R}^h$. For a given input, U and V are computed once and shared by all Double Gate layers within the same forward pass. Their values are nevertheless not fixed during training, because θ_U and θ_V are optimized by backpropagation. In parallel, the initial hidden state is obtained through a separate input projection:

$$z^0 = \mathcal{F}_{\text{in}}(W_{\text{in}}\phi + b_{\text{in}}),$$

where $W_{\text{in}} \in \mathbb{R}^{h\times d_e}$ and $b_{\text{in}} \in \mathbb{R}^h$ are trainable. Here, $\mathcal{F}_{\text{in}}(\cdot)$ denotes the activation function of the input projection and is set to the hyperbolic tangent function (tanh) in the implementation. The state z^0 is the input to the first Double Gate layer, rather than its final output. For the k-th Double Gate layer, the ordinary candidate feature is

$$h^k = \mathcal{F}_k\left(W_h^k z^k + b_h^k\right).$$

Here, $\mathcal{F}_k(\cdot)$ denotes the activation function of the ordinary hidden mapping in layer k, which is set to tanh in the implementation. The subscript k identifies the layer to which the activation is applied; the same activation type is used in all layers. Moreover, z^k is the hidden state entering layer k, whereas h^k is only the ordinary candidate update generated inside that layer. The layer output is z^{k+1}, defined below. The parameters $W_h^k \in \mathbb{R}^{h\times h}$ and $b_h^k \in \mathbb{R}^h$ are not shared across layers. The first gate uses the current hidden state to perform channel-wise interpolation between the two shared reference branches:

$$\begin{aligned} B^k &= \mathcal{S}_k\left(W_B^k z^k + b_B^k\right), \\ h_*^k &= \left(1 - B^k\right) \odot U + B^k \odot V. \end{aligned} \tag{2}$$

In Eq. (2), $\mathcal{S}_k(\cdot)$ denotes the gate activation function in layer k and is set to the sigmoid function in the implementation ($k = 0, \ldots, L-2$.) Thus, every component of B^k lies in $(0,1)$. Here, B^k denotes the branch-interpolation gate, and the parameters $W_B^k \in \mathbb{R}^{h\times h}$ and $b_B^k \in \mathbb{R}^h$ are specific to layer k. The symbol $\odot$ denotes element-wise multiplication, and 1 is the all-ones vector. Consequently, h_*^k is a continuous channel-wise mixture of U and V. Although the same U and V are reused in all Double Gate layers during one forward pass, the layer-specific gate B^k generally produces a different h_*^k at each layer. The second gate continuously fuses the ordinary candidate feature with the gated reference feature:

$$\begin{aligned} G^k &= \mathcal{S}_k\left(W_G^k z^k + b_G^k\right), \\ z^{k+1} &= \left(1 - G^k\right) \odot h^k + G^k \odot h_*^k, \end{aligned} \tag{3}$$

In Eq. (3), $W_G^k \in \mathbb{R}^{h\times h}$ and $b_G^k \in \mathbb{R}^h$ are also layer-specific. Here, G^k is termed the main reference fusion gate, as it controls the channel-wise fusion between the ordinary hidden-layer candidate h^k and the gated reference feature h_*^k. If a component of G^k approaches zero, the corresponding component of the next state is dominated by h^k; if it approaches one, the component is dominated by h_*^k. Intermediate values provide a continuous blend of the two feature paths. Therefore, z^{k+1}, rather than h^k, constitutes the final output of the k-th Double Gate layer. After the $L-1$ Double Gate updates, the network prediction is produced by a linear output layer:

$$\widehat{u}_\theta(x) = W_{\text{out}} z^{L-1} + b_{\text{out}}.$$

Here, L denotes the hidden-layer count used in the implementation, including the initial input projection; consequently, the loop performs $L-1$ Double Gate updates. The output parameters satisfy $W_{\text{out}} \in \mathbb{R}^{d_{\text{out}}\times h}$ and $b_{\text{out}} \in \mathbb{R}^{d_{\text{out}}}$. According to Eq. (1), although the internal network input is $\phi = \gamma(p)$, the prediction is written as a function of the original physical coordinate x because the normalization, Fourier encoding, and gated backbone together constitute the complete mapping $x \mapsto \widehat{u}_\theta(x)$. All linear and multilayer-perceptron parameters in the Double Gate backbone are randomly initialized and jointly updated during Adam and L-BFGS optimization, whereas the prescribed Fourier frequencies remain fixed.

3.3 Adaptive weighting scheme

The uncertainty-based adaptive weighting scheme proposed by Kendall et al. [29] is adopted and reformulated here in a more general form. Rather than assuming that every equation simultaneously contains PDE, IC, and BC terms, the weights are constructed automatically according to the active loss terms. Suppose that the current equation contains K active loss terms, denoted as

$$\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_K.$$

These loss terms typically arise from the PDE residual, initial condition, and boundary condition. For each term, we introduce a trainable parameter q_i and define the corresponding positive weight as

$$w_i = e^{-q_i}, \qquad i = 1, \ldots, K.$$

The total loss is written as

$$\mathcal{L}_{AW}(\theta, q) = \frac{1}{2}\sum_{i=1}^{K} e^{-q_i}\mathcal{L}_i(\theta) + \frac{1}{2}\sum_{i=1}^{K} q_i. \tag{4}$$

The second term in Eq. (4) penalizes the unbounded growth of the learnable parameters q_i, thereby preventing the corresponding loss weights e^{-q_i} from collapsing toward zero. During the Adam stage, the network parameters θ and the learnable weight parameters q are updated alternately: q is updated while θ is held fixed. Upon entering the L-BFGS stage, q is frozen to avoid further variation in the loss weights during quasi-Newton optimization, and only the network parameters are optimized. This training strategy is

straightforward to implement and enables fair comparisons with the fixed-weight PINN, DG-PINN, and FDG-PINN baselines.

4 Numerical experiments

This section applies FDG-PINN to several representative PDE problems, systematically evaluates its solution accuracy, and uses ablation experiments to examine the roles of Fourier features and the Double Gate architecture in high-frequency, multiscale, and strongly constrained initial-boundary-value problems.

To ensure comparability among different methods, all benchmark experiments in this section use a unified default setting. Specifically, all benchmark PDE experiments use the same five random seeds, network width 96, hidden depth 4, and the tanh activation function. Training first uses Adam for 2000 preliminary iterations with a network learning rate of 1.0×10^{-3}, followed by L-BFGS refinement for 1000 steps with learning rate 0.5, history size 50, and strong Wolfe line search. For sampling, the number of interior collocation points is set to $N_f = 10000$ by default. If the equation contains initial or boundary conditions, the numbers of initial and boundary points are set to $N_i = 2000$ and $N_b = 2000$, respectively, and all points are sampled by Latin Hypercube Sampling (LHS). All Fourier-encoded variants use the Fourier frequency set $\{1,2,4,8\}$, float64 precision, a gradient clipping threshold of 100.0, and weight decay 1.0×10^{-6}. If a specific equation requires an exception because of its input dimension or boundary form, it is stated in the corresponding subsection; otherwise, the remaining hyperparameters are kept unchanged.

The error metrics in the tables are evaluated on two types of point sets. The column L_2(training) is computed on the union of the training collocation points and the prescribed initial and boundary points, after removing duplicate coordinates. In contrast, Global L_2 and Discrete L_2 are computed on an independent uniform evaluation grid rather than on the training samples. For one-dimensional time-dependent equations, this grid contains 600 spatial points and 10 uniformly spaced time levels over the full computational time interval. Discrete L_2 is first computed on each time slice and then averaged over the 10 slices, whereas Global L_2 is computed over all space–time evaluation points at once. For steady two-dimensional Helmholtz equations, the evaluation grid is a 140×140 uniform spatial grid; in this case, Discrete L_2 and Global L_2 are computed on the same grid.

To avoid additional bias caused by manually selected frequencies, we do not tune the Fourier frequencies for individual equations. Even when the analytical solution contains higher-frequency components, the default frequency set $\{1,2,4,8\}$ is used throughout.

In addition to the proposed ablation variants, Standard PINN and ϕ-NN are included as reference models throughout this subsection. Standard PINN provides the conventional fully connected baseline, whereas ϕ-NN adopts the improved network backbone proposed in Ref. [24], which enhances information transfer by coupling auxiliary feature transformations with the hidden representations and has demonstrated improved expressive capacity for rapidly varying solutions. It should be noted that the complete method in Ref. [24] further combines automatic differentiation (AD) with numerical differentiation (ND) through a mixed differentiation strategy. To isolate the effect of the network architecture and maintain a consistent derivative-evaluation procedure across all methods, mixed differentiation is not used here; ϕ-NN and all other models compute the required derivatives solely by AD. Consequently, the accuracy reported for ϕ-NN may be lower than that of the complete method in Ref. [24], but it remains a directly comparable structural baseline under the common AD-based setting adopted in this research.

4.1 High-Frequency Transport Equation

We first consider the high-frequency transport equation as

$$u_t + cu_x = s(x,t), \qquad (x,t)\in[0,1]\times[0,1],$$

where the transport velocity is set by default to $c = 0.7$. The analytical solution is constructed using the manufactured-solution approach:

$$\begin{aligned} u(x,t) = {} & \sin\left(2\pi(4x-1.5t)\right) \\ & + 0.35\cos\left(2\pi(8x+0.5t)\right) \\ & + 0.20\sin\left(2\pi(2x+3t)\right). \end{aligned}$$

The source term $s(x,t)$ is obtained by substituting the analytical solution into $u_t + cu_x$. The initial condition is given by $u(x,0)$, and Dirichlet boundary conditions are prescribed by $u(0,t)$ and $u(1,t)$.

Table 1 Quantitative comparison and ablation analysis of FDG-PINN on the one-dimensional high-frequency transport equation

Method	PDE Loss	IC Loss	BC Loss	L_2(training)	Global L_2	Discrete L_2
PINN	2.138×10^{-2} $\pm7.503\times10^{-3}$	8.527×10^{-4} $\pm1.277\times10^{-3}$	9.936×10^{-4} $\pm9.684\times10^{-4}$	2.913×10^{-2} $\pm1.219\times10^{-2}$	2.963×10^{-2} $\pm1.052\times10^{-2}$	2.872×10^{-2} $\pm1.015\times10^{-2}$
ϕ-NN	2.804×10^{-3} $\pm6.831\times10^{-4}$	3.496×10^{-5} $\pm1.828\times10^{-5}$	5.195×10^{-5} $\pm2.347\times10^{-5}$	1.000×10^{-2} $\pm3.365\times10^{-3}$	1.093×10^{-2} $\pm4.030\times10^{-3}$	1.059×10^{-2} $\pm3.679\times10^{-3}$
DG-PINN	3.175×10^{-3} $\pm5.920\times10^{-4}$	4.829×10^{-5} $\pm1.892\times10^{-5}$	6.703×10^{-5} $\pm1.612\times10^{-5}$	1.084×10^{-2} $\pm2.429\times10^{-3}$	1.184×10^{-2} $\pm2.816\times10^{-3}$	1.160×10^{-2} $\pm2.728\times10^{-3}$
FF-PINN	8.336×10^{-4} $\pm1.113\times10^{-4}$	7.112×10^{-6} $\pm3.021\times10^{-6}$	1.455×10^{-5} $\pm6.165\times10^{-6}$	3.808×10^{-3} $\pm6.348\times10^{-4}$	4.036×10^{-3} $\pm6.871\times10^{-4}$	3.980×10^{-3} $\pm6.619\times10^{-4}$
FDG-PINN	$\mathbf{2.593\times10^{-4}}$ $\mathbf{\pm2.686\times10^{-5}}$	$\mathbf{2.474\times10^{-6}}$ $\mathbf{\pm7.089\times10^{-7}}$	$\mathbf{3.863\times10^{-6}}$ $\mathbf{\pm3.107\times10^{-7}}$	$\mathbf{2.419\times10^{-3}}$ $\mathbf{\pm2.924\times10^{-4}}$	$\mathbf{2.640\times10^{-3}}$ $\mathbf{\pm3.814\times10^{-4}}$	$\mathbf{2.592\times10^{-3}}$ $\mathbf{\pm3.719\times10^{-4}}$

Table 1 reports the mean and standard deviation over five random seeds. Standard PINN remains at a relatively large mean Global L_2 error of 2.963×10^{-2}. FF-PINN provides a larger improvement after Fourier features are introduced,

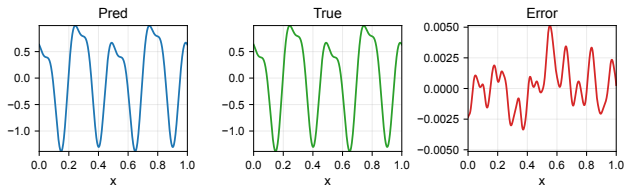


Fig. 2 Predicted solution, analytical solution, and error of FDG-PINN at $t = 0.5$ for the one-dimensional high-frequency transport equation.

while FDG-PINN achieves the lowest mean Global L_2 and Discrete L_2 errors of 2.640×10^{-3} and 2.592×10^{-3}, respectively. The corresponding solution obtained by FDG-PINN for the one-dimensional high-frequency transport equation is shown in Fig. 2.

4.2 Mixed-Frequency Burgers Equation

We next consider a one-dimensional viscous Burgers equation with a source term,

$$u_t + uu_x - \nu u_{xx} = f(x,t), \qquad x \in [-1,1], \quad t \in [0,1],$$

where $\nu = 0.01$ and

$$u(x,t) = e^{-t}\left[\sin(\pi x) + 0.10\sin(8\pi x)\right].$$

The source term, initial condition, and Dirichlet boundary values are generated from the analytical solution.

Table 2 Quantitative comparison and ablation analysis of FDG-PINN on the mixed-frequency Burgers equation

Method	PDE Loss	IC Loss	BC Loss	L_2(training)	Global L_2	Discrete L_2
PINN	9.136×10^{-4} $\pm 5.736 \times 10^{-4}$	4.086×10^{-4} $\pm 1.889 \times 10^{-4}$	3.632×10^{-5} $\pm 2.286 \times 10^{-5}$	3.324×10^{-2} $\pm 7.432 \times 10^{-3}$	3.471×10^{-2} $\pm 7.648 \times 10^{-3}$	3.817×10^{-2} $\pm 8.523 \times 10^{-3}$
ϕ-NN	9.104×10^{-2} $\pm 2.017 \times 10^{-1}$	1.221×10^{-3} $\pm 2.389 \times 10^{-3}$	1.346×10^{-4} $\pm 1.941 \times 10^{-4}$	3.422×10^{-2} $\pm 3.952 \times 10^{-2}$	3.419×10^{-2} $\pm 3.906 \times 10^{-2}$	3.528×10^{-2} $\pm 3.811 \times 10^{-2}$
DG-PINN	8.258×10^{-5} $\pm 1.218 \times 10^{-4}$	7.949×10^{-6} $\pm 1.467 \times 10^{-5}$	2.405×10^{-6} $\pm 3.332 \times 10^{-6}$	3.042×10^{-3} $\pm 3.240 \times 10^{-3}$	3.043×10^{-3} $\pm 3.152 \times 10^{-3}$	3.264×10^{-3} $\pm 3.187 \times 10^{-3}$
FF-PINN	8.165×10^{-6} $\pm 1.032 \times 10^{-6}$	4.721×10^{-7} $\pm 1.283 \times 10^{-7}$	1.419×10^{-7} $\pm 4.528 \times 10^{-8}$	9.040×10^{-4} $\pm 2.219 \times 10^{-4}$	8.676×10^{-4} $\pm 2.775 \times 10^{-4}$	8.891×10^{-4} $\pm 3.626 \times 10^{-4}$
FDG-PINN	$\mathbf{4.923 \times 10^{-6}}$ $\mathbf{\pm 3.987 \times 10^{-7}}$	$\mathbf{1.757 \times 10^{-7}}$ $\mathbf{\pm 4.926 \times 10^{-8}}$	$\mathbf{9.821 \times 10^{-8}}$ $\mathbf{\pm 2.020 \times 10^{-8}}$	$\mathbf{5.382 \times 10^{-4}}$ $\mathbf{\pm 8.324 \times 10^{-5}}$	$\mathbf{4.963 \times 10^{-4}}$ $\mathbf{\pm 9.854 \times 10^{-5}}$	$\mathbf{5.100 \times 10^{-4}}$ $\mathbf{\pm 1.300 \times 10^{-4}}$

As summarized in Table 2, all entries are reported as the mean and standard deviation over the same repeated runs. FF-PINN reduces the mean Global L_2 to 8.676×10^{-4}, and FDG-PINN further reduces it to 4.963×10^{-4}. The corresponding solution obtained by FDG-PINN for the mixed-frequency Burgers equation is shown in Fig. 4.2.

4.3 Two-Dimensional Helmholtz Equation

The Helmholtz equation is commonly used to describe acoustic waves, electromagnetic waves, and steady-state wave phenomena. As a baseline for the later high-wavenumber and mixed-frequency examples, we first consider the following equation on the two-dimensional domain $\Omega = [0,1]^2$:

$$u_{xx} + u_{yy} + k^2 u = f(x,y).$$

Here, the wavenumber is set to $k = 10$, and the analytical solution is constructed as a superposition of two oscillatory components

$$u(x,y) = \sin(8\pi x)\cos(6\pi y) + 0.35\cos(16\pi x)\sin(4\pi y).$$

The source term $f(x,y)$ is obtained by substituting the analytical solution into the equation, and the Dirichlet boundary conditions are prescribed by the exact solution. This example already contains multiple spatial-frequency components and is used to examine the ability of different network structures to represent oscillatory solutions under the standard Helmholtz setting.

Table 3 Quantitative comparison and ablation analysis of FDG-PINN on the two-dimensional Helmholtz equation ($k = 10$)

Method	L_2(training)	Train MSE	Eval MSE	Global L_2	Discrete L_2
PINN	1.014×10^{0} $\pm 9.140 \times 10^{-3}$	2.891×10^{-1} $\pm 5.236 \times 10^{-3}$	2.887×10^{-1} $\pm 5.462 \times 10^{-3}$	1.014×10^{0} $\pm 9.615 \times 10^{-3}$	1.014×10^{0} $\pm 9.615 \times 10^{-3}$
ϕ-NN	1.010×10^{0} $\pm 2.627 \times 10^{-3}$	2.869×10^{-1} $\pm 1.926 \times 10^{-3}$	2.870×10^{-1} $\pm 1.426 \times 10^{-3}$	1.011×10^{0} $\pm 2.512 \times 10^{-3}$	1.011×10^{0} $\pm 2.512 \times 10^{-3}$
DG-PINN	8.192×10^{-1} $\pm 3.146 \times 10^{-1}$	2.111×10^{-1} $\pm 1.142 \times 10^{-1}$	2.178×10^{-1} $\pm 1.111 \times 10^{-1}$	8.378×10^{-1} $\pm 3.049 \times 10^{-1}$	8.378×10^{-1} $\pm 3.049 \times 10^{-1}$
FF-PINN	1.253×10^{-2} $\pm 5.343 \times 10^{-3}$	5.060×10^{-5} $\pm 3.779 \times 10^{-5}$	5.968×10^{-5} $\pm 4.498 \times 10^{-5}$	1.362×10^{-2} $\pm 5.826 \times 10^{-3}$	1.362×10^{-2} $\pm 5.826 \times 10^{-3}$
FDG-PINN	$\mathbf{1.246 \times 10^{-2}}$ $\mathbf{\pm 4.171 \times 10^{-3}}$	$\mathbf{4.756 \times 10^{-5}}$ $\mathbf{\pm 2.655 \times 10^{-5}}$	$\mathbf{5.659 \times 10^{-5}}$ $\mathbf{\pm 3.157 \times 10^{-5}}$	$\mathbf{1.360 \times 10^{-2}}$ $\mathbf{\pm 4.556 \times 10^{-3}}$	$\mathbf{1.360 \times 10^{-2}}$ $\mathbf{\pm 4.556 \times 10^{-3}}$

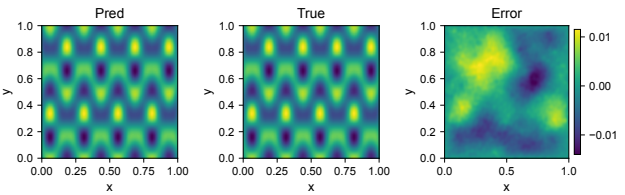


Fig. 3 Predicted field, analytical field, and error distribution of FDG-PINN for the two-dimensional Helmholtz equation ($k = 10$).

The means and standard deviations are presented in Table 3. FF-PINN gives a mean Global L_2 error of 1.362×10^{-2}, while FDG-PINN obtains a slightly lower value of 1.360×10^{-2} with a smaller standard deviation.

These results indicate that Fourier feature encoding substantially improves the ability of PINNs to approximate the solution of this equation. In contrast, the remaining methods without Fourier encoding fail to approximate the solution of the Helmholtz equation. The corresponding field obtained by FDG-PINN for the two-dimensional Helmholtz equation ($k = 10$) is shown in Fig. 3.

To examine the model's behavior at a higher wavenumber, the wavenumber is altered to k=20.

Table 4 Quantitative comparison and ablation analysis of FDG-PINN on the high-wavenumber Helmholtz equation ($k = 20$)

Method	L_2(training)	Train MSE	Eval MSE	Global L_2	Discrete L_2
PINN	1.015×10^{0} $\pm 9.766 \times 10^{-3}$	2.896×10^{-1} $\pm 5.614 \times 10^{-3}$	2.891×10^{-1} $\pm 5.836 \times 10^{-3}$	1.015×10^{0} $\pm 1.025 \times 10^{-2}$	1.015×10^{0} $\pm 1.025 \times 10^{-2}$
ϕ-NN	8.959×10^{-1} $\pm 1.828 \times 10^{-1}$	2.330×10^{-1} $\pm 8.293 \times 10^{-2}$	2.447×10^{-1} $\pm 7.371 \times 10^{-2}$	9.229×10^{-1} $\pm 1.588 \times 10^{-1}$	9.229×10^{-1} $\pm 1.588 \times 10^{-1}$
DG-PINN	1.004×10^{0} $\pm 2.872 \times 10^{-3}$	2.830×10^{-1} $\pm 1.271 \times 10^{-3}$	2.826×10^{-1} $\pm 1.720 \times 10^{-3}$	1.004×10^{0} $\pm 3.074 \times 10^{-3}$	1.004×10^{0} $\pm 3.074 \times 10^{-3}$
FF-PINN	7.281×10^{-3} $\pm 2.404 \times 10^{-3}$	1.618×10^{-5} $\pm 8.985 \times 10^{-6}$	1.911×10^{-5} $\pm 1.060 \times 10^{-5}$	7.916×10^{-3} $\pm 2.608 \times 10^{-3}$	7.916×10^{-3} $\pm 2.608 \times 10^{-3}$
FDG-PINN	$\mathbf{7.099 \times 10^{-3}}$ $\mathbf{\pm 2.801 \times 10^{-3}}$	$\mathbf{1.590 \times 10^{-5}}$ $\mathbf{\pm 1.293 \times 10^{-5}}$	$\mathbf{1.874 \times 10^{-5}}$ $\mathbf{\pm 1.510 \times 10^{-5}}$	$\mathbf{7.716 \times 10^{-3}}$ $\mathbf{\pm 3.006 \times 10^{-3}}$	$\mathbf{7.716 \times 10^{-3}}$ $\mathbf{\pm 3.006 \times 10^{-3}}$

Table 4 provides a summary of the means and standard deviations. FDG-PINN further lowers the mean Global L_2 to 7.716×10^{-3}, compared with 7.916×10^{-3} for FF-PINN. The corresponding field obtained by FDG-PINN for the high-wavenumber Helmholtz equation ($k = 20$) is shown in Fig. 4.

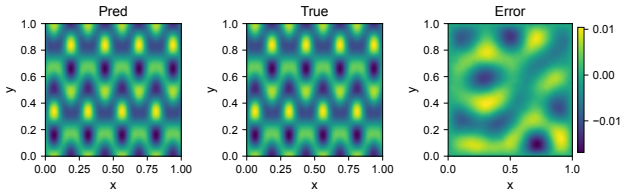


Fig. 4 Predicted field, analytical field, and error distribution of FDG-PINN for the high-wavenumber Helmholtz equation ($k = 20$).

When the equation configuration is altered by changing the wavenumber while keeping the analytical solution unchanged, results similar to those of the preceding experiment are obtained. For the Helmholtz equation with a high-wavenumber term, FDG-PINN slightly outperforms FF-PINN, while models equipped with Fourier feature encoding outperform those without it.

4.4 Mixed-Frequency Helmholtz Equation

For the mixed-frequency Helmholtz problem, we use

$$u_{xx} + u_{yy} + k^2 u = f(x,y), \qquad k = 20,$$

and introduce a 16-fold frequency ratio between the two components of the analytical solution:

$$u(x,y) = \sin(2\pi x)\cos(2\pi y) + 0.10\cos(32\pi x)\sin(2\pi y).$$

The source term and Dirichlet boundary values are prescribed from the exact solution.

Table 5 Quantitative comparison and ablation analysis of FDG-PINN on the mixed-frequency Helmholtz equation

Method	L_2(training)	Train MSE	Eval MSE	Global L_2	Discrete L_2
PINN	2.327×10^{-1} $\pm 3.445 \times 10^{-2}$	1.389×10^{-2} $\pm 3.893 \times 10^{-3}$	1.630×10^{-2} $\pm 4.648 \times 10^{-3}$	2.518×10^{-1} $\pm 3.747 \times 10^{-2}$	2.518×10^{-1} $\pm 3.747 \times 10^{-2}$
ϕ-NN	2.503×10^{-1} $\pm 5.129 \times 10^{-2}$	1.633×10^{-2} $\pm 6.479 \times 10^{-3}$	1.944×10^{-2} $\pm 7.801 \times 10^{-3}$	2.729×10^{-1} $\pm 5.632 \times 10^{-2}$	2.729×10^{-1} $\pm 5.632 \times 10^{-2}$
DG-PINN	1.828×10^{-1} $\pm 2.680 \times 10^{-2}$	8.579×10^{-3} $\pm 2.497 \times 10^{-3}$	1.001×10^{-2} $\pm 2.959 \times 10^{-3}$	1.974×10^{-1} $\pm 2.904 \times 10^{-2}$	1.974×10^{-1} $\pm 2.904 \times 10^{-2}$
FF-PINN	1.790×10^{-2} $\pm 1.085 \times 10^{-2}$	1.052×10^{-4} $\pm 1.373 \times 10^{-4}$	1.247×10^{-4} $\pm 1.627 \times 10^{-4}$	1.951×10^{-2} $\pm 1.189 \times 10^{-2}$	1.951×10^{-2} $\pm 1.189 \times 10^{-2}$
FDG-PINN	$\mathbf{1.360 \times 10^{-2}}$ $\mathbf{\pm 4.016 \times 10^{-3}}$	$\mathbf{4.987 \times 10^{-5}}$ $\mathbf{\pm 2.812 \times 10^{-5}}$	$\mathbf{5.877 \times 10^{-5}}$ $\mathbf{\pm 3.275 \times 10^{-5}}$	$\mathbf{1.476 \times 10^{-2}}$ $\mathbf{\pm 4.299 \times 10^{-3}}$	$\mathbf{1.476 \times 10^{-2}}$ $\mathbf{\pm 4.299 \times 10^{-3}}$

Table 5 provides a summary of the means and standard deviations. FDG-PINN reduces the mean Global L_2 to 1.476×10^{-2} and has a smaller standard deviation than FF-PINN. The corresponding field obtained by FDG-PINN for the mixed-frequency Helmholtz equation is shown in Fig. 5.

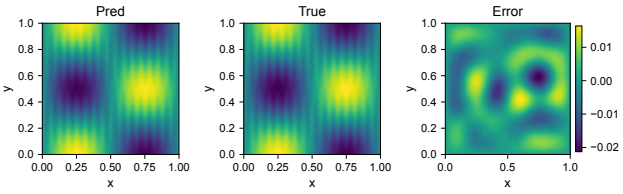


Fig. 5 Predicted field, analytical field, and error distribution of FDG-PINN for the mixed-frequency Helmholtz equation.

As the frequency ratio increases, the coexistence of mixed-frequency components enables FDG-PINN to exhibit a more pronounced advantage over the other models. In particular, the Double Gate architecture reduces the mean relative discrete L_2 error by approximately 24.35% compared with FF-PINN, demonstrating its effectiveness in capturing mixed-frequency solution structures.

4.5 Reaction-Diffusion Equation

We consider the mixed-frequency reaction-diffusion equation

$$u_t - \nu u_{xx} + \lambda u = s(x,t),$$
$$(x,t) \in [0,\pi] \times [0,1],$$

where $\nu = 0.01$, $\lambda = 0.5$, and

$$u(x,t) = e^{-t}\left(\sin x + 0.10\sin(20x)\right).$$

The source term, initial condition, and Dirichlet boundary values are generated from the analytical solution.

Table 6 Quantitative comparison and ablation analysis of FDG-PINN on the mixed-frequency reaction-diffusion equation

Method	PDE Loss	IC Loss	BC Loss	L_2(training)	Global L_2	Discrete L_2
PINN	1.786×10^{-2} $\pm2.608\times10^{-3}$	3.677×10^{-3} $\pm4.841\times10^{-4}$	5.716×10^{-5} $\pm2.039\times10^{-5}$	8.471×10^{-2} $\pm6.290\times10^{-3}$	8.454×10^{-2} $\pm6.336\times10^{-3}$	8.468×10^{-2} $\pm6.543\times10^{-3}$
ϕ-NN	1.862×10^{-2} $\pm4.754\times10^{-3}$	3.817×10^{-3} $\pm8.777\times10^{-4}$	6.005×10^{-5} $\pm2.210\times10^{-5}$	8.633×10^{-2} $\pm9.605\times10^{-3}$	8.630×10^{-2} $\pm9.547\times10^{-3}$	8.679×10^{-2} $\pm9.032\times10^{-3}$
DG-PINN	1.839×10^{-2} $\pm4.331\times10^{-3}$	3.749×10^{-3} $\pm7.524\times10^{-4}$	7.352×10^{-5} $\pm1.850\times10^{-5}$	8.524×10^{-2} $\pm8.326\times10^{-3}$	8.498×10^{-2} $\pm8.360\times10^{-3}$	8.497×10^{-2} $\pm8.258\times10^{-3}$
FF-PINN	1.235×10^{-5} $\pm2.929\times10^{-6}$	4.508×10^{-7} $\pm1.629\times10^{-7}$	1.980×10^{-7} $\pm5.991\times10^{-8}$	7.295×10^{-4} $\pm1.234\times10^{-4}$	6.211×10^{-4} $\pm1.100\times10^{-4}$	5.809×10^{-4} $\pm1.007\times10^{-4}$
FDG-PINN	$\mathbf{4.915\times10^{-6}}$ $\mathbf{\pm6.772\times10^{-7}}$	$\mathbf{1.330\times10^{-7}}$ $\mathbf{\pm3.591\times10^{-8}}$	$\mathbf{1.046\times10^{-7}}$ $\mathbf{\pm4.297\times10^{-8}}$	$\mathbf{4.082\times10^{-4}}$ $\mathbf{\pm5.002\times10^{-5}}$	$\mathbf{3.329\times10^{-4}}$ $\mathbf{\pm3.641\times10^{-5}}$	$\mathbf{3.105\times10^{-4}}$ $\mathbf{\pm3.764\times10^{-5}}$

The means and standard deviations are presented in Table 6. PINN, ϕ-NN, and DG-PINN yield mean relative discrete L_2 errors on the order of 10^{-2}, whereas FF-PINN and FDG-PINN reduce the error to the orders of 10^{-4} and 10^{-5}, respectively. Specifically, FDG-PINN reduces the mean relative discrete L_2 error by approximately 94.65% compared with FF-PINN and achieves an improvement of more than two orders of magnitude over the standard PINN. The corresponding solution obtained by FDG-PINN for the mixed-frequency reaction-diffusion equation is shown in Fig. 6.

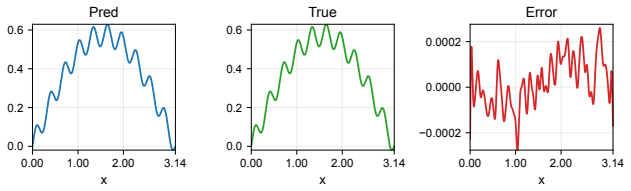


Fig. 6 Predicted solution, analytical solution, and error of FDG-PINN at $t = 0.5$ for the mixed-frequency reaction-diffusion equation.

5 Reconstruction and Prediction of Nonlinear Spatiotemporal Temperature Fields

Temperature fields are among the key physical variables describing the state of the atmospheric environment. Their spatiotemporal evolution is jointly influenced by radiation, convection, advection, turbulence, elevation, topography, and other physical mechanisms, leading to pronounced nonlinearity and rich local variability. Accurate reconstruction of temperature fields is important not only for understanding regional climate variability, but also as a foundation for weather forecasting, climate simulation, and environmental monitoring. Therefore, investigating the feasibility of using PINNs to construct continuous temperature-field models and reconstruct complex meteorological data is of considerable practical significance.

This section applies FDG-PINN to real regional daily mean temperature fields and evaluates its ability to represent nonlinear spatiotemporal fields under sparse observations. The data regions include the northern South Asian subcontinent and the North China Plain together with nearby coastal areas. The experiments first reconstruct the complete temperature field within the training window using the full field on the first day, partial boundary observations on each day, and a small number of interior observations. We then extend the time coordinate beyond the training interval on the northern South Asian dataset to examine temporal extrapolation without using any future observations.

This paper uses the ERA5-Land Hourly dataset provided by the European Centre for Medium-Range Weather Forecasts (ECMWF), which was accessed through the Google Earth Engine platform as the image collection `ECMWF/ERA5_LAND/HOURLY` [32]. The original data have an hourly temporal resolution and a spatial resolution of approximately 0.1° (about 9–11 km). The target variable is air temperature at 2 m above the surface. For each calendar day, we take the arithmetic mean of the 24 hourly temperature fields to reduce short-time-scale fluctuations such as diurnal variability and to emphasize the continuous evolution of the regional temperature field at the daily scale. Inspection of the original sampling grid shows that the northernmost and southernmost latitude rows contain only partial longitude points. Therefore, these two incomplete latitude rows are removed in both regions, and longitude and latitude are then normalized as

$$\hat{x} = \frac{x - x_{\min}}{x_{\max} - x_{\min}}, \qquad \hat{y} = \frac{y - y_{\min}}{y_{\max} - y_{\min}}.$$

The standard PINN baseline uses a four-layer tanh multilayer perceptron with 128 neurons per layer, and fixes the weights of the initial-field, boundary-observation, and interior-observation losses at $1:1:1$. FDG-PINN uses the same hidden width and depth, but introduces Fourier encoding with frequency set $\{1,2,4,8\}$, the Double Gate backbone, and adaptive loss weighting. Both models use float32 precision and the same random seed. Training consists of 2000 Adam steps followed by at most 300 L-BFGS steps. The initial condition uses all valid points on the first day; boundary conditions use a random 20% subset of the boundary points on each day; and interior observations are sampled from 10% of all non-boundary points within the training time window.

In contrast to the preceding PDE benchmark problems with prescribed analytical solutions, the training objective in the real meteorological experiments consists only of the initial-field loss $\mathcal{L}_{\mathrm{IC}}$, boundary-observation loss $\mathcal{L}_{\mathrm{BC}}$, and interior-observation loss $\mathcal{L}_{\mathrm{data}}$; it does not include explicit residuals of atmospheric governing equations. Therefore, this section mainly evaluates the representation, sparse reconstruction, and temporal extrapolation capabilities of

the FDG-PINN backbone and its adaptive loss weighting scheme for real nonlinear spatiotemporal fields.

5.1 Spatiotemporal Temperature-Field Reconstruction and Extrapolation over the Northern South Asian Subcontinent

The northern South Asian dataset covers 20.1672°–28.9707°N and 73.0779°–82.9594°E, mainly including parts of northern India and a small area of Nepal. The time span is May 1–25, 2026. The spatial grid contains 99 latitudes and 111 longitudes, giving 10989 valid points per day and 274725 records over 25 days. The temperature range is −1.933–40.062 °C.

Days 1–20 are used for field reconstruction. All 10989 points on day 1 are used as initial-condition samples. The rectangular grid contains 416 boundary points per day, from which 83 points are sampled, resulting in 1660 boundary samples over 20 days. From all interior points over the 20 days, 10% are randomly sampled, yielding 21146 interior observations. Within the training window, the mean temperature is 32.175 °C and the standard deviation is 3.546 °C. Table 7 reports the reconstruction results of the two models over all spatial points during days 1–20, where the final values of the three loss terms are computed on the standardized temperature scale.

Table 7 Temperature-field reconstruction results for days 1–20 in the northern South Asian region

Method	IC Loss	BC Loss	Data Loss	RMSE (°C)	MAE (°C)	Relative L_2
Standard PINN	2.311×10^{-2}	4.292×10^{-2}	5.417×10^{-2}	0.822	0.628	2.539×10^{-2}
FDG-PINN	2.453×10^{-3}	8.968×10^{-4}	4.491×10^{-3}	0.299	0.210	9.231×10^{-3}

Compared with the standard PINN, FDG-PINN reduces both RMSE and relative L_2 by approximately 63.6%. Fig. 7 shows the predicted field, true field, and absolute error on days 5, 10, and 20. The spatially averaged daily temperature and the daily relative L_2 errors of the two models are shown in Fig. 8(a) and Fig. 8(b), respectively. These results indicate that, under the same data volume, random seed, and optimization budget, Fourier encoding and the Double Gate backbone can reconstruct the continuous spatiotemporal temperature field of this region more accurately.

After completing training on the first 20 days, we keep the FDG-PINN parameters fixed and extrapolate the time coordinate to days 21–25. No future interior or boundary points are added in this stage, so the experiment is a fully future-observation-free temporal extrapolation. The program outputs day 20 as a transition frame at the end of the training window, but the summary metrics in Table 8 are strictly computed only over days 21–25.

The extrapolation error generally increases with temporal distance. The relative L_2 error is 0.0227 on day 21 and increases to 0.0871 and 0.0966 on days 24 and 25, respectively. The model can still maintain the spatial structure learned in the training window during the first two extrapolated days, but amplitude bias gradually accumulates as the extrapolation horizon grows. This indicates a clear validity horizon for temporal extension without future constraints.

Table 8 Future-observation-free extrapolation results of FDG-PINN in the northern South Asian region

Day	MSE (°C^2)	RMSE (°C)	MAE (°C)	Relative L_2
21	0.632	0.795	0.652	0.0227
22	0.837	0.915	0.737	0.0262
23	2.852	1.689	1.348	0.0483
24	9.392	3.065	2.487	0.0871
25	11.584	3.404	2.957	0.0966
Days 21–25	5.060	2.249	1.636	0.0641

5.2 Spatiotemporal Temperature-Field Reconstruction over the North China Plain

The North China region contains irregular land-sea interfaces such as the Bohai Sea and the Yellow Sea, so boundary points cannot be defined solely by normalized coordinates satisfying $x = 0, 1$ or $y = 0, 1$. We first construct a valid land mask from the longitude and latitude coordinates that are actually present each day. For any valid land point, if at least one of its four neighboring grid points above, below, left, and right belongs to the sea or lies outside the rectangular domain, the point is defined as a boundary point. This rule identifies both the rectangular outer boundary and the coastline boundary, while excluding sea-grid cells from training sampling and error evaluation. Fig. 9 shows the valid land region, interior points, and boundary points determined by the four-neighbor rule.

The cropped North China dataset covers 32.1148°–41.9962°N and 112.0648°–121.9463°E, including the North China Plain and surrounding land and coastal regions. The time span is May 1 to June 19, 2026, totaling 50 days. The regular coordinate range forms a 111×111 candidate grid, with 10180 valid land points and 2141 invalid sea-grid points per day, giving 509000 valid records over 50 days. The temperature range is −1.440–30.867 °C. During days 1–25, the mean RMSE between adjacent daily temperature fields is 2.742 °C, with a maximum of 5.555 °C; the corresponding values for the northern South Asian dataset are 0.920 °C and 2.048 °C. This shows that the North China samples exhibit stronger day-to-day variability during this period.

Among the 10180 valid land points each day, 615 are boundary points. During training, 123 boundary points are sampled per day, giving 2460 boundary samples over 20 days. All 10180 land points on day 1 are used as initial-condition samples, and 10% of all non-boundary land points

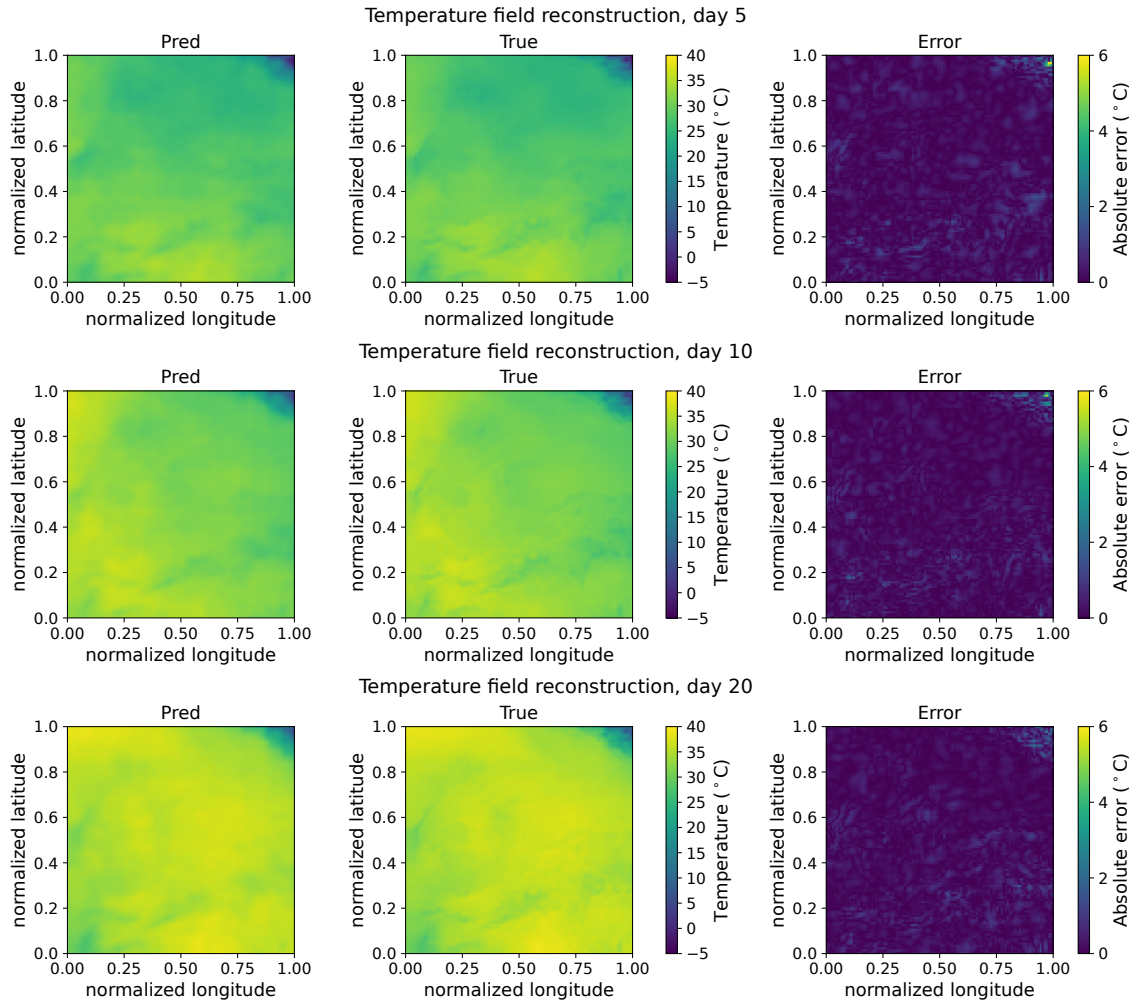


Fig. 7 FDG-PINN temperature-field reconstruction results for days 5, 10, and 20 in northern South Asia. Each row shows the predicted field, true field, and absolute error.

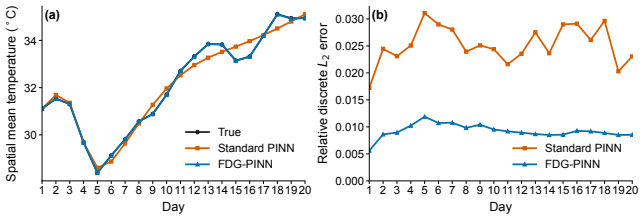


Fig. 8 Daily reconstruction statistics in the northern South Asian region: (a) spatial mean temperature of the true field, standard PINN, and FDG-PINN; (b) daily relative discrete L_2 errors of standard PINN and FDG-PINN.

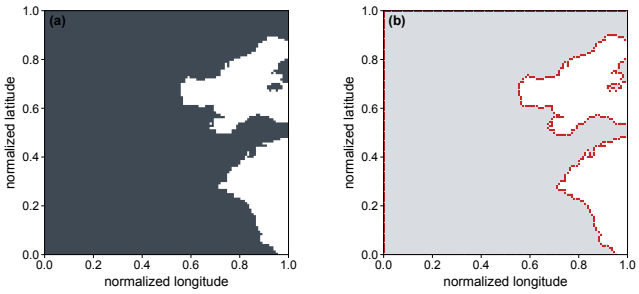


Fig. 9 Sampling-point distribution and boundary-point identification for the North China region. (a) Spatial domain containing all valid land sampling points. (b) Identified boundary sampling points, where a valid land point is classified as a boundary point if any of its four neighboring cells is sea or lies outside the rectangular domain.

over the 20 days are sampled, resulting in 19130 interior observations. Within the training window, the mean temperature is 18.989°C and the standard deviation is 4.700°C. Table 9 reports the reconstruction results of FDG-PINN and the standard PINN.

Compared with the standard PINN, FDG-PINN reduces both RMSE and relative L_2 by approximately 58.2%. In Fig. 10, sea regions are uniformly shown in white and are

Table 9 Temperature-field reconstruction results for days 1–20 in the North China region

Method	IC Loss	BC Loss	Data Loss	RMSE (°C)	MAE (°C)	Relative L_2
Standard PINN	2.521×10^{-2}	3.258×10^{-2}	4.244×10^{-2}	0.991	0.745	5.064×10^{-2}
FDG-PINN	3.390×10^{-3}	6.044×10^{-4}	4.251×10^{-3}	0.414	0.288	2.119×10^{-2}

excluded from error evaluation. Fig. 11(a) and Fig. 11(b) compare the daily spatial mean temperature and relative discrete L_2 errors of the two models, respectively. FDG-PINN maintains a lower daily reconstruction error over all 20 training days, indicating that it can still improve sparse temperature-field reconstruction in a region with irregular coastline boundaries and stronger day-to-day variability.

6 Discussion

For problems with pronounced high-frequency or mixed-frequency structures, the errors of MLP-based PINNs may not be caused merely by an insufficient number of optimization steps; they can also arise from limited representation capacity and competition among loss terms. In the numerical PDE experiments, Standard PINN, ϕ-NN, and DG-PINN still remain at relatively high errors in several Helmholtz and mixed-frequency examples.

After incorporating Fourier features and adaptive loss weighting, FF-PINN already reduces the error substantially, demonstrating the benefits of explicit frequency encoding in alleviating spectral bias and adaptive weighting in balancing the different loss components.

In most benchmark equations considered here, FDG-PINN further improves the mean Global L_2 error or reduces the variability, suggesting that the gain of the proposed method is not produced by a single module alone; rather, it is closer to a joint mechanism in which frequency inputs provide expressive bases and the gated backbone selects and recombines those bases.

It is worth noting that FDG-PINN demonstrates favorable generality across the PDE benchmark problems considered in this research. All benchmarks with prescribed analytical solutions employ the same frequency set $\{1,2,4,8\}$, the same random seed setting, an identical network architecture, and nearly identical optimization procedures. No equation-specific adjustment of the network depth or width is performed, although PDEs with different solution characteristics may benefit from different architectural configurations. Nevertheless, FDG-PINN achieves the lowest mean Global L_2 error in most of the benchmark equations, including the transport equation, elliptic Helmholtz equations, the reaction–diffusion equation, and the nonlinear Burgers equation. These results indicate that the effectiveness of FDG-PINN does not rely on architecture tuning tailored to an individual equation. Instead, under a unified experimental setting, the model exhibits good adaptability to oscillatory, nonlinear, and time-dependent mixed-frequency solutions. Therefore, the generality demonstrated here is primarily reflected in its transferability across different equation structures, rather than solely in achieving a low error for a single manufactured solution.

The real-world temperature-field experiments demonstrate that FDG-PINN can also be applied beyond PDE benchmark problems with prescribed analytical solutions. In the northern South Asian reconstruction experiment, FDG-PINN reduces RMSE from $0.822\,^{\circ}\mathrm{C}$ for the standard PINN to $0.299\,^{\circ}\mathrm{C}$, and reduces relative L_2 from 2.539×10^{-2} to 9.231×10^{-3}. In the North China experiment, the corresponding RMSE decreases from $0.991\,^{\circ}\mathrm{C}$ to $0.414\,^{\circ}\mathrm{C}$, and the relative L_2 decreases from 5.064×10^{-2} to 2.119×10^{-2}. The two regions differ in geographic extent, temperature distribution, and boundary geometry, and the North China experiment also involves an irregular valid domain induced by the land-sea mask. FDG-PINN still improves reconstruction accuracy on these real datasets, indicating that Fourier encoding and the gated backbone can model not only explicit frequency terms in manufactured solutions but also more implicit and irregular spatial variability in real temperature fields.

However, when the task is changed to extrapolative prediction, good reconstruction accuracy does not directly imply strong long-term predictive skill. In the northern South Asian experiment, the overall relative L_2 error for the fully future-observation-free extrapolation on days 21–25 is 0.0641, and the daily error increases as the extrapolation horizon grows. This phenomenon indicates that FDG-PINN has some short-term extrapolation ability, but it does not escape the general difficulty of long-range PINN forecasting. This limitation is partly because the real meteorological experiments in this paper do not include atmospheric dynamics residuals such as advection, diffusion, or energy conservation. In essence, the model here is closer to a spatiotemporal neural field constrained by an initial field, boundary observations, and sparse interior observations. It can provide short-term trend continuation, but reliable longer-range meteorological prediction probably cannot rely only on a single temperature variable and temporal-coordinate extrapolation.

In addition, FDG-PINN incorporates Fourier feature encoding, Double Gate branches, and adaptive loss weighting. Although it improves accuracy by orders of magnitude in several cases compared with the PINN, it also requires more parameters and longer training time; the tables show that FDG-PINN is usually noticeably slower than Standard PINN. Therefore, this paper emphasizes the use of additional computation to obtain more stable representation and lower error in scenarios where PINNs tend to fail, such as high-frequency problems, mixed-frequency problems, and sparse-observation reconstruction. Moreover, although the PDE benchmarks are evaluated under the same random seed setting and unified optimization settings, broader seasonal coverage, more regions, and unified comparisons with additional spatiotemporal forecasting models are still needed to further strengthen the robustness of the evidence. These

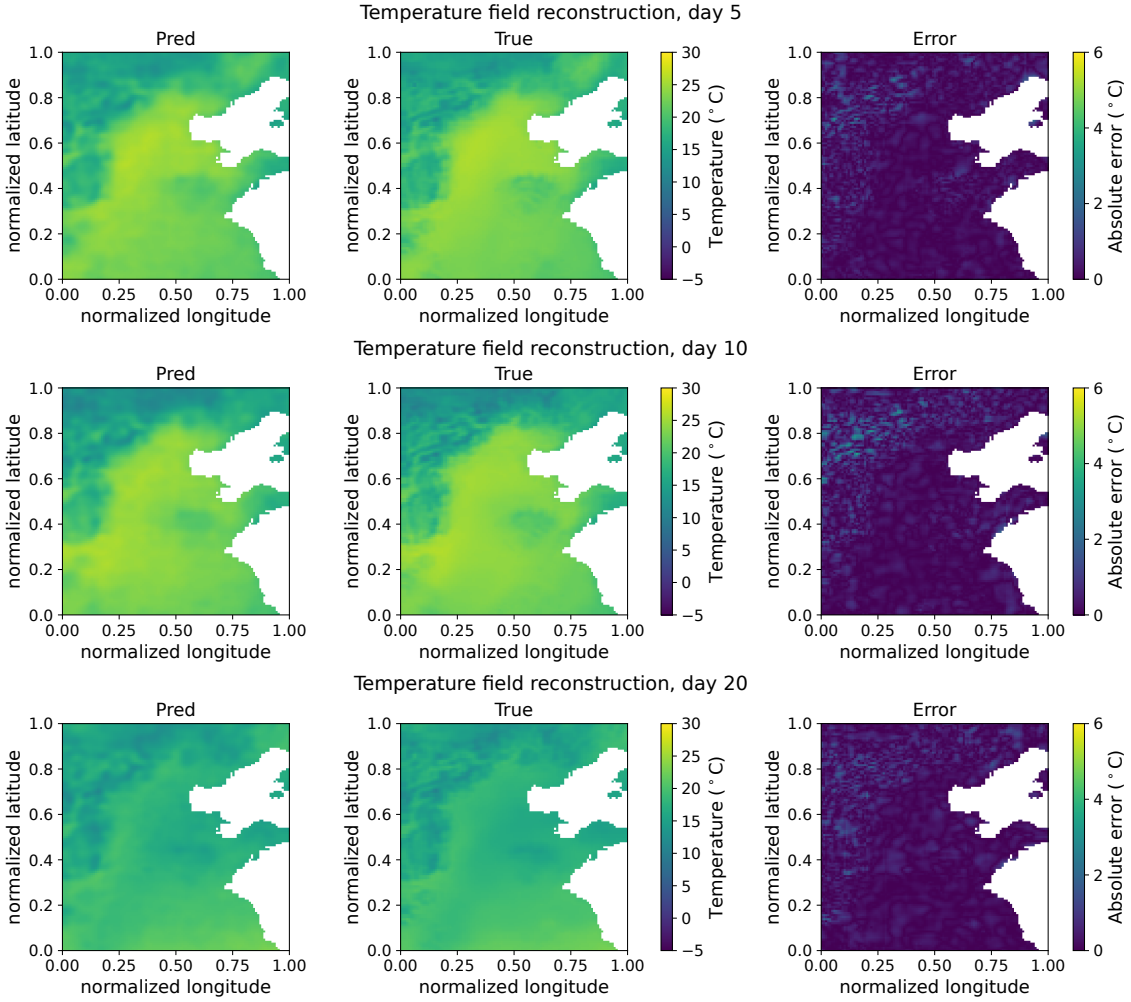


Fig. 10 FDG-PINN temperature-field reconstruction results for days 5, 10, and 20 in the North China region. White areas denote sea regions excluded from training and evaluation.

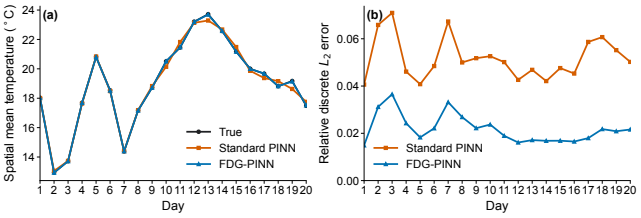


Fig. 11 Daily reconstruction statistics in the North China region: (a) spatial mean temperature of the true field, standard PINN, and FDG-PINN; (b) daily relative discrete L_2 errors of standard PINN and FDG-PINN.

extensions will also place higher demands on computational cost.

7 Conclusions

This paper proposes Fourier Double Gate PINN (FDG-PINN) to address the limitations of standard PINNs in high-frequency representation, multiscale modeling, and imbalanced loss optimization. The method integrates deterministic multiscale Fourier encoding, a Double Gate backbone, and adaptive loss weighting. As a result, the network can explicitly receive multifrequency inputs, dynamically select and combine different features in hidden layers, and automatically balance constraint terms such as PDE residuals, initial conditions, and boundary conditions during training.

The numerical experiments show that FDG-PINN achieves the lowest mean Global L_2 error in most benchmark equations, including the one-dimensional high-frequency transport equation, Helmholtz equations, the mixed-frequency reaction-diffusion equation, and the Burgers equation. The comparison with FF-PINN shows that Fourier features can improve high-frequency representation, while the comparison with DG-PINN shows that a gated structure alone is not sufficient for strongly oscillatory or mixed-frequency solutions. The advantage of FDG-PINN

mainly comes from the coupling of these two components rather than from either module in isolation.

In real temperature-field reconstruction, FDG-PINN also outperforms the standard PINN under the same sampling and optimization settings. For the first 20 days in the northern South Asian region, the RMSE and relative L_2 are $0.299\,^{\circ}\mathrm{C}$ and 9.231×10^{-3}, respectively. For the North China region, the corresponding values are $0.414\,^{\circ}\mathrm{C}$ and 2.119×10^{-2}. These results indicate that the proposed method is applicable not only to analytical manufactured solutions, but also to sparse reconstruction of real nonlinear spatiotemporal data, with a certain degree of adaptability to different regions and irregular valid boundaries.

Overall, FDG-PINN provides a relatively general PINN enhancement framework for high-frequency and mixed-frequency problems. Under a unified random seed setting and a unified frequency setting, its effectiveness is verified on multiple classes of PDEs and real temperature fields. The results also show that combining frequency encoding, gated feature selection, and adaptive loss weighting can substantially improve the representation capacity and training stability of PINNs.

Funding

This work was supported by the National Natural Science Foundation of China (Grant No. 12275179).

Conflict of interest

The authors declare that they have no conflict of interest, and all authors have approved the manuscript and agreed to its submission.

Code availability

The source code and implementation details for FDG-PINN, including the PDE benchmarks and the Spatiotemporal Temperature Fields experiments, are publicly available at https://github.com/zzzzzgz/FDG-PINN.

Data availability

The ERA5-Land Hourly open dataset used in this research is publicly available at https://developers.google.com/earth-engine/datasets/catalog/ECMWF_ERA5_LAND_HOURLY.

References

1. Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.
2. Zhao Chen, Yang Liu, and Hao Sun. Physics-informed learning of governing equations from scarce data. *Nature Communications*, 12(1):6136, 2021.
3. Maziar Raissi, Alireza Yazdani, and George Em Karniadakis. Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science*, 367(6481):1026–1030, 2020.
4. Zhiwei Shi, Zheyuan Hu, Min Lin, et al. Stochastic taylor derivative estimator: Efficient amortization for arbitrary differential operators. *Advances in Neural Information Processing Systems*, 37:122316–122353, 2024.
5. Zhen Meng, Qian Qian, Min Xu, et al. PINN-FORM: A new physics-informed neural network for reliability analysis with partial differential equation. *Computer Methods in Applied Mechanics and Engineering*, 414:116172, 2023.
6. Ying Zhang, Yong Cheng, Shuzhi Sam Ge, et al. Physics-informed neural networks-based adaptive optimized control and its application to automated surface vessels. *IEEE Transactions on Industrial Informatics*, 2025.
7. Jiaohao Xie, Jie Zhang, Yuxin Cheng, Jiazheng Zhou, Xiang Li, Xin Li, and Gezhen Mao. Real-time prediction of wellbore temperatures in deep shale gas drilling using a combination of PINN and heat transfer models. *Applied Thermal Engineering*, 279:127984, 2025.
8. Kai Luo, Jun Zhao, Yifan Wang, et al. Physics-informed neural networks for PDE problems: A comprehensive review. *Artificial Intelligence Review*, 58(10):323, 2025.
9. Rui Zhang, Yang Liu, and Hao Sun. Physics-informed multi-LSTM networks for metamodeling of nonlinear structures. *Computer Methods in Applied Mechanics and Engineering*, 369:113226, 2020.
10. Yifan Zhou, Zhen Tao, Hao Wang, et al. LSTM-PINN for steady-state electrothermal transport: Preserving multi-field consistency in strongly coupled heat and fluid flow. *arXiv preprint arXiv:2604.14201*, 2026.
11. Nguyen Anh Khoa Doan, Wolfgang Polifke, and Luca Magri. Physics-informed echo state networks. *Journal of Computational Science*, 47:101237, 2020.
12. Liu Yang, Xuhui Meng, and George Em Karniadakis. B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data. *Journal of Computational Physics*, 425:109913, 2021.
13. Zhen Liu, Yang Liu, Xing Yan, et al. Automatic network structure discovery of physics informed neural networks via knowledge distillation. *Nature Communications*, 16(1):9558, 2025.
14. Yifan Wang and Linlin Zhong. NAS-PINN: Neural architecture search-guided physics-informed neural network for solving PDEs. *Journal of Computational Physics*, 496:112603, 2024.
15. Jixuan Han, Lian Xue, Yansong Wei, et al. Physics-informed neural network-based petroleum reservoir simulation with sparse data using domain decomposition. *Petroleum Science*, 20(6):3450–3460, 2023.
16. Ameya D Jagtap, Ehsan Kharazmi, and George Em Karniadakis. Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. *Computer Methods in Applied Mechanics and Engineering*, 365:113028, 2020.
17. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.
18. Kimi Team, Guangyu Chen, Yu Zhang, Jianlin Su, Weixin Xu, Siyuan Pan, Yaoyu Wang, Yucheng Wang, Guanduo Chen, Bohong Yin, et al. Attention residuals. *arXiv preprint arXiv:2603.15031*, 2026.
19. Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing

scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.
20. Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33:7537–7547, 2020.
21. Chang Wei, Yuchen Fan, Jian Cheng Wong, Chin Chun Ooi, Heyang Wang, and Pao-Hsiung Chiu. FFV-PINN: A fast physics-informed neural network with simplified finite volume discretization and residual correction. *Computer Methods in Applied Mechanics and Engineering*, 444:118139, 2025.
22. Sifan Wang, Hanwen Wang, and Paris Perdikaris. On the eigenvector bias of fourier feature networks: From regression to solving multi-scale PDEs with physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 384:113938, 2021.
23. Sifan Wang, Yujun Teng, and Paris Perdikaris. Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM Journal on Scientific Computing*, 43(5):A3055–A3081, 2021.
24. Jun Wang, Xiaoliang Xiao, Xiaoping Feng, et al. An improved physics-informed neural network with adaptive weighting and mixed differentiation for solving the incompressible Navier–Stokes equations. *Nonlinear Dynamics*, 112(18):16113, 2024.
25. Zixue Xiang, Wei Peng, Xu Liu, and Wen Yao. Self-adaptive loss balanced physics-informed neural networks. *Neurocomputing*, 496:11–34, 2022.
26. Chen Han, Xuhui He, Zheng Wang, et al. A self-adaptive physics-informed neural network for real-time prediction of structural damage. *Mechanical Systems and Signal Processing*, 239:113322, 2025.
27. Jun Wu, Peng Zhao, Wen Shu, et al. Self-adaptive physics-informed neural networks (SA-PINN) coupled with thermal network model (TNM) for cross-scale temperature field modeling of wet friction components. *Tribology International*, page 111208, 2025.
28. Jun Wu, Peng Zhao, Wen Shu, et al. Self-adaptive physics-informed neural networks for solving the heat transfer problem in wet friction components of transmission system. *International Journal of Thermal Sciences*, 220:110390, 2026.
29. Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7482–7491, 2018.
30. Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.
31. Dong C. Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45(1–3):503–528, 1989.
32. Joaquín Muñoz-Sabater, Emanuel Dutra, Anna Agustí-Panareda, Clément Albergel, Gabriele Arduini, Gianpaolo Balsamo, Souhail Boussetta, Margarita Choulga, Shaun Harrigan, Hans Hersbach, et al. ERA5-Land: A state-of-the-art global reanalysis dataset for land applications. *Earth System Science Data*, 13:4349–4383, 2021.