Multistage physics informed neural network for solving coupled multiphysics problems in material degradation and fluid dynamics

Khadijeh, Mahmoud; Cerqueglini, Veronica; Kasbergen, Cor; Erkens, Sandra; Varveri, Aikaterini

DOI
10.1007/s00366-025-02174-4

Publication date
2025

Document Version
Final published version

Published in
Engineering with Computers

Engineering with Computers
https://doi.org/10.1007/s00366-025-02174-4

ORIGINAL ARTICLE

Multistage physics informed neural network for solving coupled multiphysics problems in material degradation and fluid dynamics

Mahmoud Khadijeh[1] · Veronica Cerqueglini[2] · Cor Kasbergen[1] · Sandra Erkens[1] · Aikaterini Varveri[1]

Received: 17 October 2024 / Accepted: 12 June 2025

Abstract
Physics Informed Neural Networks (PINNs) have been rarely applied to solve multiphysics systems due to the inherent challenges in optimizing their complex loss functions, which typically incorporate multiple physics-based terms. This study presents a multistage PINN approach designed to efficiently solve coupled multiphysics systems with strong interdependencies. The multistage PINN progressively increases the complexity of the physical system being modeled, enabling more effective capture of coupling between different physics. The computational merits of this approach are demonstrated through two illustrative applications: prediction of asphalt aging and modeling of lid-driven cavity flow. Quantitative and qualitative comparisons with standard PINN and adaptive weight PINN approaches demonstrate the enhanced precision and computational efficiency of the proposed algorithm. The multistage PINN achieves a reduction in training time of more than 90% compared to standard PINNs while maintaining better alignment with the finite element method (FEM) solutions. The improvement in computational efficiency, coupled with enhanced accuracy, positions the multistage PINN as a powerful tool for addressing complex multiphysics problems across various engineering disciplines. The method's ability to handle interactions between multiple physical processes, such as diffusion, chemical reactions, and fluid dynamics, makes it suitable for simulating long-term material behavior and complex fluid systems.

Keywords Multiphysics modeling · Multistage learning · Asphalt aging · Physics informed neural networks · Partial differential equations

1 Introduction

1.1 Advancements in multiphysics modeling: a background

Multiphysics systems, characterized by the interaction of multiple physical phenomena, are crucial in science and engineering. These systems often involve coupled partial differential equations (PDEs) that describe complex processes across various scales and domains. Traditional numerical methods for solving such systems, including finite element methods (FEM) and finite difference methods (FDM) can be computationally intensive and may struggle with highly nonlinear or long-term simulations [1].

To address these challenges, techniques such as operator splitting and sequential time-stepping have been widely employed. Operator splitting decomposes complex coupled PDEs into simpler subproblems, which are solved independently or sequentially, a method particularly effective for multiphysics problems like fluid-structure interactions and

✉ Mahmoud Khadijeh
m.khadijeh@tudelft.nl

Veronica Cerqueglini
s298323@studenti.polito.it

Cor Kasbergen
c.kasbergen@tudelft.nl

Sandra Erkens
s.m.j.g.erkens@tudelft.nl

Aikaterini Varveri
a.varveri@tudelft.nl

[1] Department of Engineering Structures, Delft University of Technology, Stevinweg 1, 2628 CN Delft, Netherlands

[2] Department of Environmental, Land and Infrastructure Engineering, Politecnico di Torino, 24 Corso Duca degli Abruzzi, 10129 Turin, Italy

Published online: 29 June 2025

porous media flow [2–4]. Sequential approaches, on the other hand, manage coupling by iterating between physical processes over discrete time steps, as seen in heat transfer and reactive transport simulations [5]. These strategies reduce computational complexity but may introduce errors at the interfaces of subproblems or require careful tuning of time steps, motivating the development of machine learning-based alternatives.

In recent years, physics informed neural networks (PINNs) have gained attention as a promising, mesh-free alternative for solving complex PDEs, including those that govern multiphysics systems. Introduced by Raissi et al. [6], PINNs use the universal approximation capabilities of neural networks while incorporating physical laws directly into the loss function. This approach allows for the simultaneous satisfaction of governing equations and boundary/initial conditions, potentially leading to more accurate and efficient solutions for complex physical systems [7–9].

PINNs face several challenges when applied to complex systems. These include difficulties in balancing multiple loss terms, slow convergence for highly nonlinear problems, and potential instabilities in long-term predictions [10]. Additionally, their training time remains a limitation compared to traditional methods for simpler forward problems. To address these issues, researchers have proposed various innovative strategies.

One such approach involves adaptive weighting schemes, which dynamically adjust the weights of different terms in the loss function during training [11]. Another strategy employs curriculum learning, gradually increasing the complexity of the problem during training [12]. Physics-constrained neural networks represent another advancement that incorporates domain knowledge directly into the network architecture [13, 14]. This approach can improve the stability and accuracy of predictions, especially for long-term simulations. Researchers have also explored domain decomposition techniques, which divide the problem domain into subdomains and train separate networks for each [15]. A more detailed discussion of these strategies, including their mathematical formulations and comparative performances, is presented in section 2.4.

1.2 PINNs in complex systems: applications and current limitations

Recent studies have revealed several limitations in the PINN methodology when applied to complex, coupled systems. For example, in fluid dynamics, PINNs have shown promise in solving coupled advection-diffusion equations and nonlinear Burgers equations [16, 17]. However, these studies revealed challenges in handling high Péclet number flows and capturing shock phenomena accurately. On the other hand, research on turbulent flows using PINNs demonstrated limitations in modeling high Reynolds number scenarios, where the complexity of the flow physics becomes particularly challenging [18, 19].

In materials science and solid mechanics, PINNs have been applied to phase-field fracture dynamics and poroelasticity problems [20]. These applications revealed challenges in handling discontinuities associated with fracture propagation and difficulties in modeling large deformations in porous media. The modeling of coupled thermo-hydro-mechanical processes in porous media using PINNs showcased their potential in complex geosystems but also exposed limitations in dealing with multiscale phenomena [21]. The interaction of processes occurring at different spatial and temporal scales proved challenging for standard PINN formulations.

In the context of materials degradation, several studies have highlighted specific limitations of PINNs. Corrosion modeling using PINNs successfully captured coupled electrochemistry and ion transport but struggled with moving boundary problems associated with corrosion front propagation [22]. Fatigue crack growth predictions using PINNs showed good agreement with experimental data but faced difficulties in accurately representing discontinuities in the solution field [23].

Polymer degradation modeling using advanced machine learning techniques, including PINNs, revealed challenges in accurately predicting complex chemical kinetics involved in the degradation process [24]. Similarly, concrete aging models using PINNs and hybrid physics-ML approaches faced difficulties in bridging multiple scales and capturing long-term behavior accurately [25]. These limitations highlight the need for advanced PINN methodologies that can address the challenges associated with complex, coupled multiphysics systems.

1.3 Objectives and contributions: PINNs for multiphysics modeling

The modeling of coupled multiphysics systems presents significant challenges for existing computational methods. PINNs have shown promise in this area, but still face limitations in handling complex, nonlinear coupled systems efficiently and accurately. These limitations include difficulty in balancing multiple physics-informed loss terms, slow convergence and instability in training for highly coupled systems, and inaccuracies in long-term predictions for evolving material properties.

The primary objective of this study is to address these challenges by developing a multistage PINN approach designed for coupled multiphysics systems with complex interactions, such as those encountered in material

degradation and fluid dynamics. The key contribution of this approach is a staged learning process that gradually incorporates different physical phenomena, enabling more effective handling of coupled multiphysics problems.

To illustrate the broad applicability of the method across various fields, the study examined asphalt aging prediction and lid-driven cavity flow. The first application tackles a complex material degradation problem involving coupled heat transfer, oxygen diffusion, and time-dependent chemical reactions. The second application addresses fluid dynamics, showcasing the method's adaptability in handling diverse physical systems governed by the Navier-Stokes equations. These phenomena are mathematically represented as a system of coupled, two-dimensional PDEs, capturing the spatial and temporal evolution of the material's properties. The multistage PINN approach is compared with other methods, including standard PINNs and adaptive weighting techniques, to highlight the advantages in terms of accuracy and computational efficiency.

2 Theoretical background

2.1 General formulation of coupled PDE systems

Coupled multiphysics problems involve the interaction of multiple physical phenomena, each described by its own set of equations. These interactions can lead to complex behaviors that cannot be captured by considering each phenomenon solely [1]. Coupled multiphysics problems can be generally expressed as a system of N interdependent PDEs [26]:

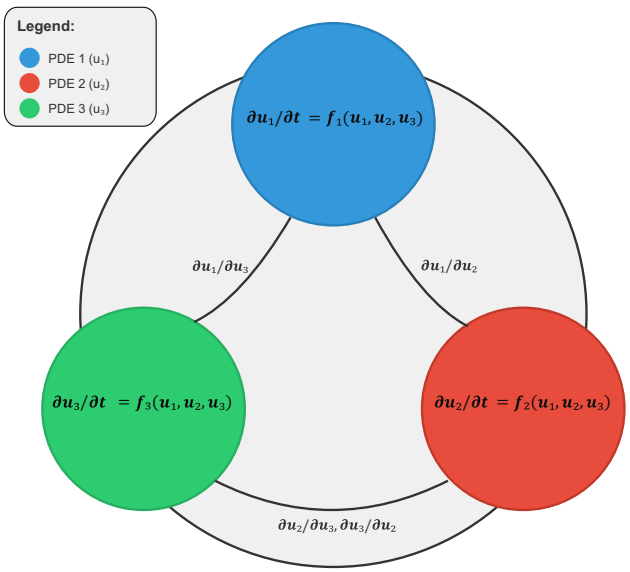

Fig. 1 Visualization of a coupled PDE system, illustrating the interdependencies between equations through shared variables and coupling terms

$$\mathcal{F}_i\left(\mathrm{u}, \frac{\partial \mathrm{u}}{\partial t}, \nabla \mathrm{u}, \nabla^2 \mathrm{u}, \mathrm{x}, t\right) = 0, \quad i = 1, \ldots, N \tag{1}$$

where $\mathrm{u} = (u_1, \ldots, u_N)$ is the vector of state variables, x is the spatial coordinate, t is time, and $\mathcal{F}_i$ are nonlinear differential operators. The coupling is shown in the dependence of each $\mathcal{F}_i$ on multiple components of u and its derivatives. These systems are typically subject to initial and boundary conditions:

$$\mathrm{u}(\mathrm{x}, 0) = \mathrm{u}_0(\mathrm{x}), \quad \mathrm{x} \in \Omega \tag{2}$$

$$\mathcal{B}_i\left(\mathrm{u}, \nabla \mathrm{u}, \mathrm{x}, t\right) = 0, \quad \mathrm{x} \in \partial\Omega, \quad t > 0 \tag{3}$$

where Ω is the spatial domain, $\partial\Omega$ is its boundary, u_0 specifies the initial state, and $\mathcal{B}_i$ are boundary condition operators. The coupling in these systems occurs in various forms within the equations [27].

Direct coupling occurs when the equation for one variable explicitly contains another variable or its derivatives. This can be represented as:

$$\mathcal{F}_i\left(\mathrm{u}, \frac{\partial \mathrm{u}}{\partial t}, \nabla \mathrm{u}, \nabla^2 \mathrm{u}, \mathrm{x}, t\right) = f_i(u_j, \frac{\partial u_j}{\partial t}, \nabla u_j, \nabla^2 u_j), \quad i \neq j \tag{4}$$

Indirect coupling arises when the equations share common parameters that depend on multiple variables. This can be expressed as:

$$\mathcal{F}_i\left(\mathrm{u}, \frac{\partial \mathrm{u}}{\partial t}, \nabla \mathrm{u}, \nabla^2 \mathrm{u}, \mathrm{x}, t\right) = g_i(\mathrm{p}(\mathrm{u})) \tag{5}$$

where $\mathrm{p}(\mathrm{u})$ is a vector of parameters that depends on multiple components of u. Boundary coupling occurs when the boundary conditions for one variable depend on another variable:

$$\mathcal{B}_i\left(\mathrm{u}, \nabla \mathrm{u}, \mathrm{x}, t\right) = h_i\left(u_j, \nabla u_j\right), \quad i \neq j, \quad \mathrm{x} \in \partial\Omega \tag{6}$$

Initial condition coupling exists when the initial conditions for different variables are interrelated:

$$u_i\left(\mathrm{x}, 0\right) = \phi_i\left(u_j(\mathrm{x}, 0)\right), \quad i \neq j, \quad \mathrm{x} \in \Omega \tag{7}$$

Figure 1 illustrates the structure and connections of such a system, highlighting the mathematical coupling between equations.

2.2 Basic principles of PINNs

PINNs represent a significant shift in scientific computing, integrating data-driven machine learning approaches with

physics based modeling. PINNs address the limitations of purely data-driven methods and the computational challenges associated with solving complex, nonlinear PDEs. Consider a general nonlinear PDE of the form:

$$\mathcal{N}u = f(x,t), \quad x \in \Omega, \quad t \in [0,T] \tag{8}$$

where $\mathcal{N}$ is a nonlinear differential operator, $u(x, t)$ is the solution, $f(x, t)$ is a known function, and T represents the final time of interest for the problem. The PINN approach approximates the solution $u(x, t)$ using a neural network $\hat{u}(x,t,\theta)$, where θ represents the network parameters. The key innovation of PINNs lies in the formulation of the loss function, which typically includes both data mismatch and PDE residual terms:

$$\mathcal{L}(\theta) = \underbrace{\mathcal{L}_{\text{data}}(\theta)}_{\substack{\text{Data-driven} \\ \text{part}}} + \underbrace{\lambda \mathcal{L}_{\text{PDE}}(\theta)}_{\substack{\text{Physical} \\ \text{model part}}} \tag{9}$$

where $\mathcal{L}\text{data}(\theta)$ measures the mismatch between network predictions and available data, which primarily consists of boundary conditions and initial conditions of the solution $u(x, t)$, and may also include additional measurements or observations when available, $\mathcal{L}_{\text{PDE}}(\theta)$ quantifies the residual of the PDE, and λ is a weighting parameter.

2.3 Formulation of standard PINN loss functions for coupled systems

A crucial feature of PINNs is their use of automatic differentiation to compute derivatives. This capability helps the handling of complex, high-order PDEs. For a neural network $\hat{u}(x,t,\theta)$, derivatives are computed as:

$$\frac{\partial \hat{u}}{\partial x} = \frac{\partial}{\partial x}\hat{u}(x,t,\theta), \quad \frac{\partial^2 \hat{u}}{\partial x^2} = \frac{\partial^2}{\partial x^2}\hat{u}(x,t,\theta) \tag{10}$$

This approach allows for efficient and accurate computation of derivatives, which is essential for evaluating PDE residuals and their gradients during the training process. For a system of N coupled equations:

$$\mathcal{N}i[u_1, ..., u_N] = f_i(x,t), \quad i = 1, ..., N \tag{11}$$

The loss function for such a system becomes:

$$\mathcal{L}(\theta) = \sum_{i=1}^{N} \left(\underbrace{\mathcal{L}_{\text{data},i}(\theta)}_{\substack{\text{Data-driven} \\ \text{part}}} + \underbrace{\lambda_i \mathcal{L}_{\text{PDE},i}(\theta)}_{\substack{\text{Physical} \\ \text{model part}}} \right) \tag{12}$$

This formulation allows PINNs to simultaneously solve multiple coupled PDEs by incorporating both the physics of the system and any available data. Algorithm 1 provides a general framework for solving coupled multiphysics systems using standard PINNs.

procedure SOLVECOUPLEDMULTIPHYSICSPINN($\mathcal{F}_i, \mathcal{B}_i, \mathbf{u}_0, \Omega, T, N_{epochs}$)
  Initialize neural network parameters θ
  for $epoch = 1$ **to** N_{epochs} **do**
    Sample collocation points $(x,t) \in \Omega \times [0,T]$
    Sample boundary points $x_b \in \partial\Omega$
    Sample initial points $x_0 \in \Omega$
    Forward pass: Compute $\mathbf{u} = \text{PINN}(x,t;\theta)$
    Compute PDE residuals: $\mathcal{R}_i = \mathcal{F}_i(\mathbf{u}, \frac{\partial \mathbf{u}}{\partial t}, \nabla\mathbf{u}, \nabla^2\mathbf{u}, x, t)$
    Compute BC residuals: $\mathcal{R}_{B_i} = \mathcal{B}_i(\mathbf{u}, \nabla\mathbf{u}, x_b, t)$
    Compute IC residuals: $\mathcal{R}_{IC} = \mathbf{u}(x_0, 0) - \mathbf{u}_0(x_0)$
    Compute total loss: $\mathcal{L} = w_1 \sum_i \|\mathcal{R}_i\|^2 + w_2 \sum_i \|\mathcal{R}_{B_i}\|^2 + w_3 \|\mathcal{R}_{IC}\|^2$
    Backpropagate: Update θ using an optimizer (e.g., Adam)
  end for
  return Trained PINN parameters θ
end procedure

Algorithm 1 PINN Algorithm for Coupled Multiphysics Systems

2.4 Recent advancements in PINNs methodologies

Recent years have seen significant developments in PINNs methodologies. These advancements aim to improve the stability, efficiency, and accuracy of PINNs in various applications.

Researchers have introduced adaptive weighting techniques to optimize the balance between different terms in the loss function. One study proposed an adaptive weighting scheme where the loss function is formulated as:

$$\mathcal{L}(\theta) = \sum_{i=1}^{N} w_i(\theta)\mathcal{L}_i(\theta) \tag{13}$$

where $w_i(\theta)$ are adaptive weights that evolve during training based on the relative magnitudes of the individual loss terms

$\mathcal{L}_i(\theta)$. This approach has shown improved convergence and accuracy in various problems by dynamically balancing the contributions of physical constraints and boundary conditions during the training [11].

Curriculum learning strategies gradually increase problem complexity during training. In the context of PINNs, this approach can be applied by simplifying equation parameters initially and progressively increasing them to their actual values [28]. Additionally, curriculum learning has been proposed to automatically adjust sample weights, emphasizing easier non-layer regions. This technique improves the network's approximation accuracy for strongly singular perturbation problems [29]. A general equation for curriculum learning in PINNs can be expressed as:

$$\mathcal{L}_{\text{total}}(\theta, t) = \alpha(t)\mathcal{L}_{\text{data}}(\theta) + \beta(t)\mathcal{L}_{\text{physics}}(\theta) + \gamma(t)\mathcal{L}_{\text{boundary}}(\theta) \tag{14}$$

where $\alpha(t)$, $\beta(t)$, and $\gamma(t)$ are time-dependent weighting functions chosen to gradually increase the importance of different loss terms, and $\mathcal{L}_{\text{data}}(\theta)$, $\mathcal{L}_{\text{physics}}(\theta)$, and $\mathcal{L}_{\text{boundary}}(\theta)$ represent the data, physics, and boundary condition loss terms, respectively. This approach can improve the learning process and the model's ability to capture complex physical behaviors, especially in problems with multiple scales or strong nonlinearities.

To better handle multiscale phenomena, multifidelity PINNs have been developed. These approaches combine data from different fidelity levels to enhance the model's ability to capture both large scale behavior and fine scale details [30]. The loss function in this case takes the form:

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{low}}(\theta) + \beta\mathcal{L}_{\text{high}}(\theta) \tag{15}$$

where $\mathcal{L}_{\text{low}}$ and $\mathcal{L}_{\text{high}}$ represent losses corresponding to low and high fidelity data, respectively, and β is a weighting factor. This is relevant for problems involving processes that occur at different spatial or temporal scales. Domain decomposition techniques have been explored to address issues related to complex geometries and improve the overall efficiency of the solution process. One approach is to divide the problem domain into subdomains and trains separate networks for each [15]. For a domain Ω divided into K subdomains, the global solution can be expressed as:

$$u(x) = \sum_{k=1}^{K} \chi_k(x) u_k(x) \tag{16}$$

where $\chi_k(x)$ are partition of unity functions and $u_k(x)$ are local solutions in each subdomain. This approach has shown improved performance for problems with complex geometries or localized phenomena.

Physics-constrained neural networks represent another advancement in incorporating domain knowledge directly into the network architecture. Research has demonstrated that custom activation functions $\sigma(x)$ can be designed based on the underlying physics [12]:

$$\sigma(x) = f(x) + g(x)\tilde{\sigma}(x) \tag{17}$$

where $f(x)$ and $g(x)$ are chosen to satisfy known physical constraints, and $\tilde{\sigma}(x)$ is a standard activation function. This approach can improve the stability and accuracy of the predictions, especially for long-term simulations.

An effective approach to further enhance the training process is to sample BCs and ICs adaptively. This ensures that the network accurately captures the boundary and initial conditions of the problem. Techniques such as importance sampling can prioritize regions with higher residuals or errors, ensuring that the network focuses on the most critical areas during training [31]. Another strategy involves gradually increasing the number of sampling points within the domain during training. This helps the network to first learn the overall structure of the solution and then refine it by focusing on finer details. Curriculum learning can be applied here, where the complexity of the problem is progressively increased by adding more sampling points or by refining the mesh used for sampling [32].

These recent advancements have significantly expanded the capabilities of PINNs in the handling of complex and interacting systems. However, there are still big challenges especially for complicated problems involving multiple physical processes. The multistage PINN approach proposed in this study builds on these developments, offering a novel strategy to address some of these challenges. It provides a robust framework for solving PDEs efficiently and accurately in complex physical systems. Table 1 provides an overview of the recent advancements in PINN methodologies to solve complex physics problems.

3 Multistaged PINNs for coupled multiphysics systems

The foundation of this approach is a neural network designed to map spatial and temporal coordinates to the relevant physical variables of the system. For a system with d spatial dimensions and n physical variables, the network can be expressed as:

$$N : \mathbb{R}^{d+1} \rightarrow \mathbb{R}^n \tag{18}$$

$$\mathrm{u} = N\left(\mathrm{x}, t; \boldsymbol{\theta}\right) \tag{19}$$

Table 1 Comparison of PINNs studies

Approach	Strengths	Limitations
Adaptive Weight PINNs [33]	Improved convergence; Better handling of stiff problems	May not fully address challenges in highly coupled systems; Potential instability in weight adaptation
Curriculum Learning PINNs [28, 29, 34]	Improved stability in training; Better performance on complex problems	Requires careful design of curriculum; May not generalize well to all problems
hp-VPINNs [35]	Improved accuracy for problems with localized features; Adaptive resolution	Increased computational complexity; May struggle with highly nonlinear coupled systems
Self-adaptive PINNs [11]	Efficient architecture optimization; Improved performance on various problems	Increased training time; May not always find optimal architecture
Multifidelity PINNs [30]	Improved accuracy with limited high-fidelity data; Efficient use of computational resources	Requires availability of multifidelity data; Complexity in balancing different fidelity levels
Physics-constrained PINNs [12]	Improved stability and accuracy for long-term predictions; Better adherence to physical laws	Requires deep understanding of underlying physics; May limit flexibility of the network
Domain Decomposition PINNs [15, 33]	Improved handling of complex geometries; Potential for parallelization	Challenges in ensuring continuity across subdomains; Increased complexity in implementation
Multistage PINNs (This study)	Improved handling of coupled multiphysics problems; Significant reduction in training time	May require careful design of stages; Potential for underestimation in some variables

where u represents the vector of physical variables, x the spatial coordinates, t the time, and $\boldsymbol{\theta}$ the network parameters. The learning process is divided into S stages, each corresponding to a level of physical complexity. In each stage s, the network focuses on a subset of the physical variables or equations:

$$\mathrm{u}^s = N^s\left(\mathrm{x}, t; \boldsymbol{\theta}^s\right), \quad s = 1, 2, \ldots, S \tag{20}$$

where u^s represents the subset of physical variables considered in stage s. The loss function for each stage, L^s, is a crucial component and includes multiple terms:

$$L^s = \lambda^s_{\mathrm{PDE}} L^s_{\mathrm{PDE}} + \lambda^s_{\mathrm{BC}} L^s_{\mathrm{BC}} + \lambda^s_{\mathrm{IC}} L^s_{\mathrm{IC}} \tag{21}$$

where L^s_{PDE} represents the PDE residuals, L^s_{BC} the boundary condition losses, L^s_{IC} the initial condition losses, and λ are weighting parameters. The PDE residual term ensures the network's predictions satisfy the governing equations:

$$L^s_{\mathrm{PDE}} = \sum_i \left\| f_i\left(\mathrm{x}, t, \mathrm{u}^s, \frac{\partial \mathrm{u}^s}{\partial \mathrm{x}}, \frac{\partial \mathrm{u}^s}{\partial t}, \ldots\right)\right\|^2 \tag{22}$$

where f_i represents the i-th PDE in the system. Boundary and initial condition losses are formulated similarly:

$$L^s_{\mathrm{BC}} = \sum_j \left\| B_j\left(\mathrm{x}, t, \mathrm{u}^s\right)\right\|^2 \tag{23}$$

$$L^s_{\mathrm{IC}} = \sum_k \left\| I_k\left(\mathrm{x}, \mathrm{u}^s(\mathrm{x}, 0)\right)\right\|^2 \tag{24}$$

where B_j and I_k represent boundary and initial conditions, respectively. For each stage s, the training process involves:

1. Initializing the network N^s(or loading weights from the previous stage for $s > 1$).
2. Generating training data appropriate for the current stage.
3. Minimizing the loss function.
4. Saving the optimized weights for potential use in the next stage.

This process is repeated for a specified number of epochs or until convergence criteria are met. To facilitate knowledge transfer between stages, the optimized weights from stage s are used to initialize the network for stage $s + 1$. This transfer allows the network to build upon previously learned features and physics.

To validate and demonstrate the efficacy of the proposed multistaged PINN approach, two multiphysics problems are presented as test cases. The first example explores the complex process of asphalt aging, a phenomenon of significant interest in civil engineering and materials science. The second example examines the fluid dynamics in a lid-driven cavity, involving the coupled Navier-Stokes equations. Both examples are implemented using PyTorch, a machine learning library that provides efficient computation and automatic differentiation capabilities, which is widely used for building PINNs models [36].

To address initialization dependence in PINNs, a consistent initialization strategy was implemented using fixed random seeds (1234) for both PyTorch and NumPy libraries across all experiments. This approach ensures reproducible results by maintaining identical initial model parameters and stochastic processes across runs.

4 Example 1: an advanced PINN-based multistage approach for modeling asphalt aging

4.1 Background on asphalt aging

Asphalt, a complex mixture of hydrocarbons derived from petroleum, is one of the most widely used materials in road construction. The viscoelastic properties of asphalt make it ideal for creating durable, flexible pavements capable of handling diverse traffic loads and environmental conditions [37, 38]. Asphalt binders, the adhesive component in asphalt mixtures, play a crucial role in determining the overall performance of road surfaces.

However, the properties of asphalt materials are not static. They evolve over time due to various environmental factors. This phenomenon, known as asphalt aging, significantly impacts the durability and functionality of pavements [39, 40]. Among the various aging mechanisms, oxidative aging stands out as a primary contributor to long-term asphalt degradation [41].

Oxidative aging is a complex process primarily driven by the interaction between asphalt binders and atmospheric oxygen and temperature. This aging mechanism can be characterized as a two-stage process [42]:

- Fast-rate reaction: Initially, there is a rapid oxidation phase where the most reactive components of the asphalt binder quickly interact with oxygen. This stage is characterized by a high reaction rate and is particularly significant in the early life of the pavement.
- Constant-rate reaction: Following the fast-rate period, the oxidation continues at a slower, more constant rate. This stage is known as the long-term aging process of the asphalt.

Temperature plays a crucial role in the aging process, with higher temperatures accelerating the oxidation reactions following the Arrhenius-type relationship [43]. The diffusion of oxygen within the asphalt matrix is also temperature-dependent. Oxygen diffusion is equally important, as the availability and concentration of oxygen directly affect the rate of oxidation [44].

A primary indicator of oxidative aging is the formation of carbonyl compounds, quantified by the carbonyl area in infrared spectroscopy [45, 46]. The carbonyl area serves as a measure of the extent of oxidation and correlates with changes in the rheological properties of the asphalt binder. The increasing in the carbonyl content typically leads to hardening and brittleness of the asphalt, affecting its viscoelastic properties and overall performance.

The governing equations of these variables (i.e., temperature, oxygen pressure, carbonyl area) are presented as follows [47]:

1- Temperature diffusion: The temperature distribution within the asphalt is governed by the heat equation:

$$\frac{\partial T}{\partial t} = \alpha \nabla^2 T \tag{25}$$

where T is temperature, t is time, ∇(gradient operator) = $\left[\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right]$, and α is the thermal diffusivity of the asphalt.

2. Oxygen pressure diffusion: The oxygen pressure within the asphalt is described by a modified diffusion equation that accounts for oxygen consumption during the oxidation process:

$$\frac{\partial P}{\partial t} = D\nabla^2 P - \frac{c_0 RT}{h} * \frac{\partial CA}{\partial t} \tag{26}$$

where P is oxygen pressure, D is the diffusion coefficient of oxygen in asphalt, R is the universal gas constant, h is the oxygen solubility in asphalt binder at a standard temperature of 30°C, CA is the carbonyl area, c_0 is a factor that converts reaction rate of CA to rate of oxygen consumption.

3. Carbonyl area formation: The formation of carbonyl compounds, represented by the carbonyl area (CA), is modeled using a kinetic equation:

$$\frac{\partial CA}{\partial t} = M_{RTFO} k_f e^{-k_f t} + k_c \tag{27}$$

where M_{RTFO} is the limiting amount of carbonyl formation due to the first-order reaction after hot mix production, k_f is the fast reaction rate constant, and k_c is the constant reaction rate constant. The rate constants k_f and k_c are temperature and pressure dependent, following the Arrhenius-type relationship [48]:

$$k_f = A_f \left(\frac{P}{P_0}\right)^a e^{-E_{af}/RT} \tag{28}$$

$$k_c = A_c \left(\frac{P}{P_0}\right)^a e^{-E_{ac}/RT} \tag{29}$$

where A_f and A_c are pre-exponential factors, P_0 is the reference pressure, a is the pressure exponent, E_{af} and E_{ac} are activation energies.

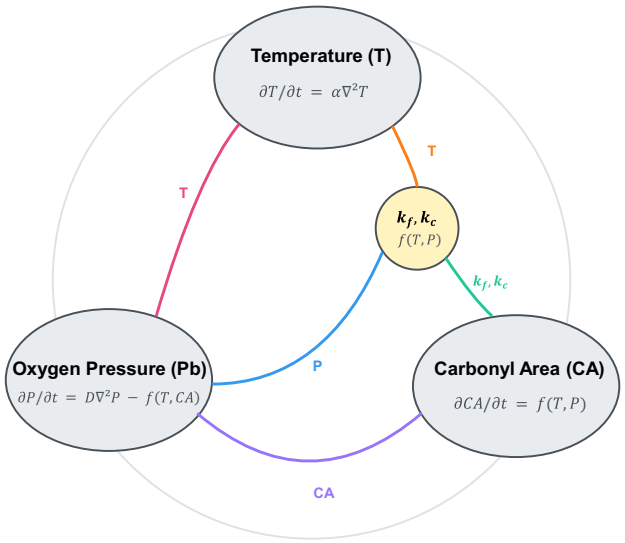

Fig. 2 Schematic representation of the coupled multiphysics system governing asphalt aging

Figure 2 shows the coupling between the three governing equations, forming a complex multiphysics system.

The temperature (T) affects both oxygen pressure (P) and carbonyl area (CA) formation. It directly influences the reaction rates k_f and k_c in the CA formation equation through the Arrhenius relationships and appears in the oxygen pressure equation, affecting the consumption term. Oxygen pressure (P) influences the carbonyl area (CA) formation through the pressure-dependent terms in the rate constants k_f and k_c, which directly affect the CA formation rate. In turn, carbonyl area (CA) formation impacts the oxygen pressure (P), as the rate of CA formation ($\partial CA/\partial t$) appears in the oxygen pressure equation, representing oxygen consumption during oxidation.

This coupling creates a feedback loop where changes in one variable can propagate through the system, affecting the others. For instance, an increase in temperature can accelerate CA formation, which in turn increases oxygen consumption, potentially leading to changes in the oxygen pressure distribution. These interactions make the aging process of asphalt a complex phenomenon that requires simultaneous consideration of all three components for accurate modeling and prediction.

4.2 Training algorithm for asphalt aging prediction

The training process follows a curriculum learning strategy, which gradually increases the complexity of the task the neural network must learn. The curriculum is divided into three stages.

In the first stage, the network learns to model only the temperature diffusion process. This allows the PINN to establish a foundation in solving the basic heat equation.

The second stage introduces a simplified version of the oxygen pressure process, coupling it with the temperature model. In this stage, a simplified oxygen pressure equation is used, where the term involving $\frac{\partial CA}{\partial t}$ is omitted, assuming a constant rate of oxygen consumption. This simplification increases the complexity as the network must now solve two interrelated PDEs simultaneously, but without the full complexity of the carbonyl area formation.

The final stage incorporates the full system, including the complete oxygen pressure equation and carbonyl area formation. At this point, the PINN is tasked with solving the complete coupled system of PDEs that describe the asphalt aging process, including the interdependencies between temperature, oxygen pressure, and carbonyl area formation.

The use of a simplified oxygen pressure equation in the second stage serves as an intermediate step between the basic heat equation and the fully coupled system. This intermediate stage helps the network to gradually adapt to the coupling between temperature and oxygen pressure before introducing the additional complexity of carbonyl area formation.

In addition to this multistage approach, an adaptive weight PINN method is also considered for comparison in this study. The adaptive weight PINN dynamically adjusts the weights of different loss components during training to enhance the balance between satisfying physical constraints and fitting boundary conditions during the training. This method provides an alternative approach to handling the complexities of the coupled PDE system in asphalt aging modeling.

While the temperature variable (T) appears to influence pressure (P) and carbonyl area (CA) more directly than vice versa, the temperature PDE residual is maintained in the loss function for all stages of the multistage PINN approach. This decision is based on several important considerations. First, real-world multiphysics systems often exhibit complex, bidirectional couplings that may not be immediately apparent. By including the temperature residual throughout, the model ensures that any small, indirect effects of P and CA on T are captured. Second, this approach enhances the overall numerical stability of the solution. The temperature residual acts as a constraint, ensuring that the temperature field remains physically consistent throughout the optimization process, even when the focus is on P and CA. Third, in coupled systems, errors in one variable can propagate and amplify in others. Maintaining the temperature residual minimizes the potential for error accumulation in the temperature field, which could otherwise lead to inaccuracies in P and CA predictions. Lastly, while the current model may suggest a primarily one-way coupling, maintaining the full

coupling in the formulation future-proofs the model. If future research reveals more complex interactions between T, P, and CA, the model structure is already equipped to accommodate these without requiring significant restructuring.

Dirichlet boundary conditions are employed for temperature (T), oxygen pressure (P) at the domain boundaries (top, bottom, left, and right edges of the asphalt sample). These conditions can be mathematically expressed as follows:

For temperature:

$$\begin{aligned} T(x,0,t) &= T_{\text{bottom}} \\ T(x,L_y,t) &= T_{\text{top}} \\ T(0,y,t) &= T_{\text{left}} \\ T(L_x,y,t) &= T_{\text{right}} \end{aligned}$$

For oxygen pressure:

$$\begin{aligned} P(x,0,t) &= P_{\text{bottom}} \\ P(x,L_y,t) &= P_{\text{top}} \\ P(0,y,t) &= P_{\text{left}} \\ P(L_x,y,t) &= P_{\text{right}} \end{aligned}$$

where L_x and L_y are the dimensions of the asphalt sample in the x and y directions, respectively. For the initial conditions, specified values for T, P, and CA throughout the domain at time $t = 0$ are given as:

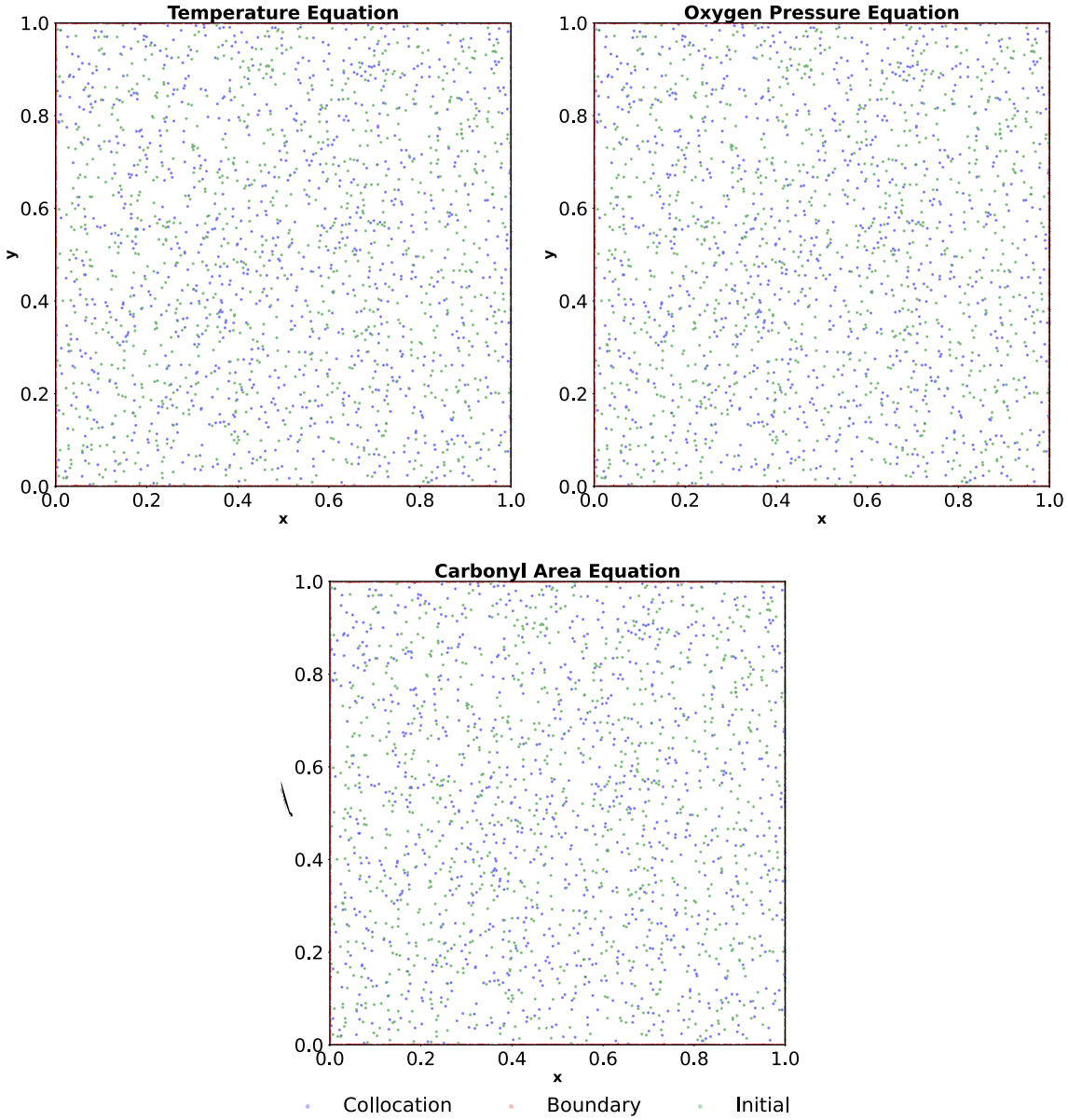

Fig. 3 Spatial distribution of collocation points, boundary conditions, and initial conditions for the coupled PDE system

Table 2 Summary of hyperparameters used in the PINN model

Hyperparameter	Value
No. of Inputs	3[a]
No. of Outputs	1–3[b]
No. of Hidden Layers	1–5[c]
No. of Hidden Neurons	10–130[c]
Activation Function	Hyperbolic Tangent (tanh)
Optimizer	Adam
Learning Rate	0.001
Batch Size	Full dataset[d]
Epochs per Stage	100,000

[a]Corresponding to spatial coordinates (x, y) and time (t)

[b]Varying by curriculum learning stage: 1 (T), 2 (T and P), and 3 (T, P, and CA)

[c]Range explored in sensitivity analysis

[d]The entire dataset is used for each iteration

$$
\begin{aligned}
T(x,y,0) &= T_{\text{init}} \\
P(x,y,0) &= P_{\text{init}} \\
CA(x,y,0) &= CA_{\text{init}}
\end{aligned}
$$

These boundary and initial conditions are incorporated into the loss function to ensure the PINN satisfies these constraints during training. Figure 3 shows the spatial distribution of the boundary, initial conditions and collocation points for each PDE.

The loss function incorporates both the physics of the problem and the boundary/initial conditions. It is formulated as a weighted sum of three components: boundary condition loss, initial condition loss, and PDE residual loss. The complete loss function of the multistage PINN for asphalt aging at the last stage can be expressed as:

$$
\begin{aligned}
\mathcal{L}_{\text{total}} = & \underbrace{w_{\text{bc}} \sum_{i\in\{T,P,CA\}} \left(\frac{1}{N_{\text{bc}}} \sum_{j=1}^{N_{\text{bc}}} \left| i_j^{\text{pred}} - i_j^{\text{true}} \right|^2 \right)}_{\text{Data-driven part}} \\
& + \underbrace{w_{\text{ic}} \sum_{i\in\{T,P,CA\}} \left(\frac{1}{N_{\text{ic}}} \sum_{k=1}^{N_{\text{ic}}} \left| i_k^{\text{pred}}(t=0) - i_k^{\text{init}} \right|^2 \right)}_{\text{Data-driven part}} \\
& + \underbrace{w_{\text{pde}} \left(\frac{1}{N_{\text{pde}}} \sum_{l=1}^{N_{\text{pde}}} \left| \frac{\partial T_l}{\partial t} - \alpha \nabla^2 T_l \right|^2 \right.}_{\text{Physical model part}} \\
& + \underbrace{\frac{1}{N_{\text{pde}}} \sum_{l=1}^{N_{\text{pde}}} \left| \frac{\partial P_l}{\partial t} - D\nabla^2 P_l + \frac{c_0 R T_l}{h} \frac{\partial CA_l}{\partial t} \right|^2}_{\text{Physical model part}} \\
& + \underbrace{\left. \frac{1}{N_{\text{pde}}} \sum_{l=1}^{N_{\text{pde}}} \left| \frac{\partial CA_l}{\partial t} - M_{\text{RTFO}} k_f e^{-k_f t} - k_c \right|^2 \right)}_{\text{Physical model part}}
\end{aligned}
\tag{30}
$$

where w_{bc}, w_{ic}, and w_{pde} are weighting factors for the boundary condition, initial condition, and PDE residual losses, respectively. $i \in \{T, P, CA\}$ represents the variables temperature, oxygen pressure, and carbonyl area. N_{bc}, N_{ic}, and N_{pde} are the numbers of boundary points, initial condition points, and collocation points, respectively. i_j^{pred} and i_j^{true} are the predicted and true values at boundary points. $i_k^{\text{pred}}(t=0)$ and i_k^{init} are the predicted and initial values at $t = 0$. The last three terms represent the PDE residuals for temperature, oxygen pressure, and carbonyl area, respectively. This multistage approach allows the PINN to progressively learn more complex physics, starting from simple temperature diffusion and building up to the fully coupled system of asphalt aging. The overall algorithm is given as Algorithm 2

Table 3 Model parameters for the asphalt aging simulation

Parameter	Symbol	Value	Unit
Thermal diffusivity	α	0.1	m^2/s
Pressure exponent	a	0.27	-
Pre-exponential factor (fast)	A_f	5.8446×10^{5}	s^{-1}
Pre-exponential factor (constant)	A_c	5.8264×10^{3}	s^{-1}
Activation energy (fast)	E_{af}	75,400	J/mol
Activation energy (constant)	E_{ac}	103,800	J/mol
Oxygen diffusion coefficient	D	1×10^{-11}	m^2/s
Oxygen consumption factor	c_0	371	mol/m^3
Oxygen solubility in bitumen	h	0.0076	$mol/(m^3 \cdot Pa)$
Universal gas constant	R	8.314	J/(mol K)

The parameters are adapted from Omairey et al. (2022) [52] for a road section on US277 in Laredo, Texas

Table 4 Boundary and initial conditions for the asphalt aging simulation

Variable	Top	Bottom	Left	Right	Initial
T (K)	303.15	303.15	274.15	274.15	295.65
P (Pa)	1×10^7	1×10^7	1×10^6	1×10^6	1×10^7
CA (a.u.)	–	–	–	–	0.94

procedure TRAINMULTISTAGEPINN(N_{epochs})
  Initialize PINN parameters θ
  for $epoch = 1$ **to** N_{epochs} **do**
    Sample collocation points (x, y, t)
    // Stage 1: Temperature Diffusion
    Forward pass: Compute $T = \text{PINN}(x, y, t; \theta)$
    Compute PDE residual: $\mathcal{R}_1 = \left|\frac{\partial T}{\partial t} - \alpha \nabla^2 T\right|$
    Compute $\mathcal{L}_{BC}, \mathcal{L}_{IC}$ for temperature
    Compute loss $\mathcal{L}_1 = \mathcal{L}_{BC} + \mathcal{L}_{IC} + \mathcal{R}_1$ (Unweighted summation)
    Backpropagate and update θ using optimizer
  end for
  for $epoch = 1$ **to** N_{epochs} **do**
    Sample collocation points (x, y, t)
    // Stage 2: Temperature and Oxygen Pressure Diffusion
    Forward pass: Compute T, $P = \text{PINN}(x, y, t; \theta)$
    Compute PDE residuals:
    $\mathcal{R}_1 = \left|\frac{\partial T}{\partial t} - \alpha \nabla^2 T\right|$
    $\mathcal{R}_2 = \left|\frac{\partial P}{\partial t} - D\nabla^2 P + \frac{c_0 RT}{h}\right|$
    Compute $\mathcal{L}_{BC}, \mathcal{L}_{IC}$ for T and P
    Compute loss $\mathcal{L}_2 = \mathcal{L}_{BC} + \mathcal{L}_{IC} + \mathcal{R}_1 + \mathcal{R}_2$ (Unweighted summation)
    Backpropagate and update θ using optimizer
  end for
  for $epoch = 1$ **to** N_{epochs} **do**
    Sample collocation points (x, y, t)
    // Stage 3: Full Coupled System
    Forward pass: Compute T, P, $CA = \text{PINN}(x, y, t; \theta)$
    Compute PDE residuals:
    $\mathcal{R}_1 = \left|\frac{\partial T}{\partial t} - \alpha \nabla^2 T\right|$
    $\mathcal{R}_2 = \left|\frac{\partial P}{\partial t} - D\nabla^2 P + \frac{c_0 RT}{h}\frac{\partial CA}{\partial t}\right|$
    $\mathcal{R}_3 = \left|\frac{\partial CA}{\partial t} - M_{RTFO} k_f e^{-k_f t} - k_c\right|$
    Compute $\mathcal{L}_{BC}, \mathcal{L}_{IC}$ for T, P, and CA
    Compute loss $\mathcal{L}_3 = \mathcal{L}_{BC} + \mathcal{L}_{IC} + \mathcal{R}_1 + \mathcal{R}_2 + \mathcal{R}_3$ (Unweighted summation)
    Backpropagate: Update θ using an optimizer (Adam)
  end for
  return Trained PINN parameters θ
end procedure

Algorithm 2 Multistage PINN Training Algorithm for Asphalt Aging

4.3 Training hyperparameters

The performance and efficiency of PINNs are significantly influenced by their architecture and training hyperparameters. To optimize the PINN model for solving the coupled PDEs governing asphalt aging, a sensitivity analysis was conducted. This analysis focused on the architectural and training hyperparameters, as detailed below:

4.3.1 Network architecture

The depth of the network was adjusted from 1 to 5 hidden layers. For each layer configuration, the width of the network was varied, with the number of neurons per layer ranging from 10 to 130. The input layer consistently comprised 3 neurons, corresponding to the spatial coordinates (x, y) and time (t). The output layer, however, varied according to the curriculum learning stage: in the first stage, it contained 1 neuron (T); in the second stage, 2 neurons (T and P); and in the third stage, 3 neurons (T, P, and CA). This progressive increase in output dimensionality aligned with the increasing complexity of the physical system being modeled across the curriculum learning stages. To ensure optimal performance and numerical stability of the PINN model, all input and output variables were scaled to the range [0, 1] prior to training. This normalization process helps to prevent issues related to widely varying magnitudes across different physical quantities, ensures consistent gradient propagation during training, and generally improves the convergence of the neural network.

4.3.2 Training parameters

The following training parameters were maintained constant across all architectural configurations to ensure a fair comparison: A fixed learning rate of 0.001 was utilized, chosen based on preliminary experiments to balance convergence speed and stability. The Adam optimizer was employed due to its effectiveness in handling complex, non-convex optimization problems commonly encountered in neural network training [49]. Hyperbolic tangent (tanh) activation functions were employed between layers due to their suitability for physics-based problems [50, 51]. Each stage of the curriculum learning process was trained for 100,000 epochs. This number of epochs was selected to ensure convergence across all model configurations. Table 2 summarize the hyperparameters adapted in this example.

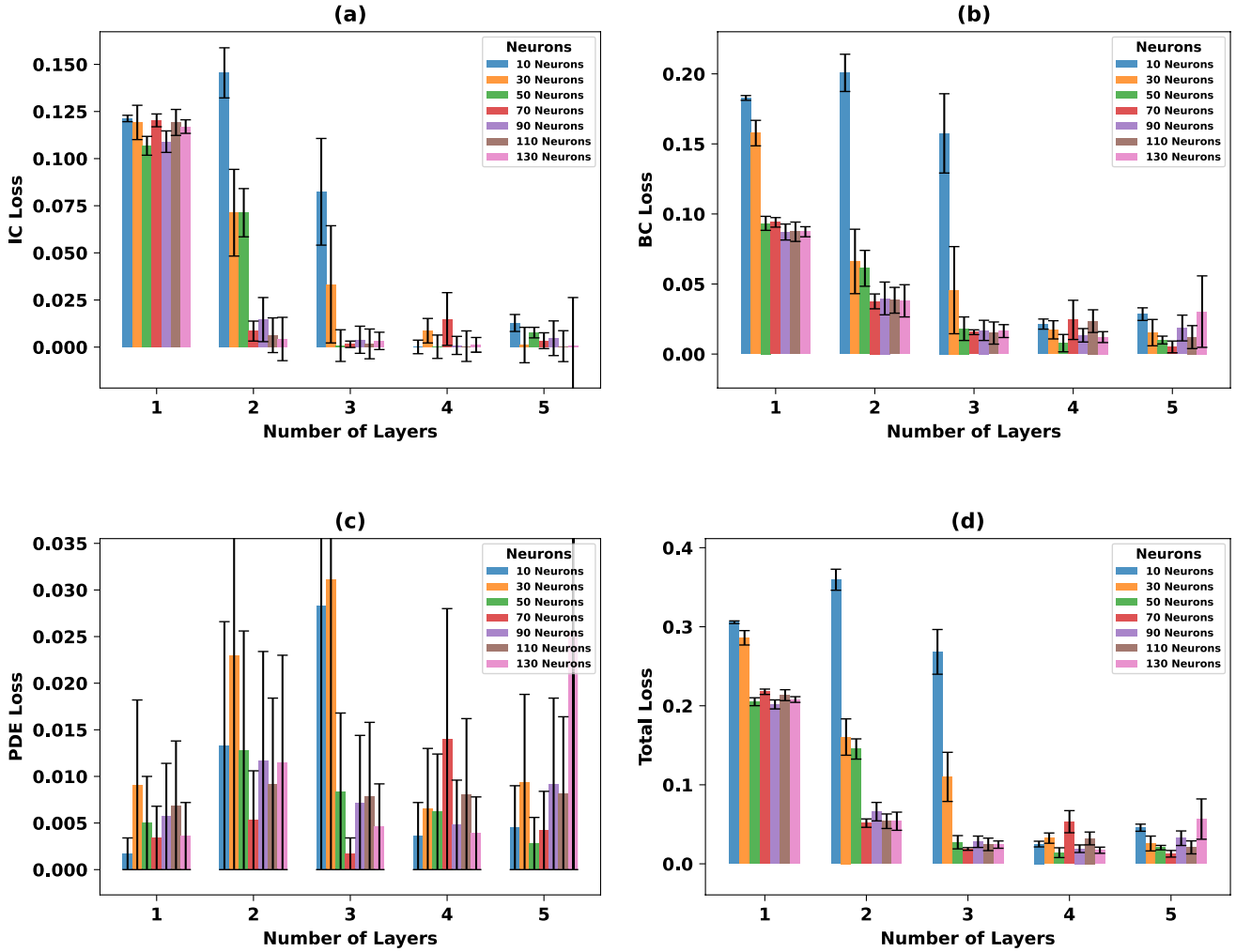

Fig. 4 Sensitivity analysis of PINN architecture for asphalt aging prediction

4.4 Model parameters

The model parameters employed in this example are derived from a road section on US277 in Laredo, Texas, as documented by Omairey et al. (2022) [52]. These parameters characterize the physical and chemical properties of the asphalt material and the environmental conditions influencing the aging process. Table 3 summarizes the key parameters used in the coupled PDE system.

The thermal diffusivity (α) governs the rate of heat propagation through the asphalt material. The pressure exponent (a) reflects the influence of oxygen pressure on oxidation reaction rates. Pre-exponential factors (A_f, A_c) and activation energies (E_{af}, E_{ac}) are components of the Arrhenius equation, describing the temperature dependence of reaction rates for fast and constant oxidation processes, respectively.

The oxygen diffusion coefficient (D) characterizes the rate of oxygen transport within the asphalt, while the oxygen consumption factor (c_0) relates the carbonyl area formation rate to oxygen consumption. The oxygen solubility (h) represents the capacity of the asphalt binder to absorb oxygen at a reference temperature of 30°C. The simulation domain is defined by spatial coordinates x and y, both ranging from 0 to 1, representing a normalized unit square. The temporal dimension t extends from 0 to 1×10^7 seconds, which is the full duration of the aging process simulation. The boundary and initial conditions are summarized in Table 4.

While these parameters are specific to the studied road section, it is important to note that they may vary for different asphalt mixtures and environmental conditions. Future studies could explore the model's sensitivity to parameter variations and extend the application to diverse asphalt compositions and climatic scenarios.

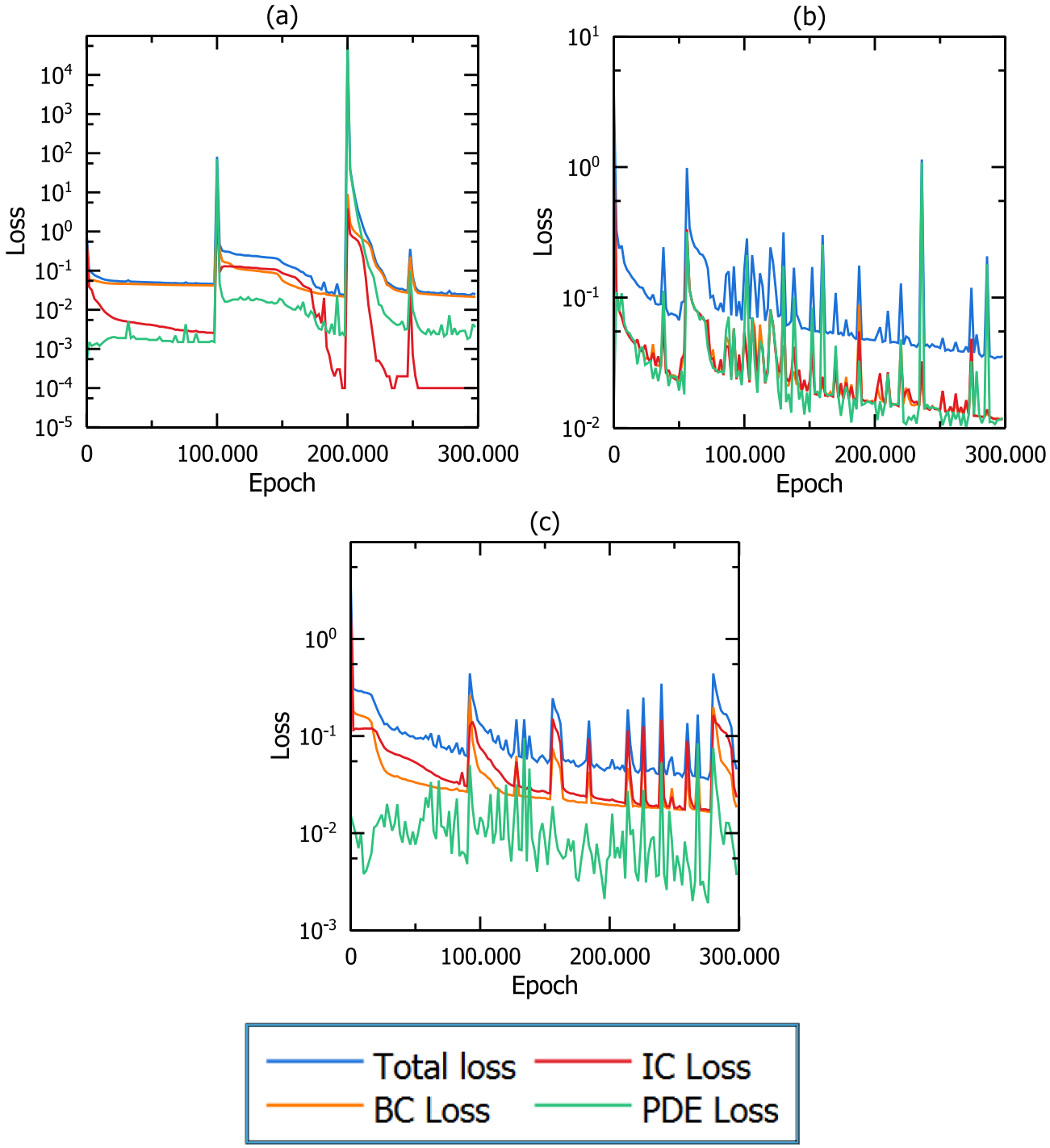

Fig. 5 The evolution of the BC, IC, PDE, and the total losses during the training process. **a** Multistage PINN **b** Adaptive Weigh PINN **c** PINN

4.5 Results: asphalt aging model

4.5.1 Sensitivity analysis

A sensitivity analysis was conducted to determine the optimal network architecture. The results of this analysis are presented in Fig. 4. Based on the analysis, a network architecture with 4 hidden layers and 10 neurons per layer was selected. This configuration demonstrates an excellent balance between model complexity and performance. The chosen architecture performs well in minimizing the initial condition (IC) loss and maintains low boundary condition (BC) and PDE losses. In addition, it was observed that

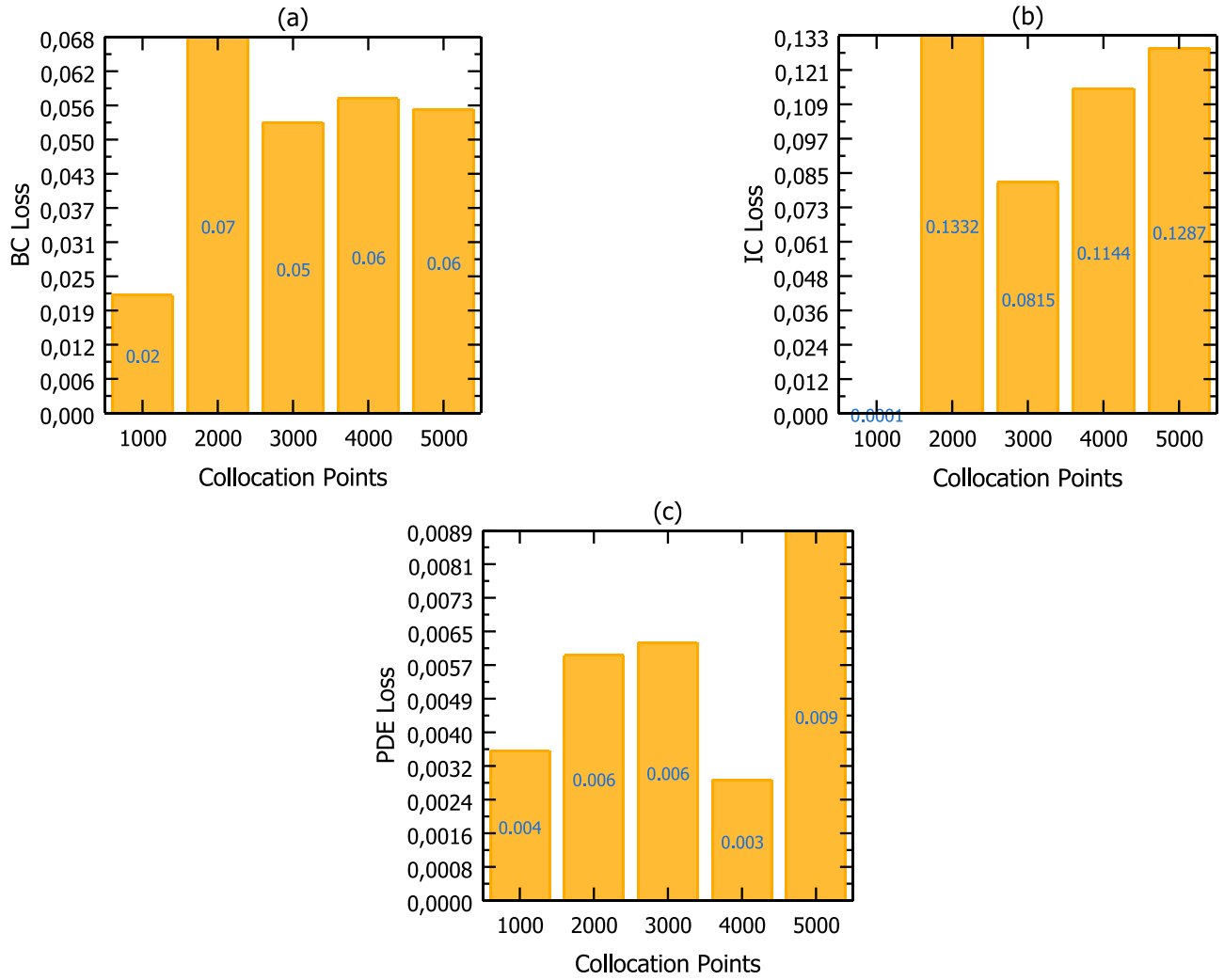

Fig. 6 Comparison of loss components **a** BC, **b** IC, and **c** PDE across different numbers of collocation points in the Multistage PINN framework

increasing the network size beyond 4 layers or 10 neurons per layer often resulted in only marginal improvements or even decreased performance in some cases, suggesting a point of diminishing returns.

Figure 5 illustrates the evolution of the loss components during the training process for the selected optimal network architecture (4 hidden layers, 10 neurons per layer).

In Stage 1 (epochs 0-100,000), which focuses on temperature diffusion, we observe a rapid initial decrease in all loss components, followed by a stable convergence. The BC loss and total loss stabilize quickly, while the IC loss continues to decrease gradually.

Stage 2 (epochs 100,000-200,000) introduces oxygen pressure diffusion. This transition is marked by a temporary spike in losses, particularly evident in the PDE loss. However, the network quickly adapts, and all loss components show a steady decline throughout this stage. Stage 3 (epochs 200,000-300,000) incorporates the full coupled system, including carbonyl area formation. Despite the increased complexity, the network maintains stable performance. The total loss and BC loss remain relatively constant, while the IC and PDE losses continue to decrease slightly, indicating the network's ability to capture the full complexity of the asphalt aging process (**Fig.**5a). The progression of losses depicted in Fig. 5b, c reveals higher loss values and greater fluctuations. This suggests that both the adaptive weight PINN and standard PINN encounter more challenges in achieving convergence.

The impact of collocation point distribution on the PINN model's performance was analyzed by varying the number of points from 1000 to 5000 (Fig. 6). The results show that 1000 collocation points provide the optimal configuration, minimizing the total loss while accurately representing BC, IC, and governing equations. At this point, additional points fail to contribute meaningful new insights into the underlying physical system may introduce unnecessary redundancy

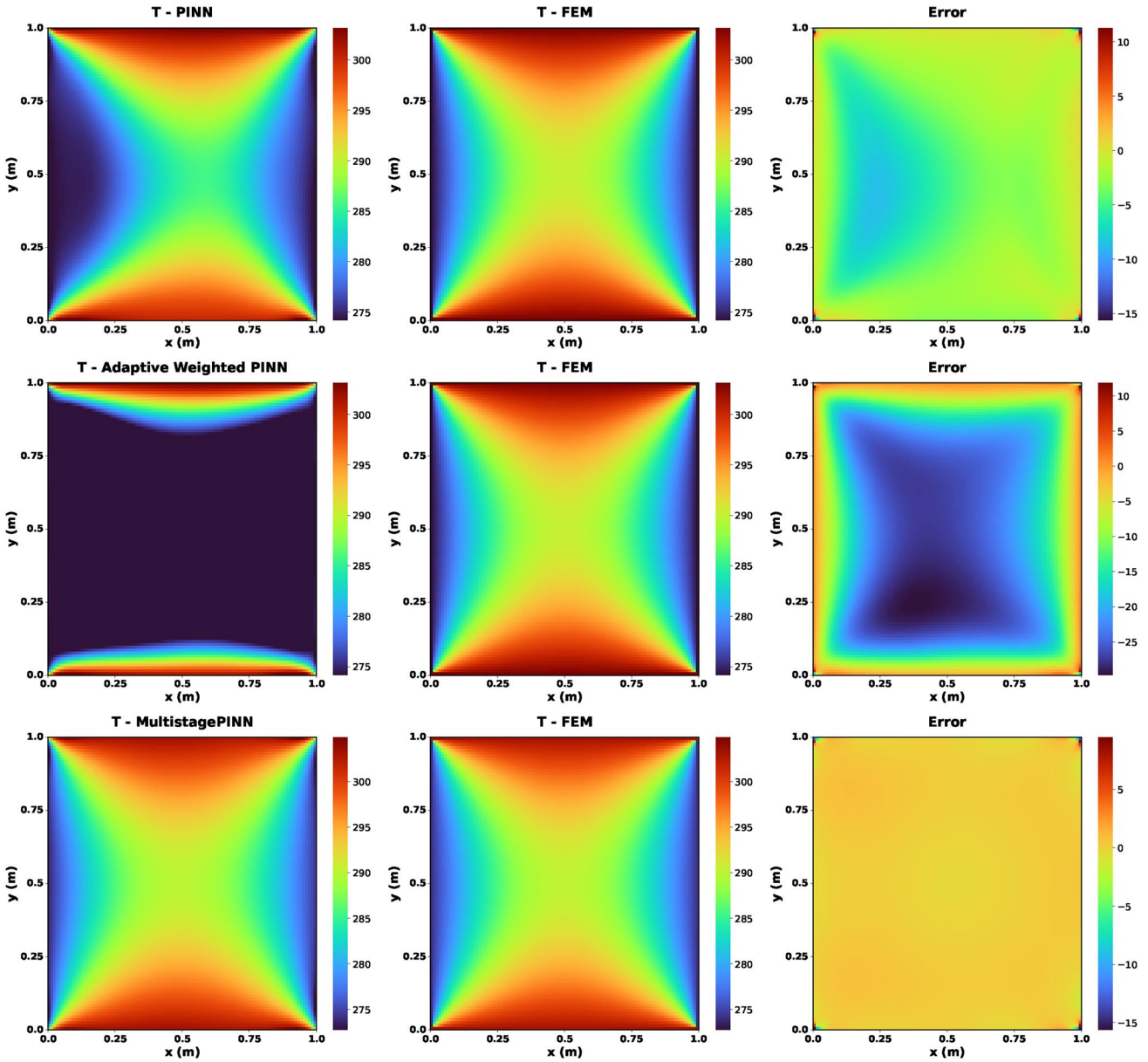

Fig. 7 A comparative visualization of temperature distributions using different computational methods: PINN, Adaptive Weight PINN, Multistage PINN, and FEM

in the data representation. This redundancy can potentially inject noise into the training process, complicating the model's ability to accurately capture the essential physics of the problem [11, 22, 35, 53].

4.5.2 Temperature distribution

Figures 7 and 8 illustrate the performance of various PINN methods compared to the FEM solution for temperature distribution. The multistage PINN method demonstrates remarkable consistency with the FEM solution across both initially and at the final time step ($t = 0$ and $t = 1$, respectively, where t represents normalized time step). In Fig. 7, the multistage PINN heatmap closely resembles the FEM result. It accurately captures the temperature gradients from the edges to the center of the domain. This visual similarity is quantitatively supported by the centerline plots in Fig. 8, where the multistage PINN curves almost perfectly overlay the FEM solutions for both time steps and along both centerlines. In contrast, the standard PINN method shows discrepancies. It overestimates the high-temperature regions at the top and bottom. This overestimation is evident in Fig. 8's centerline plots, particularly at $t = 1$ *(s)*, where the standard PINN deviates from the FEM solution, especially

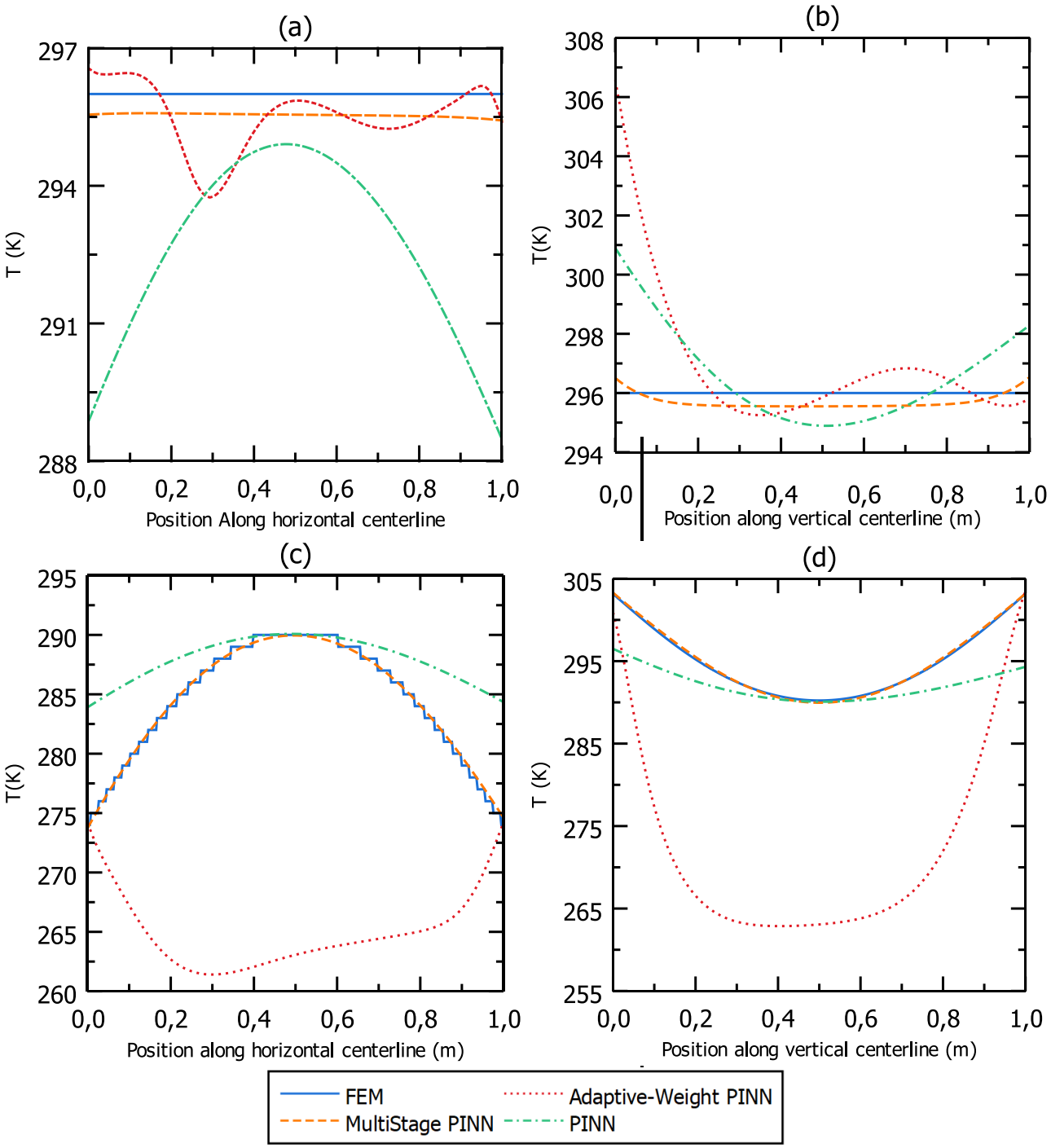

Fig. 8 A quantitative analysis of temperature distributions along the centerlines using different computational methods: PINN, adaptive weight PINN, Multistage PINN, and FEM. **a**, **b** At t = 0 (s) while **c**, **d** at t = 1 (s)

near the domain boundaries. The adaptive weight PINN performs poorly in both representations. The heatmap in Fig. 7 shows a significantly underestimated central temperature region, which is reflected in the centerline plots of Fig. 8. The centerlines plots show extreme oscillations at *t = 0 (s)* and underestimation of temperatures at the final time step.

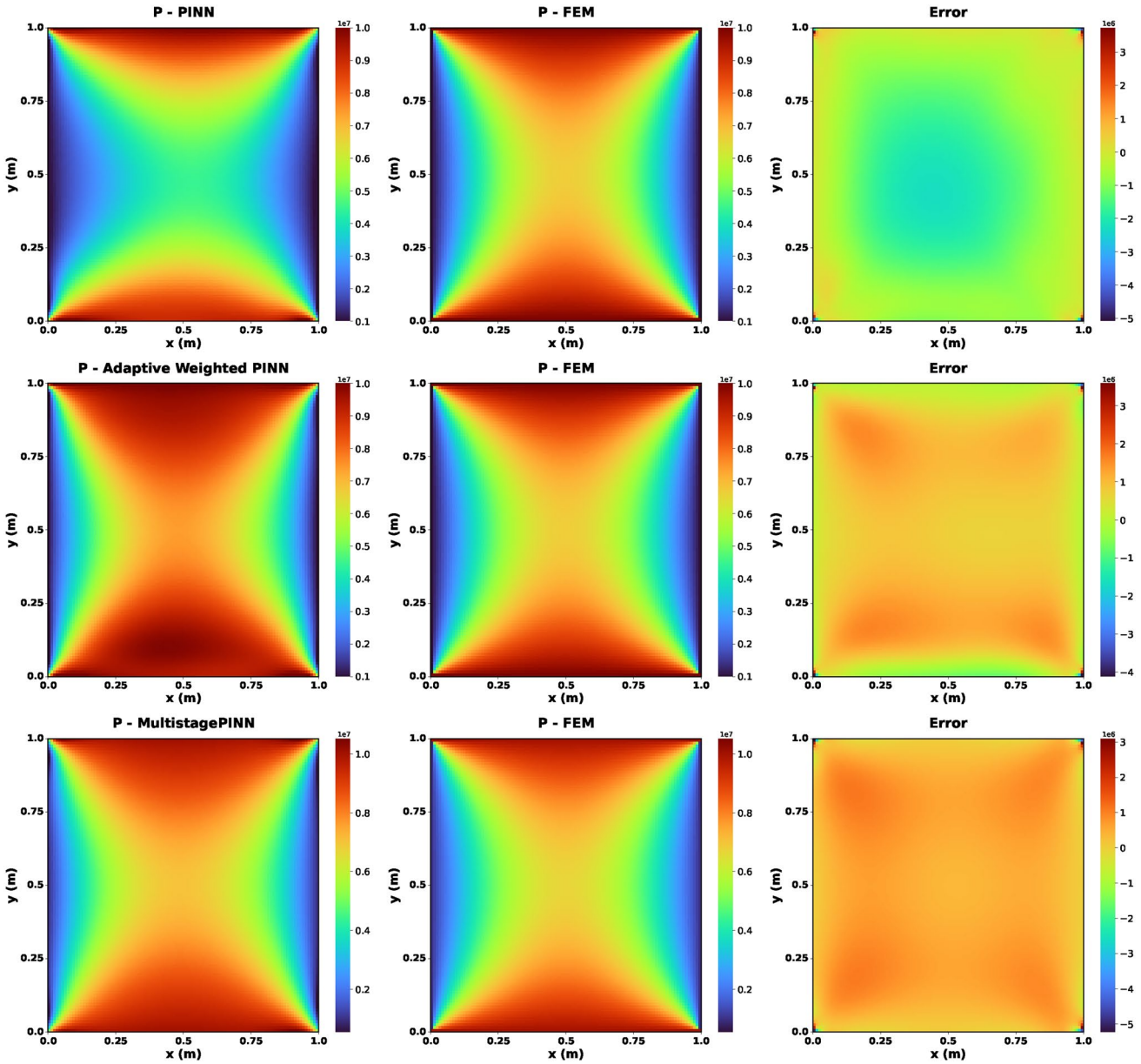

Fig. 9 A comparative visualization of oxygen distributions using different computational methods: PINN, Adaptive Weight PINN, Multistage PINN, and FEM

4.5.3 Oxygen pressure evolution

Figures 9 and 10 show the performance of various PINN methods compared to the FEM solution for oxygen pressure distribution. The multistage PINN demonstrates high accuracy in capturing the oxygen pressure distribution. The heatmap in Fig. 9 closely aligns with the FEM result. This method can accurately captures the higher pressure region at the bottom and the lower pressure areas at the left and right boundaries. The centerline plots in Fig. 10, where the multistage PINN results were plotted, show excellent agreement with the FEM solutions at both time steps and along both centerlines. The standard PINN method, while capturing the general pressure pattern, shows some limitations. The heatmap in Fig. 9 indicates an underestimation of pressure values, particularly evident in the center of the domain and near the boundaries. This underestimation is consistently reflected in Fig. 10's centerline plots, where the standard PINN predicts lower pressure values compared to the FEM solution, especially pronounced at $t = 1$ *(s)* along both horizontal and vertical centerlines. The adaptive weight PINN exhibits improved performance for oxygen pressure compared to its temperature predictions. The heatmap in Fig. 9 shows a reasonable pressure distribution, with

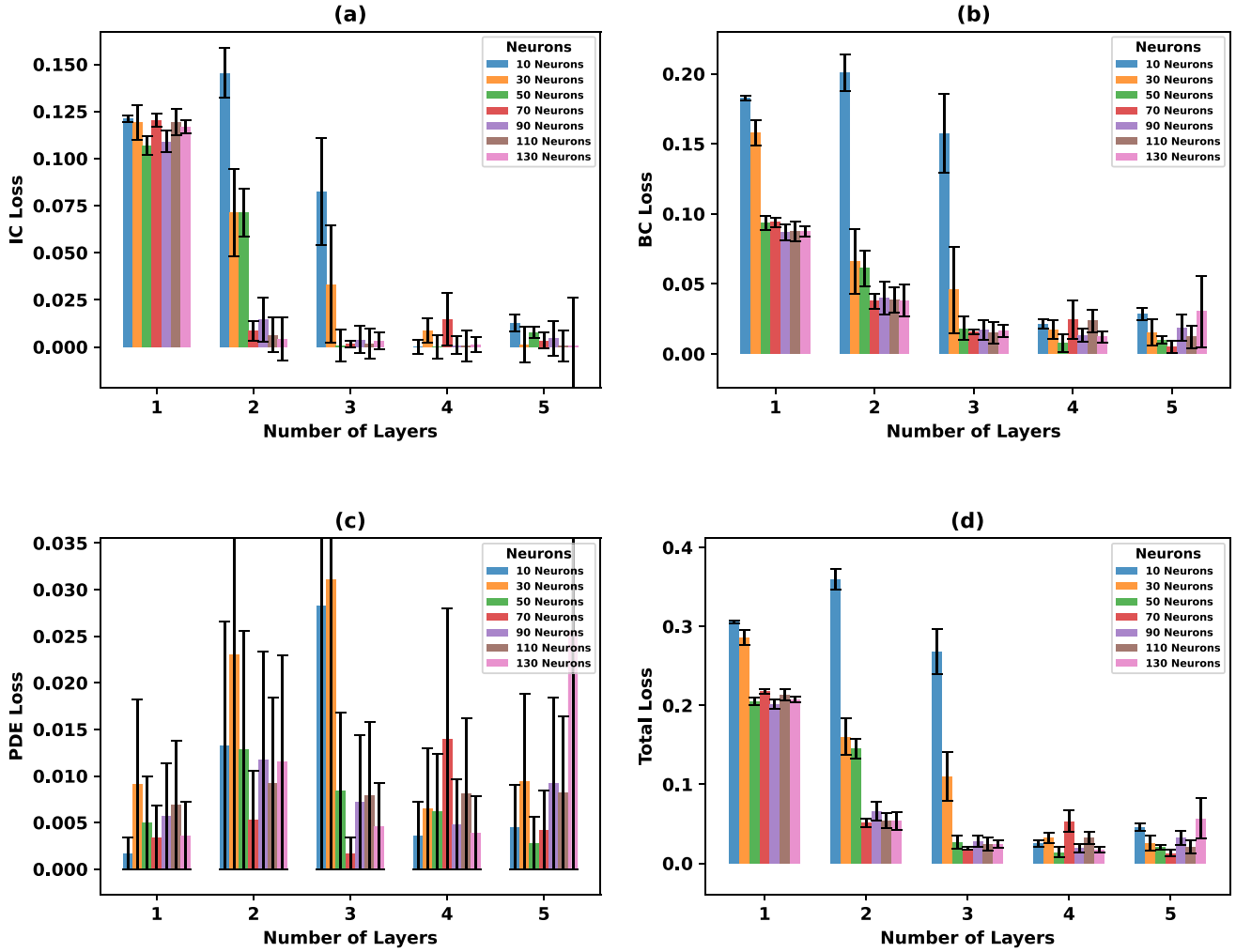

Fig. 10 A quantitative analysis of oxygen distributions along the centerlines using different computational methods: PINN, Adaptive Weight PINN, Multistage PINN, and FEM. (a) and (b) at t = 0 (s) while (c) and (d) at t = 1 (s)

some overestimation in the central region. This tendency is mirrored in the centerline plots of Fig. 10, where the results show slight oscillations at *t = 0 (s)* and overestimation of pressures at *t = 1 (s)*, particularly in the central portions of both centerlines.

4.5.4 Carbonyl area formation

Figures 11 and 12 show the carbonyl area distribution results for different PINN methods compared to the FEM solution. In Fig. 11, the multistage PINN result closely matches the FEM solution. It accurately shows higher carbonyl concentrations at the top and bottom edges, with a gradual decrease towards the center and sides. This accuracy is further confirmed in Fig. 12's centerline plots, where the multistage PINN results align closely with the FEM solutions at *t = 0 (s)*. However, at *t = 1 (s)*, it slightly underestimates the carbonyl area, particularly in the domain center. The standard PINN's heatmap, however, underestimates the carbonyl area across the entire domain. It fails to capture the detailed spatial changes seen in the FEM solution. This underestimation is evident Fig. 12's centerline plots, where the standard PINN consistently predicts lower carbonyl area values. The effect is more pronounced at *t = 1 (s)*, where it fails to capture the curvature of the profiles, especially along the vertical centerline. The adaptive weight PINN's heatmap shows a large overestimation of carbonyl area, especially in the middle of the domain. This result seems unrealistic and points to possible problems with how the method adjusts its weights for this variable.This overestimation is reflected in the centerline plots of Fig. 12, where the results show large oscillations at *t = 0 (s)* and extreme overestimation at *t = 1 (s)*.

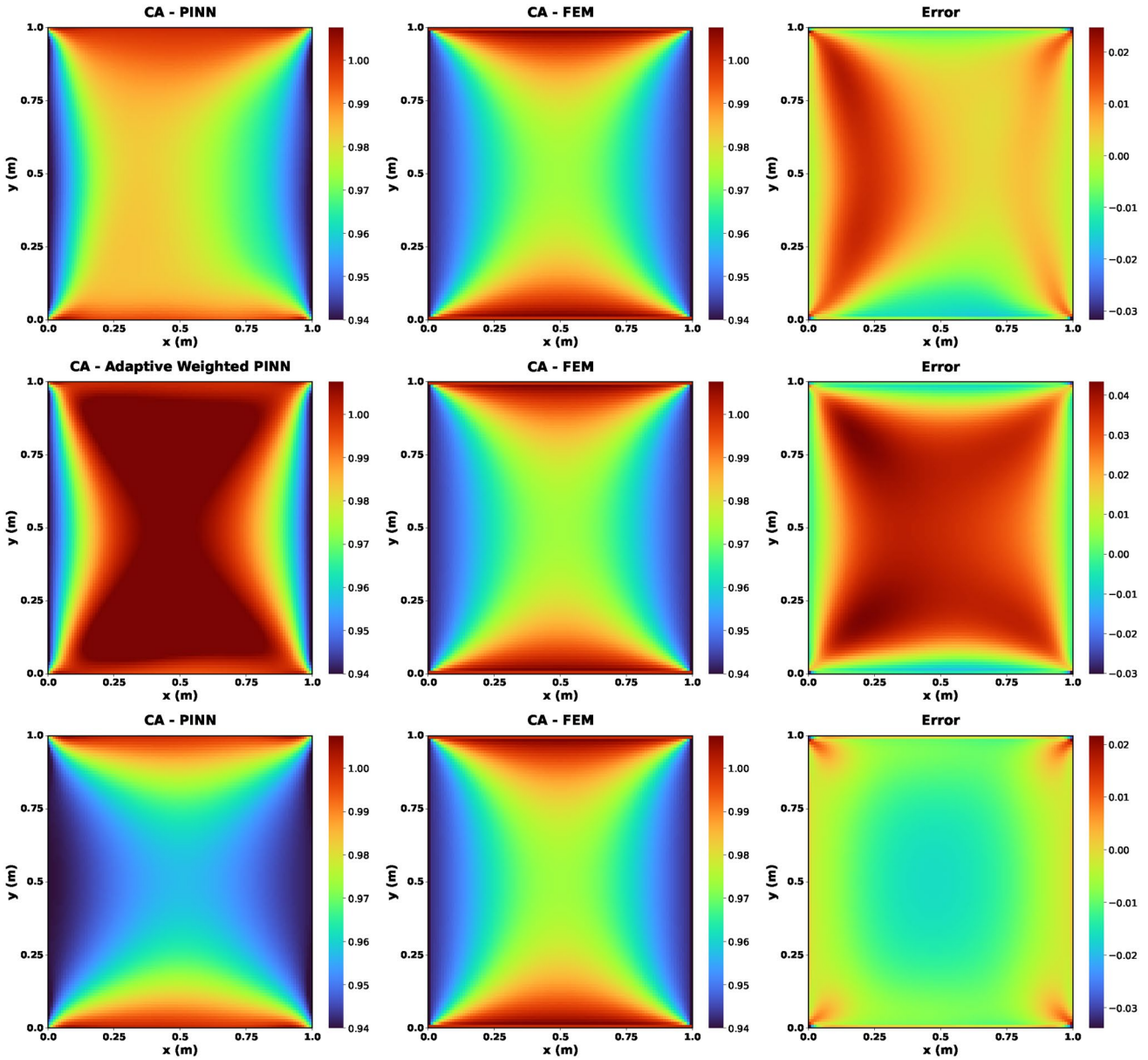

Fig. 11 A comparative visualization of carbonyl area distributions using different computational methods: PINN, Adaptive Weight PINN, Multi-stage PINN, and FEM

5 Example 2: an advanced PINN-based multistage approach for modeling Navier–Stokes equations

5.1 Background in fluid dynamics

The Navier-Stokes equations are fundamental principles in fluid dynamics, describing the motion of viscous fluid substances. These equations model the physics of numerous phenomena of scientific and engineering interest, including weather patterns, ocean currents, pipe flow, and air flow around aircraft wings.

For an incompressible Newtonian fluid in two dimensions (x, y), the Navier-Stokes equations take the following form:

Continuity equation:

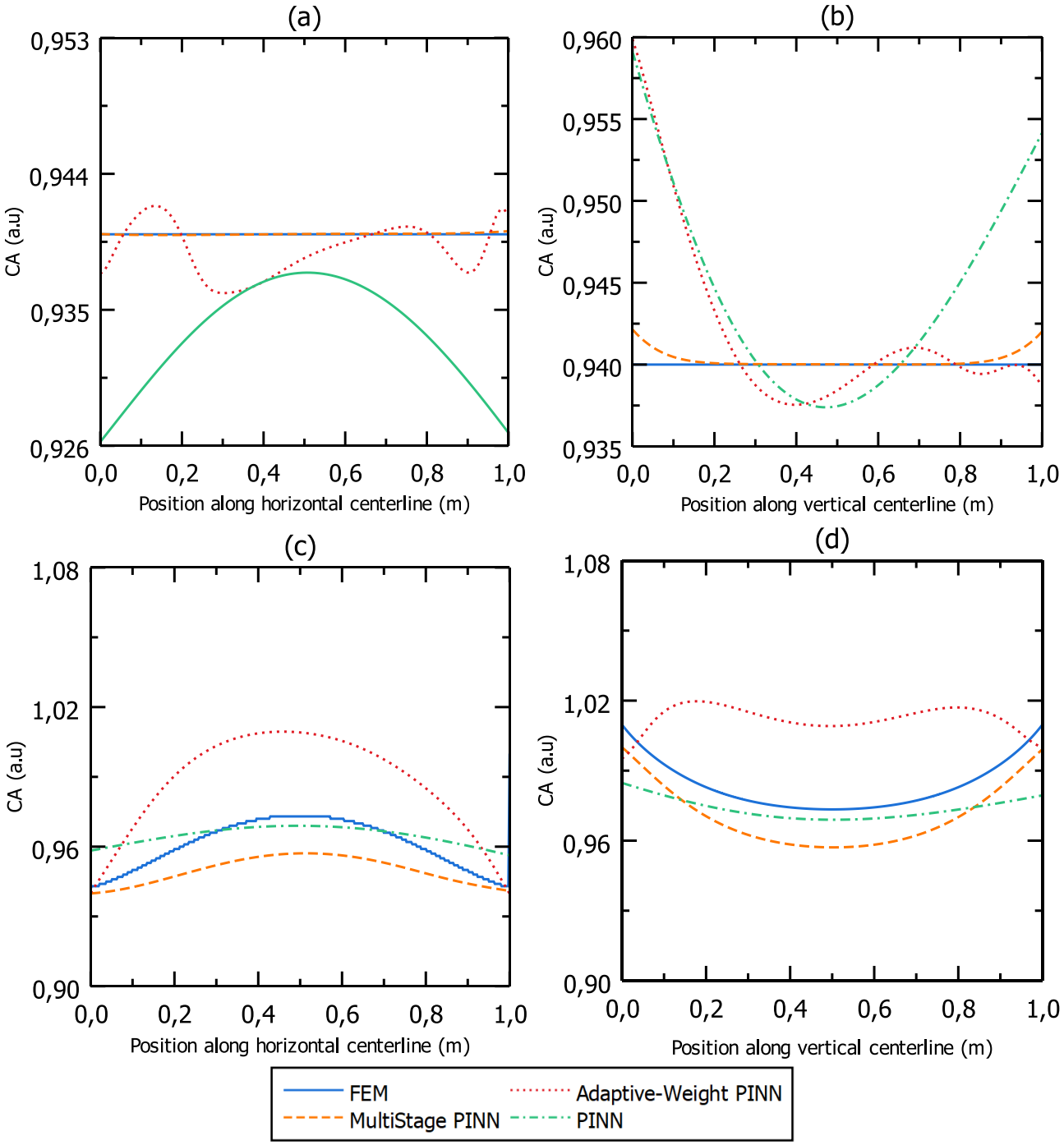

Fig. 12 A quantitative analysis of carbonyl area distributions along the centerlines using different computational methods: PINN, Adaptive Weight PINN, Multistage PINN, and FEM. **a**, **b** At t = 0 (s) while **c**, **d** At t = 1 (s)

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0 \tag{31}$$

Momentum equations:

x-direction:

$$\rho\left(\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}\right)=-\frac{\partial p}{\partial x}+\mu\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)+f_x \tag{32}$$

y-direction:

$$\rho\left(\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}\right)=-\frac{\partial p}{\partial y}+\mu\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right)+f_y \tag{33}$$

where u and v are the velocity components in the x and y directions respectively ρ is the fluid density t is time p is pressure μ is the dynamic viscosity f_x and f_y are body force components (like gravity). These PDEs are nonlinear, making them challenging to solve analytically for most real world situations. Their complexity lies in this nonlinearity, which gives rise to phenomena such as turbulence. The equations find applications across various fields, including aerospace engineering, meteorology, civil engineering, and oceanography.

In practice, the Navier–Stokes equations are often solved numerically using computational fluid dynamics (CFD) techniques [54]. These methods discretize the equations and solve them iteratively, allowing for the simulation of complex fluid flows in various scenarios.

Although standard PINNs have been previously employed to solve the Navier-Stokes equations, the objective by proposing this example is to compare the proposed multistage PINN with these established approaches and evaluate its effectiveness in handling complex fluid dynamics problems under different BCs and ICs [6].

5.2 Training algorithm for the Navier-Stokes equations

Similar to the first example, the training process for the Navier-Stokes PINN follows a curriculum learning strategy, which gradually increases the complexity of the task the neural network must learn. The curriculum is divided into three stages, each building upon the previous one to solve the full Navier-Stokes equations for incompressible fluid flow.

In the first stage, the network learns to model only the u-velocity component of the flow. This allows the PINN to establish a foundation in solving a simplified version of the momentum equation, focusing on the diffusion of the u-velocity field without considering the coupling with other velocity components or pressure.

The second stage introduces the v-velocity component, coupling it with the u-velocity model. In this stage, the network learns to solve two coupled equations simultaneously: the u-momentum and v-momentum equations. However, the pressure term is still omitted, simplifying the task compared to the full Navier-Stokes equations. This intermediate step increases the complexity as the network must now account for the interactions between the two velocity components.

The final stage incorporates the full system, including the pressure field and the continuity equation. At this point, the PINN is tasked with solving the complete coupled system of PDEs that describe incompressible fluid flow: the u-momentum equation, v-momentum equation, and the continuity equation (which enforces mass conservation). This stage introduces the pressure gradient terms in the momentum equations and requires the network to satisfy the divergence-free condition for incompressible flow.

The problem under consideration is the classical lid-driven cavity flow, a standard benchmark in computational fluid dynamics. The domain is a square cavity with sides of length $L_x = L_y = 1.0$, and the simulation time extends to $t = 1.0$(s). The boundary conditions for the velocity components (u, v) are specified as follows:

At the bottom wall $(y = 0)$ and side walls $(x = 0 \text{ and } x = L_x)$, both u and v are set to zero, implementing a no-slip condition. At the top wall $(y = L_y)$, $u = U$ and $v = 0$, where $U = 1.0$ is the constant velocity of the lid. For all walls, the pressure gradient normal to the wall is implicitly assumed to be zero (Neumann condition for pressure) through the enforcement of the incompressibility constraint.

The initial conditions at $t = 0$ are defined as: $u(x, y, 0) = U$ if $y = L_y$, and 0 otherwise; $v(x, y, 0) = 0$ everywhere; and $p(x, y, 0) = 0$ everywhere. The initial pressure is set to an arbitrary constant (zero in this case) since only pressure gradients are relevant in incompressible flow. The model architecture, hyperparameters, and collocation points are similar to those used in the asphalt aging example. The final loss function of this example and the algorithm used are summurized in Eqs. 34 and Algorithm 3, respectively.

$$
\begin{aligned}
\mathcal{L}_{\text{total}} = & \underbrace{w_{\text{bc}} \sum_{i \in \{u,v,p\}} \left(\frac{1}{N_{\text{bc}}} \sum_{j=1}^{N_{\text{bc}}} |i_j^{\text{pred}} - i_j^{\text{true}}|^2 \right)}_{\text{Boundary condition part}} \\
& + \underbrace{w_{\text{ic}} \sum_{i \in \{u,v,p\}} \left(\frac{1}{N_{\text{ic}}} \sum_{k=1}^{N_{\text{ic}}} |i_k^{\text{pred}}(t=0) - i_k^{\text{init}}|^2 \right)}_{\text{Initial condition part}} \\
& + \underbrace{w_{\text{pde}} \left(\frac{1}{N_{\text{pde}}} \sum_{l=1}^{N_{\text{pde}}} \left| \frac{\partial u_l}{\partial t} + u_l \frac{\partial u_l}{\partial x} + v_l \frac{\partial u_l}{\partial y} + \frac{\partial p_l}{\partial x} - \nu \nabla^2 u_l \right|^2 \right.}_{\text{u-momentum equation}} \\
& + \underbrace{\frac{1}{N_{\text{pde}}} \sum_{l=1}^{N_{\text{pde}}} \left| \frac{\partial v_l}{\partial t} + u_l \frac{\partial v_l}{\partial x} + v_l \frac{\partial v_l}{\partial y} + \frac{\partial p_l}{\partial y} - \nu \nabla^2 v_l \right|^2}_{\text{v-momentum equation}} \\
& + \underbrace{\left. \frac{1}{N_{\text{pde}}} \sum_{l=1}^{N_{\text{pde}}} \left| \frac{\partial u_l}{\partial x} + \frac{\partial v_l}{\partial y} \right|^2 \right)}_{\text{Continuity equation}}
\end{aligned}
\tag{34}
$$

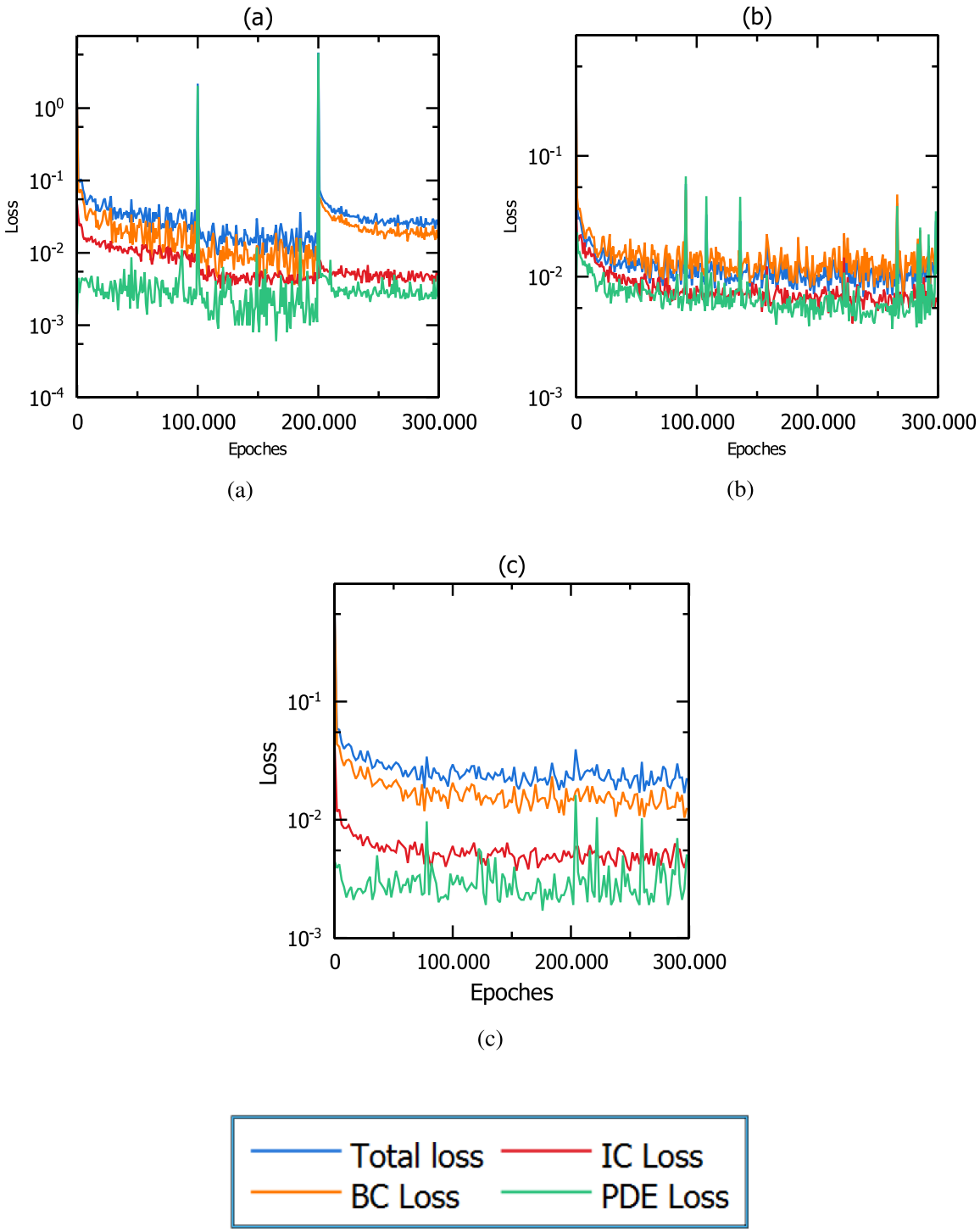

Fig. 13 The evolution of the BC, IC, PDE, and the total losses during the training process for solving the Navier-Stokes equations. **a** Multistage PINN **b** adaptive weight PINN **c** standard PINN

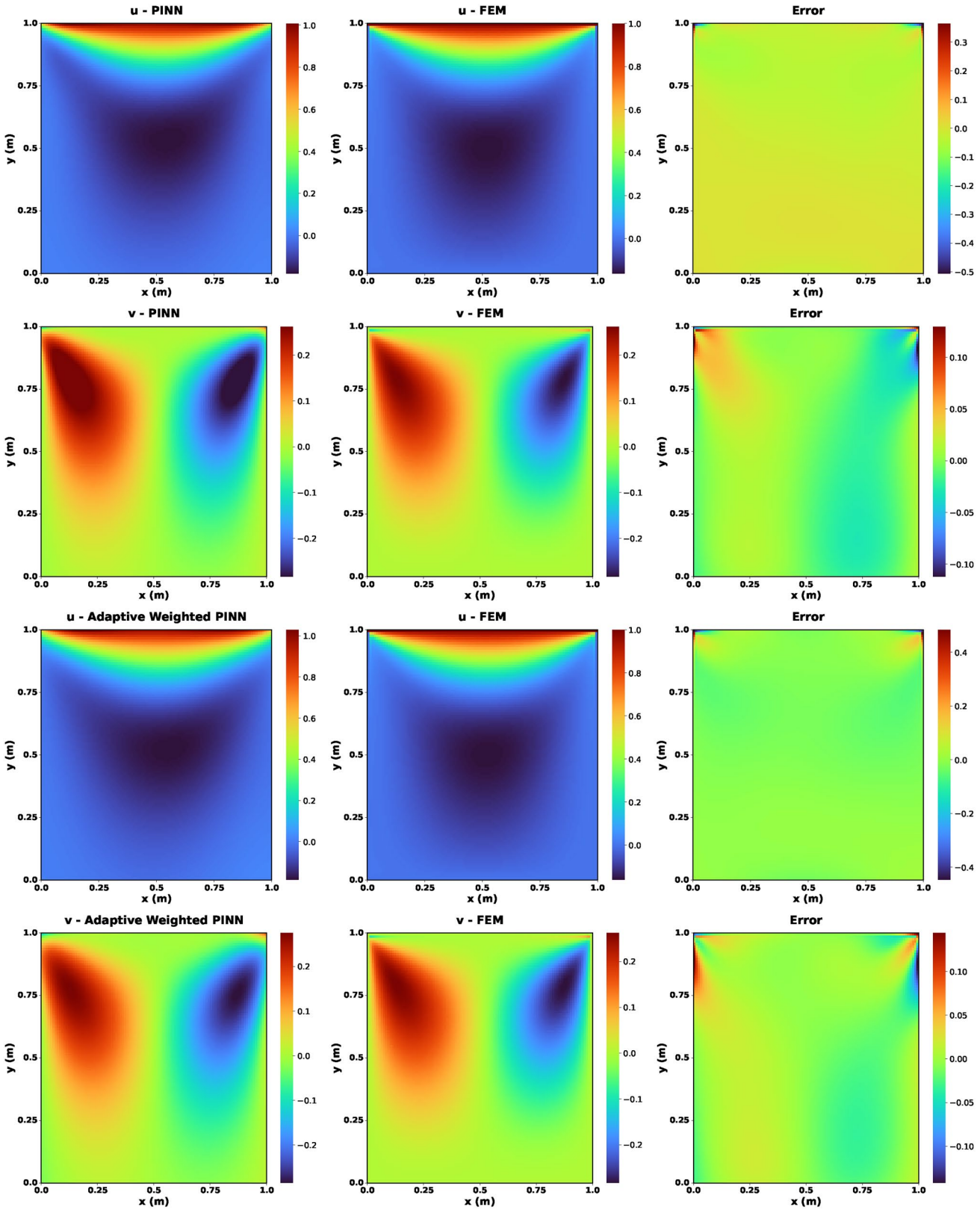

Fig. 14 A comparative visualization of velocities distributions using different computational methods: PINN, Adaptive Weight PINN, and FEM

procedure TRAINMULTISTAGEPINN($N_{epochs}, L_x, L_y, T, \nu, U$)
  Initialize PINN parameters θ
  for $stage = 1$ **to** 3 **do**
    for $epoch = 1$ **to** N_{epochs} **do**
      Sample collocation points (x, y, t)
      // Stage 1: u-velocity
      if $stage == 1$ **then**
        Forward pass: Compute $u = \text{PINN}(x, y, t; \theta)$
        Compute PDE residual: $\mathcal{R}_1 = \left|\frac{\partial u}{\partial t} - \nu \nabla^2 u\right|$
        Compute $\mathcal{L}_{BC}, \mathcal{L}_{IC}$ for u
        Compute loss $\mathcal{L}_1 = \mathcal{L}_{BC} + \mathcal{L}_{IC} + \mathcal{R}_1$ (Unweighted summation)
      end if
      // Stage 2: u-velocity and v-velocity
      if $stage == 2$ **then**
        Forward pass: Compute $u, v = \text{PINN}(x, y, t; \theta)$
        Compute PDE residuals:
        $\mathcal{R}_1 = \left|\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} - \nu \nabla^2 u\right|$
        $\mathcal{R}_2 = \left|\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} - \nu \nabla^2 v\right|$
        Compute $\mathcal{L}_{BC}, \mathcal{L}_{IC}$ for u and v
        Compute loss $\mathcal{L}_2 = \mathcal{L}_{BC} + \mathcal{L}_{IC} + \mathcal{R}_1 + \mathcal{R}_2$ (Unweighted summation)
      end if
      // Stage 3: Full Navier-Stokes system
      if $stage == 3$ **then**
        Forward pass: Compute $u, v, p = \text{PINN}(x, y, t; \theta)$
        Compute PDE residuals:
        $\mathcal{R}_1 = \left|\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + \frac{\partial p}{\partial x} - \nu \nabla^2 u\right|$
        $\mathcal{R}_2 = \left|\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + \frac{\partial p}{\partial y} - \nu \nabla^2 v\right|$
        $\mathcal{R}_3 = \left|\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right|$
        Compute $\mathcal{L}_{BC}, \mathcal{L}_{IC}$ for u, v, and p
        Compute loss $\mathcal{L}_3 = \mathcal{L}_{BC} + \mathcal{L}_{IC} + \mathcal{R}_1 + \mathcal{R}_2 + \mathcal{R}_3$ (Unweighted summation)
      end if
      Backpropagate and update θ using optimizer (Adam)
    end for
  end for
  return Trained PINN parameters θ
end procedure

Algorithm 3 Multistage PINN Training Algorithm for Navier-Stokes Equations

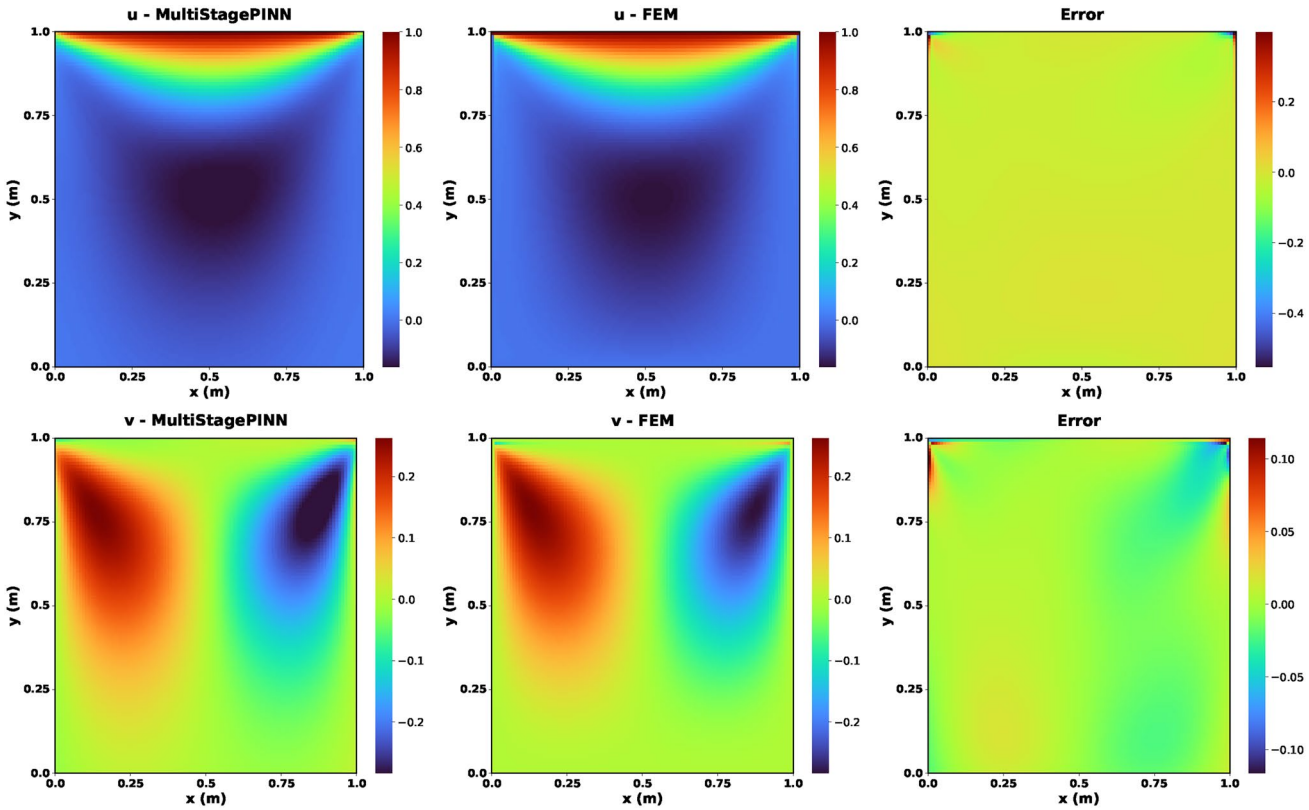

Fig. 15 A comparative visualization of velocities distributions using Multistage PINN, and FEM

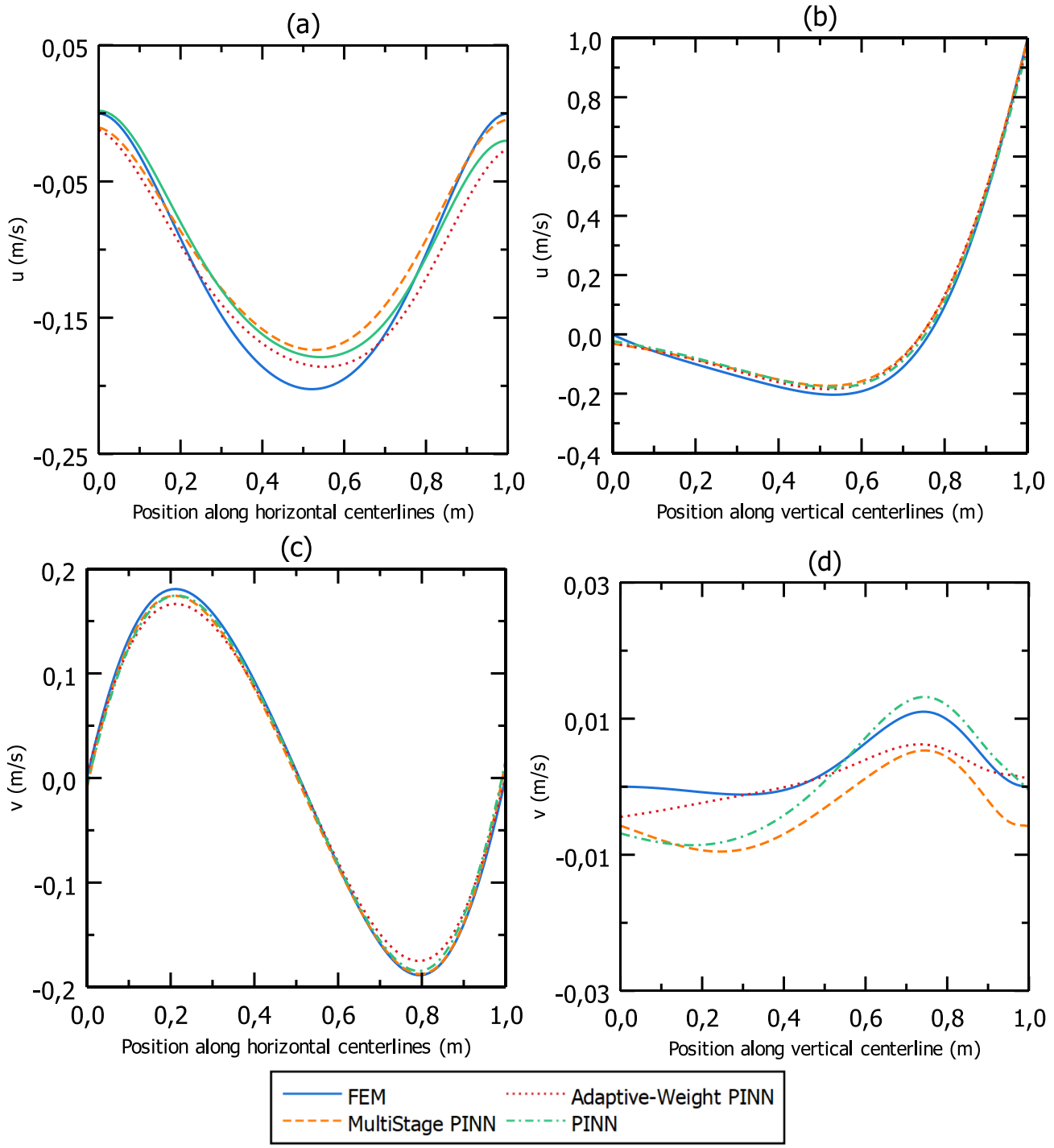

Fig. 16 A quantitative analysis of the velocities distributions along the centerlines using different computational methods: Multistage PINN, adaptive weight PINN, standard PINN and FEM. (a) and (b) at t = 0 (s) while (c) and (d) at t = 1 (s)

5.3 Results: Naviar Stokes model

Figure 13 illustrates the evolution of losses during the training process for the three PINN variants. The multistage PINN exhibits a sharp spikes in PDE loss at 100,000 and 200,000 epochs, suggesting that the gradual increase in PDE complexity allows the model to explore a wider solution space. Despite this increase, the total loss and BC loss show a consistent downward trend, indicating overall convergence. The adaptive weight PINN demonstrates a smooth convergence among all the losses. This smooth convergence suggests that dynamically adjusting the weights

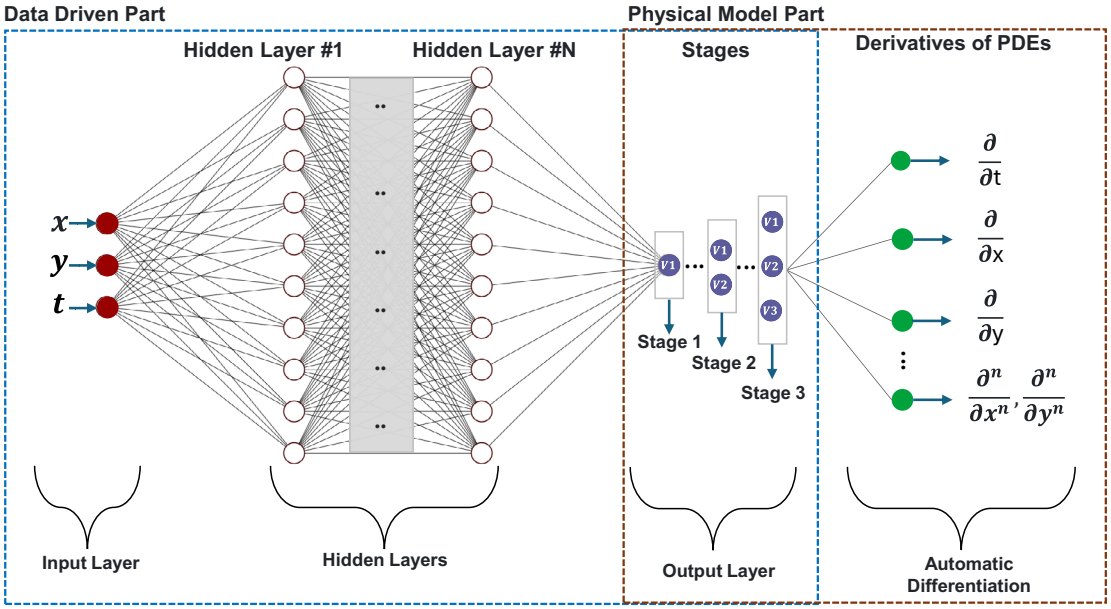

Fig. 17 Schematic representation of the multistage PINN architecture adapted in this study where v_1, v_2, v_3 are the dependent variables for each PDE used in the adapted examples

Table 5 Comparison of training times for different PINN Methods across two examples

Method	Example 1	Example 2
Multistage PINN (s)	5364	5900
Adaptive Weight PINN (s)	13,060	14,205
Standard PINN (s)	72,792	80,590

of different loss components effectively balances their contributions throughout training. The standard PINN shows characteristics intermediate between the multistage and adaptive weight PINNs. While it achieves convergence, the final loss values are generally higher than those of the other two methods, indicating potentially lower accuracy in the final predictions.

For the standard PINN, weight-adaptive PINN (Fig. 14), the u velocity predictions capture the overall pattern of the FEM results. The v-velocity predictions show good agreement in general distribution but exhibit some discrepancies near boundaries and corners. The multistage PINN (Fig. 15) demonstrates the highest accuracy among the PINN variants where the u-velocity predictions are remarkably similar to FEM results, with the error map showing minimal and uniformly distributed discrepancies across the domain. The v-velocity predictions also closely match FEM results, accurately capturing complex flow patterns. While slightly higher errors are observed near corners and edges for v-velocity, the overall agreement is excellent.

Figure 16 presents a quantitative analysis of velocities distributions along the centerlines using different computational methods. The plots show the velocity components (u and v) at different times and positions. In general, all PINN methods consistently capture the overall trends and patterns of both u and v velocities along both centerlines. However, the more complex the flow pattern (as seen in Fig 16d), the more pronounced the differences between the methods become, with advanced PINN methods showing their strength in capturing intricate details.

A notable advantage of the multistage PINN is its computational efficiency. It is reported to be faster than the other methods, including the standard PINN and adaptive weight PINN. This significant speed improvement, combined with its superior accuracy, makes the multistage PINN particularly attractive for solving complex fluid dynamics problems.

6 Discussion

Figure 17 shows a schematic representation of the multistage PINN architecture adapted in this study. The multistage PINN approach offers a significant improvement in both accuracy and computational efficiency compared to other PINN methods. As shown in Table 5, it achieves superior results in less than half the time required by the adaptive weight PINN and less than a tenth of the time needed by the standard PINN. This efficiency does not come at the cost of accuracy. In fact, the multistage approach consistently shows the smallest maximum differences from the FEM solution across all variables. The computational efficiency of the multistage PINN is attributed to its staged

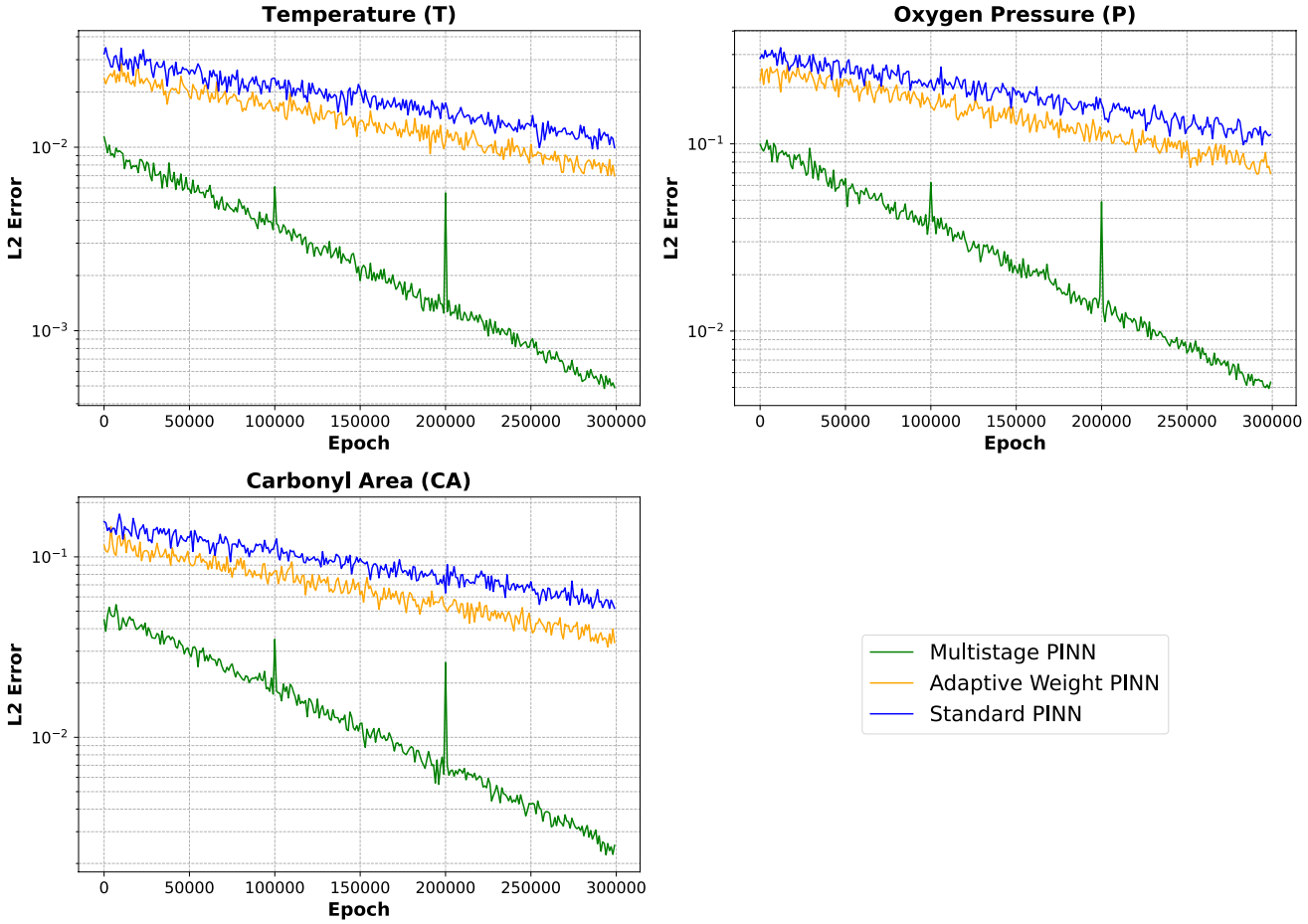

Fig. 18 L2 error analysis of PINN predictions compared to FEM for the asphalt aging example

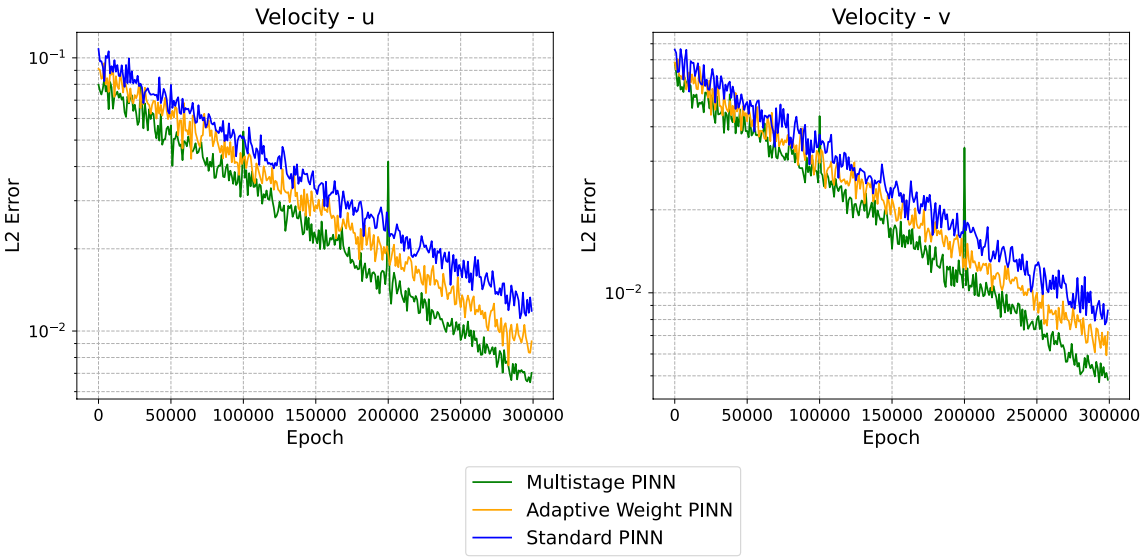

Fig. 19 L2 error analysis of PINN predictions compared to FEM for the lid-driven cavity flow

approach to optimizing the loss function. Conversely, a causal PINN method prioritizes temporal causality in time-dependent PDEs [55]. Though effective for such cases, the multistage PINN approach better addresses coupled multiphysics systems, as demonstrated by the asphalt aging and Navier–Stokes examples. For instance, in the asphalt aging example, the model first incorporates terms related to temperature diffusion, then integrates oxygen pressure, and finally adds carbonyl area formation. While the standard PINN and adaptive weight PINN struggled to accurately model the final behavior, the multistage PINN achieved high prediction accuracy with minimal error. The presence of exponential terms in k_f and k_c makes it more challenging for standard PINNs to accurately capture the final behavior of aging asphalt materials.

In the second example, although both the standard and adaptive weight approaches demonstrated high accuracy, the multistage PINN maintained the fastest training time (5900 seconds compared to 14,205 seconds and 80,590 seconds for adaptive weight and standard PINNs, respectively.

To further quantify the accuracy of the multistage PINN, the L2 error norms between the predicted solutions and FEM solutions were calculated for both example problems. Figures 18 and 19 present the L2 error analysis for (a) the asphalt aging problem and the lid-driven cavity flow problem, respectively. For the asphalt aging example, the L2 errors for T, P, and CA are consistently lower for the multistage PINN compared to the standard and adaptive weight PINNs across the simulation period. The reduced L2 error reflects the ability of the multistage approach to effectively capture the coupled interactions between temperature, oxygen diffusion, and chemical reactions through its curriculum learning strategy. Similarly, in the lid-driven cavity flow example, the L2 errors for the u and v velocities components at demonstrate that the multistage PINN achieves closer alignment with the FEM solution than its counterparts.

The choice of a fixed learning rate of 0.001 across all stages in both examples highlights the robustness of this approach in handling coupled multiphysics systems. While the fully coupled optimization in stage 3 increases the complexity of the loss function, the curriculum learning strategy combined with weight transfer between stages effectively stabilizes the training process, allowing the fixed learning rate to remain effective. This contrasts with observations in more complex coupled PDE systems, where a smaller learning rate may be necessary to prevent instability as the loss landscape becomes more challenging. The success of the fixed learning rate in this study is likely due to the moderate coupling strength and the staged approach, which avoids abrupt transitions in the optimization problem. However, for highly nonlinear or strongly coupled systems, future work could explore adaptive learning rate schedules or self-adaptive PINN techniques to further enhance training stability and efficiency.

Despite these advantages, the slight tendency to underestimate values and the variability in performance across different physical variables indicate that there is still potential for further refinement of the multistage approach. Future work could focus on optimizing the staging strategy and exploring hybrid approaches that combine the strengths of multiple PINN variants.

7 Applications and future work

7.1 Implications for pavement engineering and future research directions

The multistage PINN approach developed in this study offers several advantages. Firstly, the PINN employs a mesh-free approach, eliminating the need for complex meshing procedures that are often required in FEM. This significantly reduces pre-processing time and avoids issues related to mesh quality and refinement. Secondly, for complex, coupled systems like asphalt aging, PINN can be more computationally efficient than FEM, especially for long-term simulations. The neural network, once trained, can provide rapid predictions across the entire space-time domain without the need for time-stepping procedures typical in FEM. Finally, the PINN approach allows for adaptive resolution in both space and time without the need for remeshing, which is often required in FEM for achieving higher accuracy in specific regions.

The flexibility of the PINN approach opens up possibilities for future research. One promising direction is the incorporation of fillers into the model. This could involve adding new variables to represent filler properties (e.g., particle distribution), modifying the governing equations to account for filler-binder interactions, and exploring how fillers affect diffusion coefficients and reaction kinetics in the aging process. Such extensions would provide a more comprehensive understanding of how material composition affects long-term pavement performance.

7.2 Potential applications in other scientific domains

The concept of gradually increasing problem complexity is well established in solving nonlinear problems. This application offers several advantages in the field of scientific machine learning and computational physics. Firstly, PINNs face distinct challenges compared to traditional numerical methods, particularly in balancing multiple physics-informed loss terms and ensuring stable convergence.

The multistage approach specifically addresses these PINN-related issues. Secondly, PINNs have shown promise in solving individual PDEs but their application to coupled multiphysics systems remains relatively unexplored. Lastly, the significant reduction in training time (over 90%) achieved through this approach is noteworthy in the computationally demanding context of multiphysics simulations.

For these reasons, the PINN approach could be adapted to model various multiphysics problems. For example, it can be implemented in material degradation processes, such as corrosion in metals or polymer degradation [56, 57]. Similar to asphalt aging, these phenomena involve coupled physical and chemical processes occurring over long time scales. The method's ability to handle interactions between diffusion, chemical reactions, and mechanical stress makes it well-suited for predicting the long-term performance of diverse materials under complex environmental conditions.

In civil engineering, the method could be applied to model the aging of concrete structures. This process shares similarities with asphalt aging, involving factors such as carbonation, chloride ingress, and the effects of cyclic loading [58, 59]. The PINN approach could provide more accurate long-term predictions of concrete durability, accounting for the coupled effects of chemical reactions and mechanical stresses.

Future research could focus on adapting the multistage PINN approach to these related fields, potentially leading to improved predictive modeling across a range of materials and infrastructure applications. This could contribute to the development of more durable and sustainable engineered systems in various domains. Future research could also investigate the optimization of hyperparameters, such as the learning rate, to adapt the multistage PINN approach to more complex multiphysics systems, potentially incorporating adaptive learning rate schedules or self-adaptive techniques to improve stability and convergence in stage 3.

8 Conclusions

This study introduced a novel multistage PINN approach for modeling complex, coupled multiphysics phenomena. The method was demonstrated through its application to two distinct problems: asphalt aging prediction and lid-driven cavity flow. The multistage PINN demonstrated superior performance in predicting the evolution of multiple interconnected variables in these complex systems. Compared to standard PINN and adaptive weight PINN methods, the proposed approach achieved higher accuracy and computational efficiency, reducing training time by over 90% while maintaining better alignment with traditional numerical solutions.

The success of the multistage approach in capturing the interdependencies between multiple physical processes, from material degradation to fluid dynamics, highlights its potential for modeling a wide range of complex, coupled systems across various scientific and engineering domains. The significant reduction in training time without compromising accuracy is particularly promising for practical applications where computational efficiency is crucial. The method's ability to handle both materials science and fluid dynamics problems demonstrates its diversity and broad applicability.

Future improvements to the multistage PINN approach could include optimizing the staging strategy to better balance variable interactions and integrating adaptive learning rate schedules to enhance training stability for highly nonlinear systems. Furthermore, exploring hybrid methods that combine multistage PINNs with other advanced PINN variants, such as self-adaptive or multifidelity approaches, could further boost accuracy and efficiency.

Acknowledgements This publication is part of the project 'A multiscale approach towards future road infrastructure: How to design sustainable paving materials?' (project number 18148) of the research program NWO Talent Programme Veni AES 2020, which is awarded to A.V. and financed by the Dutch Research Council (NWO).

Author contributions M.K. and V.C. developed the primary algorithm, applied it to the two examples, and wrote the main manuscript text. A.V. provided primary supervision of the work and manuscript revision. C.K. and S.E. offered general supervision of the project.

Data availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Keyes DE, McInnes LC, Woodward C, Gropp W, Myra E, Pernice M, Bell J, Brown J, Clo A, Connors J et al (2013) Multiphysics simulations: challenges and opportunities. Int J High Perform Comput Appl 27(1):4–83. https://doi.org/10.1177/1094342012468181
2. Geiser J (2015) Recent advances in splitting methods for multiphysics and multiscale: theory and applications. J Algorithms Comput Tech 9(1):65–93
3. Geiser J (2013) Coupled Navier Stokes molecular dynamics simulation: theory and applications based on iterative operator-splitting methods. Comput Fluids 77:97–111. https://doi.org/10.1016/j.compfluid.2013.02.012
4. Rohan E, Naili S (2020) Homogenization of the fluid-structure interaction in acoustics of porous media perfused by viscous fluid. Z Angew Math Phys 71:1–28. https://doi.org/10.1007/s00033-020-01361-1
5. Hirthe EM, Graf T (2012) Non-iterative adaptive time-stepping scheme with temporal truncation error control for simulating variable-density flow. Adv Water Resour 49:46–55. https://doi.org/10.1016/j.advwatres.2012.07.021
6. Raissi M, Perdikaris P, Karniadakis GE (2019) Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J Comput Phys 378:686–707. https://doi.org/10.1016/j.jcp.2018.10.045
7. Karniadakis GE, Kevrekidis IG, Lu L, Perdikaris P, Wang S, Yang L (2021) Physics-informed machine learning. Nat Rev Phys 3(6):422–440. https://doi.org/10.1038/s42254-021-00314-5
8. Cicirello A (2024) Physics-enhanced machine learning: a position paper for dynamical systems investigations. arXiv preprint arXiv:2405.05987https://doi.org/10.48550/arXiv.2405.05987
9. Saidaoui H, Espath L, Tempone R (2024) Deep Nurbs admissible physics-informed neural networks. Eng Comput. https://doi.org/10.1007/s00366-024-02040-9
10. Wang S, Teng Y, Perdikaris P (2021) Understanding and mitigating gradient flow pathologies in physics-informed neural networks. SIAM J Sci Comput 43(5):3055–3081. https://doi.org/10.1137/20M1318043
11. Wang S, Yu X, Perdikaris P (2022) When and why Pinns fail to train: a neural tangent kernel perspective. J Comput Phys 449:110768. https://doi.org/10.1016/j.jcp.2021.110768
12. Jagtap AD, Kawaguchi K, Karniadakis GE (2020) Adaptive activation functions accelerate convergence in deep and physics-informed neural networks. J Comput Phys 404:109136. https://doi.org/10.1016/j.jcp.2019.109136
13. Wei W, Fu L-Y (2022) Small-data-driven fast seismic simulations for complex media using physics-informed Fourier neural operators. Geophysics 87(6):435–446. https://doi.org/10.1190/geo2021-0573.1
14. Nair S, Walsh TF, Pickrell G, Semperlotti F (2024) Multiple scattering simulation via physics-informed neural networks. Eng Comput. https://doi.org/10.1007/s00366-024-02038-3
15. Jagtap AD, Kharazmi E, Karniadakis GE (2020) Conservative physics-informed neural networks on discrete domains for conservation laws: applications to forward and inverse problems. Comput Methods Appl Mech Eng 365:113028. https://doi.org/10.1016/j.cma.2020.113028
16. Cai S, Wang Z, Wang S, Perdikaris P, Karniadakis GE (2021) Physics-informed neural networks for heat transfer problems. J Heat Transfer 143(6):060801. https://doi.org/10.1115/1.4050542
17. Dwivedi V, Srinivasan B (2020) Physics informed extreme learning machine (pielm) a rapid method for the numerical solution of partial differential equations. Neurocomputing 391:96–118. https://doi.org/10.1016/j.neucom.2019.12.099
18. Geneva N, Zabaras N (2020) Modeling the dynamics of pde systems with physics-constrained deep auto-regressive networks. J Comput Phys 403:109056. https://doi.org/10.1016/j.jcp.2019.109056
19. Tartakovsky AM, Marrero CO, Perdikaris P, Tartakovsky GD, Barajas-Solano D (2020) Physics-informed deep neural networks for learning parameters and constitutive relationships in subsurface flow problems. Water Resour Res 56(5):2019–026731. https://doi.org/10.1029/2019WR026731
20. Manav M, Molinaro R, Mishra S, De Lorenzis L (2024) Phase-field modeling of fracture with physics-informed deep learning. Comput Methods Appl Mech Eng 429:117104. https://doi.org/10.1016/j.cma.2024.117104
21. Haghighat E, Amini D, Juanes R (2022) Physics-informed neural network simulation of multiphase poroelasticity using stress-split sequential training. Comput Methods Appl Mech Eng 397:115141. https://doi.org/10.1016/j.cma.2022.115141
22. Pang G, Lu L, Karniadakis GE (2019) fpinns: fractional physics-informed neural networks. SIAM J Sci Comput 41(4):2603–2626. https://doi.org/10.1137/18M1229845
23. Hu L, Zhang J, Xiang Y, Wang W (2020) Neural networks-based aerodynamic data modeling: a comprehensive review. IEEE Access 8:90805–90823. https://doi.org/10.1109/ACCESS.2020.2993562
24. Lin L-C, Chen S-J, Yu H-Y (2023) Connecting structural characteristics and material properties in phase-separating polymer solutions: phase-field modeling and physics-informed neural networks. Polymers 15(24):4711. https://doi.org/10.3390/polym15244711
25. Wang S, Wang H, Perdikaris P (2021) On the eigenvector bias of Fourier feature networks: From regression to solving multi-scale pdes with physics-informed neural networks. Comput Methods Appl Mech Eng 384:113938. https://doi.org/10.1016/j.cma.2021.113938
26. Quarteroni A, Quarteroni S (2009) Numerical models for differential problems, vol 2. Springer, Berlin (**10.1007/978-3-319-49316-9**)
27. Farhat C, Zee KG, Geuzaine P (2006) Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity. Comput Methods Appl Mech Eng 195(17–18):1973–2001. https://doi.org/10.1016/j.cma.2004.11.031
28. Monaco S, Apiletti D (2023) Training physics-informed neural networks: one learning to rule them all? Results Eng 18:101023. https://doi.org/10.1016/j.rineng.2023.101023
29. Wang Y, Xu C, Yang M, Zhang J (2023) Less emphasis on hard regions: curriculum learning of Pinns for singularly perturbed convection-diffusion-reaction problems. East Asian J Appl Math. https://doi.org/10.4208/eajam.2023-062.170523
30. Patra A, Batra R, Chandrasekaran A, Kim C, Huan TD, Ramprasad R (2020) A multi-fidelity information-fusion approach to machine learn and predict polymer bandgap. Comput Mater Sci 172:109286. https://doi.org/10.1016/j.commatsci.2019.109286
31. Anticev J, Aghdaei A, Cheng W, Feng Z (2024) Sgm-pinn: Sampling graphical models for faster training of physics-informed neural networks. In: Proceedings of the 61st ACM/IEEE Design Automation Conference, pp. 1–6. https://doi.org/10.1145/3649329.3656521
32. Wu C, Zhu M, Tan Q, Kartha Y, Lu L (2023) A comprehensive study of non-adaptive and residual-based adaptive sampling for

physics-informed neural networks. Comput Methods Appl Mech Eng 403:115671. https://doi.org/10.1016/j.cma.2022.115671
33. Pu R, Feng X (2022) Physics-informed neural networks for solving coupled stokes Darcy equation. Entropy. https://doi.org/10.3390/e24081106
34. Dekhovich A, Sluiter MH, Tax DM, Bessa MA (2024) ipinns: incremental learning for physics-informed neural networks. Eng Comput. https://doi.org/10.1007/s00366-024-02010-1
35. Kharazmi E, Zhang Z, Karniadakis GEM (2021) hp-vpinns: variational physics-informed neural networks with domain decomposition. Comput Methods Appl Mech Eng 374:113547. https://doi.org/10.1016/j.cma.2020.113547
36. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, Desmaison A, Kopf A, Yang E, DeVito Z, Raison M, Tejani A, Chilamkurthy S, Steiner B, Fang L, Bai J, Chintala S (2019) Pytorch: an imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc.,???
37. Liu J, Yan K, Liu J (2018) Rheological properties of warm mix asphalt binders and warm mix asphalt binders containing polyphosphoric acid. Int J Pavement Res Tech 11(5):481–487. https://doi.org/10.1016/j.ijprt.2018.03.005
38. Zhang Z, Fang Y, Yang J, Li X (2022) A comprehensive review of bio-oil, bio-binder and bio-asphalt materials: their source, composition, preparation and performance. J Traffic Transp Eng (English Edition) 9(2):151–166. https://doi.org/10.1016/j.jtte.2022.01.003
39. Wang Z, Ye F (2020) Experimental investigation on aging characteristics of asphalt based on rheological properties. Constr Build Mater 231:117158. https://doi.org/10.1016/j.conbuildmat.2019.117158
40. Tarsi G, Varveri A, Lantieri C, Scarpas A, Sangiorgi C (2018) Effects of different aging methods on chemical and rheological properties of bitumen. J Mater Civ Eng 30(3):04018009. https://doi.org/10.1061/(ASCE)MT.1943-5533.0002206
41. Camargo I.G.d.N, Hofko B, Mirwald J, Grothe H (2020) Effect of thermal and oxidative aging on asphalt binders rheology and chemical composition. Materials 13(19):4438. https://doi.org/10.3390/ma13194438
42. Jing R, Varveri A, Liu X, Scarpas A, Erkens S (2019) Laboratory and field aging effect on bitumen chemistry and rheology in porous asphalt mixture. Transp Res Rec 2673(3):365–374. https://doi.org/10.1177/0361198119833362
43. Wang J, Wang T, Hou X, Xiao F (2019) Modelling of rheological and chemical properties of asphalt binder considering Sara fraction. Fuel 238:320–330. https://doi.org/10.1016/j.fuel.2018.10.126
44. Petersen JC, Glaser R (2011) Asphalt oxidation mechanisms and the role of oxidation products on age hardening revisited. Road Mater Pavement Des 12(4):795–819. https://doi.org/10.1080/14680629.2011.9713895
45. Luo X, Gu F, Lytton RL (2015) Prediction of field aging gradient in asphalt pavements. Transp Res Rec 2507(1):19–28. https://doi.org/10.3141/2507-0
46. Liu M, Lunsford K, Davison R, Glover C, Bullin J (1996) The kinetics of carbonyl formation in asphalt. AIChE J 42(4):1069–1076. https://doi.org/10.1002/aic.690420417
47. Omairey EL, Gu F, Zhang Y (2021) An equation-based multiphysics modelling framework for oxidative ageing of asphalt pavements. J Clean Prod 280:124401. https://doi.org/10.1016/j.jclepro.2020.124401
48. Kohout J (2021) Modified Arrhenius equation in materials science, chemistry and biology. Molecules 26(23):7162. https://doi.org/10.3390/molecules26237162
49. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980, https://doi.org/10.48550/arXiv.1412.6980
50. Markidis S (2021) The old and the new: can physics-informed deep-learning replace traditional linear solvers? Front Big Data 4:669097. https://doi.org/10.3389/fdata.2021.669097
51. Lu L, Meng X, Mao Z, Karniadakis GE (2021) Deepxde: a deep learning library for solving differential equations. SIAM Rev 63(1):208–228 10.48550/arXiv.1907.04502
52. Omairey EL, Zhang Y, Soenen H, Carbonneau X (2023) Parametric analysis and field validations of oxidative ageing in asphalt pavements using multiphysics modelling approaches. Int J Pavement Eng 24(2):2020267. https://doi.org/10.1080/10298436.2021.2020267
53. McClenny LD, Braga-Neto UM (2023) Self-adaptive physics-informed neural networks. J Comput Phys 474:111722. https://doi.org/10.1016/j.jcp.2022.111722
54. Sheng W (2020) A revisit of Navier Stokes equation. Eur J Mech B Fluids 80:60–71. https://doi.org/10.1016/j.euromechflu.2019.12.005
55. Wang S, Sankaran S, Perdikaris P (2024) Respecting causality for training physics-informed neural networks. Comput Methods Appl Mech Eng 421:116813. https://doi.org/10.1016/j.cma.2024.116813
56. Scheiner S, Hellmich C (2009) Finite volume model for diffusion- and activation-controlled pitting corrosion of stainless steel. Comput Methods Appl Mech Eng 198(37):2898–2910. https://doi.org/10.1016/j.cma.2009.04.012
57. Zhang L-W, Ye J-Y (2023) Cavitation impact damage of polymer: a multi-physics approach incorporating phase-field. Comput Methods Appl Mech Eng 417:116420. https://doi.org/10.1016/j.cma.2023.116420
58. Gawin D, Pesavento F, Schrefler BA (2003) Modelling of hygrothermal behaviour of concrete at high temperature with thermochemical and mechanical material degradation. Comput Methods Appl Mech Eng 192(13):1731–1771. https://doi.org/10.1016/S0045-7825(03)00200-7
59. Uthaman S, Vishwakarma V (2023) Assessment of causes and consequences of concrete deterioration and its remediation. J Build Eng 79:107790. https://doi.org/10.1016/j.jobe.2023.107790