



# Neural operators for adaptive control of freeway traffic<sup>☆</sup>

Kaijing Lv<sup>a</sup>, Junmin Wang<sup>a</sup>, Yihuai Zhang<sup>b</sup>, Huan Yu<sup>b,\*</sup>

<sup>a</sup> School of Mathematics and Statistics, Beijing Institute of Technology, Beijing, 100081, China

<sup>b</sup> Hong Kong University of Science and Technology (Guangzhou), Nansha, Guangzhou, Guangdong, 511400, China



## ARTICLE INFO

### Article history:

Received 20 October 2024

Received in revised form 15 May 2025

Accepted 8 August 2025

### Keywords:

Traffic flow model  
 $2 \times 2$  hyperbolic system  
 PDE backstepping  
 Neural operators  
 Adaptive control

## ABSTRACT

The uncertainty in human driving behaviors leads to stop-and-go traffic congestion on freeway. The freeway traffic dynamics are governed by the Aw–Rascle–Zhang (ARZ) traffic Partial Differential Equation (PDE) models with unknown relaxation time. Motivated by the adaptive traffic control problem, this paper presents a neural operator (NO) based adaptive boundary control design for the coupled  $2 \times 2$  hyperbolic systems with uncertain spatially varying in-domain coefficients and boundary parameter. In traditional adaptive control for PDEs, solving backstepping kernel online can be computationally intensive, as it updates the estimation of coefficients at each time step. To address this challenge, we use operator learning, i.e. DeepONet, to learn the mapping from system parameters to the kernels functions. DeepONet, a class of deep neural networks designed for approximating operators, has shown strong potential for approximating PDE backstepping designs in recent studies. Unlike previous works that focus on approximating single kernel equation associated with the scalar PDE system, we extend this framework to approximate PDE kernels for a class of the first-order coupled  $2 \times 2$  hyperbolic kernel equations. Our approach demonstrates that DeepONet is nearly two orders of magnitude faster than traditional PDE solvers for generating kernel functions, while maintaining a loss on the order of  $10^{-3}$ . In addition, we rigorously establish the system's stability via Lyapunov analysis when employing DeepONet-approximated kernels in the adaptive controller. The proposed adaptive control is compared with reinforcement learning (RL) methods. Our approach guarantees stability and does not rely on initial values, which is essential for rapidly changing traffic scenarios. This is the first time this operator learning framework has been applied to the adaptive control of the ARZ traffic model, significantly enhancing the real-time applicability of this design framework for mitigating traffic congestion.

© 2025 Elsevier Ltd. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

## 1. Introduction

Stop-and-go traffic congestion is a very common phenomenon in major cities around the world. The traffic congestion on highways leads to many unsafe driving behaviors, as well as increased fuel emissions, environmental pollution, and increased commuting time (Belletti, Huo, Litrico, & Bayen, 2015; De Palma & Lindsey, 2011). The traffic congestion is characterized by the propagation of shock waves on road, caused by delayed driver response. There have been many studies on traffic stabilization using PDE models, such as the first-order hyperbolic PDE model proposed by

Lighthill and Whitham and Richards (LWR) (Lighthill & Whitham, 1955; Richards, 1956) to describe traffic density waves on highways. Then Aw and Rascle (2000) and Zhang (2002) proposed the second-order nonlinear hyperbolic PDE model to describe the evolution of velocity and density states in traffic flow. The ARZ model is a  $2 \times 2$  hyperbolic PDE system and widely used for describing dynamics of the stop-and-go traffic oscillations. In this paper, we adopt the ARZ model and develop adaptive boundary control designs for traffic stabilization.

### 1.1. PDE backstepping for traffic control

The control strategy for freeway traffic congestion is usually based on static road infrastructure to regulate traffic flow, such as ramp metering and varying speed limits. Various traffic boundary control designs have been proposed to smooth traffic in the works of Bekiaris-Liberis and Delis (2019), Zhang, Prieur, and Qiao (2019) as well as Karafyllis, Bekiaris-Liberis, and Papa-georgiou (2018). While Bekiaris-Liberis and Delis utilize Adaptive Cruise Control vehicles for in-domain actuation as control inputs (Bekiaris-Liberis & Delis, 2019), Karafyllis et al. design a

<sup>☆</sup> This work is supported by the National Natural Science Foundation of China under Grant 62203131, 92471108 and 12131008. The material in this paper was partially presented at the 5th IFAC Workshop on Control of Systems Governed by Partial Differential Equations (CPDE), June 18–June 20, 2025, Beijing, China. This paper was recommended for publication in revised form by Associate Editor Rafael Vazquez under the direction of Editor Miroslav Krstic.

\* Corresponding author.

E-mail addresses: [kjlv@bit.edu.cn](mailto:kjlv@bit.edu.cn) (K. Lv), [jmwang@bit.edu.cn](mailto:jmwang@bit.edu.cn) (J. Wang), [yizhang169@connect.hkust-gz.edu.cn](mailto:yizhang169@connect.hkust-gz.edu.cn) (Y. Zhang), [huanyu@ust.hk](mailto:huanyu@ust.hk) (H. Yu).

boundary feedback law to manage inlet demand [Karafyllis et al. \(2018\)](#). The boundary control strategy using PDE backstepping is first proposed in [Yu and Krstic \(2019\)](#) to stabilize the linearized ARZ system, including full state feedback and output feedback. Recent efforts ([Burkhardt, Yu, & Krstic, 2021](#); [Yu, Auriol, & Krstic, 2022](#); [Yu & Krstic, 2018a, 2018b, 2022](#); [Zhang, Yu, Auriol, & Pereira, 2023](#)) have further developed backstepping controllers for various traffic scenarios including multi-lane, multi-class and mixed-autonomy traffic. This paper primarily focuses on adaptive control of traffic PDE systems with uncertain parameters.

In traffic flow modeling, relaxation time is a critical parameter representing drivers' reaction delays to evolving traffic conditions. However, heterogeneity and unpredictability of individual driver behavior makes it impossible to obtain the relaxation time in practice. This uncertainty in relaxation time can significantly impact the stability and performance of traffic systems. Traditional control methods struggle to handle such uncertainties, making it difficult to ensure system stability and optimal performance under varying traffic conditions. To address these challenges, we adopt adaptive control strategies that allow for real-time adjustment of the controller gains to accommodate unknown or time-varying system characteristics, ensuring the desired system performance.

Early developments in adaptive control for PDEs, as in [Logemann and Townley \(1997\)](#), addressed systems stabilized via high-gain feedback, under a relative degree one condition. While these approaches ensured parameter identifiability, they required control input to be applied throughout the spatial domain. Considerable progress has been achieved in the adaptive stabilization of PDEs with uncertain parameters, especially for hyperbolic and parabolic systems ([Böhm, Demetriou, Reich, & Rosen, 1998](#); [Hong & Bentsman, 1994](#); [Krstic & Smyshlyayev, 2008a](#); [Smyshlyayev & Krstic, 2007](#)). Adaptive control methods ([Belhadjoudja, Maghenem, Witrant, & Prieur, 2023](#); [Di Meglio, Bresch-Pietri, & Aarsnes, 2014](#); [Kawan, Mironchenko, & Zamani, 2022](#)) can be categorized into Lyapunov-based design, identifier-based design and swapping-based design. After a decade of research, advancements in adaptive control have begun to be applied to coupled hyperbolic PDEs ([Anfinson & Aamo, 2019](#)).

Although adaptive control for PDE systems with unknown parameters has been extensively studied ([Anfinson & Aamo, 2018](#); [Auriol, 2020](#); [Hu, Di Meglio, Vazquez, & Krstic, 2015](#); [Krstic & Smyshlyayev, 2008b](#); [Smyshlyayev, Cerpa, & Krstic, 2010](#); [Wang & Krstic, 2020](#)) and was first applied for the ARZ PDE model in [Yu and Krstic \(2018a\)](#). The practical implementation of the adaptive controller for the traffic systems still faces challenge. This is because the adaptive control process simultaneously requires the estimation of unknown system parameters and PDE states. After each time step, it is necessary to recalculate the solution to the PDE corresponding to the gain kernel function in order to update the estimated system parameter functions. This places extremely high demands on real-time computation. The computational resources required for calculation of the gain function increase significantly with spatial sampling precision when applying traditional finite difference and finite element methods. In this paper, we adopt neural operators to accelerate computation of adaptive PDE backstepping controllers.

## 1.2. Advances in machine learning for PDE traffic control

With rapid advances in machine learning, data-driven methods for solving, modeling and control of PDEs have received widespread attention including physics-informed learning, reinforcement learning and operator learning. Physics-Informed Neural Networks (PINNs) directly incorporates physical constraints into neural networks training by embedding the physical laws

of PDEs into the loss function. This enables PINNs to solve PDEs without large amounts of training data. Mowlavi and Nabi extend PINNs method to PDE optimal control problems in [Mowlavi and Nabi \(2023\)](#). Zhao proposed a novel hybrid Traffic state estimation (TSE) approach called Observer-Informed Deep Learning (OIDL), which integrates a PDE observer and deep learning paradigm to estimate spatial-temporal traffic states from boundary sensing data in [Zhao and Yu \(2023\)](#). However, PINNs need to be retrained for each new set of boundary and initial conditions, which poses limitations in adaptive control applications.

Reinforcement learning (RL) has also been increasingly applied for PDE control problems, particularly in boundary and feedback control. RL continuously optimizes strategies to achieve real-time control of complex PDE systems. In the field of traffic management, researchers have been applying RL to various traffic issues. Wu et al. used the city mobility traffic micro-simulator SUMO to design a deep RL framework for hybrid autonomous traffic in various experimental scenarios ([Wu, Kreidieh, Vinitzky, & Bayen, 2017](#)). Under the same framework, [Qu, Yu, Zhou, Lin, and Wang \(2020\)](#) proposed a reinforcement learning-based car-following model for electric, connected, and automated vehicles to reduce traffic oscillations and improve energy efficiency. [Yu, Park, Bayen, Moura, and Krstic \(2021\)](#) presented the exploration using RL for traffic PDE boundary control. However, RL has limited generalization ability in practical applications. RL may perform well under the specific initial conditions. However, for initial conditions outside the training range, there may be performance degradation or even failure. RL may be sensitive to hyperparameters and exhibit unpredictable behavior, making it difficult to ensure consistent and stable performance in different scenarios.

Traditional neural networks typically learn mappings between finite dimensional Euclidean spaces, but with the advancement of research, this method has been extended to the field of NO ([Lu, Jin, Pang, Zhang, & Karniadakis, 2021](#)). NO-based learning methods focus on mapping between function spaces and are specifically designed for solving PDEs and dynamical systems. Compared with traditional machine learning methods, NO have two unique advantages. Firstly, theoretically speaking, NO can learn the mapping of the entire system parameter set, rather than being limited to a single system parameter like standard neural networks. Secondly, from an empirical perspective, research work ([Lu et al., 2021](#); [Shi et al., 2022](#)) has shown that NO have significantly better accuracy than traditional deep learning methods when simulating complex functions. Therefore, NO not only solves individual equation instances, but can also handle the problems of the entire PDE family.

Recent research has effectively utilized DeepONet for one-dimensional transport PDEs ([Bhan, Shi, & Krstic, 2023](#)), reaction-diffusion equations and observer designs ([Krstic, Bhan, & Shi, 2024](#)), as well as for hyperbolic PDEs with delay ([Qi, Zhang, & Krstic, 2024](#)), parabolic PDEs with delays ([Wang, Diagne, & Krstić, 2025b](#)),  $2 \times 2$  hyperbolic PDEs ([Wang, Diagne, & Krstic, 2025a](#)), traffic flow ([Zhang, Zhong, & Yu, 2024](#)) and cascaded parabolic PDEs ([Lv, Wang, & Cao, 2024](#)). In contrast to the approximate backstepping transformations used in [Bhan et al. \(2023\)](#), [Krstic et al. \(2024\)](#), [Lv et al. \(2024\)](#), [Qi et al. \(2024\)](#), [Wang et al. \(2025a, 2025b\)](#), [Zhang et al. \(2024\)](#), this paper adopts the exact backstepping transformation, referred to as a gain-only approach. The gain-only approach focuses on approximating a 1D gain kernel, simplifying network design, reducing training set size and time, and easing the derivation of the perturbed target system, which have been successfully used in gain scheduling that adjust controller gains based on current states of nonlinear PDE system ([Lamarque, Bhan, Vazquez, & Krstic, 2024](#)) and several benchmark unstable PDEs ([Vazquez & Krstic, 2024](#)). A recently developed method based on power series approximations ([Vazquez,](#)

Chen, Qiao, & Krstic, 2023), along with its MATLAB extension (Lin, Vazquez, & Krstic, 2024), shows promise as a tool for generating training datasets. The application of NO-approximated gain kernels becomes even more valuable for adaptive control, where the kernel must be recomputed online at each time step to accommodate updated estimates of the plant parameters. This was first explored for first-order hyperbolic PDE in Lamarque, Bhan, Shi, and Krstic (2025) and extended to the reaction–diffusion equation in Bhan, Shi, and Krstic (2025). Different from Bhan et al. (2025), Lamarque et al. (2025), where the kernel equation involves a single kernel, in this work, we extend the results of Lamarque et al. (2025) to the ARZ traffic models which involved the coupled heterogeneous hyperbolic PDEs. The technical challenges arise from both the more complex kernel computations and the proof analysis of the higher-order PDE systems with the approximated controllers.

**Contributions:** The main contributions are summarized as follows:

- We present an NO-based adaptive control method to stabilize the ARZ traffic PDE model with unknown relaxation time. Additionally, we extend stability schemes for more general  $2 \times 2$  hyperbolic systems with uncertain spatially varying in-domain coefficients and boundary parameter. Compared to the relevant works (Bhan et al., 2025; Lamarque et al., 2025), which approximate single kernel, a key technical challenge is dealing with the approximation of coupled  $2 \times 2$  Goursat-form PDE kernels in the stabilization of coupled  $2 \times 2$  hyperbolic PDEs.
- To address the computational challenges associated with solving gain kernel equations, we integrate DeepONet into the adaptive control framework. It is shown that the NO is almost two orders of magnitude faster than the PDE solver in solving kernel functions, and the loss remains on the order of  $10^{-3}$ . To the best of our knowledge, this is the first study to integrate DeepONet with adaptive control in traffic flow systems, demonstrating its potential to improve the computational efficiency of control schemes in congested traffic scenarios.
- Through comparative experiments with RL, it has been proven that our method does not rely on initial values compared to RL and provides a model-based solution with guaranteed stability. In addition, we theoretically prove the system's stability through Lyapunov analysis when replacing with the DeepONet approximation kernels in the adaptive controller.

**Organization of paper:** The paper is organized as follows. Section 2 introduces ARZ traffic PDE model and a nominal adaptive backstepping control scheme designed for  $2 \times 2$  hyperbolic PDEs. Section 3 gives a series of properties for the gain kernel and its time derivative and introduces the approximation of feedback kernel operators. Section 4 presents the stabilization achieved through the application of approximate controller gain functions via DeepONet. Numerical simulations are presented in Section 5. Section 6 presents the conclusion.

**Notation.** We present the nomenclature for kernel learning with exact and approximate operators in Table 1. We define the  $L^2$ -norm for  $\chi(x) \in L^2[0, 1]$  as  $\|\chi\|_2 = \int_0^1 |\chi(x)|^2 dx$ . We use  $\|\cdot\|_\infty$  for the infinity-norm, that is  $\|\chi\|_\infty = \sup_{x \in [0, 1]} |\chi(x)|$ . We set  $\|\chi\|_1 = \int_0^1 |\chi(x)| dx$ .

## 2. Nominal adaptive control design

### 2.1. ARZ PDE traffic model

The ARZ PDE model is used to describe the formation and dynamics of the traffic oscillations which refer to variations of

**Table 1**

Nomenclature for kernel learning with exact and approximate operators.

exact operator	$\mathcal{K}$
neural operator	$\hat{\mathcal{K}}$
unknown model parameters	$(c_1, c_2, c_3, c_4, r)$
estimated model parameters	$(\hat{c}_1, \hat{c}_2, \hat{c}_3, \hat{c}_4, \hat{r})$
exact kernel	$(K^u, K^m) = \mathcal{K}(c_1, c_2, c_3, c_4, r)$
exact estimated kernel	$(\check{K}^u, \check{K}^m) = \mathcal{K}(\hat{c}_1, \hat{c}_2, \hat{c}_3, \hat{c}_4, \hat{r})$
approximate estimated kernel	$(\hat{K}^u, \hat{K}^m) = \hat{\mathcal{K}}(\hat{c}_1, \hat{c}_2, \hat{c}_3, \hat{c}_4, \hat{r})$

traffic density and speed around equilibrium values. It consists of a set of  $2 \times 2$  hyperbolic PDEs for traffic density and velocity. The ARZ model of  $(\rho(x, t), v(x, t))$ -system is given by

$$\begin{aligned} \partial_t \rho + \partial_x(\rho v) &= 0, \\ \partial_t(v - V(\rho)) + v \partial_x(v - V(\rho)) &= \frac{V(\rho) - v}{\tau}, \\ \rho(0, t) &= \frac{q^*}{v(0, t)}, \\ v(L, t) &= U(t) + v^*, \end{aligned} \quad (1)$$

where  $(x, t) \in [0, L] \times \mathbb{R}_+$ ,  $\rho(x, t)$  represents the traffic density,  $v(x, t)$  represents the traffic speed, and  $\tau$  denotes the relaxation time, which refers to the time required for driver behavior to adapt to equilibrium. This parameter is used to describe the process by which vehicle speed adjusts to match the traffic density. The variable  $p(\rho)$ , defined as the traffic system pressure, is related to the density by the equation

$$p(\rho) = c_0(\rho)^\gamma,$$

and  $c_0, \gamma \in \mathbb{R}_+$ . The equilibrium velocity–density relationship  $V(\rho)$  is given in Greensfield model:

$$V(\rho) = v_f \left( 1 - \left( \frac{\rho}{\rho_m} \right)^\gamma \right),$$

where  $v_f$  is the velocity of free flow,  $\rho_m$  is the maximum density of free flow, and  $(\rho^*, v^*)$  are the equilibrium points of the system with  $v^* = V(\rho^*)$ . We consider a constant traffic flux  $q^* = \rho^* v^*$  entering the domain from  $x = 0$  and there is a Varying Speed Limit(VSL) boundary control at the outlet.  $U(t)$  is defined as variation from steady state velocity. The VSL at outlet shows  $v^*$  with  $U(t)$  which we will design later. We can apply the change of coordinates introduced in Yu and Krstic (2018b) to rewrite it in the Riemann coordinates and then map it to a decoupled first-order  $2 \times 2$  hyperbolic system.

$$\begin{aligned} \partial_t u_1 + v^* \partial_x u_1 &= 0, \\ \partial_t m_1 - (\gamma p^* - v^*) \partial_x m_1 &= c(x) u_1, \\ u_1(0, t) &= r_0 m_1(0, t), \\ m_1(L, t) &= U(t), \end{aligned} \quad (2)$$

where

$$c(x) = -\frac{1}{\tau} \exp\left(-\frac{x}{\tau v^*}\right), r_0 = \frac{\rho^* V(\rho^*) + v^*}{v^*}.$$

The relaxation time  $\tau$  describes how fast drivers adapt their speed to equilibrium speed–density relations. Its value is usually difficult to measure in practice and is easily affected by various external factors. Therefore, we propose adaptive control law. Motivated by the second-order ARZ model, we first propose NO-based adaptive design for a more general framework of  $2 \times 2$  hyperbolic PDEs with spatially varying coefficients, as the linearized ARZ model (2) is a special case of such systems.

## 2.2. Adaptive control for coupled $2 \times 2$ hyperbolic PDEs

We consider the first-order coupled  $2 \times 2$  hyperbolic PDE system with four spatially variable coefficients,

$$\begin{aligned} \partial_t u(x, t) + \lambda \partial_x u(x, t) &= c_1(x)u(x, t) + c_2(x)m(x, t), \\ \partial_t m(x, t) - \mu \partial_x m(x, t) &= c_3(x)u(x, t) + c_4(x)m(x, t), \\ u(0, t) &= rm(0, t), \\ m(1, t) &= U(t), \end{aligned} \quad (3)$$

where  $t \in \mathbb{R}_+$  is the time,  $x \in [0, 1]$  is the space, the states are given by  $u, m$  and the initial conditions are  $u(x, 0) = u_0(x)$ ,  $m(x, 0) = m_0(x)$  where  $u_0, m_0 \in L^2([0, 1])$ . The positive transport speeds  $\lambda, \mu \in \mathbb{R}$  are known. We assume the spatially variable coefficients  $c_1(x), c_2(x), c_3(x), c_4(x) \in C([0, 1])$  and boundary coefficient  $r \in \mathbb{R}$  are unknown.

Note that system (3) is  $2 \times 2$  hyperbolic system with spatially variable coefficients in domain, which is different from the system in Anfinen and Aamo (2018) with the constant coefficients. System (3) is a direct extension of the system in Anfinen and Aamo (2018), where the difference lies in the designed adaptive update law.

To ensure the well-posedness of the kernel PDEs, the adaptive control estimation requires bounded assumptions. Our basic assumption is as follows.

**Assumption 1.** Bounds are known on all uncertain parameters, that is, there exists some constants  $\bar{c}_i, i = 1 \dots 4$ , and  $\bar{r}$  so that

$$\|c_i\|_\infty \leq \bar{c}_i, i = 1 \dots 4, |r| \leq \bar{r}. \quad (4)$$

We first propose an adaptive control design using passive identifier design method, which includes the exact estimated backstepping kernels  $\check{K}^u, \check{K}^m$ .

We consider the identifier

$$\begin{aligned} \partial_t \hat{u}(x, t) &= -\lambda \partial_x \hat{u}(x, t) + \hat{c}_1(x, t)u(x, t) \\ &\quad + \hat{c}_2(x, t)m(x, t) + \rho e_1(x, t)\|\varpi(t)\|^2, \\ \partial_t \hat{m}(x, t) &= \mu \partial_x \hat{m}(x, t) + \hat{c}_3(x, t)u(x, t) \\ &\quad + \hat{c}_4(x, t)m(x, t) + \rho e_2(x, t)\|\varpi(t)\|^2, \\ \hat{u}(0, t) &= \frac{\hat{r}u(0, t) + u(0, t)m^2(0, t)}{1 + m^2(0, t)}, \\ \hat{m}(1, t) &= U(t), \end{aligned} \quad (5)$$

where  $\rho > 0$ ,

$$e_1(x, t) = u(x, t) - \hat{u}(x, t), e_2(x, t) = m(x, t) - \hat{m}(x, t), \quad (6)$$

are errors between  $u$  and  $m$  and their estimates  $\hat{u}$  and  $\hat{m}$ ,  $\hat{c}_i$  and  $\hat{r}$  are estimates  $c_i$  and  $r$ . We define

$$\varpi(x, t) = [u(x, t), m(x, t)]^T,$$

for some initial conditions

$$\hat{u}_0, \hat{m}_0 \in L^2([0, 1]).$$

The error signals (6) can straightforwardly be shown to have dynamics

$$\begin{aligned} \partial_t e_1(x, t) &= -\lambda \partial_x e_1(x, t) + \tilde{c}_1(x, t)u(x, t) \\ &\quad + \tilde{c}_2(x, t)m(x, t) - \rho e_1(x, t)\|\varpi(t)\|^2, \end{aligned} \quad (7)$$

$$\begin{aligned} \partial_t e_2(x, t) &= \mu \partial_x e_2(x, t) + \tilde{c}_3(x, t)u(x, t) \\ &\quad + \tilde{c}_4(x, t)m(x, t) - \rho e_2(x, t)\|\varpi(t)\|^2, \end{aligned} \quad (8)$$

$$e_1(0, t) = \frac{\tilde{r}(t)m(0, t)}{1 + m^2(0, t)}, \quad (9)$$

$$e_2(1, t) = 0, \quad (10)$$

where

$$\tilde{r} = r - \hat{r}, \quad \tilde{c}_i = c_i - \hat{c}_i, \quad i = 1, \dots, 4.$$

We choose the following update laws

$$\begin{aligned} \hat{c}_{1t}(x, t) &= \text{Proj}_{\bar{c}_1} \{ \gamma_1 e^{-\gamma x} e_1(x, t)u(x, t), \hat{c}_1(x, t) \}, \\ \hat{c}_{2t}(x, t) &= \text{Proj}_{\bar{c}_2} \{ \gamma_2 e^{-\gamma x} e_1(x, t)m(x, t), \hat{c}_2(x, t) \}, \\ \hat{c}_{3t}(x, t) &= \text{Proj}_{\bar{c}_3} \{ \gamma_3 e^{\gamma x} e_2(x, t)u(x, t), \hat{c}_3(x, t) \}, \\ \hat{c}_{4t}(x, t) &= \text{Proj}_{\bar{c}_4} \{ \gamma_4 e^{\gamma x} e_2(x, t)m(x, t), \hat{c}_4(x, t) \}, \\ \hat{r}(t) &= \text{Proj}_{\bar{r}} \{ \gamma_5 e_1(0, t)m(0, t), \hat{r}(t) \}, \end{aligned} \quad (11)$$

where  $\gamma, \gamma_1, \gamma_2, \gamma_3, \gamma_4, \gamma_5 > 0$  are scalar design gains. Proj denotes the projection operator

$$\text{Proj}_{\bar{\omega}}\{\tau, \hat{\omega}\} = \begin{cases} 0 & |\hat{\omega}| \geq \bar{\omega} \text{ and } \hat{\omega}\tau \geq 0, \\ \tau & \text{otherwise.} \end{cases}$$

The projection operator  $\text{Proj}_{\bar{\omega}}\{\tau, \hat{\omega}\}$  is designed to constrain the update of parameter  $\hat{\omega}$ , ensuring that it does not exceed the predefined bound  $\bar{\omega}$ . The adaptive laws (11) have the following properties for all  $t > 0$  (Krstic, 2009)

$$-\tilde{\omega}^T \text{Proj}_{\bar{\omega}}\{\tau, \hat{\omega}\} \leq -\tilde{\omega}^T \tau. \quad (12)$$

**Lemma 1** (Properties of passive identifier). Consider the system (3) and the identifier (5), with an arbitrary initial condition  $\hat{u}_0 = \hat{u}(\cdot, 0)$ ,  $\hat{m}_0 = \hat{m}(\cdot, 0)$  such that  $\|\hat{u}_0\| < \infty$ ,  $\|\hat{m}_0\| < \infty$ , along with the update law (11) with an arbitrary Lipschitz initial conditions satisfying the bounds (4), guarantees the following properties

$$\|\hat{c}_i(\cdot, t)\|_\infty \leq \bar{c}_i, \quad \forall i = 1, \dots, 4, \quad |\hat{r}| \leq \bar{r}, \quad (13)$$

$$\|e_1\|, \|e_2\| \in L^\infty \cap L^2, \quad (14)$$

$$\|e_1\| \|\varpi\|, \|e_2\| \|\varpi\| \in L^2, \quad (15)$$

$$|e_1(0, \cdot)|, |e_1(1, \cdot)|, |e_2(0, \cdot)|, |e_1(0, \cdot)u(0, \cdot)| \in L^2, \quad (16)$$

$$\|\partial_t \hat{c}_i\|, |\hat{r}| \in L^2, \quad (17)$$

$$\frac{\tilde{r}m(0, \cdot)}{\sqrt{1 + m^2(0, \cdot)}} \in L^2. \quad (18)$$

**Proof.** The proof can be found in Appendix. ■

Considering the plant (3) with unknown parameters  $c_i, i = 1, \dots, 4$  and  $r$ , we will design a nominal adaptive control law to achieve global stability.

We consider the following adaptive backstepping transformation

$$\begin{aligned} w(x, t) &= \hat{u}(x, t), \\ z(x, t) &= \hat{m}(x, t) - \int_0^x \check{K}^u(x, \xi, t) \hat{u}(\xi, t) d\xi \\ &\quad - \int_0^x \check{K}^m(x, \xi, t) \hat{m}(\xi, t) d\xi = T[\hat{u}, \hat{m}](x, t), \end{aligned} \quad (19)$$

where the kernels  $\check{K}^u$  and  $\check{K}^m$  satisfy the following kernel functions

$$\begin{aligned} \mu \check{K}_x^u(x, \xi, t) &= \lambda \check{K}_\xi^u(x, \xi, t) + \hat{c}_3(\xi, t) \check{K}^m(x, \xi, t) \\ &\quad + (\hat{c}_1(\xi, t) - \hat{c}_4(\xi, t)) \check{K}^u(x, \xi, t), \\ \mu \check{K}_x^m(x, \xi, t) &= -\mu \check{K}_\xi^m(x, \xi, t) + \hat{c}_2(\xi, t) \check{K}^u(x, \xi, t), \\ \check{K}^u(x, x, t) &= -\frac{\hat{c}_3(x, t)}{\lambda + \mu}, \\ \check{K}^m(x, 0, t) &= \frac{\lambda \hat{r}(t)}{\mu} \check{K}^u(x, 0, t). \end{aligned} \quad (20)$$



The coupled  $2 \times 2$  Goursat-form PDEs, governed by two gain kernels, are defined over the triangular domain  $\mathcal{T}_1$ , given by:

$$\mathcal{T}_1 = \{(x, \xi) \mid 0 \leq \xi \leq x \leq 1\}. \quad (21)$$

The kernel functions  $\check{K}^u(x, \xi, t)$  and  $\check{K}^m(x, \xi, t)$  are computed on-line by solving the equation of  $2 \times 2$  hyperbolic PDEs whose dynamics depend on the unknown parameters  $\hat{c}_i(x, t)$ ,  $i = 1, \dots, 4$ , and  $\hat{r}(t)$ . These parameters are continuously estimated and updated via the adaptive laws (11), so that at each time step, the kernel functions must be continually recalculated according to the new  $\hat{c}_i(x, t)$  and  $\hat{r}(t)$  estimate.

Using the transformation (19), we get the following target system

$$\begin{aligned} w_t(x, t) = & -\lambda w_x(x, t) + \hat{c}_1 w(x, t) + \hat{c}_1 e_1(x, t) + \hat{c}_2 z(x, t) \\ & + \int_0^x \omega(x, \xi, t) w(\xi, t) d\xi \\ & + \int_0^x \kappa(x, \xi, t) z(\xi, t) d\xi \\ & + \hat{c}_2 e_2(x, t) + \rho e_1(x, t) \|\varpi(t)\|^2, \\ z_t(x, t) = & \mu z_x(x, t) + \hat{c}_4 z(x, t) - \lambda \check{K}^u(x, 0, t) r(t) e_2(0, t) \\ & - \lambda \check{K}^u(x, 0, t) \tilde{r}(t) z(0, t) + \lambda \check{K}^u(x, 0, t) e_1(0, t) \\ & - \int_0^x \check{K}_t^u(x, \xi, t) w(\xi, t) d\xi \\ & - \int_0^x \check{K}_t^m(x, \xi, t) T^{-1}[w, z](\xi, t) d\xi \\ & + T[\hat{c}_1 e_1 + \hat{c}_2 e_2, \hat{c}_3 e_1 + \hat{c}_4 e_2](x, t) \\ & + \rho T[e_1, e_2](x, t) \|\varpi(t)\|^2, \\ w(0, t) = & r(t) z(0, t) + r(t) e_2(0, t) - e_1(0, t), \\ z(1, t) = & 0, \end{aligned} \quad (22)$$

where the coefficient  $\omega$  and  $\kappa$  are chosen to satisfy

$$\begin{aligned} \omega(x, \xi, t) = & \hat{c}_2(x, t) \check{K}^u(x, \xi, t) + \int_\xi^x \kappa(x, s, t) \check{K}^u(s, \xi, t) ds, \\ \kappa(x, \xi, t) = & \hat{c}_2(x, t) \check{K}^m(x, \xi, t) + \int_\xi^x \kappa(x, s, t) \check{K}^m(s, \xi, t) ds. \end{aligned}$$

From the boundary condition of (3), (19) and boundary condition of (22), the nominal stabilizing controller is straightforwardly derived as follows

$$U(t) = \int_0^1 \check{K}^u(1, \xi, t) \hat{u}(\xi, t) d\xi + \int_0^1 \check{K}^m(1, \xi, t) \hat{m}(\xi, t) d\xi. \quad (23)$$

Next, we present the stability of exact adaptive backstepping control, which serves as a guide for what we aim to achieve under the NO-based approximate adaptive backstepping design.

**Theorem 1.** [Stability of exact adaptive backstepping control] Consider the plant (3) in feedback with the adaptive control law (23) along with the update law for  $\hat{c}_1, \hat{c}_2, \hat{c}_3, \hat{c}_4, \hat{r}$  given by (11) and the passive identifier  $\hat{u}, \hat{m}$  given by (5) satisfies the following properties for all solutions for all time:

$$\begin{aligned} \|u\|, \|m\|, \|\hat{u}\|, \|\hat{m}\|, \|u\|_\infty, \|m\|_\infty, \|\hat{u}\|_\infty, \|\hat{m}\|_\infty &\in L^2 \cap L^\infty, \\ \|u\|, \|m\|, \|\hat{u}\|, \|\hat{m}\|, \|\hat{u}\|_\infty, \|\hat{m}\|_\infty, \|\hat{u}\|_\infty, \|\hat{m}\|_\infty &\mapsto 0. \end{aligned}$$

**Proof.** The proof of this theorem follows a similar approach to the proof of Theorem 9.1 in Anfinson and Aamo (2019). ■

In summary, the exact adaptive backstepping feedback law (23) can achieve global stability at the equilibrium point, with the system states  $u(x, t), m(x, t)$  converging pointwise to zero.

However, this method is computationally intensive because it requires solving Volterra Eqs. (20) at each time step  $t$ . To simplify the computation, we propose using the NO  $\hat{\mathcal{K}} : (\hat{c}_1, \hat{c}_2, \hat{c}_3, \hat{c}_4, \hat{r}) \mapsto (\hat{K}^u, \hat{K}^m)$  to approximate the exact adaptive backstepping gain operator  $\mathcal{K} : (\hat{c}_1, \hat{c}_2, \hat{c}_3, \hat{c}_4, \hat{r}) \mapsto (\check{K}^u, \check{K}^m)$ . This approach allows for neural network evaluation at each time step instead of solving the complex equations. In the following section, we will introduce how to approximate the operator  $\mathcal{K}$  using DeepONet and use the resulting approximate control gain functions for boundary stabilization of plant (3). And we will use the universal approximation theorem of DeepONet (Deng, Shin, Lu, Zhang, & Karniadakis, 2022) to derive the stability theorem from control gain kernel approximations.

### 3. Neural operator for approximating gain kernels

By proving the continuity and boundedness mentioned above, it can be further deduced that for a set of continuous coefficients within a certain supremely bounded norm, there exists a NO with arbitrary accuracy.

#### 3.1. Properties of the gain kernel functions

**Lemma 2.** Let  $\|\hat{c}_i(\cdot, t)\|_\infty \leq \bar{c}_i, |\hat{r}| \leq \bar{r}, \forall (x, t) \in [0, 1] \times \mathbb{R}_+$ . Then, for any fixed  $t \in \mathbb{R}_+$  and for any  $\lambda, \mu \in \mathbb{R}_+$ ,  $\hat{c}_i \in C([0, 1])$  and  $r \in \mathbb{R}_+, t \geq 0$ , the gain kernels  $\check{K}^u, \check{K}^m$  satisfying the PDE systems (20), have unique  $C(\mathcal{T}_1)$  solutions with the property

$$|\check{K}^u(x, \xi, t)| \leq \bar{K}, \quad (24)$$

$$|\check{K}^m(x, \xi, t)| \leq \bar{K}, \quad (25)$$

$$\begin{aligned} \|\check{K}_t^u(x, \xi, t)\| \leq & M_1 \|\hat{c}_{1t}\| + M_2 \|\hat{c}_{2t}\| + M_3 \|\hat{c}_{3t}\| \\ & + M_4 \|\hat{c}_{4t}\| + M_5 \|\hat{r}\|, \end{aligned} \quad (26)$$

$$\begin{aligned} \|\check{K}_t^m(x, \xi, t)\| \leq & M_6 \|\hat{c}_{1t}\| + M_7 \|\hat{c}_{2t}\| + M_8 \|\hat{c}_{3t}\| \\ & + M_9 \|\hat{c}_{4t}\| + M_{10} \|\hat{r}\|, \end{aligned} \quad (27)$$

where  $\bar{K} > 0, M_j > 0, j = 1, \dots, 10$  are constants depending on the parameter bounds (4).

**Proof.** Fix time  $t \geq 0$ . The existence of a unique, bounded solution to kernel Eqs. (20) are guaranteed by Coron, Vazquez, Krstic, and Bastin (2013, Theorem A.1). Furthermore, this result ensures that the solution satisfies uniform bounds of the form

$$|\check{K}^u(x, \xi, t)| \leq \bar{K}, |\check{K}^m(x, \xi, t)| \leq \bar{K}, \quad t \geq 0,$$

where  $\bar{K}$  is a constant determined by the compactness of the admissible parameter set  $\hat{c}_1, \dots, \hat{c}_4, \hat{r}$ .

Differentiating Eq. (20) with respect to time yields the following system in terms of  $\check{K}_t^u$  and  $\check{K}_t^m$ :

$$\begin{aligned} \mu \check{K}_{tx}^u - \lambda \check{K}_{t\xi}^u = & (\hat{c}_1 - \hat{c}_4) \check{K}_t^u + \hat{c}_3 \check{K}_t^m \\ & + (\hat{c}_{1t} - \hat{c}_{4t}) \check{K}^u + \hat{c}_{3t} \check{K}^m, \\ \mu \check{K}_{tx}^m + \mu \check{K}_{t\xi}^m = & \hat{c}_2 \check{K}_t^u + \hat{c}_{2t} \check{K}^u, \\ \check{K}_t^m(x, 0) = & \frac{\lambda \hat{r}}{\mu} \check{K}_t^u(x, 0) + \frac{\lambda \hat{r}}{\mu} \check{K}^u(x, 0), \\ \check{K}_t^u(x, x) = & -\frac{\hat{c}_{3t}}{\lambda + \mu}. \end{aligned} \quad (28)$$

Applying again the result of Coron et al. (2013, Theorem A.1) to the Eqs. (28), one obtains the existence and uniqueness of a bounded solution  $(\check{K}_t^u, \check{K}_t^m)$ , with bounds of the form (26)–(27). ■

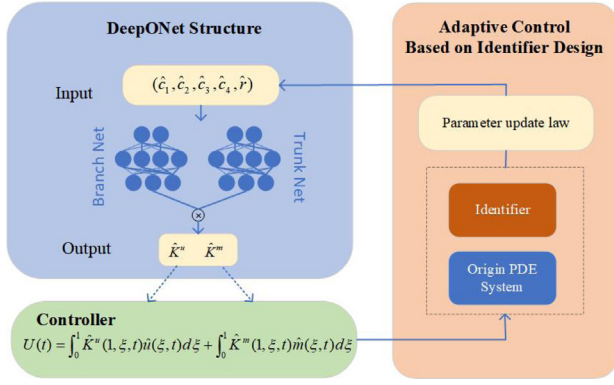


Fig. 1. The operator learning framework for adaptive control.

### 3.2. Approximation of the neural operator

In the following discussion, we will first introduce the universal approximation theorem of DeepONet. This theorem demonstrates DeepONet's ability to approximate operators, enabling us to use it to learn the backstepping gain kernel mapping of PDEs and theoretically guarantee the stability of the adaptive control system.

**Theorem 2** (DeepONet universal approximation theorem (Deng et al., 2022)). Let  $X \subset \mathbb{R}^{d_x}$  and  $Y \subset \mathbb{R}^{d_y}$  be compact sets of vectors  $x \in X$  and  $y \in Y$ , respectively. Let  $\mathcal{U} : X \mapsto U \subset \mathbb{R}^{d_u}$  and  $\mathcal{V} : Y \mapsto V \subset \mathbb{R}^{d_v}$  be sets of continuous functions  $u(x)$  and  $m(y)$ , respectively. Let  $\mathcal{U}$  be also compact. Assume the operator  $\mathcal{G} : \mathcal{U} \mapsto \mathcal{V}$  is continuous. Then for all  $\epsilon > 0$ , there exist  $n^*, p^* \in \mathbb{N}$  such that for each  $n \geq n^*, p \geq p^*$ , there exist  $\theta^{(i)}, \vartheta^{(i)}$  for neural networks  $f^{\mathcal{N}}(\cdot; \theta^{(i)}), g^{\mathcal{N}}(\cdot; \vartheta^{(i)}), i = 1, \dots, p$ , and  $x_j \in X, j = 1, \dots, n$ , with corresponding  $\mathbf{u}_n = (u(x_1), u(x_2), \dots, u(x_n))^T$ , such that

$$|\mathcal{G}(u)(y) - \mathcal{G}_{\mathbf{N}}(\mathbf{u}_n)(y)| \leq \epsilon$$

where

$$\mathcal{G}_{\mathbf{N}}(\mathbf{u}_n)(y) = \sum_{i=1}^p g^{\mathcal{N}}(\mathbf{u}_n; \vartheta^{(i)}) f^{\mathcal{N}}(y; \theta^{(i)}),$$

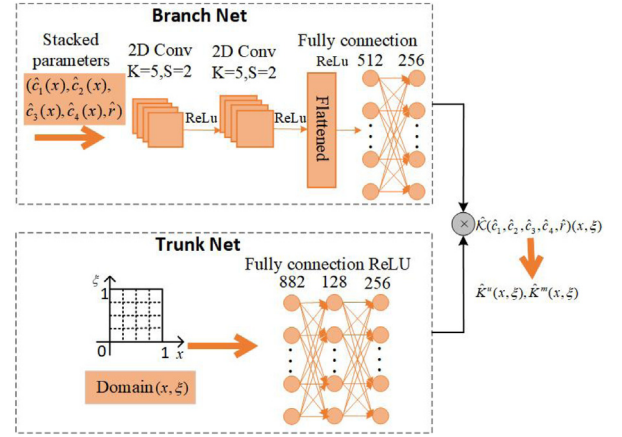
for all functions  $u \in \mathcal{U}$  and for all values  $y \in Y$  of  $\mathcal{G}(u)$ .

Fig. 1 presents a schematic diagram of the control circuit, illustrating the utilization of neural operators to accelerate the generation process of gain kernel functions in PDE adaptive control.

As illustrated in Fig. 2, the DeepONet consists of two sub-networks: a Branch Net and a Trunk Net. The Branch Net takes a five-channel 2D input and extracts spatial features using two convolutional layers, each with a kernel size of 5 and a stride of 2. The output is flattened and passed through two fully connected layers with 512 and 256 neurons, respectively, each followed by a ReLU activation. The Trunk Net receives spatial coordinate input and processes it through three fully connected layers with 884, 128 and 256 neurons, also activated by ReLU functions. The outputs from both networks are then combined via inner product to produce the approximated kernel function value  $\hat{\mathcal{K}}$ .

**Theorem 3.** [Existence of a neural operator to approximating the kernels] Fix  $t > 0$ . Fix a compact set  $K \subset (C([0, 1]))^4 \times \mathbb{R}$  and define the operator  $\mathcal{K} : K \mapsto (C(\mathcal{T}_1))^2$

$$\mathcal{K}(\hat{c}_1, \hat{c}_2, \hat{c}_3, \hat{c}_4, \hat{r})(\cdot, t) := (\tilde{K}^u(x, \xi), \tilde{K}^m(x, \xi)).$$

Fig. 2. The DeepONet structure for operator  $\hat{\mathcal{K}}$ .

Then, for all  $\epsilon > 0$ , there exists a neural operator  $\hat{\mathcal{K}} : K \mapsto (C(\mathcal{T}_1))^2$  such that for all  $(x, \xi) \in \mathcal{T}_1$ ,

$$|\mathcal{K}(\hat{c}_1, \hat{c}_2, \hat{c}_3, \hat{c}_4, \hat{r})(\cdot, t) - \hat{\mathcal{K}}(\hat{c}_1, \hat{c}_2, \hat{c}_3, \hat{c}_4, \hat{r})(\cdot, t)| \leq \epsilon.$$

**Proof.** The continuity of the operator  $\mathcal{K}$  is derived from Lemma 2. And this result is based on Theorem 2.1 proposed by B. Deng et al. in their study (Deng et al., 2022). ■

### 4. Stabilization under DeepONet-approximated gain feedback

We will demonstrate that although the adaptive controller uses approximate estimated kernel functions, the stability of the system is still guaranteed. Based on Theorem 1, we present the system stability proof for the adaptive backstepping controller using approximate estimation kernels  $\hat{K}^u, \hat{K}^m$  in the following theorem.

**Theorem 4** (Stabilization under approximate adaptive backstepping control). For all  $\bar{c}_1, \bar{c}_2, \bar{c}_3, \bar{c}_4, \bar{r} > 0$ , there exists a constant  $\epsilon_0 > 0$  such that for all NO approximations  $\hat{K}^u, \hat{K}^m$  of accuracy  $\epsilon \in (0, \epsilon_0)$  provided by Theorem 3, the plant (3) in feedback with the adaptive control law

$$U(t) = \int_0^1 \hat{K}^u(1, \xi, t) \hat{u}(\xi, t) d\xi + \int_0^1 \hat{K}^m(1, \xi, t) \hat{m}(\xi, t) d\xi, \quad (29)$$

along with the update law for  $\hat{c}_1, \hat{c}_2, \hat{c}_3, \hat{c}_4$  and  $\hat{r}$  given by (11) with any Lipschitz initial condition  $\hat{c}_{10} = \hat{c}_1(\cdot, 0), \hat{c}_{20} = \hat{c}_2(\cdot, 0), \hat{c}_{30} = \hat{c}_3(\cdot, 0), \hat{c}_{40} = \hat{c}_4(\cdot, 0)$  such that  $\|\hat{c}_{10}\| \leq \bar{c}_1, \|\hat{c}_{20}\| \leq \bar{c}_2, \|\hat{c}_{30}\| \leq \bar{c}_3, \|\hat{c}_{40}\| \leq \bar{c}_4$ , and  $|\hat{r}(0)| \leq \bar{r}$ . The passive identifier  $\hat{u}, \hat{m}$  given by (5) with any initial condition  $\hat{u}_0 = \hat{u}(\cdot, 0), \hat{m}_0 = \hat{m}(\cdot, 0)$  such that  $\|\hat{u}_0\| < \infty, \|\hat{m}_0\| < \infty$ , the following properties hold:

$$\|u\|, \|m\|, \|\hat{u}\|, \|\hat{m}\|, \|u\|_\infty, \|m\|_\infty, \|\hat{u}\|_\infty, \|\hat{m}\|_\infty \in L^2 \cap L^\infty, \\ \|u\|_\infty, \|m\|_\infty, \|\hat{u}\|_\infty, \|\hat{m}\|_\infty \mapsto 0.$$

Moreover, for the equilibrium  $(u, m, \hat{u}, \hat{m}, \hat{c}_1, \hat{c}_2, \hat{c}_3, \hat{c}_4, \hat{r}) = (0, 0, 0, 0, c_1, c_2, c_3, c_4, r)$  the following global stability estimate holds

$$S(t) \leq 2 \frac{k_2}{k_1} \theta_2 S(0) e^{\theta_1 k_2 S(0)}, \quad t > 0, \quad (30)$$

where

$$S(t) := \|u\|^2 + \|m\|^2 + \|\hat{u}\|^2 + \|\hat{m}\|^2 + \|\tilde{c}_1\|^2 + \|\tilde{c}_2\|^2 \\ + \|\tilde{c}_3\|^2 + \|\tilde{c}_4\|^2 + \tilde{r}^2,$$

and  $k_1, k_2, \theta_1$  and  $\theta_2$  are strictly positive constants.

**Proof.** This proof mainly refers to [Anfinson and Aamo \(2019, Chapter 9\)](#), and makes necessary supplements to the gain approximation error while reducing repetition.

#### A. DeepONet-perturbed target system

We consider the following adaptive backstepping transformation [\(19\)](#)

$$\begin{aligned} w(x, t) &= \hat{u}(x, t), \\ z(x, t) &= \hat{m}(x, t) - \int_0^x \check{K}^u(x, \xi, t) \hat{u}(\xi, t) d\xi \\ &\quad - \int_0^x \check{K}^m(x, \xi, t) \hat{m}(\xi, t) d\xi =: T[\hat{u}, \hat{m}](x, t), \end{aligned} \quad (31)$$

where  $\check{K}^u$  and  $\check{K}^m$  are exact solutions to kernel function [\(20\)](#). The transformation is an invertible backstepping transformation, with inverse in the same form

$$\begin{aligned} \hat{u}(x, t) &= w(x, t), \\ \hat{m}(x, t) &= z(x, t) + \int_0^x \check{L}^u w(\xi, t) d\xi \\ &\quad + \int_0^x \check{L}^m z(\xi, t) d\xi =: T^{-1}[w, z](x, t) \end{aligned} \quad (32)$$

where  $T^{-1}$  is an operator similar to  $T$ . From [Lemma 2](#),  $\check{K}^u, \check{K}^m$  are continuous, there exist unique continuous inverse kernels  $\check{L}^u, \check{L}^m$  defined on  $\mathcal{T}_1$  and there exists a constant  $\bar{L}$  so that  $\|\check{L}^u\|_\infty \leq \bar{L}$ ,  $\|\check{L}^m\|_\infty \leq \bar{L}$ . We will derive the DeepONet-perturbed target system with exact estimated kernels. Because the controller we have chosen is [\(29\)](#), where the kernels  $\hat{K}^u$  and  $\hat{K}^m$  are approximated by NO. This transformation lead to the following target system

$$\begin{aligned} w_t(x, t) &= -\lambda w_x(x, t) + \hat{c}_1 w(x, t) + \hat{c}_1 e_1(x, t) + \hat{c}_2 z(x, t) \\ &\quad + \int_0^x \omega(x, \xi, t) w(\xi, t) d\xi \\ &\quad + \int_0^x \kappa(x, \xi, t) z(\xi, t) d\xi \\ &\quad + \hat{c}_2 e_2(x, t) + \rho e_1(x, t) \|\varpi(t)\|^2, \end{aligned} \quad (33)$$

$$\begin{aligned} z_t(x, t) &= \mu z_x(x, t) + \hat{c}_4 z(x, t) - \lambda \check{K}^u(x, 0, t) r(t) e_2(0, t) \\ &\quad - \lambda \check{K}^u(x, 0, t) \tilde{r}(t) z(0, t) + \lambda \check{K}^u(x, 0, t) e_1(0, t) \\ &\quad - \int_0^x \check{K}_t^u(x, \xi, t) w(\xi, t) d\xi \\ &\quad - \int_0^x \check{K}_t^m(x, \xi, t) T^{-1}[w, z](\xi, t) d\xi \\ &\quad + T[\hat{c}_1 e_1 + \hat{c}_2 e_2, \hat{c}_3 e_1 + \hat{c}_4 e_2](x, t) \\ &\quad + \rho T[e_1, e_2](x, t) \|\varpi(t)\|^2, \end{aligned} \quad (34)$$

$$w(0, t) = r(t) z(0, t) + r(t) e_2(0, t) - e_1(0, t), \quad (35)$$

$$\begin{aligned} z(1, t) &= - \int_0^1 \check{K}^u(1, \xi, t) w(\xi, t) d\xi \\ &\quad - \int_0^1 \check{K}^m(1, \xi, t) T^{-1}[w, z](\xi, t) d\xi := \Gamma(t), \end{aligned} \quad (36)$$

where

$$\begin{aligned} \omega(x, \xi, t) &= \hat{c}_2(x, t) \check{K}^u(x, \xi, t) + \int_\xi^x \kappa(x, s, t) \check{K}^u(s, \xi, t) ds, \\ \kappa(x, \xi, t) &= \hat{c}_2(x, t) \check{K}^m(x, \xi, t) + \int_\xi^x \kappa(x, s, t) \check{K}^m(s, \xi, t) ds. \end{aligned}$$

The main difference between the current system [\(33\)–\(36\)](#) and the system described in [\(22\)](#) is the perturbation  $\Gamma(t)$  in the boundary conditions [\(36\)](#). This difference is due to the controller [\(29\)](#) using an approximated estimated kernels  $\hat{K}^u$  and  $\hat{K}^m$  instead

of the exact estimated kernels  $\check{K}^u$  and  $\check{K}^m$ . The specific derivation process of [\(36\)](#) is as follows

$$\begin{aligned} z(1, t) &= \hat{m}(1, t) - \int_0^1 \check{K}^u(1, \xi, t) \hat{u}(\xi, t) d\xi \\ &\quad - \int_0^1 \check{K}^m(1, \xi, t) \hat{m}(\xi, t) d\xi \\ &= U(t) - \int_0^1 \check{K}^u(1, \xi, t) \hat{u}(\xi, t) d\xi \\ &\quad - \int_0^1 \check{K}^m(1, \xi, t) \hat{m}(\xi, t) d\xi \\ &= \int_0^1 \hat{K}^u(1, \xi, t) \hat{u}(\xi, t) d\xi + \int_0^1 \hat{K}^m(1, \xi, t) \hat{m}(\xi, t) d\xi \\ &\quad - \int_0^1 \check{K}^u(1, \xi, t) \hat{u}(\xi, t) d\xi - \int_0^1 \check{K}^m(1, \xi, t) \hat{m}(\xi, t) d\xi \\ &= - \int_0^1 \check{K}^u(1, \xi, t) \hat{u}(\xi, t) d\xi - \int_0^1 \check{K}^m(1, \xi, t) \hat{m}(\xi, t) d\xi. \end{aligned}$$

By using [\(32\)](#), we obtained [\(36\)](#).

In the following part, we introduce the spatial  $L^2$  boundedness and regulation of plant and observer states.

We use the following Lyapunov function candidate

$$V(t) := V_4(t) + aV_5(t), \quad (37)$$

and

$$V_4(t) := \|w(t)\|_{-\delta}^2 = \int_0^1 e^{-\delta x} w^2(x, t) dx, \quad (38)$$

$$V_5(t) := \|z(t)\|_k^2 = \int_0^1 e^{kx} z^2(x, t) dx, \quad (39)$$

where  $a, \delta, k > 0$ . We deduce that there exist positive constants  $a_1, a_2 > 0$  such that

$$a_1(\|w(t)\| + \|z(t)\|)^2 \leq V(t) \leq a_2(\|w(t)\| + \|z(t)\|)^2.$$

Before we start the formal calculations of the Lyapunov function, we present the formulas derived from [Lemma 2](#)

$$\begin{aligned} \|\check{K}^u\|_\infty &\leq \bar{K}, \|\check{K}^m\|_\infty \leq \bar{K}, \\ \|\tilde{K}^u\|_\infty &\leq \epsilon, \|\tilde{K}^m\|_\infty \leq \epsilon, \\ \|w(t)\| &= \|\hat{u}(t)\|, \\ \|z(t)\| &\leq (1 + \bar{K})\|\hat{m}(t)\| + \bar{K}\|\hat{u}(t)\|, \\ \|\hat{m}(t)\| &\leq (1 + \bar{L})\|z(t)\| + \bar{L}\|w(t)\|, \\ |\Gamma(t)| &\leq \epsilon \bar{F}(\|w\| + \|z\|), \end{aligned}$$

where

$$\bar{F} = 1 + \bar{L}.$$

The key difference is  $z^2(1, t) = \Gamma(t)^2 \neq 0$ . This leads to the terms  $\frac{1}{a_1} \mu e^k \epsilon^2 \bar{F}^2 V(t)$  in [\(40\)](#) as follows. There exist positive constants  $h_1, h_2, \dots, h_6$  and nonnegative, integrable function  $g_1, g_2, \dots, g_5$  such that

$$\begin{aligned} \dot{V}_4(t) &\leq h_1 z^2(0, t) - [\lambda \delta - h_2] V_4(t) + h_3 V_5(t) \\ &\quad + g_1(t) V_4(t) + g_2(t), \\ \dot{V}_5(t) &\leq \mu e^k z^2(1, t) - [\mu - e^k h_4 \bar{r}^2(t)] z^2(0, t) + h_5 V_4 \\ &\quad - [k\mu - h_6] V_5 + g_3(t) V_4 + g_4(t) V_5 + g_5(t) \\ &\leq \frac{1}{a_1} \mu e^k \epsilon^2 \bar{F}^2 V(t) - [\mu - e^k h_4 \bar{r}^2(t)] z^2(0, t) + h_5 V_4 \\ &\quad - [k\mu - h_6] V_5 + g_3(t) V_4 + g_4(t) V_5 + g_5(t), \end{aligned} \quad (40)$$

where

$$\begin{aligned}
g_1(t) &= e^{2\delta} \rho^2 \|e_1(t)\|^2 \|\varpi(t)\|^2, \\
g_2(t) &= (\bar{c}_1^2 + 4e^{-\delta}) \|e_1(t)\|^2 + (\bar{c}_2^2 + 4e^{-\delta}) \|e_2(t)\|^2 \\
&\quad + 3\lambda e_1^2(0, t) + 3\lambda \bar{r}^2 e_2^2(0, t), \\
g_3(t) &= e^{\delta+k} \left\| \hat{K}_t^u(t) \right\|^2 + 2e^{\delta+k} \left\| \hat{K}_t^m(t) \right\|^2 A_3^2, \\
g_4(t) &= 2\rho^2 e^{\delta+2k} (A_1^2 \|e_1(t)\|^2 + A_2^2 \|e_2(t)\|^2) \|\varpi(t)\|^2 \\
&\quad + 2e^k \left\| \hat{K}_t^m(t) \right\|^2 A_4^2, \\
g_5(t) &= \lambda^2 \bar{K}^2 \bar{r}^2 e^k e_1^2(0, t) + \lambda^2 \bar{K}^2 e^k e_2^2(0, t) \\
&\quad + 2e^k (A_1^2 \bar{c}_1^2 \|e_1(t)\|^2 + A_1^2 \bar{c}_2^2 \|e_2(t)\|^2 + A_2^2 \bar{c}_3^2 \|e_1(t)\|^2 \\
&\quad + A_2^2 \bar{c}_4^2 \|e_2(t)\|^2) + 4e^{-\delta} (\|e_1(t)\|^2 + \|e_2(t)\|^2),
\end{aligned}$$

where  $A_i > 0, i = 1 \dots 4, \delta \geq 1$ . Thus, we obtain the following upper bound calculation

$$\dot{V}(t) \leq - \left[ d - \frac{1}{a_1} \mu e^k \epsilon^2 \bar{r}^2 \right] V(t) + g_6(t)V(t) + g_7(t), \quad (41)$$

for positive constant  $d$  and the nonnegative, integrable functions  $g_6(t)$  and  $g_7(t)$

$$g_6(t) = \max \{g_1(t) + ag_3(t), g_4(t)\}, \quad (42)$$

$$\begin{aligned}
g_7(t) &= \frac{2\bar{r}^2 z(0, t)^2}{1 + m(0, t)^2} + 8b\bar{r}^2 e_2(0, t)^2 + g_2(t) + ag_5(t) \\
&\quad + b \frac{\bar{r}^2 z(0, t)^2}{1 + m(0, t)^2} z(0, t)^2,
\end{aligned} \quad (43)$$

where  $a, b$  are positive constants. We introduce the positive constant

$$\epsilon_0 := \frac{\sqrt{a_1(2d-1)}}{\sqrt{2\mu e^k \bar{r}^2}}.$$

Thus, if we choose  $\epsilon \in (0, \epsilon_0)$  we have  $d - \frac{1}{a_1} \mu e^k \epsilon^2 \bar{r}^2 > 1/2 > 0$ .

It then follows from Lemma 12 in [Anfinson and Aamo \(2017\)](#) that

$$V \in L^1 \cap L^\infty,$$

and hence

$$\|w\|, \|z\| \in L^2 \cap L^\infty.$$

Due to the invertibility of the backstepping transformation

$$\|\hat{u}\|, \|\hat{m}\| \in L^2 \cap L^\infty.$$

From [Lemma 1](#),  $\|e_1\|, \|e_2\| \in L^2 \cap L^\infty$ , it follows that

$$\|u\|, \|m\| \in L^2 \cap L^\infty.$$

#### B. Pointwise-in-space boundedness and regulation

The paper ([Vazquez, Krstic, & Coron, 2011](#)) proved that the system (3) is equivalent to the following system through an invertible backstepping transformation.

$$\psi_t(x, t) = -\lambda \psi_x(x, t) + h_1(x)\zeta(0, t), \quad (44)$$

$$\zeta_t(x, t) = \mu \zeta_x(x, t), \quad (45)$$

$$\psi(0, t) = r\zeta(0, t), \quad (46)$$

$$\zeta(1, t) = U(t) - \int_0^1 (G_1(\xi)u(\xi) - G_2(\xi)m(\xi))d\xi, \quad (47)$$

for some bounded functions  $h_1, G_1, G_2$  of the unknown parameters. Eq. (44)–(47) can be explicitly be solved for  $t > \lambda^{-1} + \mu^{-1}$  to yield

$$\psi(x, t) = r\zeta(1, t - \mu^{-1} - \lambda^{-1}x)$$

$$+ \lambda^{-1} \int_0^x h_1(\tau)\zeta(1, t - \mu^{-1} - \lambda^{-1}(x - \tau))d\tau, \quad (48)$$

$$\zeta(x, t) = \zeta(1, t - \mu^{-1}(1 - x)). \quad (49)$$

From (47), the control law  $U(t)$  and  $\|u\|, \|m\|, \|\hat{u}\|, \|\hat{m}\| \in L^2 \cap L^\infty$ , it follows that  $\zeta(1, \cdot) \in L^2 \cap L^\infty$ . Since  $\zeta$  and  $\psi$  are simple, cascaded transport equations, this implies

$$\|\psi\|_\infty, \|\zeta\|_\infty \in L^2 \cap L^\infty, \quad \|\psi\|_\infty, \|\zeta\|_\infty \rightarrow 0.$$

With the invertibility of the transformation, then yields

$$\|u\|_\infty, \|m\|_\infty \in L^2 \cap L^\infty, \quad \|u\|_\infty, \|m\|_\infty \rightarrow 0.$$

From the structure of the identifier (3), we will also have  $\hat{u}(x, \cdot), \hat{m}(x, \cdot) \in L^2 \cap L^\infty$ , and hence

$$\|\hat{u}\|_\infty, \|\hat{m}\|_\infty \in L^2 \cap L^\infty, \quad \|\hat{u}\|_\infty, \|\hat{m}\|_\infty \mapsto 0.$$

#### C. Global stability

Here, we will prove the global stability of the system, specifically by proving (30), and thus we introduce the following function

$$\begin{aligned}
S(t) &:= \|u\|^2 + \|m\|^2 + \|\hat{u}\|^2 + \|\hat{m}\|^2 + \|\tilde{c}_1\|^2 + \|\tilde{c}_2\|^2 \\
&\quad + \|\tilde{c}_3\|^2 + \|\tilde{c}_4\|^2 + \bar{r}^2.
\end{aligned}$$

The goal of the proof is to demonstrate the existence of a function  $\theta$  such that the following inequality holds.

$$S(t) \leq \theta(S(0)), \quad t \geq 0.$$

We will reuse the Lyapunov function from [Appendix](#) to show that the system's state remains stable over time.

$$\begin{aligned}
V_1(t) &= V_2(t) + \gamma_1^{-1} \|\tilde{c}_1\|^2 + \gamma_2^{-1} \|\tilde{c}_2\|^2 + \gamma_3^{-1} \|\tilde{c}_3\|^2 \\
&\quad + \gamma_4^{-1} \|\tilde{c}_4\|^2 + \frac{\lambda}{2\gamma_5} \bar{r}^2(t),
\end{aligned}$$

where

$$V_2(t) = \int_0^1 e^{-\gamma x} e_1^2(x, t)dx + \int_0^1 e^{\gamma x} e_2^2(x, t)dx,$$

leads to the following upper bound:

$$\begin{aligned}
\dot{V}_1(t) &\leq -\lambda e^{-\gamma} e_1^2(1, t) - \lambda e_1^2(0, t)m^2(0, t) - \lambda \gamma e^{-\gamma} \|e_1(t)\|^2 \\
&\quad - 2\rho e^{-\gamma} \|e_1(t)\|^2 \|\varpi(t)\|^2 - \mu e_2^2(0, t) \\
&\quad - \mu \gamma \|e_2(t)\|^2 - 2\rho e^{\gamma} \|e_2(t)\|^2 \|\varpi(t)\|^2,
\end{aligned} \quad (50)$$

which shows that  $V_1(t)$  is non-increasing and hence bounded. Thus implies that the  $V_1(t) < V_1(0)$  and limit  $\lim_{t \rightarrow \infty} V_1(t) = V_{1,\infty}$  exists. By integrating (50) from zero to infinity, we obtain the following upper bound:

$$\begin{aligned}
&\lambda e^{-\gamma} \int_0^\infty e_1^2(1, \tau)d\tau + \lambda \int_0^\infty e_1^2(0, \tau)m^2(0, \tau)d\tau \\
&\quad + \lambda \gamma e^{-\gamma} \int_0^\infty \|e_1(\tau)\|^2 d\tau \\
&\quad + 2\rho e^{-\gamma} \int_0^\infty \|e_1(\tau)\|^2 \|\varpi(\tau)\|^2 d\tau + \mu \int_0^\infty e_2^2(0, \tau)d\tau \\
&\quad + \mu \gamma \int_0^\infty \|e_2(\tau)\|^2 d\tau + 2\rho e^{\gamma} \int_0^\infty \|e_2(\tau)\|^2 \|\varpi(\tau)\|^2 d\tau \\
&\leq V_1(0).
\end{aligned}$$

From (42) and (43), it can be concluded that there are constants  $\theta_1 > 0$  and  $\theta_2 > 1$  such that

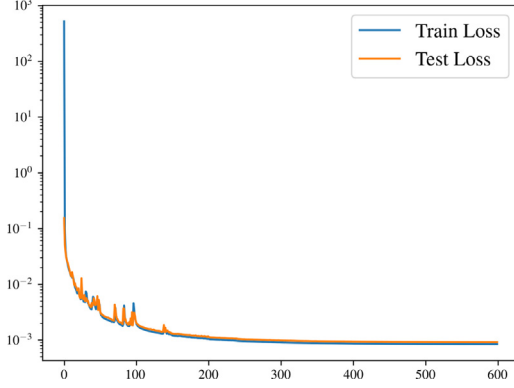
$$\|g_6\|_1 \leq \theta_1 V_1(0), \quad (51)$$

$$\|g_7\|_1 \leq \theta_2 V_1(0). \quad (52)$$



**Table 2**  
Comparison of computation time of kernels  $\check{K}^u$  and  $\check{K}^m$ .

Spatial step size	Numerical solver (s)	NO(s)	Speedup	Error
$dx=0.01$	$8.221 \times 10^{-3}$	$4.38 \times 10^{-3}$	$2\times$	0.024
$dx=0.05$	$2.432 \times 10^{-2}$	$4.42 \times 10^{-3}$	$58\times$	0.031
$dx=0.001$	$8.701 \times 10^{-1}$	$4.513 \times 10^{-3}$	$192\times$	0.037
$dx=0.0005$	3.191	$4.631 \times 10^{-3}$	$689\times$	0.045



**Fig. 3.** The train and test loss for neural operator  $\hat{\mathcal{K}}$ .

Recalling (41), we have that

$$\dot{V}(t) \leq -\frac{1}{2}V(t) + g_6(t)V(t) + g_7(t).$$

We also have from Lemma B.6 in Krstic, Kokotovic, and Kanelakopoulos (1995) that

$$V(t) \leq (e^{-\frac{1}{2}t}V(0) + \|g_7\|_1)e^{\|g_6\|_1}. \quad (53)$$

We then introduce the function

$$V_6(t) := V_1(t) + V(t).$$

Noticing that

$$V_1(t) \leq V_1(0) \leq \theta_2 V_1(0)e^{\theta_1 V_1(0)}, \quad (54)$$

we achieve from (53), (54), (51) and (52) the following

$$\begin{aligned} V_6(t) &= V(t) + V_1(t) \\ &\leq (e^{-\frac{1}{2}t}V(0) + \|g_7\|_1)e^{\|g_6\|_1} + \theta_2 V_1(0)e^{\theta_1 V_1(0)} \\ &\leq (\theta_2 V(0) + \theta_2 V_1(0))e^{\|g_6\|_1} + \theta_2 V_1(0)e^{\theta_1 V_1(0)} \\ &\leq 2\theta_2 V_6(0)e^{\theta_1 V_6(0)}. \end{aligned}$$

This Lyapunov functional can be represented by an equivalent norm, and the bounds of this equivalent norm are determined by two positive constants  $k_1 > 0$  and  $k_2 > 0$ .

$$k_1 S(t) \leq V_6(t) \leq k_2 S(t).$$

So we have

$$S(t) \leq 2 \frac{k_2}{k_1} \theta_2 S(0) e^{\theta_1 k_2 S(0)}. \quad \blacksquare$$

## 5. Simulations

This section will present and analyze the performance of the proposed NO-based adaptive controllers for two PDE models: (i) a general  $2 \times 2$  hyperbolic system (3) (ii) the ARZ PDE system. Through these examples, we will demonstrate the effectiveness of the NO-based adaptive control design.

### 5.1. Simulation of the coupled $2 \times 2$ hyperbolic system

#### A. Simulation configuration

The coefficients are defined as  $c_1(x) = \cos(\sigma_1 \cos^{-1}(x))$ ,  $c_2(x) = \cos(\sigma_2 \cos^{-1}(x))$ ,  $c_3(x) = \sin(1 - \sigma_3 x) + 1$  and  $c_4(x) = \cos(\sigma_4 x)$  with the shape parameters  $\sigma_1, \sigma_2, \sigma_3$  and  $\sigma_4$ . Although this paper uses specific Chebyshev polynomial forms, sine functions, and cosine functions, our framework is applicable to any compact set of continuous functions. We use the first order finite difference scheme to solve PDEs, where the time step  $dt = 0.005$  s, the spatial step  $dx = 0.05$  m, the total time  $T = 10$  s, and the length  $L = 1$  m. The initial conditions are  $u_0 = \sin(2\pi x)$ ,  $m_0 = x$ .

#### B. Dataset generation and NO training

We choose 10 sets of  $(c_1, c_2, c_3, c_4, r)$  randomly sampled with  $\sigma_1 \sim U(3.5, 4.5)$ ,  $\sigma_2 \sim U(0.8, 1)$ ,  $\sigma_3 \sim U(20, 21)$ ,  $\sigma_4 \sim U(10, 11)$  and  $r \sim U(2, 5)$ , where  $U(a, b)$  denotes the uniform distribution over the interval  $[a, b]$ . We simulate trajectories using adaptive control methods and calculate the corresponding kernel functions using numerical solvers. In practice, during DeepONet design, we choose a sufficiently expressive architecture to ensure accurate approximation of the kernel functions as shown in Fig. 2. Each trajectory was sampled at 1000 time points, resulting in a dataset of 10000 sets of  $(\hat{c}_1, \hat{c}_2, \hat{c}_3, \hat{c}_4, \hat{r}, \check{K}^u, \check{K}^m)$  for training. We trained the model on an Nvidia RTX 4060 Ti GPU. After 600 epochs of training, the  $L^2$  error of neural operator  $\hat{\mathcal{K}}$  reached  $1.2 \times 10^{-3}$ , and the test error was  $1.1 \times 10^{-3}$ , as shown in Fig. 3.

#### C. Computation time comparison

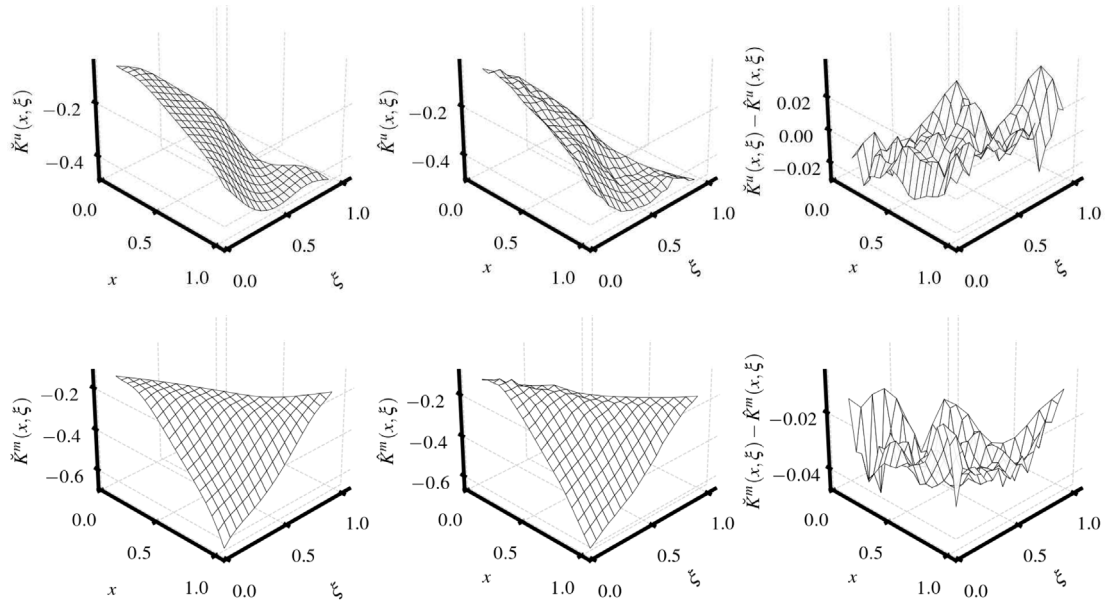
Table 2 provides a comparison of the computation time of solving kernels at each time step using the numerical solver and the trained DeepONet model. We can see that as the sampling accuracy improves, the acceleration obtained by the NO becomes substantial. We computed the average absolute error  $\int_{\xi}^1 \int_0^x (|\check{K}^u - \hat{K}^u| + |\check{K}^m - \hat{K}^m|) d\xi dx$  between numerical solutions and NO solutions with different step sizes. Although the error slightly increases with the decrease of step size, they are quite small at all step sizes. Because adaptive control requires calculating control gain at every step of updating parameter estimation, quickly solving the kernel function can help improve the performance of adaptive control.

#### D. Simulation results

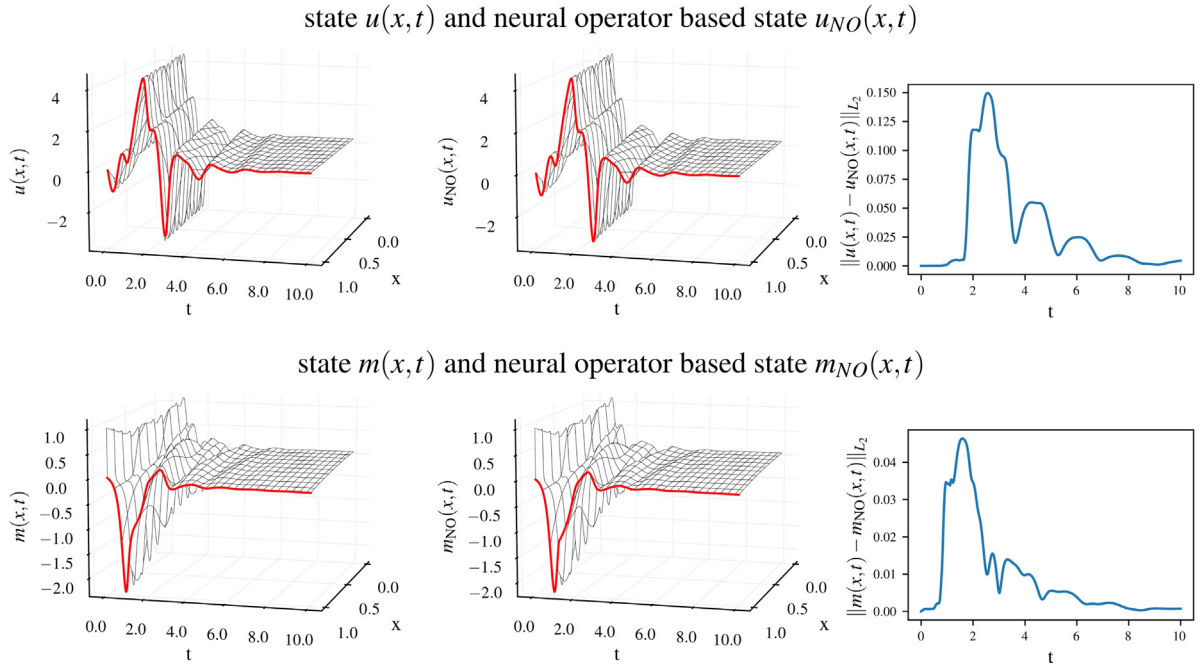
We test the performance of the closed-loop system stability with test values  $(\sigma_1 = 4, \sigma_2 = 0.9, \sigma_3 = 20.1, \sigma_4 = 10.1, r = 4)$  unseen during training. Fig. 4 shows the kernels  $\check{K}^u, \check{K}^m$  calculated by the numerical solver, the kernels  $\check{K}^u, \check{K}^m$  learned by DeepONet, and the error between them. In Fig. 5, we demonstrate closed-loop stability with the NO approximated kernel function for the control feedback law. Figs. 4 and 5 confirm that the kernels  $\hat{K}^u, \hat{K}^m$  approximated by NO can effectively simulate the backstepping kernels  $\check{K}^u, \check{K}^m$  while maintaining the stability of the system. All estimated parameters  $\hat{c}_i$  and  $\hat{r}$  are shown in Fig. 6. We emphasize that although in adaptive control the system parameters  $\hat{c}_1, \hat{c}_2, \hat{c}_3, \hat{c}_4$  and  $\hat{r}$  may not precisely converge to their true values, this does not affect the control performance. This phenomenon is not a problem but rather a characteristic of adaptive control. The goal of adaptive control is not perfect system identification, but rather the estimation of parameters that ensure system stability.

#### E. Comparative experiment with RL

We will evaluate the performance of NO-based adaptive control method and RL method for stabilization results under different initial conditions. In this work, we implement the Proximal Policy Optimization (PPO) algorithm. The PDE state is discretized and used as the observation input to a neural network policy. The output of the policy network determines the boundary control action at each time step. The PPO algorithm is trained to minimize a cumulative cost function, which achieves regulation of the



**Fig. 4.** The first row shows the exact estimated kernels  $\check{K}^u$  and  $\check{K}^m$ . The second row shows the NO estimated kernels  $\hat{K}^u$  and  $\hat{K}^m$ . The last row shows the kernel errors  $\check{K}^u - \hat{K}^u$  and  $\check{K}^m - \hat{K}^m$ . All kernels are plotted at the final time  $T$ .



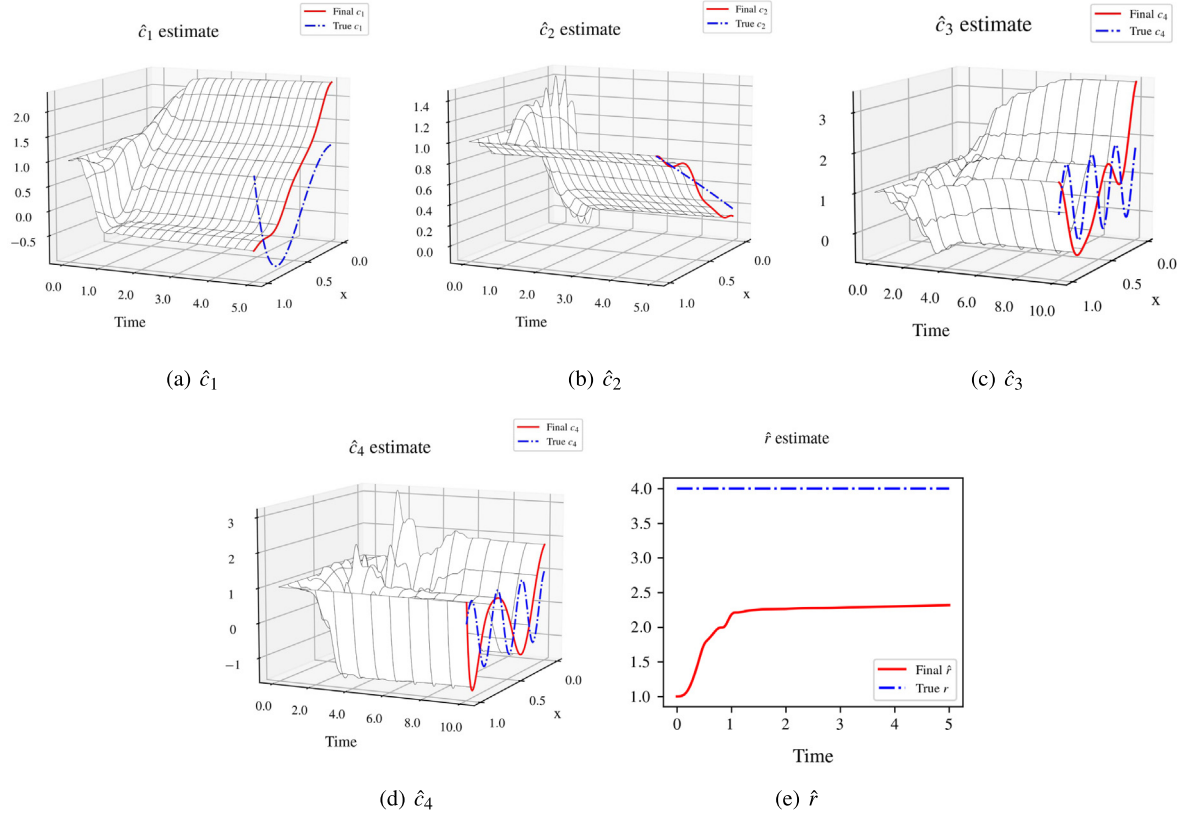
**Fig. 5.** Simulation of the close-loop system with feedback controller (23) and (29). The left columns of the first and second rows show close-loop system states  $u(x, t)$ ,  $m(x, t)$  with the analyzed kernels  $\check{K}^u$  and  $\check{K}^m$ . The middle columns of the first and second rows show close-loop system states  $u_{NO}(x, t)$ ,  $m_{NO}(x, t)$  with the approximated kernels  $\hat{K}^u$  and  $\hat{K}^m$ . The right columns of the first and second rows show the errors between  $u(x, t)$  and  $u_{NO}(x, t)$ , and between  $m(x, t)$  and  $m_{NO}(x, t)$ , respectively.

traffic states to a spatially uniform density and velocity. We use the standard clipped surrogate objective for policy updates. We choose the initial condition of state  $u$  is a sine function

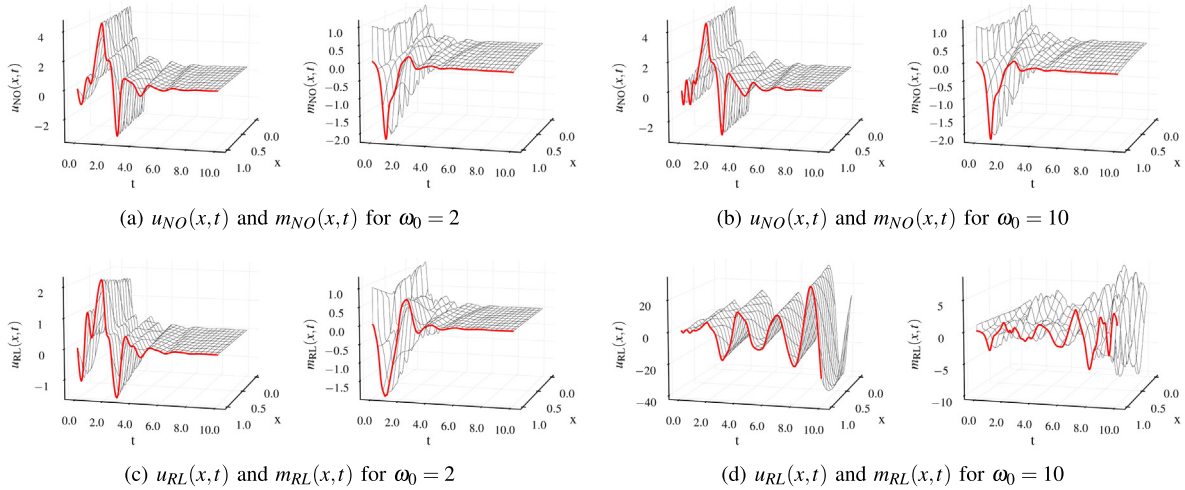
$$u_0 = \sin(\omega_0 \pi x), \quad (55)$$

where  $\omega_0$  is the frequency of a sine wave. To evaluate the performance of these two methods, we train DeepONet and RL at the same frequency  $\omega_0 = 2$ , ensuring all other parameters remained consistent with those in Fig. 4. In the testing phase, we will use sine initial conditions of different frequencies

$\omega_0 = 2, 10$  to verify the model stability of NO-based adaptive control and RL. Fig. 7 shows the stabilization results of the RL and NO control under different initial conditions. The comparative experiments highlight a significant advantage of the NO-based adaptive control method, which consistently demonstrates robustness across different initial conditions. Specifically, the NO-based adaptive control method maintains system stability without requiring retraining even when the initial conditions are changed. This characteristic underscores its adaptability in dynamic environments. In contrast, the RL method shows a significant dependency on initial conditions. Although it performs



**Fig. 6.**  $\hat{c}_1$ ,  $\hat{c}_2$ ,  $\hat{c}_3$ ,  $\hat{c}_4$  and  $\hat{r}$  estimates using NO approximated kernels. The blue dashed lines represent the true parameters, and the red solid lines represent the estimated parameters.



**Fig. 7.** Comparison of system stability between NO-based adaptive control and RL under different initial conditions (55). (a) State evolution under NO-based adaptive control when the parameter of initial condition (55) is  $\omega_0 = 2$ . (b) State evolution under NO-based adaptive control when the parameter of initial condition (55) is  $\omega_0 = 10$ . (c) State evolution under RL when the parameter of initial condition (55) is  $\omega_0 = 2$ . (d) State evolution under RL when the parameter of initial condition (55) is  $\omega_0 = 10$ .

well under specific conditions encountered during training, it is unstable when faced with unforeseen initial conditions ( $\omega_0 = 10$ ). In real-world scenarios where initial conditions are often variable and unpredictable, DeepONet ensures stability and adaptability without the need for retraining. In summary, this demonstrates DeepONet's potential for more reliable applications in adaptive control systems, where maintaining performance across diverse conditions is crucial.

## 5.2. Application simulation of the ARZ traffic system

### A. NO-based adaptive controller

Following the steps in Section 2, we can obtain the adaptive controller for ARZ traffic system (1) as follows

$$U(t) = \int_0^1 \check{K}^{u_1}(1, \xi, t) u_1(\xi, t) d\xi + \int_0^1 \check{K}^{m_1}(1, \xi, t) m_1(\xi, t) d\xi, \quad (56)$$

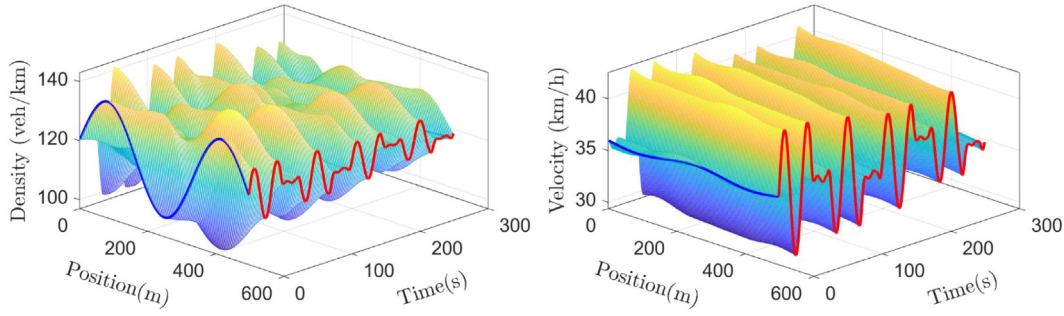


Fig. 8. Density and velocity evolution of open-loop ARZ traffic system.

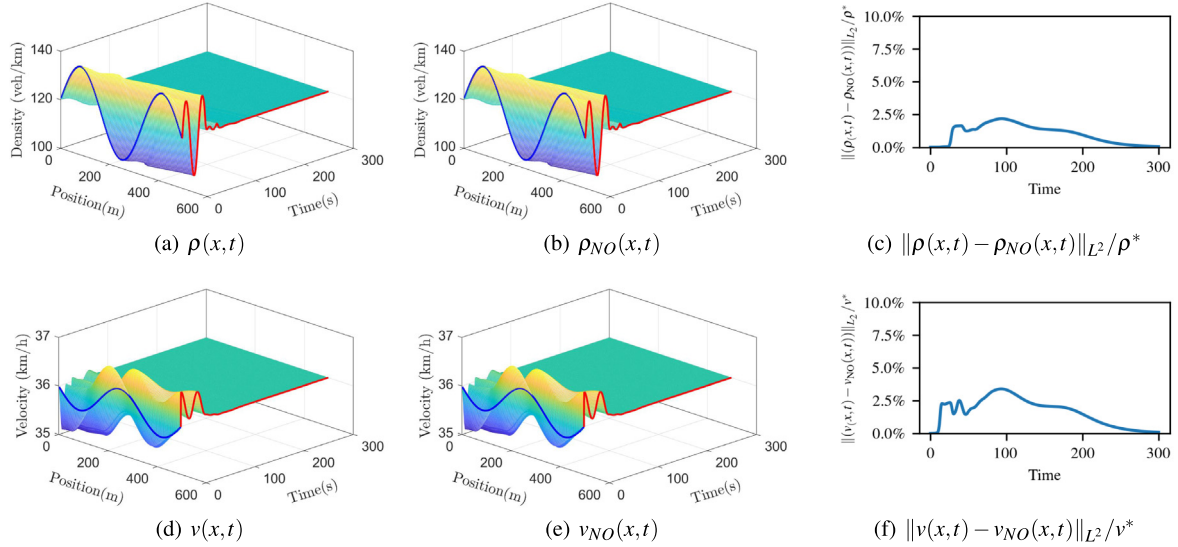


Fig. 9. Simulation of density and velocity with feedback controller (56) and (58). The left columns of the first and second rows show close-loop system states  $\rho(x, t)$ ,  $v(x, t)$ . The middle columns of the first and second rows show close-loop system states  $\rho_{NO}(x, t)$ ,  $v_{NO}(x, t)$ . The left columns of the first and second rows show the relative  $L^2$  error between  $\rho(x, t)$  and  $\rho_{NO}(x, t)$ , and between  $v(x, t)$  and  $v_{NO}(x, t)$ , respectively.

with the parameter update law

$$\hat{c}_t(x, t) = \text{Proj}_{\hat{c}} \left\{ \gamma_3 e^{\gamma x} \varepsilon_1(x, t) u_1(x, t), \hat{c}(x, t) \right\}, \quad (57)$$

where

$$\varepsilon_1(x, t) = m_1(x, t) - \hat{m}_1(x, t),$$

and the kernels satisfy the following kernel functions

$$\begin{aligned} (\gamma p^* - v^*) \check{K}_x^{u_1}(x, \xi, t) &= v^* \check{K}_\xi^{u_1}(x, \xi, t) \\ &\quad + \hat{c}(x, t) \check{K}^{m_1}(x, \xi, t), \\ (\gamma p^* - v^*) \check{K}_x^{m_1}(x, \xi, t) &= -(\gamma p^* - v^*) \check{K}_\xi^{m_1}(x, \xi, t), \\ \check{K}^{u_1}(x, x, t) &= -\frac{\hat{c}(x, t)}{\gamma p^*}, \\ \check{K}^{m_1}(x, 0, t) &= \frac{v^*}{\gamma p^* - v^*} r \check{K}^{u_1}(x, 0, t). \end{aligned}$$

According to the approximation of NO in Theorem 3, we get the NO-based adaptive controller

$$U(t) = \int_0^1 \hat{K}^{u_1}(1, \xi, t) u_1(\xi, t) d\xi + \int_0^1 \hat{K}^{m_1}(1, \xi, t) m_1(\xi, t) d\xi. \quad (58)$$

### B. Simulation results

Then, we analyze the performance of the proposed NO-based adaptive control law for the ARZ traffic PDE system through

simulations on a  $L=600$  m road over  $T=300$ s. The parameters are set as follows: free-flow velocity  $v_m = 40$  m/s, maximum density  $\rho_m = 160$  veh/km, equilibrium density  $\rho^* = 120$  veh/km, driver reaction time  $\tau = 60$  s. Let  $\gamma = 1$ . Initial conditions are sinusoidal inputs  $\rho(x, 0) = \rho^* + 0.1 \sin(\frac{3\pi x}{L}) \rho^*$  and  $v(x, 0) = v^* - 0.01 \sin(\frac{3\pi x}{L}) v^*$  to mimic stop-and-go traffic. Recent advances in traffic sensing technologies, such as connected vehicles (CVs), loop detectors, and roadside sensors, provide increasingly dense and accurate measurements. To generate a sufficient dataset for training, we use 10 different  $c(x)$  functions with  $\tau \in U[50, 70]$  and simulate the resulting PDEs under the adaptive controller for  $T = 300$  seconds. We sub-sample each  $(c, \hat{K}^{u_1}, \hat{K}^{m_1})$  pair every 0.1 s, resulting in a total of 30,000 distinct  $(c, \hat{K}^{u_1}, \hat{K}^{m_1})$  pairs for training the NO. Using the trained NO, we simulate with the same parameters. Fig. 8 shows the ARZ system is open-loop unstable. Figs. 9 show the density and velocity of ARZ traffic system. The blue line indicates the initial condition, whereas the red line represents the boundary condition of the system. The results indicate that both the NO-based adaptive method and the adaptive backstepping control method effectively stabilize the transportation system. The traffic density and velocity converge to the equilibrium values of  $\rho^* = 120$  veh/km and  $v^* = 36$  m/s, respectively. As shown in Fig. 10, the boundary control input constructed using the DeepONet-based kernels achieves stabilization performance comparable to that of the exact controller, indicating that the learned kernels are sufficiently accurate for practical



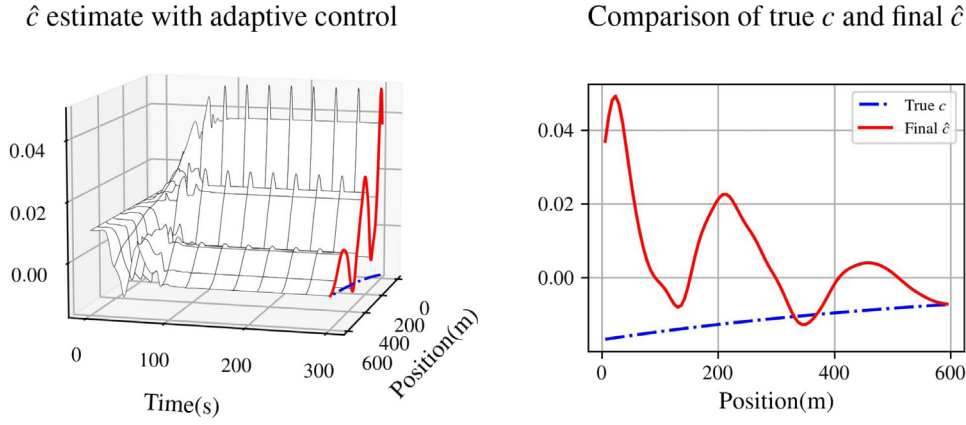


Fig. 10. Parameter estimation of  $\hat{c}$  in ARZ traffic system and comparison of true  $c$  and  $\hat{c}$ .

Table 3

Computation time and mean square errors (MSE) of density and velocity for the nominal and NO-based adaptive controllers in traffic control.

Method	Average Computation time (s)	MSE %	
		Density $\rho$	Velocity $v$
Nominal Adaptive Controller	1.51	0	0
NO-based Adaptive Controller	0.043	0.021	0.011

traffic control applications and effectively alleviate traffic congestion. The maximum error does not exceed 10%. The estimated parameter  $\hat{c}$  is shown in Fig. 10.

Table 3 presents the computation times for both the nominal adaptive controller and the NO-based adaptive controller. As the baseline result, the nominal adaptive control method is compared with the NO-based adaptive control method. Notably, the NO-based adaptive control method not only achieves significantly faster average computation times but also maintains superior accuracy with lower mean square errors. These advantages of the NO-based adaptive control method not only enhance computational efficiency but also make it highly suitable for real-time traffic system applications. The NO method's efficiency and accuracy represent a substantial advancement, promising more effective and scalable traffic control strategies in practical scenarios.

## 6. Conclusion

We develop a NO-based adaptive boundary control design for a  $2 \times 2$  linear first-order hyperbolic system. Compared with the previous studies (Lamarque et al., 2025), (Bhan et al., 2025) that primarily focused on approximating a single kernel PDE, this work accelerates the computation of  $2 \times 2$  coupled Goursat-form PDEs. In this paper, the DeepONet is used to learn the adaptive control gains for stabilizing the traffic PDE system, and it is shown that under the DeepONet-approximated kernels the stabilization of  $2 \times 2$  hyperbolic PDEs can still be achieved with significant improvement for computational speeds. Experimental results show that compared to traditional numerical solvers, our method improves computational efficiency by two orders of magnitude. Additionally, compared with RL, the NO-based adaptive control strategy is independent of the system's initial conditions, making it more robust for rapidly changing traffic scenarios. Our method significantly accelerates the process of obtaining adaptive controllers in PDE systems, greatly improving the real-time applicability of adaptive control strategies for mitigating traffic congestion. In the future, we will incorporate real traffic data into the training of the neural operator.

## Appendix. Proof of Lemma 1

The proof of this lemma follows a similar approach to the proof of Lemma 9.1 in Anfinssen and Aamo (2019). Property (13) follows trivially from projection in (11) and Lemma A.1 in Anfinssen and Aamo (2019). The result can be easily obtained using the following Lyapunov function candidate:

$$V_1(t) = V_2(t) + \gamma_1^{-1} \|\tilde{c}_1\|^2 + \gamma_2^{-1} \|\tilde{c}_2\|^2 + \gamma_3^{-1} \|\tilde{c}_3\|^2 + \gamma_4^{-1} \|\tilde{c}_4\|^2 + \frac{\lambda}{2\gamma_5} \tilde{r}^2(t), \quad (\text{A.1})$$

where

$$V_2(t) = \int_0^1 e^{-\gamma x} e_1^2(x, t) dx + \int_0^1 e^{\gamma x} e_2^2(x, t) dx.$$

Computing the time derivative of (A.1) along (7)–(10) as

$$\begin{aligned} \dot{V}_1(t) = & 2 \int_0^1 \left( e^{-\gamma x} e_1(x, t) e_{1t}(x, t) + e^{\gamma x} e_2(x, t) e_{2t}(x, t) \right) dx \\ & + 2\gamma_1^{-1} \int_0^1 \tilde{c}_1(x, t) \tilde{c}_{1t}(x, t) dx \\ & + 2\gamma_2^{-1} \int_0^1 \tilde{c}_2(x, t) \tilde{c}_{2t}(x, t) dx \\ & + 2\gamma_3^{-1} \int_0^1 \tilde{c}_3(x, t) \tilde{c}_{3t}(x, t) dx \\ & + 2\gamma_4^{-1} \int_0^1 \tilde{c}_4(x, t) \tilde{c}_{4t}(x, t) dx + \lambda \gamma_5^{-1} \tilde{r}(t) \dot{\tilde{r}}(t). \end{aligned}$$

Substituting into the dynamics (7)–(10) and integrating by parts, we obtain

$$\begin{aligned} \dot{V}_1(t) = & -\lambda e^{-\gamma} e_1^2(1, t) + \lambda e_1^2(0, t) - \lambda \gamma \int_0^1 e^{-\gamma x} e_1^2(x, t) dx \\ & + 2 \int_0^1 e^{-\gamma x} e_1(\tilde{c}_1 u + \tilde{c}_2 m) dx \\ & - 2\rho \int_0^1 e^{-\gamma x} e_1^2(x, t) \|\varpi(t)\|^2 dx \\ & - \mu e_2^2(0, t) - \mu \gamma \int_0^1 e^{\gamma x} e_2^2(x, t) dx \\ & + 2 \int_0^1 e^{\gamma x} e_2(\tilde{c}_3 u + \tilde{c}_4 m) dx \\ & - 2\rho \int_0^1 e^{\gamma x} e_2^2(x, t) \|\varpi(t)\|^2 dx \end{aligned}$$

$$\begin{aligned}
& + 2\gamma_1^{-1} \int_0^1 \tilde{c}_1(x, t) \tilde{c}_{1t}(x, t) dx \\
& + 2\gamma_2^{-1} \int_0^1 \tilde{c}_2(x, t) \tilde{c}_{2t}(x, t) dx \\
& + 2\gamma_3^{-1} \int_0^1 \tilde{c}_3(x, t) \tilde{c}_{3t}(x, t) dx \\
& + 2\gamma_4^{-1} \int_0^1 \tilde{c}_4(x, t) \tilde{c}_{4t}(x, t) dx + \lambda \gamma_5^{-1} \tilde{r}(t) \dot{\tilde{r}}(t).
\end{aligned}$$

Inserting the adaptive laws (11). By using the property (12), we have

$$\begin{aligned}
\tilde{c}_1(x, t) \tilde{c}_{1t}(x, t) &= -\tilde{c}_1(x, t) \hat{c}_{1t}(x, t) \\
&= -\tilde{c}_1(x, t) \text{Proj}_{\tilde{c}_1}(\gamma_1 e^{-\gamma x} e_1 u, \hat{c}_1(x, t)) \\
&\leq -\tilde{c}_1(x, t) \gamma_1 e^{-\gamma x} e_1 u.
\end{aligned} \tag{A.2}$$

Similarly for  $\tilde{c}_2$ ,  $\tilde{c}_3$ ,  $\tilde{c}_4$ , and  $\tilde{r}$ . Then we have

$$\begin{aligned}
\dot{V}_1(t) &\leq -\lambda e^{-\gamma} e_1^2(1, t) + \lambda e_1^2(0, t) - \lambda \gamma \int_0^1 e^{-\gamma x} e_1^2(x, t) dx \\
&\quad - 2\rho \int_0^1 e^{-\gamma x} e_1^2(x, t) \|\varpi(t)\|^2 dx \\
&\quad - \mu e_2^2(0, t) - \mu \gamma \int_0^1 e^{\gamma x} e_2^2(x, t) dx \\
&\quad - 2\rho \int_0^1 e^{\gamma x} e_2^2(x, t) \|\varpi(t)\|^2 dx - \lambda \tilde{r}(t) e_1(0, t) m(0, t).
\end{aligned} \tag{A.3}$$

From the boundary condition (9), we have

$$e_1(0, t) - \tilde{r}(t) m(0, t) = -e_1(0, t) m^2(0, t). \tag{A.4}$$

For the second and last terms in (A.3), substituting (A.4), we have

$$\begin{aligned}
&\lambda e_1^2(0, t) - \lambda \tilde{r}(t) e_1(0, t) m(0, t) \\
&= \lambda e_1(0, t) (e_1(0, t) - \tilde{r}(t) m(0, t)) \\
&= -\lambda e_1^2(0, t) m^2(0, t).
\end{aligned} \tag{A.5}$$

By substituting (A.5), we obtain

$$\begin{aligned}
\dot{V}_1(t) &\leq -\lambda e^{-\gamma} e_1^2(1, t) - \lambda e_1^2(0, t) m^2(0, t) - \lambda \gamma e^{-\gamma} \|e_1(t)\|^2 \\
&\quad - 2\rho e^{-\gamma} \|e_1(t)\|^2 \|\varpi(t)\|^2 - \mu e_2^2(0, t) \\
&\quad - \mu \gamma \|e_2(t)\|^2 - 2\rho e^{\gamma} \|e_2(t)\|^2 \|\varpi(t)\|^2.
\end{aligned} \tag{A.6}$$

From (A.6), we obtain that  $V_1$  is bounded. By the definitions of  $V_1$  and  $V_2$ , it follows that  $\|e_1\|, \|e_2\| \in L^\infty$ . When (A.6) is integrated over time from zero to infinity, we conclude the results that  $\|e_1\|, \|e_2\| \in L^2$ , (15), and  $e_1(1, \cdot), e_2(0, \cdot), |e_1(0, \cdot) m(0, \cdot)| \in L^2$ . From above results and the adaptive laws (11), we derive that (16). we choose the Lyapunov function candidate  $V_3(t) = \frac{1}{2} \gamma_5^{-1} \tilde{r}^2(t)$ , and use the property (12), we find

$$\dot{V}_3(t) \leq -\tilde{r}(t) e_1(0, t) m(0, t) \leq -\frac{\tilde{r}^2(t) m^2(0, t)}{1 + m^2(0, t)}. \tag{A.7}$$

This implies that  $V_3$  is upper-bounded, and hence we have  $V_3 \in L^\infty$ . By integrating (A.7) from zero to infinity, we obtain (18). Using (9) and (A.4), we derive that

$$\begin{aligned}
e_1^2(0, t) &= e_1(0, t) (\tilde{r}(t) m(0, t) - e_1(0, t) m^2(0, t)) \\
&= \frac{\tilde{r}^2(t) m^2(0, t)}{1 + m^2(0, t)} - e_1^2(0, t) m^2(0, t),
\end{aligned} \tag{A.8}$$

and from  $|e_1(0, \cdot) m(0, \cdot)| \in L^2$  and (18), we have  $e_1(0, \cdot) \in L^2$ .

## References

- Anfinsen, H., & Aamo, O. M. (2017). Model reference adaptive control of  $n + 1$  coupled linear hyperbolic PDEs. *Systems & Control Letters*, 109, 1–11.
- Anfinsen, H., & Aamo, O. M. (2018). Adaptive control of linear  $2 \times 2$  hyperbolic systems. *Automatica*, 87, 69–82.
- Anfinsen, H., & Aamo, O. M. (2019). *Adaptive control of hyperbolic PDEs*. Springer.
- Auriol, J. (2020). Output feedback stabilization of an underactuated cascade network of interconnected linear PDE systems using a backstepping approach. *Automatica*, 117, Article 108964.
- Aw, A., & Rascle, M. (2000). Resurrection of “second order” models of traffic flow. *SIAM Journal on Applied Mathematics*, 60(3), 916–938.
- Bekiaris-Liberis, N., & Delis, A. (2019). Feedback control of freeway traffic flow via time-gap manipulation of acc-equipped vehicles: A pde-based approach. *IFAC-PapersOnLine*, 52(6), 1–6.
- Belhadjoudja, M. C., Maghenem, M., Witrant, E., & Prieur, C. (2023). Adaptive stabilization of the Kuramoto-Sivashinsky equation subject to intermittent sensing. In *2023 American control conference* (pp. 1608–1613). IEEE.
- Belletti, F., Huo, M., Litrico, X., & Bayen, A. M. (2015). Prediction of traffic convective instability with spectral analysis of the Aw–Rascle–Zhang model. *Physics Letters A*, 379(38), 2319–2330.
- Bhan, L., Shi, Y., & Krstic, M. (2023). Neural operators for bypassing gain and control computations in PDE backstepping. *IEEE Transactions on Automatic Control*, 69(8), 5310–5325.
- Bhan, L., Shi, Y., & Krstic, M. (2025). Adaptive control of reaction–diffusion PDEs via neural operator-approximated gain kernels. *Systems & Control Letters*, 195, Article 105968.
- Böhm, M., Demetriou, M., Reich, S., & Rosen, I. (1998). Model reference adaptive control of distributed parameter systems. *SIAM Journal on Control and Optimization*, 36(1), 33–81.
- Burkhardt, M., Yu, H., & Krstic, M. (2021). Stop-and-go suppression in two-class congested traffic. *Automatica*, 125, Article 109381.
- Coron, J.-M., Vazquez, R., Krstic, M., & Bastin, G. (2013). Local exponential  $h^2$  stabilization of a  $2 \times 2$  quasilinear hyperbolic system using backstepping. *SIAM Journal on Control and Optimization*, 51(3), 2005–2035.
- De Palma, A., & Lindsey, R. (2011). Traffic congestion pricing methodologies and technologies. *Transportation Research Part C: Emerging Technologies*, 19(6), 1377–1399.
- Deng, B., Shin, Y., Lu, L., Zhang, Z., & Karniadakis, G. E. (2022). Approximation rates of DeepONets for learning operators arising from advection–diffusion equations. *Neural Networks*, 153, 411–426.
- Di Meglio, F., Breschi-Pietri, D., & Aarsnes, U. J. F. (2014). An adaptive observer for hyperbolic systems with application to underbalanced drilling. *IFAC Proceedings Volumes*, 47(3), 11391–11397.
- Hong, K. S., & Bentsman, J. (1994). Direct adaptive control of parabolic systems: Algorithm synthesis and convergence and stability analysis. *IEEE Transactions on Automatic Control*, 39(10), 2018–2033.
- Hu, L., Di Meglio, F., Vazquez, R., & Krstic, M. (2015). Control of homodirectional and general heterodirectional linear coupled hyperbolic PDEs. *IEEE Transactions on Automatic Control*, 61(11), 3301–3314.
- Karafyllis, I., Bekiaris-Liberis, N., & Papageorgiou, M. (2018). Feedback control of nonlinear hyperbolic PDE systems inspired by traffic flow models. *IEEE Transactions on Automatic Control*, 64(9), 3647–3662.
- Kawan, C., Mironchenko, A., & Zamani, M. (2022). A Lyapunov-based ISS small-gain theorem for infinite networks of nonlinear systems. *IEEE Transactions on Automatic Control*, 68(3), 1447–1462.
- Krstic, M. (2009). *Delay compensation for nonlinear, adaptive, and PDE systems*. Springer.
- Krstic, M., Bhan, L., & Shi, Y. (2024). Neural operators of backstepping controller and observer gain functions for reaction–diffusion PDEs. *Automatica*, 164, Article 111649.
- Krstic, M., Kokotovic, P. V., & Kanellakopoulos, I. (1995). *Nonlinear and adaptive control design*. John Wiley & Sons, Inc.
- Krstic, M., & Smyshlyaev, A. (2008a). Adaptive boundary control for unstable parabolic PDEs—Part I: Lyapunov design. *IEEE Transactions on Automatic Control*, 53(7), 1575–1591.
- Krstic, M., & Smyshlyaev, A. (2008b). Backstepping boundary control for first-order hyperbolic PDEs and application to systems with actuator and sensor delays. *Systems & Control Letters*, 57(9), 750–758.
- Lamarque, M., Bhan, L., Shi, Y., & Krstic, M. (2025). Adaptive neural-operator backstepping control of a benchmark hyperbolic PDE. *Automatica*, 177, Article 112329.
- Lamarque, M., Bhan, L., Vazquez, R., & Krstic, M. (2024). Gain scheduling with a neural operator for a transport PDE with nonlinear recirculation. arXiv preprint arXiv:2401.02511.
- Lighthill, M. J., & Whitham, G. B. (1955). On kinematic waves II. A theory of traffic flow on long crowded roads. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 229(1178), 317–345.
- Lin, X., Vazquez, R., & Krstic, M. (2024). Towards a MATLAB toolbox to compute backstepping kernels using the power series method. arXiv preprint arXiv:2403.16070.

- Logemann, H., & Townley, S. (1997). Adaptive stabilization without identification for distributed parameter systems: An overview. *IMA Journal of Mathematical Control and Information*, 14, 175–206.
- Lu, L., Jin, P., Pang, G., Zhang, Z., & Karniadakis, G. E. (2021). Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3), 218–229.
- Lv, K., Wang, J., & Cao, Y. (2024). Neural operator approximations for boundary stabilization of cascaded parabolic PDEs. *International Journal of Adaptive Control and Signal Processing*.
- Mowlavi, S., & Nabi, S. (2023). Optimal control of PDEs using physics-informed neural networks. *Journal of Computational Physics*, 473, Article 111731.
- Qi, J., Zhang, J., & Krstic, M. (2024). Neural operators for PDE backstepping control of first-order hyperbolic PIDE with recycle and delay. *Systems & Control Letters*, 185, Article 105714.
- Qu, X., Yu, Y., Zhou, M., Lin, C.-T., & Wang, X. (2020). Jointly dampening traffic oscillations and improving energy consumption with electric, connected and automated vehicles: A reinforcement learning based approach. *Applied Energy*, 257, Article 114030.
- Richards, P. I. (1956). Shock waves on the highway. *Operations Research*, 4(1), 42–51.
- Shi, Y., Li, Z., Yu, H., Steeves, D., Anandkumar, A., & Krstic, M. (2022). Machine learning accelerated pde backstepping observers. In *2022 IEEE 61st conference on decision and control* (pp. 5423–5428). IEEE.
- Smyshlyaev, A., Cerpa, E., & Krstic, M. (2010). Boundary stabilization of a 1-D wave equation with in-domain antidamping. *SIAM Journal on Control and Optimization*, 48(6), 4014–4031.
- Smyshlyaev, A., & Krstic, M. (2007). Adaptive boundary control for unstable parabolic PDEs—Part II: Estimation-based designs. *Automatica*, 43(9), 1543–1556.
- Vazquez, R., Chen, G., Qiao, J., & Krstic, M. (2023). The power series method to compute backstepping kernel gains: theory and practice. In *2023 62nd IEEE conference on decision and control* (pp. 8162–8169). IEEE.
- Vazquez, R., & Krstic, M. (2024). Gain-only neural operators for PDE backstepping. *arXiv preprint arXiv:2403.19344*.
- Vazquez, R., Krstic, M., & Coron, J.-M. (2011). Backstepping boundary stabilization and state estimation of a  $2 \times 2$  linear hyperbolic system. In *2011 50th IEEE conference on decision and control and European control conference* (pp. 4937–4942). IEEE.
- Wang, S., Diagne, M., & Krstic, M. (2025a). Backstepping neural operators for  $2 \times 2$  hyperbolic PDEs. *Automatica*, 178, Article 112351.
- Wang, S., Diagne, M., & Krstic, M. (2025b). Deep learning of delay-compensated backstepping for reaction-diffusion PDEs. *IEEE Transactions on Automatic Control*.
- Wang, J., & Krstic, M. (2020). Event-triggered backstepping control of  $2 \times 2$  hyperbolic PDE-ODE systems. *IFAC-PapersOnLine*, 53(2), 7551–7556.
- Wu, C., Kreidieh, A., Vinitsky, E., & Bayen, A. M. (2017). Emergent behaviors in mixed-autonomy traffic. In *Conference on robot learning* (pp. 398–407). PMLR.
- Yu, H., Auriol, J., & Krstic, M. (2022). Simultaneous downstream and upstream output-feedback stabilization of cascaded freeway traffic. *Automatica*, 136, Article 110044.
- Yu, H., & Krstic, M. (2018a). Adaptive output feedback for aw-rascl-zhang traffic model in congested regime. In *2018 annual American control conference* (pp. 3281–3286). IEEE.
- Yu, H., & Krstic, M. (2018b). Varying speed limit control of Aw-Rascl-Zhang traffic model. In *2018 21st international conference on intelligent transportation systems* (pp. 1846–1851). IEEE.
- Yu, H., & Krstic, M. (2019). Traffic congestion control for Aw-Rascl-Zhang model. *Automatica*, 100, 38–51.
- Yu, H., & Krstic, M. (2022). *Traffic congestion control by PDE backstepping*. Springer.
- Yu, H., Park, S., Bayen, A., Moura, S., & Krstic, M. (2021). Reinforcement learning versus PDE backstepping and PI control for congested freeway traffic. *IEEE Transactions on Control Systems Technology*, 30(4), 1595–1611.
- Zhang, H. M. (2002). A non-equilibrium traffic model devoid of gas-like behavior. *Transportation Research Part B: Methodological*, 36(3), 275–290.
- Zhang, L., Prieur, C., & Qiao, J. (2019). PI boundary control of linear hyperbolic balance laws with stabilization of ARZ traffic flow models. *Systems & Control Letters*, 123, 85–91.

Zhang, Y., Yu, H., Auriol, J., & Pereira, M. (2023). Mean-square exponential stabilization of mixed-autonomy traffic PDE system. *arXiv preprint arXiv:2310.15547*.

Zhang, Y., Zhong, R., & Yu, H. (2024). Neural operators for boundary stabilization of stop-and-go traffic. In *6th annual learning for dynamics & control conference* (pp. 554–565). PMLR.

Zhao, C., & Yu, H. (2023). Observer-informed deep learning for traffic state estimation with boundary sensing. *IEEE Transactions on Intelligent Transportation Systems*.



**Kaijing Lv** received the B.S. degree in Mathematics from Hebei University of Technology, Tianjin, China, in 2021. She is currently a Ph.D. student in the School of Mathematics and Statistics at Beijing Institute of Technology. Her research interests include the control theory of distributed parameter systems and intelligent control.



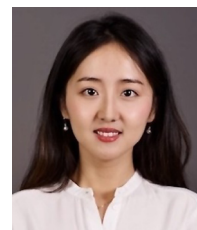
**Jun-Min Wang** received the B.S. degree in Mathematics from Shanxi University in 1995, and M.S. degree in Applied Mathematics from Beijing Institute of Technology in 1998, respectively, and Ph.D. degree in Applied Mathematics from the University of Hong Kong in 2004.

From 1998 to 2000, he worked as a Teaching Assistant at the Beijing Institute of Technology. He was a Postdoctoral Fellow at the University of the Witwatersrand, South Africa from 2004 to 2006. From 2006 to 2009 he was an Associate Professor at the Beijing Institute of Technology. Since 2009, he has been with Beijing Institute of Technology as a Full Professor in Applied Mathematics and Control Theory. His research interests focus on the control theory of distributed parameter systems.

Dr. Wang is in the editorial board of Control Theory and Technology. He is a Senior Member of IEEE, a member of CAA and CSIAM, and the recipient for New Century Excellent Talents Program from Ministry of Higher Education of China.



**Yihui Zhang** received his B.E. degree in Vehicle Engineering from Southwest University in 2019, and his M.S. degree in Vehicle Engineering from South China University of Technology in 2022. He is currently a Ph.D. student in the Thrust of Intelligent Transportation at the Hong Kong University of Science and Technology (Guangzhou). His doctoral research focuses on distributed parameter systems, learning and control for dynamical systems, and their applications in transportation systems.



**Dr. Huan Yu** is an Assistant Professor at the Hong Kong University of Science and Technology (Guangzhou), jointly appointed with the Intelligent Transportation Thrust and the Robotics and Autonomous Systems Thrust. She received her Ph.D. in mechanical and aerospace engineering from the University of California, San Diego. She was a visiting scholar at University of California, Berkeley and Massachusetts Institute of Technology. Dr. Yu's research integrates control theory, transportation science, and machine learning to advance urban mobility—from connected autonomous ground vehicles to emerging low-altitude air transportation systems.