Journal of Computational Physics 538 (2025) 114161

Contents lists available at ScienceDirect

Journal of Computational Physics

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

A coordinate transformation-based physics-informed neural networks for hyperbolic conservation laws

Yuanhong Chen [a,b], Zhen Gao [a,b,*], Jan S. Hesthaven [c], Yifan Lin [d], Xiang Sun [a,b]

[a] School of Mathematical Sciences, Ocean University of China, Qingdao postcode, PR China
[b] Laboratory of Marine Mathematics, Ocean University of China, Qingdao postcode, PR China
[c] Chair of Computational Mathematics and Simulation Science, École Polytechnique F édérale de Lausanne, 1015 city, Switzerland
[d] School of Physical and Mathematical Sciences, Nanjing Tech University, Nanjing postcode, PR China

ARTICLE INFO

Keywords:
Physics-informed neural network
Coordinate transformation
Hyperbolic conservation laws
Characteristic curves

ABSTRACT

Hyperbolic conservation laws play a critical role in various fields, including aerodynamics, physics, and oceanography. However, traditional physics-informed neural networks (PINNs), despite their remarkable capabilities in solving partial differential equations (PDEs), often struggle to accurately resolve these problems. To address this challenge, a coordinate transformation-based PINN (CT-PINN) algorithm for hyperbolic conservation laws is proposed, which uses coordinate transformations along characteristic curves to prevent the generation and propagation of discontinuities. The coordinate transformation transforms subdomains divided along characteristic curves into regular domains governed by the corresponding transformed PDEs. The CT-PINN framework simultaneously learns the characteristic curves and the transformed solutions by optimizing a loss function that integrates both the transformed PDEs and the characteristic equations. Due to the equivalence between solutions in the transformed and original domains, predictions in arbitrary coordinates can be obtained without the need for interpolation. Moreover, different PINN architectures can be applied for each subdomain, with hyperparameters flexibly adjusted to enhance accuracy. The proposed method has been evaluated on a range of hyperbolic conservation laws, including the convection equation, the Burgers equation, the shallow water wave equation, the traffic flow equation and the Euler equation. The results demonstrate that CT-PINN can accurately solve the characteristic equation and PDEs, and effectively capture shock waves without transition points, outperforming traditional numerical approaches.

1. Introduction

The hyperbolic conservation laws find widespread application in fields such as aerodynamics, physics, and oceanography, which are particularly useful for simulating wave phenomena, traffic phenomena, turbulent phenomena, and other complex physical processes [1–3]. Therefore, the accurate resolution of these issues holds significant practical significance. In particular, the solutions of the nonlinear hyperbolic conservation laws typically exhibit discontinuous and smooth small-scale structures. Accurately capturing the discontinuities while avoiding pseudo-oscillations or over-smoothing phenomena poses a significant challenge for numerical solution. To this end, a number of effective numerical methods have been developed for solving hyperbolic conservation laws [4–6],

---

[*] Corresponding author.

E-mail addresses: chenyuanhong@stu.ouc.edu.cn (Y. Chen), zhengao@ouc.edu.cn (Z. Gao), Jan.Hesthaven@epfl.ch (J.S. Hesthaven), linyifan@njtech.edu.cn (Y. Lin), sunxiang@ouc.edu.cn (X. Sun).

https://doi.org/10.1016/j.jcp.2025.114161
Received 11 October 2024; Received in revised form 1 February 2025; Accepted 3 June 2025

including the total variation diminishing schemes [7], the Godunov schemes [8], the essential non-oscillatory scheme [9], the weighted essential non-oscillatory scheme [10,11] and its variants [12–14], the discontinuous Galerkin method [15,16], the entropy conserving schemes [17] among others. All of these methods rely heavily on the mesh discretisation of the solution region, and are constrained by the curse of dimensionality.

The rapid development of neural network (NN) algorithms and computer hardware has led to the widespread application of NN-based methods across various fields [18–20], including computer vision, natural language processing, data mining, and reinforcement learning, achieving remarkable results. These methods have also emerged as a promising alternative to classical numerical methodologies for PDEs [21–23]. The Deep Ritz method [24] employs a variational formulation to derive solutions by minimizing the general energy function. The PINN [23] is trained to derive solutions to PDEs by minimizing the sum of the residuals of the PDEs and the initial/boundary conditions. The deep operator network (DeepONet) [25] is proposed for PDEs with varying parameters, which establish a continuous mapping between the inputs and outputs of a given PDE. The randomized NN with Petrov-Galerkin methods, proposed in Shang et al. [26], uses randomized NNs to approximate unknown functions and allows flexibility in the choice of test functions. Furthermore, numerous other NN-based methodologies have been proposed for PDEs, including the deep Galerkin method [27], weak adversarial networks [28], etc. The NN-based methods are straightforward to implement and provide real-time predictive capabilities.

Despite the increasing number of methods for solving PDEs, the PINN has garnered significant interest due to its capacity to solve PDEs with minimal or no training data, relying instead on the governing equations and initial/boundary conditions [29–33]. The PINNs are straightforward to implement, as they incorporate physical laws directly into the loss function, with derivatives computed through automatic differentiation [34], eliminating the need for mesh discretization. Despite its advantages, PINNs face certain limitations and practical challenges [35–39], particularly when solving PDEs with low regularity, where the approximated solutions may fail to converge to the exact ones [40]. These challenges arise because PINN necessitates derivatives of a particular order, and significant changes in these derivatives can destabilize the neural network. Nevertheless, PINN has been shown to be an effective approach for inverse problems, especially those involving the development of flow structures [41]. However, the traditional PINN methodologies can face difficulties in forward problems, particularly those involving shockwaves in hyperbolic conservation equations. In the vicinity of the shockwave, increasing or decreasing the gradient can result in an increase in the loss function, which may lead to an unsuitable training behaviour [42].

To develop the PINN to learn the states of interest for the hyperbolic conservation laws, in Mao et al. [41], the training data is clustered around the high gradient regions, with the objective of capturing discontinuities for contact discontinuities and two-dimensional oblique shockwave problems. Nevertheless, an understanding of the locations of discontinuities is of paramount importance. The conservative PINN is proposed in Jagtap et al. [43], where the different PINNs for subdomains that do not overlap are employed to solve the conservation laws. It is observed that the implementation of an appropriate domain decomposition enhances the accuracy of the results, which also necessitates the priori knowledge. Patel et al. [44] establish a connection between classical finite volume methods and PINN by integrating entropy inequalities into NNs. A PINN with equation weights [42] imposes different constraints on the shockwave and smooth regions yields remarkable results in capturing the shockwave. However, accurately capturing the shockwave location is challenging. The Lagrangian PINN [45] represents a promising avenue to address the challenges posed by convection-dominated equations, the PDEs are reformulated within the Lagrangian frame, which can reduce the complexity and assist with training of convection-dominated PINNs. Subsequently, a two-branch network is employed to simultaneously solve for the state variables on the characteristic and the low-dimensional characteristic curves. However, this approach requires interpolation between Eulerian and Lagrangian coordinates, and it inevitably encounters mesh management challenges.

The focus of this work is on addressing the challenging problem of solving hyperbolic conservation laws using PINN. By employing the characteristic curve, a significant physical quantity, we develop a PINN model based on a coordinate transformation [46,47] that effectively addresses the challenge of discontinuities, which are inherently difficult to capture and track. The CT-PINN comprises two branch networks, one of which is employed to learn the evolution of the specified characteristic curves, thereby guiding the coordinate transformation. The inputs to this network are the initial point of the curve and the time. The second branch network is utilized to learn the transformed solution, with inputs are the transformed coordinates along the characteristic curve. Instead of requiring prior knowledge of characteristic speed, the proposed method simultaneously trains two branch networks to identify the characteristic curves and transformed solutions. The CT-PINN's loss function is defined by the residuals of both the characteristic equation and the transformed PDEs. To address discontinuities at the initial interface, we introduce new strategies by modifying initial conditions and incorporating the Rankine-Hugoniot conditions, Generalized Riemann Invariants, and isentropic law into the loss function to learn the physical quantities in unknown regions. During the prediction phase, for arbitrary spatio-temporal coordinates, the transformed coordinates along the predicted characteristic curves are fed into the second branch to derive solutions without the need for interpolation. Furthermore, the regions separated by characteristic curves can be mapped as regular regions and subjected to different equations, which affords the option of having a separate network on each subdomain, thereby greatly enhancing the flexibility of network architecture and approximate accuracy. As demonstrated by a number of hyperbolic conservation laws, including PDEs such as the Burgers equation, the shallow water wave equation, the traffic flow equation and the Euler equation, the proposed method can allow for the avoidance of the generation and propagation of discontinuities in the transformed solution. The CT-PINN is capable of efficiently handling hyperbolic conservation equations that accurately capture discontinuities.

The paper is structured as follows, Section 2 introduces the problem and reviews the traditional PINN method. In Section 3, the difficulty of PINN in capturing shock waves is demonstrated using the Burgers equation and the implementation of the complete

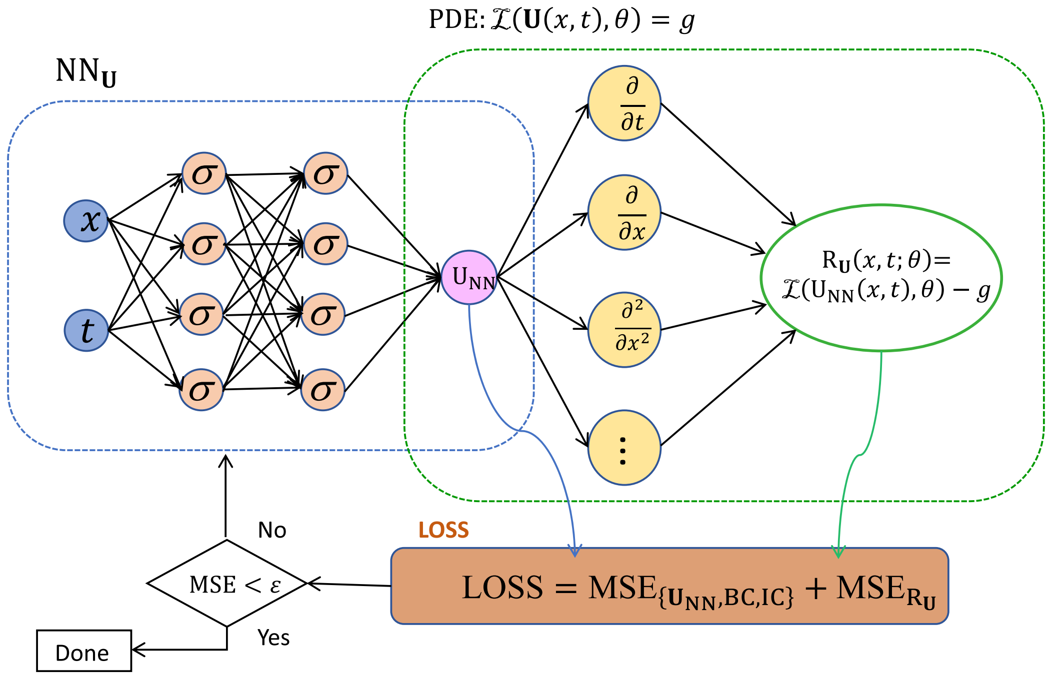

Fig. 1. Achitecture of the PINN.

algorithm for CT-PINN is discussed. One- and two-dimensional test cases are solved to verify the proposed method in Section 4. The conclusions and future directions are discussed in Section 5.

2. Problem setup and PINN review

The hyperbolic conservation laws can usually be expressed in the following form

$$\mathbf{U}_t + \nabla \cdot \mathbf{F}(\mathbf{U}) = 0, (x,t) \in \mathcal{D} \times [0,T], \tag{1}$$

where $\mathbf{U}$ is the conservative variables with respect to time t and space x, $\mathbf{F}(\mathbf{U})$ is the flux function, $\mathcal{D} \subset \mathbb{R}^d$ denotes the simulation domain. For different problems, the corresponding initial and boundary conditions are indicated to solve Eq. (1)

$$\begin{aligned} &\mathrm{BCs}(\mathbf{U},x,t) = 0, (x,t) \in \partial\mathcal{D} \times [0,T], \\ &\mathrm{ICs}(\mathbf{U},x,0) = 0, (x,t) \in \mathcal{D}. \end{aligned} \tag{2}$$

In addition to the traditional numerical algorithms that have been developed for solving hyperbolic conservation laws that have attracted considerable attention, PINN, a popular and straightforward machine learning method for solving PDEs, has also been employed to address such problems. The schematic diagram of the PINN for time-dependent PDEs $\mathcal{L}(\mathbf{U}(x,t),\theta) = g$ is shown in Fig. 1. The most significant distinction between the PINN and the traditional data-driven NN is the formulation of the loss function, which is dependent on the control equation and the initial/boundary conditions. The calculation of the loss function requires the sampling of the spatio-temporal domain to generate the N_f residual points $\{x_f^{(i)}, t_f^{(i)}\}_{i=1}^{N_f}$, the N_{ic} initial training points $\{x_{ic}^{(i)}, 0\}_{i=1}^{N_{ic}}$, and the N_{bc} boundary training points $\{x_{bc}^{(i)}, t_{bc}^{(i)}\}_{i=1}^{N_{bc}}$. Then, the loss function is determined by the initial/boundary conditions and the control equation, as follows

$$\mathrm{LOSS} = \mathrm{MSE}_{\{\mathbf{U}_{\mathrm{NN}},\mathrm{BC,IC}\}} + \mathrm{MSE}_{\mathrm{R}_{\mathbf{U}}}, \tag{3}$$

and each term is given by

$$\mathrm{MSE}_{\mathrm{R}_{\mathbf{U}}} = eta_r \frac{1}{N_f} \sum_{i=1}^{N_f} \left| \mathcal{L}(\mathbf{U}_{\mathrm{NN}}\left(x_f^{(i)}, t_f^{(i)}\right), \theta) - g \right|^2,$$

$$\begin{aligned} \mathrm{MSE}_{\{\mathbf{U}_{\mathrm{NN}},\mathrm{BC,IC}\}} &= \eta_{ic} \frac{1}{N_{ic}} \sum_{i=1}^{N_{ic}} \left| \mathrm{ICs}(\mathbf{U}_{\mathrm{NN}}(x_{ic}^{(i)}, 0), x_{ic}^{(i)}, 0) \right|^2 \\ &+ \eta_{bc} \frac{1}{N_{bc}} \sum_{i=1}^{N_{bc}} \left| \mathrm{BCs}(\mathbf{U}_{\mathrm{NN}}(x_{bc}^{(i)}, t_{bc}^{(i)}), x_{bc}^{(i)}, t_{bc}^{(i)}) \right|^2, \end{aligned}$$

where $\mathbf{U}_{\mathrm{NN}}$ represents the output of the PINN, which is used to approximate the state variable $\mathbf{U}$, trained by selecting an appropriate optimizer to minimize the loss function. η_r, η_{ic} and η_{bc} are the residual, initial and boundary weights, respectively. Consequently, the output of the PINN is constrained by the physical principles that permit it to approximate the PDE solutions without the necessity of training data, in contrast to the data-driven deep learning algorithms.

Traditional numerical solvers require discretizing the spatio-temporal domain and approximating the derivative using solutions at the discrete points. In contrast to the numerical solvers, PINN employs automatic differentiation to compute the derivative values, thereby enabling the mapping of spatio-temporal coordinates to PDE solutions without spatio-temporal domain discretization. Once the PINN has been trained, one can rapidly evaluate solution values at any given spatio-temporal point.

Although the PINNs are highly effective in numerous applications, they are often infeasible due to the fact that they identify solutions that differ significantly from the true solutions, particularly when addressing discontinuous or large gradient problems. The issue cannot typically be resolved by selecting an appropriate network structure, modifying the optimization algorithm, or tuning the hyperparameters of the NN, as the specific order of derivative required for NN training is necessary. As a result, solving hyperbolic conservation equations using PINNs inevitably poses a significant challenge, given that the regularity of solutions to such problems is often poor. The solutions of hyperbolic conservation laws may be discontinuous, regardless of the smoothness of the initial conditions. Furthermore, the impact of the discontinuous interfaces makes it difficult not only to capture the solution near the discontinuities but also adversely affects convergence within the smooth region. However, characteristic curves can be employed to capture the position of discontinuous interfaces. In this paper, the CT-PINN is proposed by using the coordinate transformation based on characteristic curves, which can be effectively applied to the hyperbolic conservation laws.

3. Proposed methods

The scalar nonlinear conservation law in Eq. (1) can be written as

$$u_t + f(u)_x = 0, \quad (x,t) \in [0,L] \times [0,T]. \tag{4}$$

For example, in the linear advection equation, $f(u) = au$, and in the inviscid Burgers equation, $f(u) = u^2/2$. The hyperbolic conservation law exhibits several important properties that offer invaluable insights into the study of the behaviour of the solution. Of particular note is the existence of characteristic curves, denoted as $\pi(t)$, along which the PDE can be transformed into an ordinary differential equation (ODE). Furthermore, the characteristic curve satisfies the ODE

$$\pi'(t) = f'(u) \triangleq \lambda(u), \quad \pi(0) = x_0 \in (0,L), \tag{5}$$

where $\lambda(u)$ represents the characteristic speed, which is a function of the solution u, and x_0 is the initial point.

By selecting an appropriate initial point x_0, a characteristic curve, $\pi(t)$, can be derived from Eq. (5), This subsequently facilitates the establishment of a coordinate transformation as follows:

$$\tilde{x}(t) = \begin{cases} \dfrac{\pi(0)}{\pi(t)} x, & \text{if } 0 \le x < \pi(t), \\ L - \dfrac{L-\pi(0)}{L-\pi(t)}(L-x), & \text{if } \pi(t) < x \le L. \end{cases} \tag{6}$$

This transformation maps subdomain $[0,\pi(t)]$ to $[0,\pi(0)]$ and subdomain $[\pi(t),L]$ to $[\pi(0),L]$, and it can be observed that $[0,\pi(0)]$ and $[\pi(0),L]$ are both regular regions that are independent of $\pi(t)$. Let $\tilde{u}(\tilde{x},t) = u(x,t)$, the derivatives of u can be calculated by the derivative of $\tilde{u}$:

$$\frac{\partial u}{\partial t} = \begin{cases} \dfrac{\partial \tilde{u}}{\partial t} - \lambda(\tilde{u}(\pi(0),t)) \dfrac{\tilde{x}}{\pi(t)} \dfrac{\partial \tilde{u}}{\partial \tilde{x}}, & \text{if } 0 < \tilde{x} < \pi(0), \\ \dfrac{\partial \tilde{u}}{\partial t} - \lambda(\tilde{u}(\pi(0),t)) \dfrac{L-\tilde{x}}{L-\pi(t)} \dfrac{\partial \tilde{u}}{\partial \tilde{x}}, & \text{if } \pi(0) < \tilde{x} < L, \end{cases} \tag{7}$$

and

$$\frac{\partial u}{\partial x} = \begin{cases} \dfrac{\pi(0)}{\pi(t)} \dfrac{\partial \tilde{u}}{\partial \tilde{x}}, & \text{if } 0 < \tilde{x} < \pi(0), \\ \dfrac{L-\pi(0)}{L-\pi(t)} \frac{\partial \tilde{u}}{\partial \tilde{x}}, & \text{if } \pi(0) < \tilde{x} < L. \end{cases} \tag{8}$$

Then Eq. (4) can be converted in the new coordinate system to

$$\begin{cases} \dfrac{\partial \tilde{u}}{\partial t} + \left(\lambda(u) \dfrac{\pi(0)}{\pi(t)} - \lambda(\tilde{u}(\pi(0),t)) \dfrac{\tilde{x}}{\pi(t)} \right) \dfrac{\partial \tilde{u}}{\partial \tilde{x}} = 0, & \text{if } 0 < \tilde{x} < \pi(0), \\ \dfrac{\partial \tilde{u}}{\partial t} + \left(\lambda(u) \dfrac{L-\pi(0)}{L-\pi(t)} - \lambda(\tilde{u}(\pi(0),t)) \dfrac{L-\tilde{x}}{L-\pi(t)} \right) \dfrac{\partial \tilde{u}}{\partial \tilde{x}} = 0, & \text{if } \pi(0) < \tilde{x} < L. \end{cases} \tag{9}$$

The aforementioned presents a mapping related to a characteristic curve. In fact, depending on the initial conditions and characteristic speed, one can select different characteristic curves to avoid the generation and shift of discontinuities or to simplify the structure of

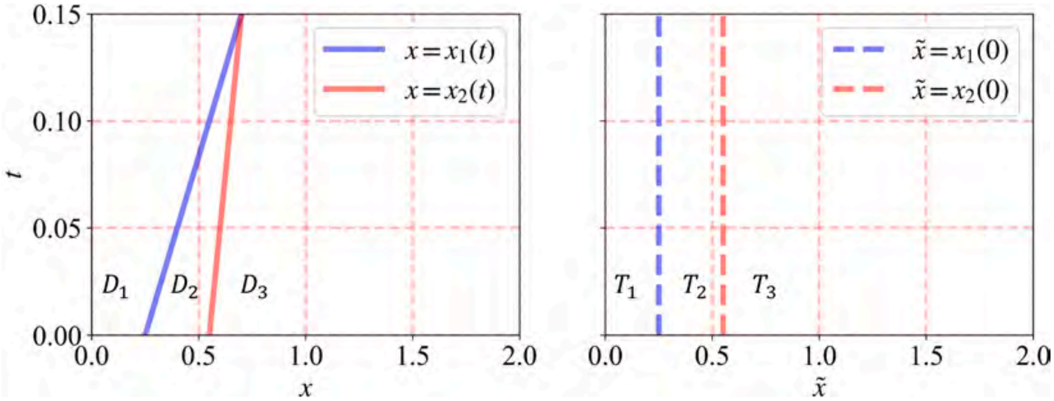

Fig. 2. Time evolution of the physical (left) and the transformed (right) domains.

the solution. For example, consider the Burgers equation with the following initial condition:

$$u(x,0) = \begin{cases} 3, & \text{if } 0 \le x < \pi_1(0), \\ 3 - 2\dfrac{x - \pi_1(0)}{\pi_2(0) - \pi_1(0)}, & \text{if } \pi_1(0) \le x < \pi_2(0), \\ 1, & \text{if } \pi_2(0) \le x \le L, \end{cases} \tag{10}$$

where $\pi_1(0) = 0.25$ and $\pi_2(0) = 0.55$. In this case, the characteristic velocity is $\lambda(u) = u$, and thus, higher values of $u(x,0)$ will propagate at a faster speed than lower values of $u(x,0)$. The propagation of $u(x,0)$ in interval $[0, \pi_1(0)]$ is faster than that in interval $[\pi_1(0), \pi_2(0)]$, and interval $[\pi_1(0), \pi_2(0)]$ is in turn faster than interval $[\pi_2(0), L]$. Therefore, interval $[\pi_1(0), \pi_2(0)]$ is a compressed region that becomes steeper and narrower over time, and there exists a time t_c when the characteristic curves emanating from $\pi_1(0)$ intersect the characteristic curves emanating from $\pi_2(0)$, leading to the generate of discontinuous interface. Therefore, the characteristic curves with the initial values $\pi_1(0)$ and $\pi_2(0)$ are selected and the following coordinate transformation is performed for time $t < t_c$:

$$\tilde{x}(t) = \begin{cases} \dfrac{\pi_1(0)}{\pi_1(t)} x, & \text{if } 0 \le x < \pi_1(t), \\ \pi_1(0) + \dfrac{\pi_2(0) - \pi_1(0)}{\pi_2(t) - \pi_1(t)}\big(x - \pi_1(t)\big), & \text{if } \pi_1(t) \le x < \pi_2(t), \\ L - \dfrac{L - \pi_2(0)}{L - \pi_2(t)}(L - x), & \text{if } \pi_2(t) \le x \le L. \end{cases} \tag{11}$$

The transformation maps

$$\begin{aligned} D_1 &:= \{(x,t) \mid 0 \le x < \pi_1(t), 0 \le t < t_c\} \longrightarrow T_1 := \{(\tilde{x},t) \mid 0 \le \tilde{x} < \pi_1(0), 0 \le t < t_c\}, \\ D_2 &:= \{(x,t) \mid \pi_1(t) \le x < \pi_2(t), 0 \le t < t_c\} \longrightarrow T_2 := \{(\tilde{x},t) \mid \pi_1(0) \le \tilde{x} < \pi_2(0), 0 \le t < t_c\}, \\ D_3 &:= \{(x,t) \mid \pi_2(t) \le x \le L, 0 \le t < t_c\} \longrightarrow T_3 := \{(\tilde{x},t) \mid \pi_2(0) \le \tilde{x} \le L, 0 \le t < t_c\}, \end{aligned}$$

which are shown in Fig. 2, transforms the Burgers equation into

$$\begin{cases} \dfrac{\partial \tilde{u}}{\partial t} + \left(\tilde{u}(\tilde{x},t)\dfrac{\pi_1(0)}{\pi_1(t)} - \tilde{u}\big(\pi_1(0),t\big)\dfrac{\tilde{x}}{\pi_1(t)}\right)\dfrac{\partial \tilde{u}}{\partial \tilde{x}} = 0, & \text{if } 0 < \tilde{x} < \pi_1(0), \\ \dfrac{\partial \tilde{u}}{\partial t} = 0, & \text{if } \pi_1(0) \le \tilde{x} < \pi_2(0), \\ \dfrac{\partial \tilde{u}}{\partial t} + \left(\tilde{u}(\tilde{x},t)\dfrac{L - \pi_2(0)}{L - \pi_2(t)} - \tilde{u}\big(\pi_2(0),t\big)\dfrac{L - \tilde{x}}{L - \pi_2(t)}\right)\dfrac{\partial \tilde{u}}{\partial \tilde{x}} = 0, & \text{if } \pi_2(0), 0 \le \tilde{x} < L. \end{cases} \tag{12}$$

When $t \ge t_c$, the characteristic curve emanating from the discontinuity is selected and the transformation of the form Eq. (6) is used. The structures of the solutions in the physical domain and the transform domain at varying times are presented in Fig. 3. It can be observed that the transformed solution is maintained at the initial solution. This is because $\tilde{u}$ remains constant in the regions T_1 and T_3, so $\frac{\partial \tilde{u}}{\partial \tilde{x}} = 0$. Combined with Eq. (12), in the transformed domain, $\frac{\partial \tilde{u}}{\partial \tilde{t}} = 0$, meaning the solution in the transformed domain is time-independent.

There are many numerical methods that consider the important physical quantity of characteristic curves, however, such methods, often require interpolation between Eulerian and Lagrangian grids, and inevitably face difficulties in grid management, especially when the gradient changes drastically or shockwave occurs, the spatial step size needs particularly small otherwise the numerical schemes unstable or grid entanglement. To avoid this issue, we apply PINN to identify the solution in the new coordinates. The

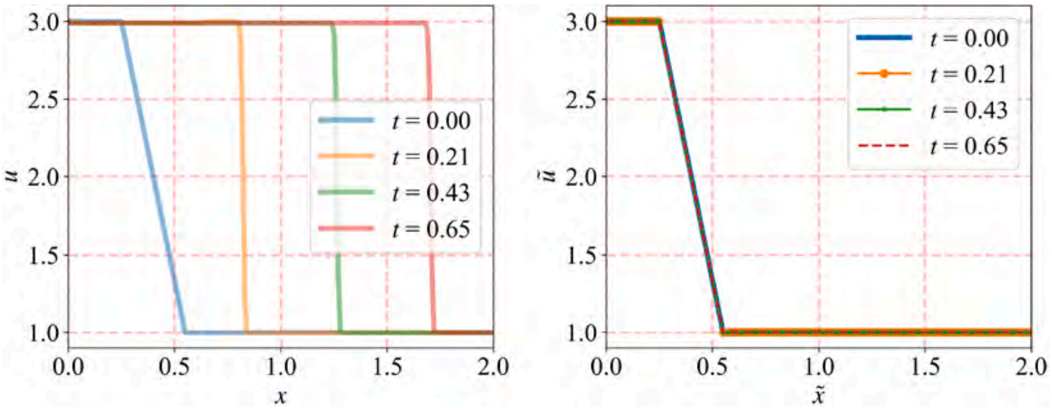

Fig. 3. The solutions in the physical domain (left) and the transformed domain (right) at different times.

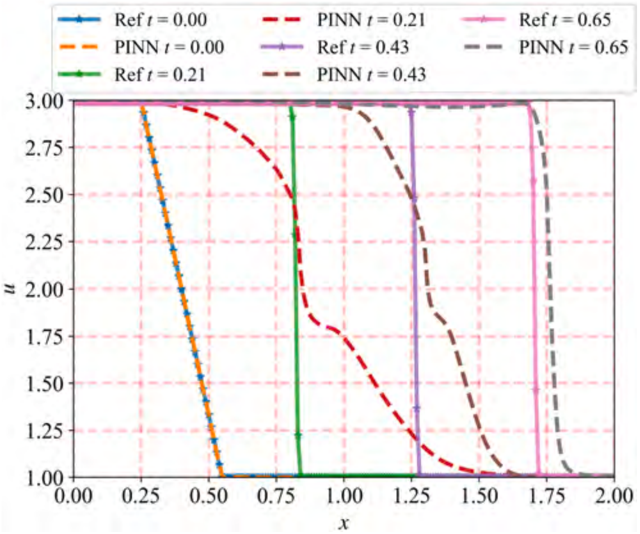

Fig. 4. The reference solutions and the corresponding predicted solutions by PINN at the test time instants $t = 0, 0.21, 0.43, 0.65$.

PINN samples the residuals in the spatio-temporal domain to learn the mapping from the coordinate points to the function values, effectively avoiding the interpolation between Eulerian and Lagrangian grids. Furthermore, PINN replaces numerical differentiation with auto-differentiation, so that the computation of derivatives is unaffected by the mesh discretization.

The generation and movement of discontinuities present a significant challenge to the training of the network in PINN, which are primary reason for its unavailability. We demonstrate this by solving the aforementioned Burgers equation using PINN. After setting and training the network with the loss function based on Eq. (3), an approximation of the PDE solution is obtained, as depicted in Fig. 4. It is evident that there is a significant discrepancy between the exact solution and the predicted solution by PINN. However, for hyperbolic problems, this issue can be addressed through the implementation of appropriate transformations along the characteristic curves. Therefore, in order to develop PINN effectively in hyperbolic problems, this paper proposes the CT-PINN, in which PINNs are used to learn the evolution of the characteristic curves in addition to the solution of the equations. The structure of CT-PINN is illustrated in Fig. 5 and comprises two principal components. One is the branch network $\mathbf{U}_{\mathrm{NN}}$, which is employed to learn solutions following coordinate transformation. The other is the branch network π_{NN}, which enables the model of the variation of characteristic curves, has the following form

$$\pi_{\mathrm{NN}} = \phi(x,t) * t + x,$$

where $\phi(x,t)$ is the output of the last hidden layer.

Define $\mathrm{R}_x := \pi'(t) - \lambda(u)$, the loss function of the CT-PINN is

$$\mathrm{LOSS}_{\mathrm{CT-PINN}} = \mathrm{MSE}_{\{\tilde{\mathbf{U}}_{\mathrm{NN}},\mathrm{BC},\mathrm{IC}\}} + \mathrm{MSE}_{\mathrm{R}_{\tilde{\mathbf{U}}}} + \mathrm{MSE}_{\mathrm{R}_x}, \tag{13}$$

where $\mathrm{MSE}_{\mathrm{R}_x} = \eta_{rx} \frac{1}{N_x} \sum_{i=1}^{N_x} \left| \mathrm{R}_x\left(x^{(i)}, t^{(i)}\right) \right|^2$ is the residual associated with the characteristic equation.

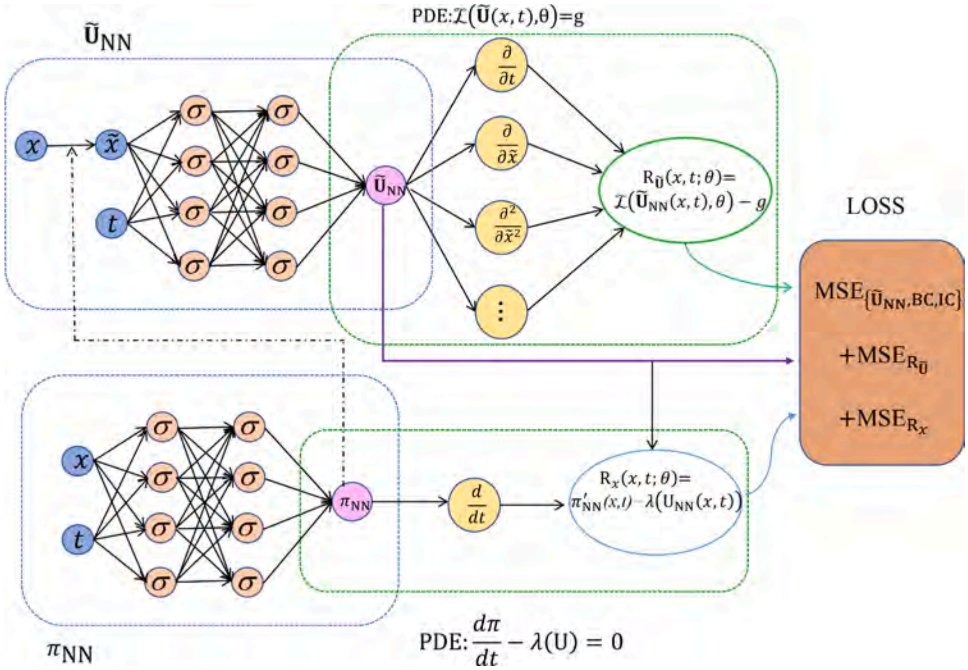

Fig. 5. Achitectures of CT-PINN, π_{NN} is the approximated characteristic curve, which is employed to map x to $\tilde{x}$. $\tilde{\mathbf{U}}_{\text{NN}}$ is a function of $\tilde{x}$ and t and is applied to approximate the transformed solution.

For the hyperbolic system described by Eq. (1), the Jacobian matrix $\frac{\partial \mathbf{F}(\mathbf{U})}{\partial \mathbf{U}}$ is characterized by real eigenvalues and corresponding right eigenvectors, which allows the system to be reformulated in the characteristic form:

$$\frac{\partial \mathbf{W}}{\partial t}+\mathbf{\Lambda}(\mathbf{U})\frac{\partial \mathbf{W}}{\partial x}=\mathbf{0}, \tag{14}$$

where $\mathbf{W}=[w_1,w_2,\ldots,w_n]$ denotes the characteristic variables obtained through a transformation of $\mathbf{U}$, and $\mathbf{\Lambda}(\mathbf{U})$ represents a diagonal matrix containing the eigenvalues. Consequently, the system (14) is fully decoupled, allowing each variable to be treated as a scalar problem that can be solved using the aforementioned method. For nonlinear systems, however, the coordinate transformation can be applied directly to the variables of the governing equation Eq. (1), with the eigenvalues computed to guide the coordinate transformation.

In practical applications, the discontinuities contained in the initial conditions of hyperbolic conservation laws can lead to complex wave structures, including shock waves, contact discontinuities, and rarefaction waves. These wave structures pose significant challenges in capturing their evolutions. To address these challenges, the following discussion focuses on the scenario where the initial conditions include discontinuities. Without loss of generality, we consider the case of a single discontinuity in the initial conditions. As an illustrative example, consider the one-dimensional Euler system:

$$\frac{\partial}{\partial t}\begin{bmatrix}\rho\\ \rho u\\ E\end{bmatrix}+\frac{\partial}{\partial x}\begin{bmatrix}\rho u\\ \rho u^2+p\\ u(E+p)\end{bmatrix}=0, \tag{15}$$

with the initial conditions:

$$(\rho(x,0),u(x,0),p(x,0))=\begin{cases}(\rho_L,u_L,p_L), & \text{if } 0\le x<x^d,\\ (\rho_R,u_R,p_R), & \text{if } x^d<x\le L,\end{cases}$$

where ρ is the density, u is the velocity, p is the pressure, and $E=\frac{p}{\gamma-1}+\frac{1}{2}\rho u^2$ represents the total energy per unit volume. The parameter γ denotes the ratio of specific heats. The eigenvalues of the flux Jacobian matrix for the Euler system (15) are given by:

$$\lambda_1=u-c,\quad \lambda_2=u,\quad \lambda_3=u+c,$$

where $c=\sqrt{\gamma p/\rho}$ is the speed of sound.

The solution structure of the original variables $\mathrm{V}=(\rho,u,p)$ consists of three waves originating from the point x^d, as illustrated in Fig. 6. Each wave corresponds to a characteristic value and divides the domain into four states. From left to right, these states are V_{L} (left-side data), $\mathrm{V}_{\mathrm{L}}^*$ (left star region), $\mathrm{V}_{\mathrm{R}}^*$ (right star region), and V_{R} (right-side data), with the star region divided into $\mathrm{V}_{\mathrm{L}}^*$ and $\mathrm{V}_{\mathrm{R}}^*$ by the middle wave. The middle wave is a contact discontinuity, while the left and right waves can manifest as either shock waves or rarefaction waves.

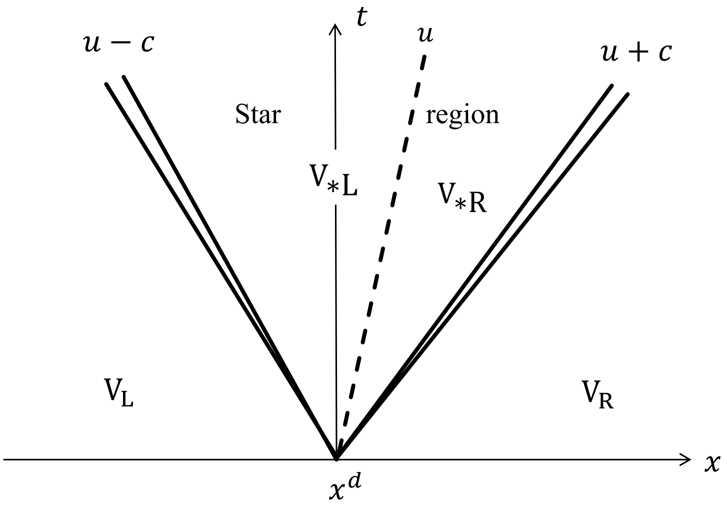

Fig. 6. Structure of the solution of the Riemann problem on the $x-t$ plane for the one-dimensional Euler equations.

Consider the following initial conditions:

$$(\rho(x,0),u(x,0),p(x,0)) = \begin{cases} (1,0,1), & \text{if } 0 \le x < x^d, \\ (0.125,0,0.1), & \text{if } x^d < x \le L. \end{cases} \tag{16}$$

In this case, the left wave is a rarefaction wave, and the right wave is a shock wave. To perform CT-PINN, four characteristic lines $\pi_i(t) = \pi_i(0) + s_i t$ are selected for the coordinate transformation, where s_i represents the wave speed. It is important to note that these speeds are not necessarily equal to the characteristic speeds defined by the eigenvalues. The transformation maps:

$$\begin{aligned}
D_1 &:= \{(x,t) \mid 0 \le x < \pi_1(t), 0 \le t < T\} \longrightarrow T_1 := \{(\tilde{x},t) \mid 0 \le \tilde{x} < \pi_1(0), 0 \le t < T\}, \\
D_2 &:= \{(x,t) \mid \pi_1(t) \le x < \pi_2(t), 0 \le t < T\} \longrightarrow T_2 := \{(\tilde{x},t) \mid \pi_1(0) \le \tilde{x} < \pi_2(0), 0 \le t < T\}, \\
D_3 &:= \{(x,t) \mid \pi_2(t) \le x < \pi_3(t), 0 \le t < T\} \longrightarrow T_3 := \{(\tilde{x},t) \mid \pi_2(0) \le \tilde{x} < \pi_3(0), 0 \le t < T\}, \\
D_4 &:= \{(x,t) \mid \pi_3(t) \le x < \pi_4(t), 0 \le t < T\} \longrightarrow T_4 := \{(\tilde{x},t) \mid \pi_3(0) \le \tilde{x} < \pi_4(0), 0 \le t < T\}, \\
D_5 &:= \{(x,t) \mid \pi_4(t) \le x \le L, 0 \le t < T\} \longrightarrow T_5 := \{(\tilde{x},t) \mid \pi_4(0) \le \tilde{x} \le L, 0 \le t < T\}.
\end{aligned}$$

In fact, $\pi_1(0) = \pi_2(0) = \pi_3(0) = \pi_4(0)$, causing the middle section of the domain to collapse into the discontinuity point $x = x^d$, which is represented as a single value. This single point is insufficient for training a PINN model to accurately predict the solutions in T_2, T_3, and T_4.

To address this issue, one can select small values for dx_1, dx_2, dx_3, and dx_4, such that $\pi_1(0) = dx_1$, $\pi_2(0) = \pi_1(0) + dx_2$, $\pi_3(0) = \pi_2(0) + dx_3$, and $\pi_4(0) = \pi_3(0) + dx_4$. Subsequently, the initial condition can be modified by replacing the discontinuity between u_L and u_R with a linear variation between the two fixed points $\pi_1(0)$ and $\pi_2(0)$ [48]. Furthermore, constant values can be defined for the intervals $[\pi_2(0), \pi_3(0)]$ and $[\pi_3(0), \pi_4(0)]$ as follows:

$$(\rho(x,0),u(x,0),p(x,0)) = \begin{cases} (\rho_L, u_L, p_L), & \text{if } 0 \le x < \pi_1(0), \\ \Bigg(\rho_L + \dfrac{\rho_{*L} - \rho_L}{\pi_2(0) - \pi_1(0)}(x - \pi_1(0)), & \\ \quad u_L + \dfrac{u_* - u_L}{\pi_2(0) - \pi_1(0)}(x - \pi_1(0)), & \\ \quad p_L + \dfrac{p_* - p_L}{\pi_2(0) - \pi_1(0)}(x - \pi_1(0)) \Bigg), & \text{if } \pi_1(0) \le x < \pi_2(0), \\ (\rho_{*L}, u_*, p_*), & \text{if } \pi_2(0) \le x < \pi_3(0), \\ (\rho_{*R}, u_*, p_*), & \text{if } \pi_3(0) \le x < \pi_4(0), \\ (\rho_R, u_R, p_R), & \text{if } \pi_4(0) \le x \le L. \end{cases}$$

In the case that the solution for the star point values is available, we need only to use the CT-PINN to predict the evolution of the solution over time. However, in many instances, the star point values may be uncertain. In such cases, we incorporate the isentropic law, the Generalised Riemann Invariants, and the Rankine-Hugoniot conditions into the loss function to derive the star point values. Specifically, based on these principles, we can derive that p_* satisfies $F(p^*) = 0$. Once p_* is computed, we can determine $u_* = g_u(p_*)$, $\rho_{*L} = g_{\rho_L}(p_*)$, and $\rho_{*R} = g_{\rho_R}(p_*)$, as detailed in Section 4.5. Therefore, to estimate the star point values, the loss function of the CT-PINN is designed as follows:

$$\text{LOSS}_{\text{CT−PINN}} = \text{MSE}_{\{\tilde{\text{U}}_{\text{NN}},\text{BC},\text{IC}\}} + \text{MSE}_{\text{R}_{\tilde{\text{U}}}} + \text{MSE}_{\text{R}_x} + \eta_{star}\text{MSE}_{\{\text{star}\}},$$

with η_{star} denotes weight and

$$\begin{aligned}\mathrm{MSE}_{\{\mathrm{star}\}} = & \left|F\left(\tilde{p}_{NN}(\pi_2(0),0\right)\right|^2 \\ & + \left|\tilde{u}_{NN}(\pi_2(0),0) - g_u\left(\tilde{p}_{NN}(\pi_2(0),0)\right)\right|^2 \\ & + \left|\tilde{\rho}_{NN}(\pi_2(0),0) - g_{\rho_L}(\tilde{p}_{NN}\left(\pi_2(0),0)\right)\right|^2 \\ & + \left|\tilde{\rho}_{NN}(\pi_3(0),0) - g_{\rho_L}(\tilde{p}_{NN}\left(\pi_2(0),0)\right)\right|^2.\end{aligned}$$

The aforementioned strategy for addressing initial discontinuities can be readily generalized to broader classes of hyperbolic conservation laws. In particular, it can be observed that under the original coordinate system, the regions divided by the characteristic curve can be transformed into regular regions in the new coordinates. This allows for greater flexibility in selecting the structure of the CT-PINN, for example, different sub-networks that can be selected to approximate the solution in different regions. Furthermore, the hyperparameters of the corresponding sub-networks can be appropriately adjusted and simplified in the case of a straightforward structure of the solutions in the subdomains, which is more conducive to the training of the CT-PINN. In some cases, the transformed equation may become time-independent, with the time derivative $\tilde{u}_t = 0$, either globally or within specific subdomains. In these regions, the initial solution and the characteristic curves can be directly fitted during training, eliminating the need to solve the transformed equations. And in certain test cases, it is crucial to ensure that the output values remain non-negative. To achieve this, we can reasonably adjust the structure of the output layer based on the initial conditions and the properties of the physical quantities. For instance, in addressing Euler equations with initial conditions (16), the activation function of the output layer can be configured to be a ReLU function and a non-negative constant a incorporated (for example, setting $a = 0$ for velocity, while for pressure and density, a can be set to 0.05).

4. Numerical examples

In this section, the performance of the proposed model is validated using several numerical examples. For all cases, a feedforward NN with 2 hidden layers and a tanh activation function is utilized to predict the characteristic curves, and all hidden layers in every branch contain 50 neurons. The hyperparameters in Eq. (13) of the CT-PINN framework are set to $[\eta_r, \eta_{ic}, \eta_{bc}, \eta_{rx}] = [1, 100, 20, 20]$. Training is initially performed using the Adam optimizer with $10,000$ iterations and the learning rate is 0.01, followed by the L-BFGS algorithm, a quasi-Newton method, with 50 iterations. This combination of optimization settings is applied consistently across all numerical experiments presented in this study. The accuracy of the trained CT-PINN is quantified using the L_2 relative error, which is calculated as follows:

$$\varepsilon_{\mathrm{RE}}(t_k) = \frac{\|\mathbf{U}_{\mathrm{NN}}(t_k) - \mathbf{U}_{\mathrm{ref}}(t_k)\|_2}{\|\mathbf{U}_{\mathrm{ref}}(t_k)\|_2}, \quad t_k \in [0, T].$$

4.1. 1D convection equation

First, consider the simple 1D linear convection equation

$$\begin{aligned}& u_t + cu_x = 0, \quad x \in [0, L], \quad t \in [0, T], \\ & u(x = 0, t) = u(x = L, t), \quad t \in [0, T],\end{aligned}$$

where $L = 2\pi$, the constant convection velocity is $c = 4$, and the final time is set to $T = 1$. The analytical solution for this problem is $u(x,t) = u_0(x - ct, t)$, which is contingent upon the initial condition $u_0(x) = 0.5 + \exp((-(x - 1.232)^2)/(0.03^2))$. This PDE describes the evolution of a steep wave translating over time, where the solution remains smooth. In such cases, PINNs generally achieve acceptable accuracy, however, the steepness of the wave can still pose challenges during training. We chose $\pi(0) = 1.232$, corresponding to the peak position of the wave, to perform a coordinate transformation as in Eq. (6). For training the CT-PINN, we uniformly selected 256 spatial points in the original coordinate system and 100 time points, resulting in a total of 25,600 residual points. The architecture of the branch used to predict the transformed solution contains 4 hidden layers.

To verify the proposed method's ability to accurately capture the peak, we tested the trained model on 1024 spatial points and compared the CT-PINN predictions in the original coordinates with the true solutions, as shown in left sub-figure of Fig. 7. The results demonstrate excellent agreement. Furthermore, the structure of the transformed solutions is also presented in Fig. 7, where the peak is concentrated at a single point, indicating that the CT-PINN accurately predicts the characteristics. To explore the performance of both PINN and the proposed CT-PINN on such smooth solutions, we trained a traditional PINN model using the same network configuration and calculated the L_2 relative errors at the test points for both methods.

The relative errors of different models on 256 spatial points and on 1024 spatial points are shown in Fig. 8, the CT-PINN outperforms the PINN in two cases, which is understandable since the transformed solution exhibits less variation, making its structure easier to capture. It can be seen that when the test space point is 256, the errors of both models are relatively stable. However, when the test space point is 1024, there is a drop in PINN at the beginning and then it gradually stabilizes. This is related to the structure of the solution, since the solution changes in a very small region and the problem of overfitting may occur if the selected residues are not fine enough. However, the introduction of feature lines in CT-PINN will avoid this problem. Unlike the PINN, the error of the CT-PINN shows a trend of increasing over time. This occurs because the distribution of residual points in the CT-PINN is not uniform,

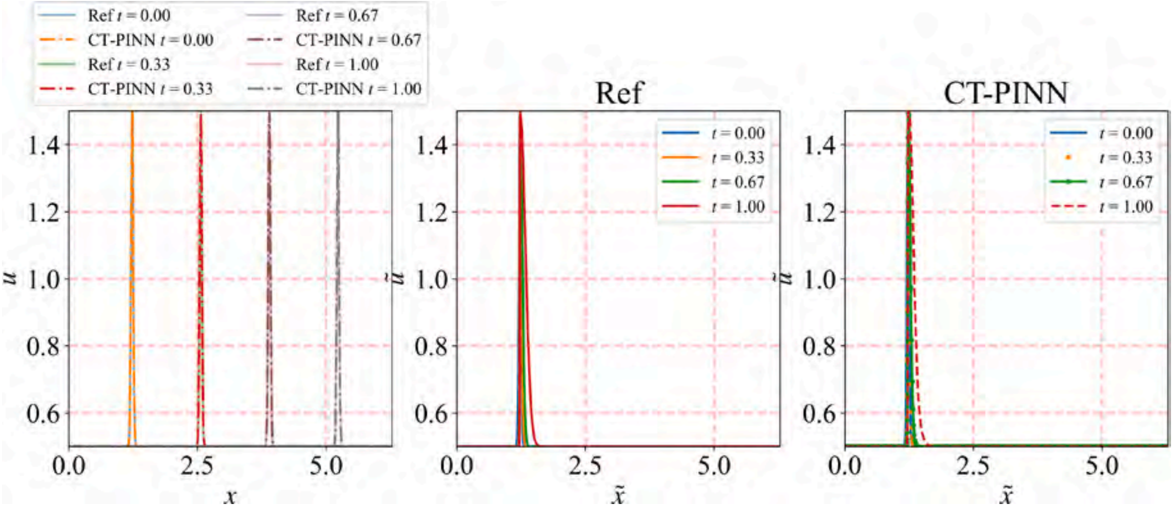

Fig. 7. **Left**: Reference and predicted solutions. **Middle**: Transformed reference solutions. **Right**: Transformed predicted solutions. Solutions are displayed for $t = 0, 0.33, 0.67, 1$.

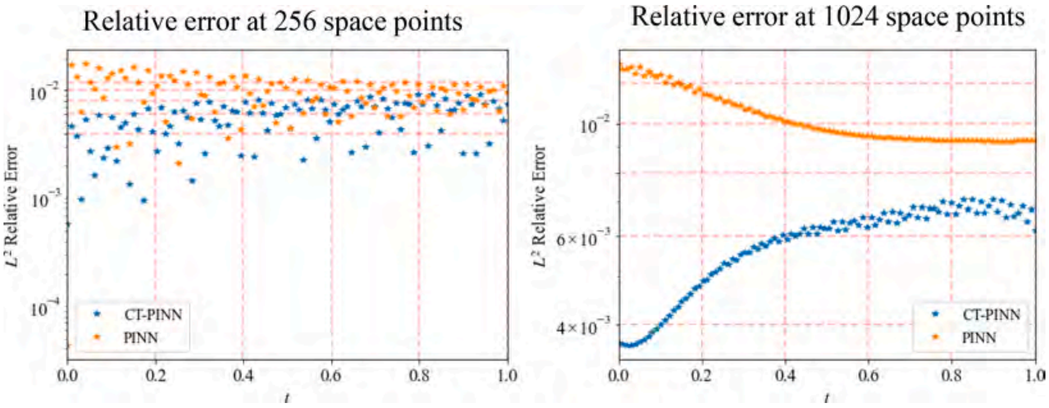

Fig. 8. The relative errors of different models in time region $[0, 1]$ for Example 4.1.

the coordinate transformation maps points with $x < \pi(t)$ to $\tilde{x} < \pi(0)$, resulting in fewer points with $\tilde{x} > \pi(0)$ in the transformed space over time, which may affect the accuracy in this region.

4.2. Inviscid Burgers equation

Let us now consider the one-dimensional Burgers equation:

$$u_t + u \cdot u_x = 0, \quad x \in [0, L], \quad t \in [0, T],$$
$$u(x = 2, t) = 3, \quad t \in [0, T],$$

with the initial condition Eq. (10), the final time is $T = 0.65$, and the starting points of characteristic curves are $\pi_1(0) = 0.25$ and $\pi_2(0) = 0.55$. In this scenario, although the initial conditions are smooth, a shock wave will form at time t_c. We use this example to demonstrate the proposed method's ability to capture shock waves. The finite volume method with a spatial step size of 201 and a time step size of $5e - 5$ is employed to compute the reference solutions.

To train the proposed model, we uniformly selected 201 points in space and 66 time points to compute the residuals of the transformed PDE. The architecture of CT-PINN is the same as that in Example 4.1. In Fig. 9, the solutions at different times in various coordinate systems are shown. It is observed that in the original coordinates, the finite volume method requires several transition points near the shock wave to prevent numerical oscillations. However, the CT-PINN accurately captures the shock wave without the need for transition points, which is superior to the traditional finite volume method. The reason for this performance is that, in the transformed coordinate system, the structure of the solution is consistent with the initial conditions, as illustrated in the right panel of Fig. 9. Therefore, in the transformed coordinates, the generation and movement of shock waves are avoided, and accurate modeling of the evolution of characteristic curves ensures a good representation of the original solution.

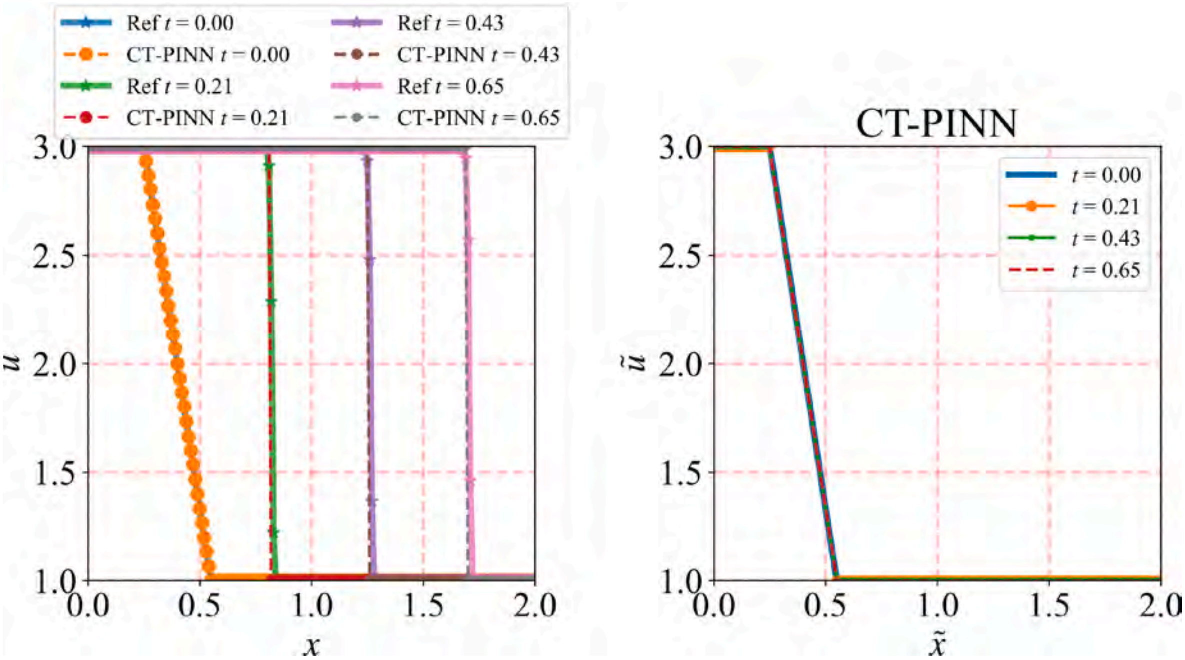

Fig. 9. Left: Comparison between the reference solutions and the corresponding predicted solutions and **Right**: transformed predicted solutions at the test time instants $t = 0, 0.21, 0.43, 0.65$.

4.3. *Traffic flow problem*

The proposed method is then applied to the traffic flow model [49,50], which is used to study the main characteristics of traffic flows and their relationships, as follows

$$
\begin{aligned}
&u_t + (1 - 2u)u_x = 0, \quad x \in [0, L], \quad t \in [0, T],\\
&u(x = 2, t) = 3, \quad t \in [0, T],
\end{aligned}
$$

where $L \approx 20.5647$, $T = 2/5L$. Consider an initial condition with discontinuities as follows:

$$
u(x, 0) = \begin{cases} 1, & \text{if } 0.4L \leqslant x \leqslant 0.8L,\\ \frac{1}{2}, & \text{else }, \end{cases}
$$

where two significant discontinuities exist ahead of and behind the congestion. As cars in the low-density region decelerate, the left discontinuity propagates leftward, leading to congestion when entering the high-density area. As cars accelerate and move from the high-density region to the low-density area, the vehicle density gradually decreases, and the right discontinuity evolves into a rarefaction wave.

In this scenario, we consider three characteristic curves corresponding to the positions of the left discontinuity and the endpoints of the rarefaction wave, which divide the transformed coordinate system into four subdomains $T_1 = \{(\tilde{x}, t) \mid 0 \leq \tilde{x} < \pi_1(0), 0 \leq t \leq T\}$, $T_2 = \{(\tilde{x}, t) \mid \pi_1(0) \leq \tilde{x} < \pi_2(0), 0 \leq t \leq T\}$, $T_3 = \{(\tilde{x}, t) \mid \pi_2(0) \leq \tilde{x} \leq \pi_3(0), 0 \leq t \leq T\}$, and $T_4 = \{(\tilde{x}, t) < \pi_3(0) \leq \tilde{x} \leq L, 0 \leq t \leq T\}$. To prevent the CT-PINN network from approximating discontinuous solutions, we apply separate PINN to each subdomain. Due to the regularity of the transformed network, this is easily achievable. Additionally, the solution structure in each transformed subdomain is simple, allowing the PINN network to be simplified to improve training efficiency, each PINN contains 3 hidden layers.

To model the evolution of the rarefaction wave, we modify the initial condition as:

$$
u_0(x) = \begin{cases} u_L, & \text{if } 0.4L \leq x < \pi_2(0),\\ u_L + \dfrac{u_R - u_L}{\pi_3(0) - \pi_2(0)}(x - \pi_2(0)), & \text{if } \pi_2(0) \leq x < \pi_3(0),\\ u_R, & \text{otherwise.} \end{cases}
$$

Here, $u_L = 1$,$u_R = \frac{1}{2}$, $\pi_2(0) = 0.8L + dx_1$, $\pi_2(0) = 0.8L + dx_1$ and $\pi_3(0) = 0.8L + dx_2$, where dx_1 and dx_2 are small values that can be positive or negative. However, for simplicity, we adopt an alternative approach, we take the solution at $t_{\text{in}} = 5\Delta t$ as the new initial condition, where $\Delta t = T/101$. This results in $\pi_1(0) = 8.0614$, and $\pi_3(0) = 16.4517$. Subsequently, the following coordinate transformation

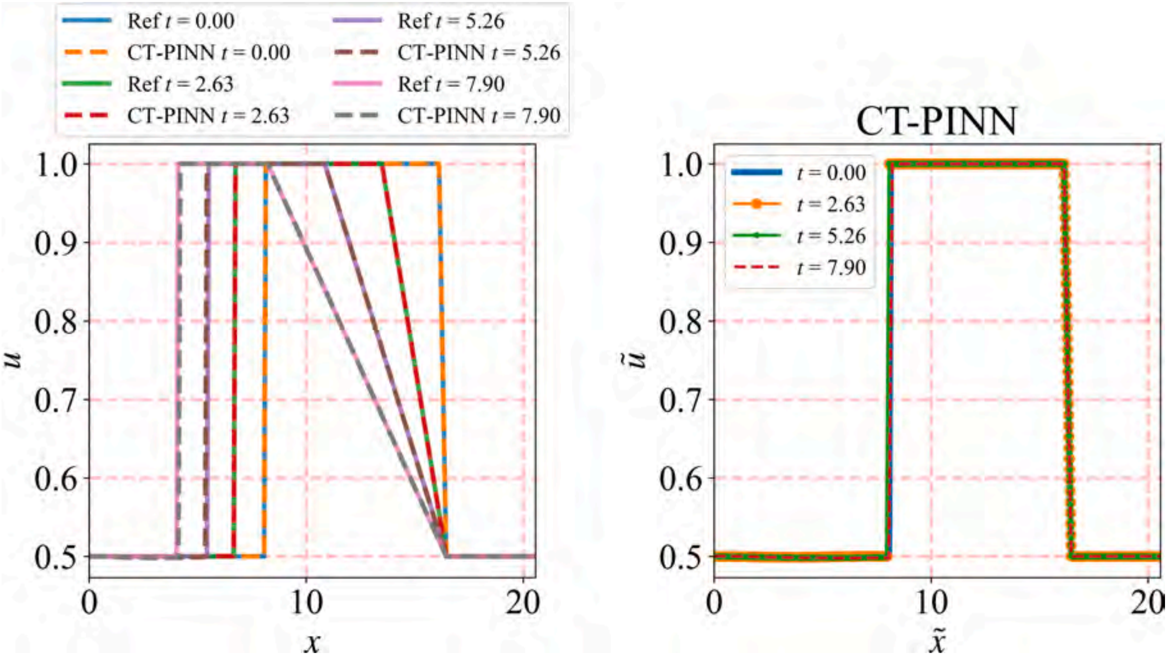

Fig. 10. **Left**: Comparison of the reference solutions and the corresponding predicted solutions and **Right**: transformed predicted solutions at the test time instants $t = 0, 2.63, 5.26, 7.90$.

is applied:

$$\tilde{x}(t) = \begin{cases} \dfrac{\pi_1(0)}{\pi_1(t)} x, & \text{if } 0 \le x < \pi_1(t), \\ \pi_1(0) + \dfrac{\pi_2(0) - \pi_1(0)}{\pi_2(t) - \pi_1(t)} \big(x - \pi_1(t)\big), & \text{if } \pi_1(t) \le x < \pi_2(t), \\ \pi_2(0) + \dfrac{\pi_3(0) - \pi_2(0)}{\pi_3(t) - \pi_2(t)} \big(x - \pi_2(t)\big), & \text{if } \pi_2(t) \le x < \pi_3(t), \\ L - \dfrac{L - \pi_3(0)}{L - \pi_3(t)} (L - x), & \text{if } \pi_3(t) < x \le L - t_{\text{in}}. \end{cases}$$

To train the CT-PINN model, we selected 251 uniformly distributed spatial points, with 5 residual points located within T_3, and 97 time points. And since the initial rarefaction wave region is relatively small, we increased the weight for this region by a factor of ten to enhance the representation of the rarefaction wave when calculating the initial condition constraints in the loss function.

The results predicted by the CT-PINN and the reference solutions at different times are displayed on the right side of Fig. 10. The CT-PINN method effectively captures both the leftward discontinuity and the leftward rarefaction wave. The transformed solutions are shown in the right subfigure of Fig. 10, where the independent NN structures for each region enable accurate approximation of piecewise smooth functions in both space and time. Consequently, the transformed solutions are more manageable, and the L_2 relative errors estimated by CT-PINN are displayed in left subfigure of Fig. 11, which generally fall within the range of 1e−04 to 1e−03. However, at some time points, the error reaches 1e−02, primarily due to errors in the characteristic lines, as shown in the right side of Fig. 11. Points whose distance from the $x_i(t)$ is less than the prediction error of the characteristic curve, may be mapped to incorrect subdomains. This issue is challenging to avoid, as the precision of NNs is inherently limited, but it is acceptable given the context.

4.4. 1D shallow water equation

Next, let's focus on the resolution of the 1D shallow water equation [48]

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{U})}{\partial x} = \mathbf{S}, \tag{17}$$

where

$$\mathbf{U} = \begin{pmatrix} h \\ hu \end{pmatrix}, \quad \mathbf{F}(\mathbf{U}) = \begin{pmatrix} hu \\ hu^2 + \frac{1}{2} g h^2 \end{pmatrix}, \quad \mathbf{S}(\mathbf{U}) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

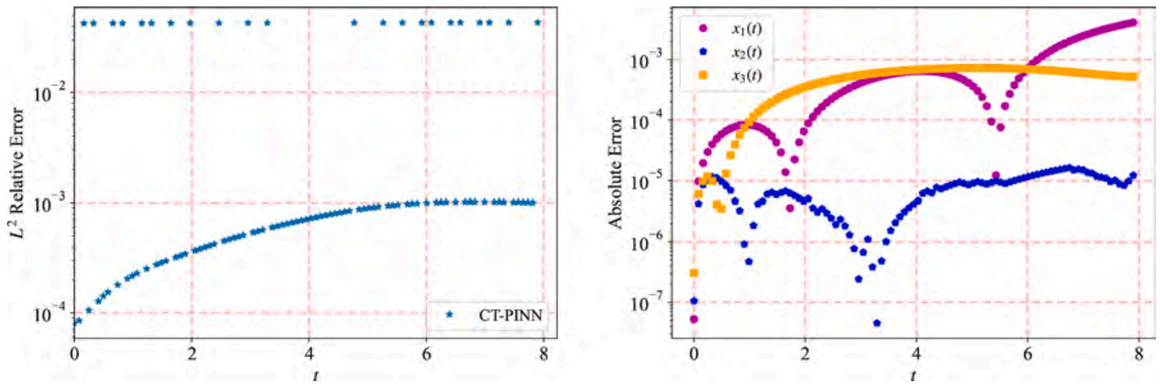

Fig. 11. **Left**: The relative errors of CT-PINN in time region $[0, 2/5L]$ and **Right**: the absolute errors of predicted characteristic curves of CT-PINN for Example 4.3.

$\mathbf{U}$ is the vector of conserved variables, where $h(x,t)$ is the depth and $u(x,t)$ is the depth averaged velocity. $\mathbf{F}(\mathbf{U})$ are the physical fluxes and $\mathbf{S}(\mathbf{U})$ are the source terms, and the ICs is

$$h(x,0) = \begin{cases} 1, & \text{if } 0 \le x \le 0.5, \\ 0.2, & \text{if } 0.5 < x \le 1, \end{cases} \qquad u(x,0) = 0, \quad 0 \le x \le 1.$$

This problem describes a scenario where a barrier is placed at 0.5 m in a 1 m long rectangular water tank. The upstream water depth $h_1 = 1$ m, and the downstream water depth $h_2 = 0.2$ m. After the barrier is removed, the changes in velocity and water depth in $T = 0.1$ seconds. The safety of dams poses a potential threat to human lives and property, making dam break calculations crucial for flood disaster research.

System (17) can be written in the quasi-linear form

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{J}(\mathbf{U})\frac{\partial \mathbf{U}}{\partial x} = \mathbf{S},$$

and the Jacobian matrix $\mathbf{J}(\mathbf{U})$ of the convective fluxes is given by

$$\mathbf{J}(\mathbf{U}) = \frac{\partial \mathbf{F}(\mathbf{U})}{\partial \mathbf{U}},$$

and can be diagonalizable with real eigenvalues $\lambda_1 = u - \sqrt{gh}$, $\lambda_2 = u - \sqrt{gh}$. Let $\boldsymbol{w} = \{w_1, w_2\} = \{u - 2\sqrt{gh}, u + 2\sqrt{gh}\}$. Then the following system respected to $\boldsymbol{w}$ is derived

$$\frac{\partial \boldsymbol{w}}{\partial t} + \boldsymbol{\Lambda}\frac{\partial \boldsymbol{w}}{\partial x} = \mathbf{0},$$

where $\boldsymbol{\Lambda} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$.

In this problem, to accurately capture the different discontinuities, we considered three characteristic curves and applied the following coordinate transformations for the variables w_1:

$$\tilde{x}_1(t) = \begin{cases} \dfrac{\pi_1(0)}{\pi_1(t)} x, & \text{if } 0 \le x < \pi_1(t), \\ \pi_1(0) + \dfrac{\pi_2(0) - \pi_1(0)}{\pi_2(t) - \pi_1(t)}\big(x - \pi_1(t)\big), & \text{if } \pi_1(t) \le x < \pi_2(t), \\ \pi_2(0) + \dfrac{\pi_3(0) - \pi_2(0)}{\pi_3(t) - \pi_2(t)}\big(x - \pi_2(t)\big), & \text{if } \pi_2(t) \le x < \pi_3(t), \\ L - \dfrac{L - \pi_3(0)}{L - \pi_3(t)}(L - x), & \text{if } \pi_3(t) < x \le L, \end{cases} \tag{18}$$

and for the variables w_2:

$$\tilde{x}_2(t) = \begin{cases} \dfrac{\pi_1(0)}{\pi_3(t)} x, & \text{if } 0 \le x < \pi_3(t), \\ L - \dfrac{L - \pi_3(0)}{L - \pi_3(t)}(L - x), & \text{if } \pi_3(t) < x \le L. \end{cases} \tag{19}$$

Transformation Eq. (18) divides the original region into four subdomains $T_1 = \{(\tilde{x}, t) \mid 0 \le \tilde{x} < \pi_1(0), 0 \le t \le T\}$, $T_2 = \{(\tilde{x}, t) \mid \pi_1(0) \le \tilde{x} < \pi_2(0), 0 \le t \le T\}$, $T_3 = \{(\tilde{x}, t) \mid \pi_2(0) \le \tilde{x} < \pi_3(0), 0 \le t \le T\}$, and $T_4 = \{(\tilde{x}, t) \mid \pi_3(0) \le \tilde{x} \le L, 0 \le t \le T\}$, while transformation Eq. (19) divides the original region into two subdomains. Similarly, to accurately simulate the distribution of w_1 in the rarefaction wave of subdomain T_2 and in subdomain T_3, and to simplify the process, we used the solution at $t = 0.004$ as the initial condition.

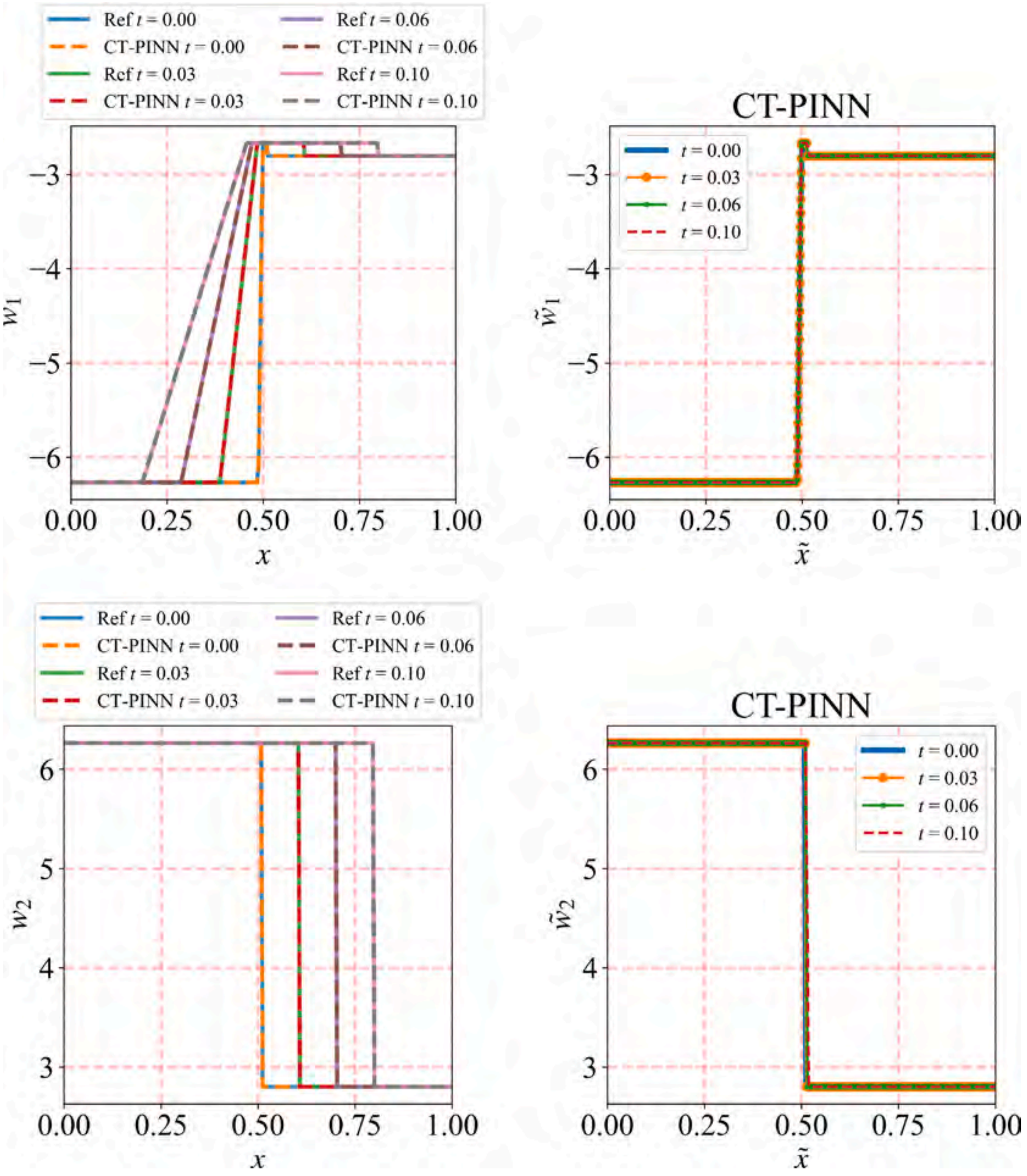

Fig. 12. **Left**: Comparison of the reference solutions and the corresponding predicted solutions and **Right**: transformed predicted solutions at the test time instants $t = 0, 0.03, 0.06, 0.1$.

When computing the initial condition constraints, the weights for subdomain T_2 and in subdomain T_3 are increased. The number of spatial residual points is 251, and the number of temporal residual points is 97. Since the initial condition contains discontinuities, for each subdomain, we selected an independent PINN network with three hidden layers for training.

Fig. 12 presents a comparison between the CT-PINN predictions and the reference solutions, along with the structure of the transformed solutions. It is evident that the spatiotemporal evolution of the transformed solutions no longer includes moving of discontinuities, and the proposed method accurately describes the evolution of w_1 and w_2. To further validate the accuracy of the proposed method in predicting velocity and depth, we display the CT-PINN predictions for different variables at the final time step, along with the corresponding relative errors, in Fig. 13. The results show an excellent fit between the CT-PINN predictions and the analytical solutions, with relative errors generally below the order of $1\text{e}{-03}$.

4.5. One-dimensional Euler equations

Considering the one-dimensional Euler equations with initial condition (16), where $L = 2$ and the final time $T = 0.2$. The unknown state V_{*L} is connected to the left data state V_L using the isentropic relation and the Generalized Riemann Invariants for the left wave.

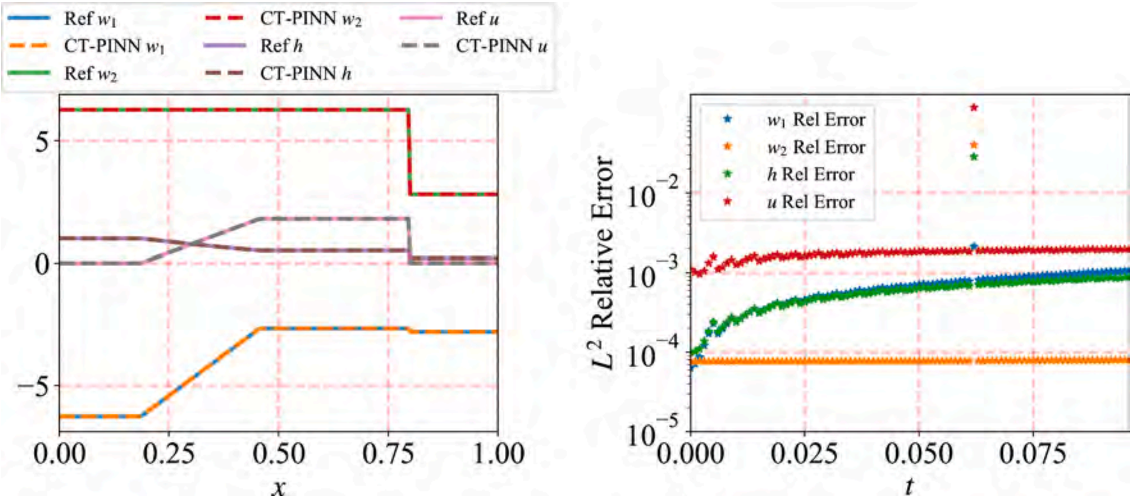

Fig. 13. Left: CT-PINN predictions for different variables at the final time and **Right**: the relative errors of CT-PINN in time region $[0, 0.1]$ for Example 4.4.

From the isentropic law, the pressure p^* satisfies

$$p^* = p_L\left(\frac{\rho_{*L}}{\rho_L}\right)^{\gamma}.$$

By evaluating the Generalized Riemann Invariants at the left data state, we obtain

$$u_L - \frac{2c_L}{\gamma-1} = u_* - \frac{2c_{*L}}{\gamma-1},$$

where $c_L = \sqrt{\frac{\gamma p_L}{\rho_L}}$ and $c_{*L} = \sqrt{\frac{\gamma p^*}{\rho_{*L}}}$ denote the sound speeds in the left data state and the left star region, respectively.

For the right shock wave, the Rankine-Hugoniot conditions yield the following relations:

$$\rho_R(u_R - s_4) = \rho_{*R}(u_{*R} - s_4),$$

$$p_R + \rho_R(u_R - s_4)^2 = p^* + \rho_{*R}(u_{*R} - s_4)^2,$$

$$E_R + \frac{p_R}{\rho_R} + \frac{1}{2}(u_R - s_4)^2 = E_{*R} + \frac{p^*}{\rho_{*R}} + \frac{1}{2}(u_{*R} - s_4)^2,$$

where s_4 is the shock speed. Using these principles, it can be shown that the pressure p^* satisfies the nonlinear equation

$$F(p^*) = f(p^*, p_L, \rho_L) + f(p^*, p_R, \rho_R) - u_L + u_R = 0,$$

where

$$f(p^*, p_L, \rho_L) = \frac{2c_L}{\gamma-1}\left[\left(\frac{p^*}{p_L}\right)^{\frac{\gamma-1}{2\gamma}} - 1\right],$$

$$f(p^*, p_R, \rho_R) = \frac{p^* - p_R}{\rho_R c_R\left[\frac{\gamma+1}{2\gamma}\left(\frac{p^*}{p_R}\right) + \frac{\gamma-1}{2\gamma}\right]^{1/2}}.$$

And once p^* is determined, the corresponding values of u_*, ρ_{*L}, and ρ_{*R} can be computed as follows:

$$u_* = g_u(p^*) = u_L - f(p^*, p_L, \rho_L),$$

$$\rho_{*L} = g_{\rho_L}(p^*) = \rho_L\left(\frac{p^*}{p_L}\right)^{1/\gamma},$$

$$\rho_{*R} = g_{\rho_R}(p^*) = \rho_R \cdot \frac{\frac{p^*}{p_R} + \frac{\gamma-1}{\gamma+1}}{\frac{\gamma-1}{\gamma+1}\frac{p^*}{p_R} + 1}.$$

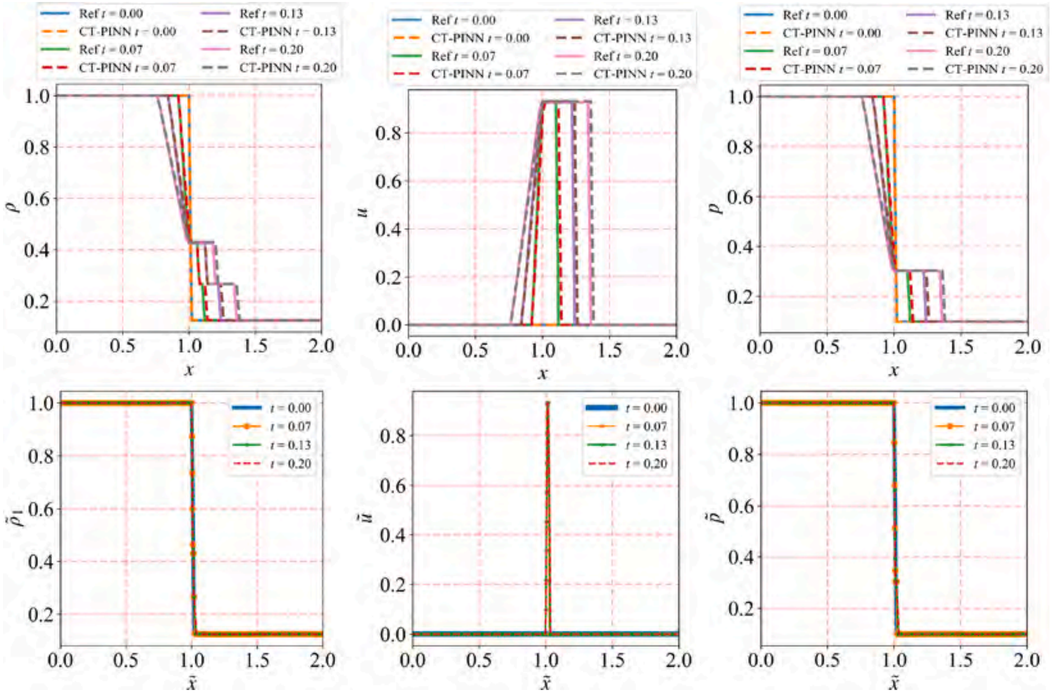

Fig. 14. **Top**: Comparison between the reference solutions and the corresponding predicted solutions and **Bottom**: transformed predicted solutions at the test time instants $t = 0, 0.07, 0.13, 0.2$.

Furthermore, by defining $s_1 = u_L - c_L$, $s_2 = u_* - c_*$, $s_3 = u_*$, and $s_4 = u_R + c_R\left[\frac{\gamma+1}{2\gamma}\frac{p_*}{p_R} + \frac{\gamma-1}{2\gamma}\right]^{\frac{1}{2}}$, the following characteristic transformations are applied:

$$\tilde{x}(t) = \begin{cases} \dfrac{\pi_1(0)}{\pi_1(t)}x, & \text{if } 0 \le x < \pi_1(t), \\ \pi_1(0) + \dfrac{\pi_2(0) - \pi_1(0)}{\pi_2(t) - \pi_1(t)}\big(x - \pi_1(t)\big), & \text{if } \pi_1(t) \le x < \pi_2(t), \\ \pi_2(0) + \dfrac{\pi_3(0) - \pi_2(0)}{\pi_3(t) - \pi_2(t)}\big(x - \pi_2(t)\big), & \text{if } \pi_2(t) \le x < \pi_3(t), \\ \pi_3(0) + \dfrac{\pi_4(0) - \pi_3(0)}{\pi_4(t) - \pi_3(t)}\big(x - \pi_3(t)\big), & \text{if } \pi_3(t) \le x < \pi_4(t), \\ L - \dfrac{L - \pi_4(0)}{L - \pi_4(t)}(L - x), & \text{if } \pi_4(t) < x \le L. \end{cases}$$

To implement the CT-PINN method, we first select $N_x = N_t = 101$ residual points in the spatial and temporal dimensions. The parameters are set as $\eta_{\text{star}} = 20$, $dx_1 = 0.01$, $dx_2 = 0.005$, and $dx_3 = 0.005$. With this configuration, we have $dx_1 + dx_2 + dx_3 = L/(N_x - 1)$, ensuring that the function values at the spatial residual points remain unaffected. In regions T_3 and T_4, where the solution is constant, a smaller step size is adopted. Separate PINN models are trained for each subdomain, and the number of neurons in the hidden layers of each model is reduced to 10. Additionally, to provide sufficient initial information for the interval $[\pi_1(0), \pi_4(0)]$, we uniformly select five residual points in each subinterval. And to ensure adequate initial information for the interval $[\pi_1(0), \pi_4(0)]$, five residual points are uniformly sampled within each subinterval. To enhance the influence of these points, their weights in the initial condition constraints are increased by a factor of 20.

Table 1 presents the star values estimated by CT-PINN, the corresponding exact solutions, and the absolute errors between them. The results demonstrate that CT-PINN provides accurate estimates, with max absolute errors on the order of $1\text{e}-3$. To further evaluate the performance of CT-PINN, Fig. 14 compares the estimated solutions and the exact solutions at different time steps. The high degree of agreement between the two highlights the method's capability to handle problems with discontinuities effectively.

4.6. 2D convection equation

In the final example, we consider a two-dimensional convection equation

$$\begin{aligned} &u_y(x,y,t) + cu_x(x,y,t) + cu_y(x,y,t) = 0, \quad (x,y) \in [0,L]^2, \quad t \in [0,T], \\ &u(x=0,y,t) = u(x,y=0,t), \quad (x,y) \in [0,L]^2, \quad t \in [0,T], \\ &u(x=L,y,t) = u(x,y=L,t), \quad (x,y) \in [0,L]^2, \quad t \in [0,T], \end{aligned}$$

Table 1
Reference and CT-PINN solutions for pressure, velocity, and density with absolute errors.

Model	p_*	u_*	ρ_{*L}	ρ_{*R}
Reference	0.3031	0.9275	0.4263	0.2656
CT-PINN	0.3030	0.9313	0.4292	0.2663
Absolute error	0.0001	0.0038	0.0029	0.0007

with two different initial conditions: an initial condition containing a discontinuity

$$u(x,y,0) = \begin{cases} 2, & \text{if } 0 \le x \le 3, 0 \le y \le 3, \\ 1, & \text{else }, \end{cases} \tag{20}$$

and a smooth Gaussian initial condition

$$u(x,y,0) = exp(-((x-1.5)^2 + (y-1.5)^2)), \tag{21}$$

where $L = 6$, $T = 2$, and the velocity c is set to 1. For each initial condition, we applied the following coordinate transformations

$$\tilde{x}(t) = \begin{cases} \dfrac{\pi_1(0)}{\pi_1(t)} x, & \text{if } 0 \le x < \pi_1(t), \\ L - \dfrac{L-\pi_1(0)}{L-\pi_1(t)}(L-x), & \text{if } \pi_1(t) < x \le L, \end{cases}$$

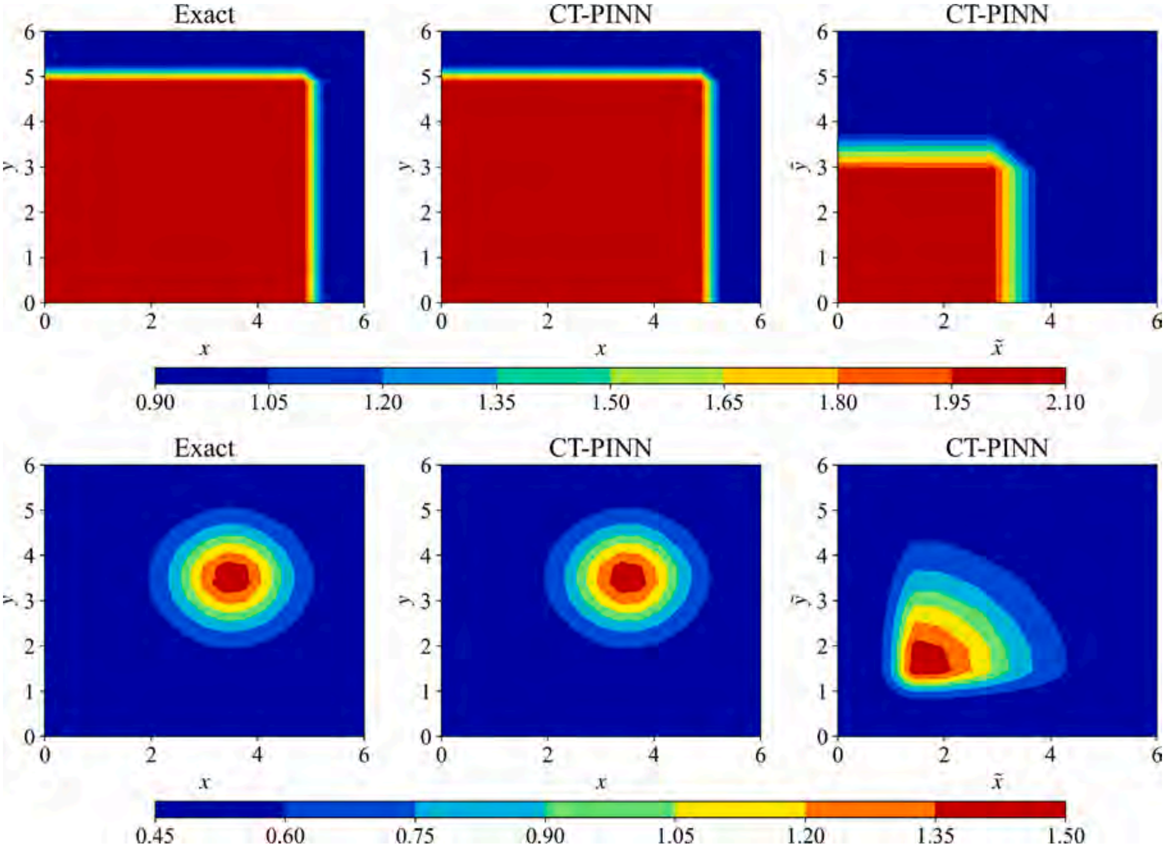

Fig. 15. **Top**: The reference solution (left), the corresponding predicted solution (middle), and the corresponding transformed predicted solution (right) for initial condition Eq. (20). **Bottom**: the reference solution (left), the corresponding predicted solution (middle), and the corresponding transformed predicted solution (right) for initial condition Eq. (21).

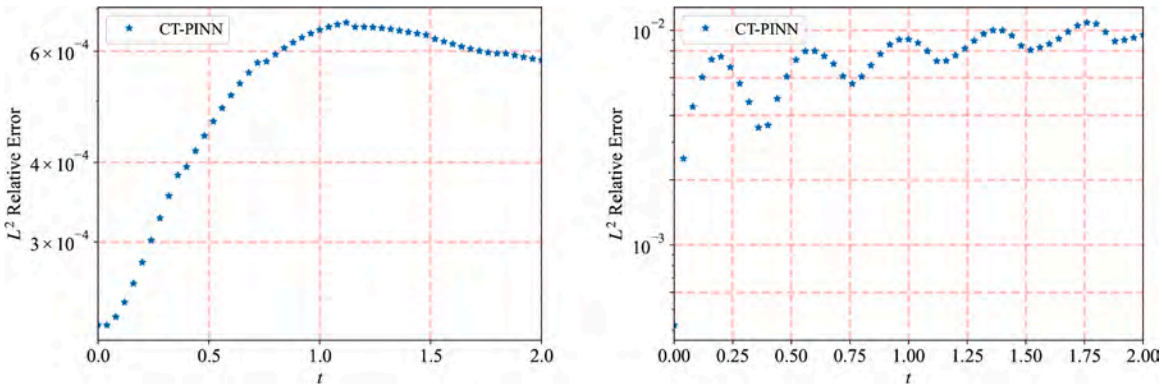

Fig. 16. **Left**: The relative errors of CT-PINN in time region $[0, 2]$ for initial condition Eq. (20) and **Right**: for initial condition Eq. (21).

Table 2
The mean relative errors of different hyper-parameters for the Example 4.2.

η_{ic} / η_{rx}	1	10	100	1000
1	Inf	0.0065	0.0059	0.0064
20	0.0657	0.0043	0.0043	0.0065
40	0.6137	0.0185	0.0055	0.0065
60	0.4633	0.0044	0.0043	0.0043

$$\tilde{y}(t) = \begin{cases} \dfrac{\pi_1(0)}{\pi_1(t)} y, & \text{if } 0 \le y < \pi_1(t), \\[2ex] L - \dfrac{L - \pi_1(0)}{L - \pi_1(t)}(L - y), & \text{if } \pi_1(t) < y \le L, \end{cases}$$

with $\pi_1(0) = 3$ and $\pi_1(0) = 1.5$ for initial conditions Eqs. (20) and (21), respectively. We select 32×32 residual points in space and 101 residual points in time for training. In the case of the discontinuous initial condition, we employed two independent PINN networks to approximate the solutions in subdomain $T_1 = \{(\tilde{x}, \tilde{y}, t) \mid 0 \le \tilde{x} < \pi_1(0), 0 \le \tilde{y} < \pi_1(0), 0 \le t \le T\}$ and subdomain $T_2 = \{(\tilde{x}, \tilde{y}, t) \mid \pi_1(0) \le \tilde{x} < L \ or\ \pi_1(0) \le \tilde{y} < L, 0 \le t \le T\}$, respectively. The reference solutions, along with CT-PINN predictions at the final time step in different coordinate systems, are displayed in Fig. 15. As shown in the figure, the CT-PINN method demonstrates excellent agreement with the reference solutions, achieving highly accurate and reliable predictions. The discrepancies are minimal, and the method effectively captures the essential features of the solution in both cases. The small L_2 relative errors observed in Fig. 16, further confirm the precision and robustness of the proposed approach.

4.7. Effect of hyperparameter selection on CT-PINN performance

In this section, we investigate the impact of hyperparameter selection on the performance of CT-PINN using the Burgers equation. Selecting appropriate hyperparameters is crucial during CT-PINN training, but identifying an optimal combination remains a challenging task. Table 2 presents the mean relative error for the Burgers equation under various hyperparameter configurations. All training is conducted using the Adam optimizer with 10,000 iterations and a learning rate of 0.01, followed by the L-BFGS algorithm with 50 iterations.

The results show that inappropriate hyperparameter choices may lead to the network failing to converge. Generally, low weights on initial condition constraints tend to result in poor accuracy. This is because the proposed method achieves good accuracy only when the initial conditions are well fitted. Additionally, the number of residual points for the initial conditions is relatively small compared with those for the governing equations, making it necessary to assign higher weights to the initial condition constraints. By contrast, the impact of characteristic equation constraints is less significant when other parameters remain fixed.

In the case of establishing different models for distinct subdomains, it is essential to account for variations in the size of the subdomains. Consider the adjusted initial conditions, where interval $[\pi_i(0), \pi_{i+1}(0)]$ may be very small. In such cases, given the importance of initial constraints, two potential improvements are proposed: firstly, the refinement of initial residual points in smaller intervals, and secondly, the increase of the weight of initial condition constraints in those intervals.

5. Conclusion

In this paper, a coordinate transformation-based framework is proposed for PINNs aimed at solving hyperbolic conservation laws that may feature discontinuous solutions such as shock waves and contact waves, which could lead to the failure of the traditional PINN. The coordinate transformations are based on specific characteristic curves, which can be selected according to the initial conditions. Following the coordinate transformations, the subdomains bounded by the characteristic curves are transformed into regular regions. The PDEs adhered to by each region can also be derived using the chain rule. The CT-PINN learns both the characteristic curves and the solution of the transformed equations, with a loss function constructed by the characteristic equations, transformed equations, and the initial/boundary conditions. To deal with discontinuous interfaces of initial conditions, Generalized Riemann Invariants, isentropic laws and Rankine-Hugoniot conditions are encoded into the loss function to learn the evolution of the solution. The transformed solution exhibits a simplified structure within each subdomain, effectively avoiding the generation and propagation of shock waves. Furthermore, the transformed subdomain's regularity facilitates the selection of an independent PINN for each subdomain, enhancing the precision of the prediction. As the NN is an end-to-end function, transforming the original coordinates based on the predicted characteristic curves and feeding the transformed coordinates into the trained network can directly compute the state variables at the original coordinates without the need for interpolation. Automatic differentiation also avoids the need for high resolution in discrete formats in numerical methods that can cause grid management problems.

The proposed method is numerically tested on the one- and two-dimensional convection equations, the Burgers' equation, the shallow water wave equation, and the traffic flow problem, demonstrating its effectiveness in solving hyperbolic conservation laws. In particular, the performance of CT-PINN on the Burgers equation for shock waves shows that it can capture shock waves without transition points and has the potential to capture discontinuities more accurately than mesh-based numerical methods. The ability of CT-PINN to accurately learn the characteristics is an important reason for the success of the approach, and the accuracy of the approximated characteristics can be illustrated in an application to the traffic flow equation, with an absolute error of the characteristics of less than $1\mathrm{e}{-03}$.

Although the proposed method is validated, the structure of the CT-PINN is not sufficiently explored. In addition to the hyperparameters of the CT-PINN, the input can also be selected based on the PDE. For instance, in a subdomain where the time derivative is zero, one can exclude time as an input and instead utilize the initial conditions to train the corresponding sub-net for this subdomain. Therefore, future work may be required to investigate this further. Moreover, the training of neural networks is sometimes time-consuming, and in settings such as multi-queries and real-time simulations where the PDEs need to be solved repeatedly, introducing the parameters into the proposed framework to build alternative models for the parameters of the whole parameter domain is a further direction.

CRediT authorship contribution statement

Yuanhong Chen: Writing – original draft, Methodology, Data curation; **Zhen Gao:** Writing – review & editing, Supervision, Funding acquisition; **Jan S. Hesthaven:** Writing – review & editing; **Yifan Lin:** Writing – review & editing, Methodology; **Xiang Sun:** Writing – review & editing.

Data availability

Data will be made available on request.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors would like to acknowledge the funding support of this research by the National Natural Science Foundation of China (12371435, 12201592), the Taishan Scholars Program (tsqn202211059) and the Shandong Provincial Natural Science Foundation (ZR2023MA043, ZR2022QA006).

References

[1] P.J. Olver, Introduction to Partial Differential Equations, 1, Springer, 2014. https://link.springer.com/book/10.1007/978-3-319-02099-0.
[2] E. Godlewski, P.-A. Raviart, Numerical Approximation of Hyperbolic Systems of Conservation Laws, 118, 2021. https://link.springer.com/book/10.1007/978-1-0716-1344-3.
[3] L.C. Evans, Partial Differential Equations, 19, American Mathematical Society, 2022. https://bookstore.ams.org/gsm-19-r/.
[4] E.F. Toro, Riemann solvers and numerical methods for fluid dynamics, Springer Berlin, Heidelberg, 1997. https://link.springer.com/book/10.1007/b79761.
[5] R.J. LeVeque, Finite Volume Methods for Hyperbolic Problems, 31, Cambridge University Press, 2002. https://doi.org/10.1017/CBO9780511791253.
[6] J. VonNeumann, R.D. Richtmyer, A method for the numerical calculation of hydrodynamic shocks, J. Appl. Phys. 21 (3) (1950) 232–237. https://pubs.aip.org/aip/jap/article/21/3/232/159292.

[7] D. Zhang, C. Jiang, D. Liang, L. Cheng, A review on TVD schemes and a refined flux-limiter for steady-state calculations, J. Comput. Phys. 302 (2015) 114–154. https://linkinghub.elsevier.com/retrieve/pii/S0021999115005707.

[8] A. Harten, P.D. Lax, B. van Leer, On upstream differencing and godunov-type schemes for hyperbolic conservation laws, SIAM Rev. 25 (1983) 53–79. https://api.semanticscholar.org/CorpusID:17009651.

[9] A. Harten, B. Engquist, S. Osher, S.R. Chakravarthy, Uniformly high order accurate essentially non-oscillatory schemes, 111, J. Comput. Phys. 71 (1987) 231–303. https://api.semanticscholar.org/CorpusID:120579203.

[10] G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, J. Comput. Phys. 126 (1995) 202–228. https://api.semanticscholar.org/CorpusID:14446714.

[11] X.-D. Liu, S. Osher, T.F. Chan, Weighted essentially non-oscillatory schemes, J. Comput. Phys. 115 (1994) 200–212. https://api.semanticscholar.org/CorpusID:260607468.

[12] A.K. Henrick, T.D. Aslam, J.M. Powers, Mapped weighted essentially non-oscillatory schemes: achieving optimal order near critical points, J. Comput. Phys. 207 (2005) 542–567. https://api.semanticscholar.org/CorpusID:14972332.

[13] X. Luo, S.-p. Wu, An improved WENO-Z+ scheme for solving hyperbolic conservation laws, J. Comput. Phys. 445 (2021) 110608. https://www.sciencedirect.com/science/article/pii/S0021999121005039.

[14] G.A. Gerolymos, D. Sénéchal, I. Vallet, C.W. Shu, Jiang-Shu, G.S. Jiang, Liu, T. Titarev, E.F. Toro, Very-high-order weno schemes, J. Comput. Phys. 228 (2009) 8481–8524. https://api.semanticscholar.org/CorpusID:18342658.

[15] B. Cockburn, G.E. Karniadakis, C.-W. Shu, Discontinuous Galerkin methods: theory, computation and applications, 2011. https://api.semanticscholar.org/CorpusID:118000895.

[16] J.S. Hesthaven, T. Warburton, Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications, 2007. https://api.semanticscholar.org/CorpusID:118098020.

[17] F. Ismail, P.L. Roe, Affordable, entropy-consistent Euler flux functions II: entropy production at shocks, J. Comput. Phys. 228 (2009) 5410–5436. https://api.semanticscholar.org/CorpusID:32563154.

[18] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015) 770–778. https://api.semanticscholar.org/CorpusID:206594692.

[19] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, Commun. ACM 60 (2012) 84–90. https://api.semanticscholar.org/CorpusID:195908774.

[20] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, in: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, NIPS'14, MIT Press, 2014, p. 3104–3112. https://dl.acm.org/doi/10.5555/2969033.2969173.

[21] J. Schmidhuber, Deep learning in neural networks: an overview, Neural Netw. 61 (2014) 85–117. https://api.semanticscholar.org/CorpusID:11715509.

[22] C. Beck, M. Hutzenthaler, A. Jentzen, B. Kuckuck, An overview on deep learning-based approximation methods for partial differential equations, Discrete Contin. Dyn. Syst. - B 28 (6) (2023) 3697–3746. https://www.aimsciences.org/article/id/639b367cb2114e413cb39c48.

[23] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686–707. https://api.semanticscholar.org/CorpusID:57379996.

[24] E. Weinan, T. Yu, The deep ritz method: a deep learning-based numerical algorithm for solving variational problems, Commun. Math. Stat. 6 (2017) 1–12. https://api.semanticscholar.org/CorpusID:2988078.

[25] L. Lu, P. Jin, G. Pang, Z. Zhang, G.E. Karniadakis, Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators, Nat. Mach. Intell. 3 (2019) 218–229. https://api.semanticscholar.org/CorpusID:233822586.

[26] Y. Shang, F. Wang, J. Sun, Randomized neural network with Petrov–Galerkin methods for solving linear and nonlinear partial differential equations, Commun. Nonlinear Sci. Numer. Simul. 127 (2023) 107518. https://api.semanticscholar.org/CorpusID:261758442.

[27] J.A. Sirignano, K.V. Spiliopoulos, DGM: A deep learning algorithm for solving partial differential equations, J. Comput. Phys. 375 (2017) 1339–1364. https://api.semanticscholar.org/CorpusID:126213877.

[28] Y. Zang, G. Bao, X. Ye, H. Zhou, Weak adversarial networks for high-dimensional partial differential equations, J. Comput. Phys. 411 (2019) 109409. https://api.semanticscholar.org/CorpusID:197935272.

[29] S. Cai, Z. Mao, Z. Wang, M. Yin, G.E. Karniadakis, Physics-informed neural networks (PINNs) for fluid mechanics: a review, Acta Mech. Sin. 37 (2021) 1727–1738. https://api.semanticscholar.org/CorpusID:234789992.

[30] X. Jin, S. Cai, H. Li, G.E. Karniadakis, NSFnets (Navier–Stokes flow nets): physics-informed neural networks for the incompressible Navier–Stokes equations, J. Comput. Phys. 426 (2020) 109951. https://api.semanticscholar.org/CorpusID:212726434.

[31] G. Pang, L. Lu, G.E. Karniadakis, fPINNs: fractional physics-informed neural networks, SIAM J. Sci. Comput. 41 (4) (2019) A2603–A2626. https://doi.org/10.1137/18M1229845.

[32] E. Kharazmi, Z. Zhang, G.E.M. Karniadakis, hp-VPINNs: variational physics-informed neural networks with domain decomposition, Comput. Methods Appl. Mech. Eng. 374 (2021) 113547. https://www.sciencedirect.com/science/article/pii/S0045782520307325.

[33] Z. Gao, L. Yan, T. Zhou, Failure-informed adaptive sampling for PINNs, part II: combining with re-sampling and subset simulation, Commun. Appl. Math. Comput. 6 (2024) 1720–1741.

[34] A.G. Baydin, B.A. Pearlmutter, A.A. Radul, J.M. Siskind, Automatic differentiation in machine learning: a survey, J. Mach. Learn. Res. 18 (153) (2018) 1–43. https://jmlr.csail.mit.edu/papers/v18/17-468.HTML.

[35] S. Wang, X. Yu, P. Perdikaris, When and why PINNs fail to train: a neural tangent kernel perspective, J. Comput. Phys. 449 (2020) 110768. https://api.semanticscholar.org/CorpusID:220845655.

[36] O. Fuks, H.A. Tchelepi, Limitations of physics informed machine learning for nonlinear two-phase transport in porous media, J. Mach. Learn. Model. Comput. 1 (2020) 19–37. https://api.semanticscholar.org/CorpusID:221835740.

[37] S. Wang, Y. Teng, P. Perdikaris, Understanding and mitigating gradient flow pathologies in physics-informed neural networks, SIAM J. Sci. Comput. 43 (2021) A3055–A3081. https://api.semanticscholar.org/CorpusID:263888201.

[38] A.S. Krishnapriyan, A. Gholami, S. Zhe, R.M. Kirby, M.W. Mahoney, Characterizing Possible Failure Modes in Physics-informed Neural Networks, in: Neural Information Processing Systems, 2021. https://api.semanticscholar.org/CorpusID:237386144.

[39] Z. Gao, L. Yan, T. Zhou, Failure-informed adaptive sampling for PINNs, SIAM J. Sci. Comput. 45 (4) (2023) A1971–A1994. https://doi.org/10.1137/22M1527763.

[40] T. Luo, Q. Zhou, On residual minimization for PDEs: failure of PINN, modified equation, and implicit bias, arXiv:2310.18201 (2023). https://api.semanticscholar.org/CorpusID:264555485.

[41] Z. Mao, A.D. Jagtap, G.E. Karniadakis, Physics-informed neural networks for high-speed flows, Comput. Methods Appl. Mech. Eng. 360 (2020) 112789. https://api.semanticscholar.org/CorpusID:212755458.

[42] L. Liu, S. Liu, H. Xie, F. Xiong, T. Yu, M. Xiao, L. Liu, H. Yong, Discontinuity computing using physics-informed neural networks, J. Sci. Comput. 98 (2022) 1–38. https://api.semanticscholar.org/CorpusID:249461894.

[43] A.D. Jagtap, E. Kharazmi, G.E. Karniadakis, Conservative physics-informed neural networks on discrete domains for conservation laws: applications to forward and inverse problems, Comput. Methods Appl. Mech. Eng. 365 (2020) 113028. https://api.semanticscholar.org/CorpusID:216333271.

[44] R.G. Patel, I. Manickam, N.A. Trask, M.A. Wood, M. Lee, I. Tomas, E.C. Cyr, Thermodynamically consistent physics-informed neural networks for hyperbolic systems, J. Comput. Phys. 449 (2020) 110754. https://api.semanticscholar.org/CorpusID:228083710.

[45] R. Mojgani, M. Balajewicz, P. Hassanzadeh, Kolmogorov n–width and Lagrangian physics-informed neural networks: a causality-conforming manifold for convection-dominated PDEs, Comput. Methods Appl. Mech. Eng. 404 (2023) 115810. https://www.sciencedirect.com/science/article/pii/S0045782522007666.

[46] J.L. Gracia, E. O'Riordan, A singularly perturbed convection-diffusion problem with a moving pulse, J. Comput. Appl. Math. 321 (2017) 371–388. https://www.sciencedirect.com/science/article/pii/S0377042717301127.

[47] P. Solán-Fustero, J.L. Gracia, A. Navas-Montilla, P. García-Navarro, A POD-based ROM strategy for the prediction in time of advection-dominated problems, J. Comput. Phys. 471 (2022) 111672. https://api.semanticscholar.org/CorpusID:252677345.

[48] E. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction, Springer Berlin, Heidelberg, 2009. https://link.springer.com/book/10.1007/b79761.

[49] P.-E. Mazaré, A.H. Dehwah, C.G. Claudel, A.M. Bayen, Analytical and grid-free solutions to the Lighthill–Whitham–Richards traffic flow model, Transp. Res. Part B 45 (10) (2011) 1727–1748. https://www.sciencedirect.com/science/article/pii/S0191261511001044.

[50] W. Imran, H.K.-P.E. Zawar, T.A. Gulliver, K. Khattak, S. saeid, M.S. Aslam, Macroscopic traffic flow characterization for stimuli based on driver reaction, Civ. Eng. J. 7 (2020). https://civilejournal.org/index.php/cej/article/view/2504.