Transportation Research Part C 170 (2025) 104928

Contents lists available at ScienceDirect

Transportation Research Part C

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

Mitigating stop-and-go traffic congestion with operator learning

Yihuai Zhang [a], Ruiguo Zhong [a], Huan Yu [a,b,*]

[a] Thrust of Intelligent Transportation, The Hong Kong University of Science and Technology (Guangzhou), Nansha, Guangzhou, 511458, China
[b] Department of Civil and Environmental Engineering, The Hong Kong University of Science and Technology, Hong Kong Special Administrative Region of China

ARTICLE INFO

Keywords:
Freeway traffic control
Partial differential equations (PDEs)
Neural operators
Backstepping control

ABSTRACT

This paper presents a novel neural operator learning framework for designing boundary control to mitigate stop-and-go congestion on freeways. The freeway traffic dynamics are described by second-order coupled hyperbolic partial differential equations (PDEs), i.e. the Aw–Rascle–Zhang (ARZ) macroscopic traffic flow model. The proposed framework learns feedback boundary control strategies from the closed-loop PDE solution using backstepping controllers, which are widely employed for boundary stabilization of PDE systems. The PDE backstepping control design is time-consuming and requires intensive depth of expertise, since it involves constructing and solving backstepping control kernels. Existing machine learning methods for solving PDE control problems, such as physics-informed neural networks (PINNs) and reinforcement learning (RL), face the challenge of retraining when PDE system parameters and initial conditions change. To address these challenges, we present neural operator (NO) learning schemes for the ARZ traffic system that not only ensure closed-loop stability robust to parameter and initial condition variations but also accelerate boundary controller computation. The first scheme embeds NO-approximated control gain kernels within a analytical state feedback backstepping controller, while the second one directly learns a boundary control law from functional mapping between model parameters to closed-loop PDE solution. The stability guarantee of the NO-approximated control laws is obtained using Lyapunov analysis. We further propose the physics-informed neural operator (PINO) to reduce the reliance on extensive training data. The performance of the NO schemes is evaluated by simulated and real traffic data, compared with the benchmark backstepping controller, a Proportional Integral (PI) controller, and a PINN-based controller. The NO-approximated methods achieve a computational speedup of approximately 300 times with only a 1% error trade-off compared to the backstepping controller, while outperforming the other two controllers in both accuracy and computational efficiency. The robustness of the NO schemes is validated using real traffic data, and tested across various initial traffic conditions and demand scenarios. The results show that neural operators can significantly expedite and simplify the process of obtaining controllers for traffic PDE systems with great potential application for traffic management.

1. Introduction

Stop-and-go traffic oscillations are a common phenomenon on freeways, causing increased travel time, fuel consumption, and traffic accidents (Belletti et al., 2015; De Palma and Lindsey, 2011; Schönhof and Helbing, 2007; Siri et al., 2021; Flynn et al., 2009).

* Corresponding author at: Thrust of Intelligent Transportation, The Hong Kong University of Science and Technology (Guangzhou), Nansha, Guangzhou, 511458, China.
E-mail address: huanyu@ust.hk (H. Yu).

https://doi.org/10.1016/j.trc.2024.104928
Received 1 May 2024; Received in revised form 23 September 2024; Accepted 6 November 2024
Available online 22 November 2024

Freeway traffic control is focused on designing control strategies to mitigate stop-and-go traffic congestion, mainly implemented by road-based traffic management systems such as ramp metering or varying speed limits. The ramp metering controls the ramp inflow to the mainline, while varying speed limits regulate the speed of the mainline vehicle (Hoogendoorn et al., 2016; Horowitz et al., 2005; Papamichail et al., 2010). In recent decades, many studies have focused on vehicle-based control using connected automated vehicles (Lee et al., 2024; Zhao et al., 2023; Zheng et al., 2020; Avedisov et al., 2020). Vehicle cruising speed control algorithms are developed to optimize car-following behaviors of a platoon of vehicles. Compared with emerging vehicle-based control methods that are developed based on individual vehicles, road-based traffic management and control utilize aggregated values such as traffic speed and flow to regulate the whole traffic on the road. This paper will mainly focus on designing control strategies to suppress spatial–temporal traffic oscillations via ramp metering. We focus on using second-order macroscopic Partial Differential Equations (PDEs) models, due to its simplicity and analytic capability in modeling freeway traffic. An operator learning framework based on neural operators will be developed to achieve boundary stabilization of stop-and-go traffic on the freeway.

1.1. Freeway traffic control

Vehicular traffic dynamics on the highway are often described using macroscopic traffic models at an aggregated level. PDEs that are continuous in time and space are used to describe the spatial and temporal evolution of the macroscopic traffic variables, that is, density, speed, and flow rate. Macroscopic traffic models are categorized by the first-order density PDE model and second-order density and speed PDE model. The Lighthill and Whitham and Richard (LWR) model (Lighthill and Whitham, 1955; Richards, 1956; Whitham, 2011) is widely used for modeling the density evolution on the road section, but cannot capture stop-and-go oscillations on the freeway. The second-order Aw–Rascle–Zhang (ARZ) model (Aw and Rascle, 2000; Zhang, 2002) allows the traffic speed has its own dynamical acceleration equation describing speed evolution as a function of density, local speed and their gradients, and therefore is adopted to describe the stop-and-go oscillations.

Control designs for freeway traffic stabilization using macroscopic PDE models can be classified into two categories: "discretize then design" and "design then discretize", based on whether numerical discretization of the PDE model is applied before or after the control design. The choice between the two approaches depends on various factors, including computational efficiency, accuracy requirements, and ease of implementation.

Discretize then design refers to the application of numerical discretization of traffic PDE model in time or space first and then designing control algorithms for the discretized models. Discretized traffic PDE models in time are Ordinary Differential Equations (ODEs) and the ones both in space and time are difference equations, which can reduce the difficulty of next-step control designs, facilitate modular designs, and increase scalability for network problems. The first-order Cell Transmission Model (CTM) is the discretized LWR model (Daganzo, 1994), in which the road section is divided into many "cells" and the propagation of traffic density in cells depends on the inflow and outflow of each cell, i.e., supply and demand. The second-order discrete METANET model is derived by discretizing and extending the Payne–Whitham model (Kotsialos et al., 2002; Wang et al., 2022b). Scaling up the CTM to the network level, the link-node cell transmission model was proposed by Muralidharan et al. (2009) and the supply and demand relations still hold in the complex network road geometry.

Based on the discretized models, various control strategies have been proposed for freeway traffic control. The classical feedback control ALINEA and PI-ALINEA have been proposed to resolve downstream bottlenecks using local ramp metering (Papageorgiou et al., 1991; Wang et al., 2014). Additionally, Müller et al. (2015) designed an integral controller for variable speed limits to prevent congestion formation at active bottlenecks. Carlson et al. (2011) designed feedback mainstream traffic flow control on motorways using METANET that achieved the same performance of the optimal control approach. To solve the on-ramp metering control problem, the asymmetric cell transmission model (ACTM) was proposed to reduce the total time spent on a given road section (Gomes and Horowitz, 2006) and then extended to network traffic (Muralidharan and Horowitz, 2012). Besides, model predictive control (Liu et al., 2016; Bellemans et al., 2006; Muralidharan and Horowitz, 2015), distributed control (Čičić et al., 2021; Reilly and Bayen, 2015), event-triggered control (Ferrara et al., 2015, 2016), reinforcement learning (Pan et al., 2021; Han et al., 2022) can also be applied for traffic control design.

Although adoption of discretized models in the modeling of macroscopic traffic makes the model simple and reduces the computational burden, the discretized cells inevitably generate errors in actual applications (Mohan and Ramadurai, 2013). In addition, CTM assumes that the density and speed in each cell is uniform and vehicles are assumed to have instantaneous acceleration and deceleration, which is also another unrealistic phenomenon (Daganzo, 1994). Furthermore, assuming that traffic is uniformly distributed within each cell violates the "causality" property. If the inflow to a cell has stopped or is declining, then the cell needs to redistribute the traffic uniformly towards backward. It is not consistent with the property that vehicles are influenced only by traffic ahead and not by traffic behind (Carey, 2021). Nonlinear traffic dynamics are not well captured by most "discretize then design" methods.

Design then discretize approaches avoid introducing the numerical approximation errors before control designs, therefore leading to more accurate control solutions. Control designs are directly proposed for the LWR PDE model or the ARZ PDE model that are continuous in time and space. In particular, the "design then discretize" control approaches mainly includes Lyapunov-based design (Bastin and Coron, 2016), backstepping design (Krstic and Smyshlyaev, 2008b), optimal control design, (Delle Monache et al., 2017; Colombo and Groli, 2004), distributed control (Bekiaris-Liberis and Delis, 2021; Qi et al., 2023), and learning-based control approaches (Belletti et al., 2018; Yu et al., 2022b).

The "design then discretize" approaches also offer greater adaptability for different control objectives of traffic (i.e., traffic stabilization (Yu and Krstic, 2019), and traffic regulation (Delle Monache et al., 2017)) or different traffic scenarios (i.e.,pure

Table 1
Comparison between different methods and models.

Reference	Model	Control methods	High efficiency	Theoretical stability guarantee	Easy implementation
Papageorgiou et al. (1991)	CTM	Integral control	✓		✓
Gomes and Horowitz (2006)	ACTM	Optimal control			✓
Ferrara et al. (2015)	CTM	Event-triggered MPC			✓
Carlson et al. (2011)	METANET	Integral control	✓		✓
Han et al. (2022)	METANET	Reinforcement learning	✓		✓
Belletti et al. (2018)	LWR	Reinforcement learning	✓		✓
Karafyllis and Papageorgiou (2019)	LWR	Feedback control		✓	
Delle Monache et al. (2017)	LWR	Optimal control			✓
Zhang et al. (2019)	ARZ	PI control		✓	
Yu and Krstic (2019)	ARZ	Backstepping		✓	
This paper	ARZ	Backstepping + NO	✓	✓	✓

traffic (Karafyllis and Papageorgiou, 2019), multi-class traffic (Burkhardt et al., 2021)), as modifications can be directly applied to continuous modeling without affecting the discretization schemes. The preservation of continuity in time also offers flexibility for control design.

Among these PDE-based control methods, they are all dealing with PDEs whose computational burden and problem formulation is higher compared with ODEs. Even for the basic feedback controller in Zhang et al. (2019), one needs to solve linear matrix inequalities (LMIs) to get the control gains for the traffic system. Solving LMIs would be time-consuming and need specific domain knowledge of it that also raises the threshold for using this method. Reinforcement learning can partially tackle this problem but it needs retraining for different parameters and traffic scenarios. However, for large-scale of traffic such as link-level traffic, the "design then discretize" method may present implementation challenges. The dynamics of traffic would be more complex in cascaded traffic scenarios (Yu and Krstic, 2022), therefore, the control design and solving analytical model problems would be more computationally intensive and technically demanding. In this paper, we design the boundary controller to mitigate traffic oscillations using the backstepping method and then we design the operator learning framework to reduce the computational burden for the traffic system. The comparison between different control methods and models is shown in Table 1.

PDE backstepping control has been widely studied for boundary stabilization of the hyperbolic PDE models (Krstic and Smyshlyaev, 2008b,a; Vazquez et al., 2011; Anfinsen and Aamo, 2019; Zhang et al., 2024b). Krstic and Smyshlyaev (2008a) first proposed the backstepping controller for stabilization of hyperbolic PDEs by simply actuating the boundary conditions, for example, flow input or speed at the boundaries of the ARZ model. Over the past decades, backstepping approaches have been extended for robust control design by Auriol and Di Meglio (2020) and Karafyllis and Krstic (2019), output disturbance rejection by Lamare and Di Meglio (2016), and adaptive design by Anfinsen and Aamo (2019) with respect to parameter uncertainty and disturbances.

Motivated by the stop-and-go traffic modeled by the ARZ PDEs, Yu and Krstic (2022) first applied backstepping method for congested freeway traffic control problems and then extended to multi-lane, and multi-class traffic PDE models. The control objective of the freeway traffic using backstepping method is to stabilize the traffic states at their equilibrium points. Unlike the results in Delle Monache et al. (2017) whose objective is to regulate the outflow to a desired outflow, backstepping method aims to make the full states of density and speed stay at a spatially uniform value. The ARZ PDE system is transformed into an exponentially stable target system using the backstepping transformation along with the controller design. The control gains are obtained by solving the kernel equations of the transformation. The boundary control law then is constructed with the backstepping gain kernels and system states.

1.2. Neural operators for control of PDEs

Over the recent decades, machine learning (ML) methods have emerged as powerful tools for solving complex algebraic equations and PDEs. Physics-informed neural network (PINN) has ability to solve forward problem of PDEs (Lu et al., 2021b). PINN can be used for traffic state and fundamental diagram estimation, which can learn a functional form of the fundamental diagram and solve the first-order and second-order traffic flow models (Shi et al., 2022a; Zhang et al., 2024a; Zhao and Yu, 2023). However, PINN cannot solve some multi-scale dynamical PDE systems because it is sensitive to hyper-parameters, and it only learns the solution of a single PDE system (Sun et al., 2020). It cannot generalize to PDEs with different parameters, initial conditions, and boundary conditions.

Compared with PINN and other traditional ML methods, neural operators (NO) present exciting advances due to their ability to learn the operator mapping of functionals (Kovachki et al., 2023). The standard frameworks for NO are DeepONet (Lu et al., 2021a) and Fourier Neural Operator (FNO) (Li et al., 2021). The performance of DeepONet and FNO is comparable for relatively simple settings, but the performance of FNO deteriorates greatly for complex geometries (Lu et al., 2022). A significant advantage of DeepONet and related structures is the ability to freely discretize output functions (Lin et al., 2021). This flexibility allows the network to predict the values of the output functions at any given domain Zhang et al. (2023). To make neural operators more accurate and robust, the extended DeepONet and FNO were proposed (Lu et al., 2022). With further consideration of the absence of training data and the generality of neural operators, Wang proposed physics-informed DeepONets to learn the solution of arbitrary PDEs (Wang et al., 2021b). These neural operators offer distinctive advantages compared to other traditional ML methods due to

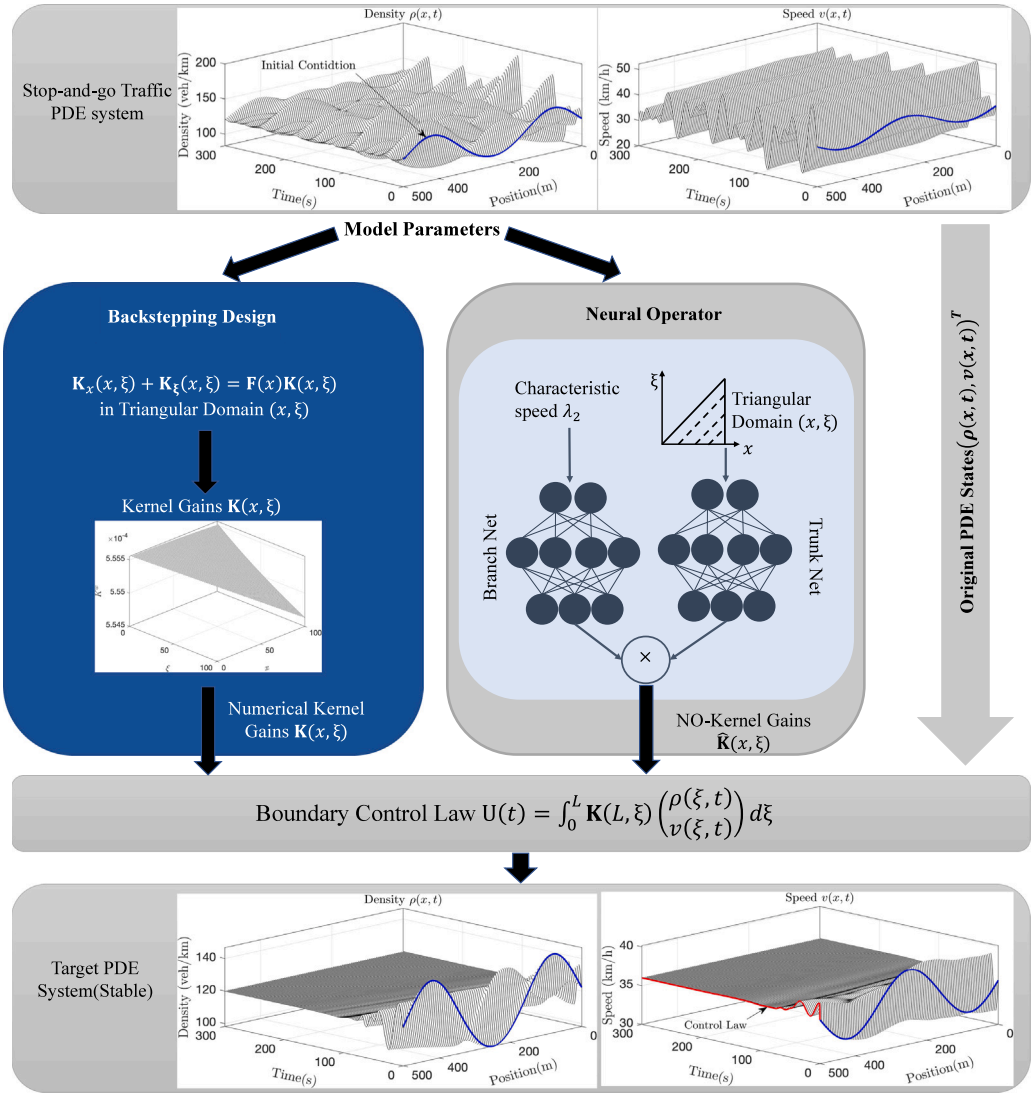


Fig. 1. The diagram of the backstepping method and the proposed neural operator framework.

their simplicity in solving complex problems such as climate forecasting (Pathak et al., 2022), multiphase flow (Wen et al., 2022), heterogeneous material modeling (You et al., 2022). Therefore, it is of great value for solving PDEs and backstepping kernel equations using neural operators. Bhan et al. (2023) adopted neural operators for nonlinear adaptive control. The operator learning framework was proposed to accelerate nonlinear adaptive control. In addition, they apply the operator learning method for bypassing gain and control computations in PDE control (Bhan et al., 2024, 2023). With the operator learning framework, there is no need to compute backstepping kernel gains by solving kernel equations numerically.

All of the aforementioned NO-based results demonstrated the capability of neural operators in accelerating computational speed and analytical derivation for PDE control design. However, to the author's knowledge, PDE boundary control using neural operators has not been studied in traffic control problems before. The neural operator framework has great potential to derive methodological learning-based control for the freeway traffic stabilization problem, which will be extensively discussed in this paper.

1.3. Contributions

To address the limitations of existing traffic control methods based on macroscopic PDE models in mitigating stop-and-go traffic oscillations, we identify key constraints that motivate our proposed operator learning framework and the main contributions.

- "Discretize then Design" control methods are inherently limited by their discretization schemes, often resulting in discretization traffic states errors, loss of continuity between cells, and difficulties in handling nonlinearities of traffic dynamics. Our proposed Neural Operator (NO) scheme is notably invariant to discretization.

- Backstepping, as a representative method of "design then discretize", offers unique advantages in free traffic control—providing a closed-form solution with lower errors and enhanced robustness, and maintaining the continuity of traffic states, compared to "discretize then design" methods However, the backstepping method incurs a significant computational burden due to the complexity of solving the control gain kernels for PDEs. The PDE control design also demands a deep level of expertise, complicating its practical implementation. The proposed NO scheme is designed to accelerate the computational process and bypass the need for analytical design through data-driven training.
- Although PINN can partially expedite the computation process for solving specific PDEs, the method is restricted to learning PDE solution of one instance, characterized by a single set of boundary and initial conditions. This limitation renders it ineffective in adapting to changes of traffic patterns and demands. Conversely, the proposed NO methods are designed to learn functional mappings from PDE model parameters to the control gains. These mappings remain invariant regardless of changes in traffic's initial and boundary conditions or system parameters, thereby providing a more flexible and robust solution for freeway traffic control.

To overcome the above problems, we proposed NO-based methods to approximate the operator mapping from congested wave speeds to control gain kernels. This approach significantly reduces computational burden and allows closed-loop results to generalize across varying traffic demands and patterns. Author's previous result on RL traffic control (Yu et al., 2022b) trained a reinforcement learning boundary controller using the ARZ PDE model as a simulator. The results are obtained for one-instance and there needs retraining given different traffic conditions. Different from the focus of this paper on traffic stabilization, another line of research has explored learning-based approaches for traffic state estimation (Shi et al., 2022b) using FNO and using PINN (Zhao and Yu, 2023). Authors investigated the PDE observer design using partial measurement information to infer the full spatial–temporal traffic states.

Two mappings, the NO-approximated gain kernels and the NO-approximated control law, are developed to improve the computation speed of gain kernels and control law based our previous results in Zhang et al. (2024d). Different with our previous results, we further developed the physics-informed neural operator (PINO) framework to learn the mapping from system parameters to the backstepping kernels to deal with limited data scenarios. The operator learning framework based on the backstepping method is shown in Fig. 1. To the best of author's knowledge, this is the first result for the application of operator learning in traffic control, in particular for boundary control of the macroscopic ARZ PDE model. The theoretical contribution lies in proposing the novel Lyapunov analysis for neural operators and proving the stability of the NO-based closed-loop traffic system. Extensive simulation results demonstrate that the NO-approximated methods significantly accelerate the computation of control laws while successfully stabilizing the traffic system under varying conditions. This approach presents an efficient, accurate, and robust solution for traffic control.

The structure of the paper is as follows: Section 2 presents the design of a boundary controller for the ARZ traffic model utilizing the backstepping method. In Section 3, the neural operator is introduced to approximate the backstepping kernels, followed by a Lyapunov analysis of the NO-based kernels. Additionally, the NO-approximated backstepping control law is developed, and it is demonstrated that practical exponential stability of the system is achieved. The extension to PINO is also discussed to illustrate its effectiveness in the absence of input–output data. Section 4 details the experiments conducted on the neural operators for kernels and control law, along with a comparative analysis of the NO-based methods, the PI controller, the PINN-based controller, and the backstepping controller. Finally, Section 5 provides the conclusion of the paper.

2. Boundary control of the ARZ PDE model

The macroscopic traffic dynamics on a given road are described by the nonlinear coupled hyperbolic PDEs, i.e., the ARZ model. The model is defined by:

$$\partial_t \rho + \partial_x(\rho v) = 0, \tag{1}$$

$$\partial_t(v - V(\rho)) + v\partial_x(v - V(\rho)) = \frac{V(\rho) - v}{\tau}, \tag{2}$$

where $\rho(x,t)$ is the traffic density, $v(x,t)$ denotes the traffic speed, defined in the spatial and time domain $(x,t) \in [0,L]\times[0,+\infty)$. The reaction time τ denotes how long it takes for drivers' behavior adapting to equilibrium density-speed relation $V(\rho)$. The fundamental diagram $V(\rho)$ describes the relation between the traffic density and speed. The fundamental diagram should guarantee the flow function $q(\rho) = \rho V(\rho)$ to be strictly concave(i.e., $q''(\rho) < 0$). It can be selected as the Greenshield's model:

$$V(\rho) = v_f\left(1 - \left(\frac{\rho}{\rho_m}\right)^{\gamma}\right), \tag{3}$$

where v_f is the maximum speed for the traffic flow, ρ_m denotes the maximum density. It should be noted that the proposed control design does not restrict the choice of fundamental diagram as long as the flow-density relation $Q(\rho)$ is twice differentiable and concave. We define the equilibrium state of the system as $(\rho^\star, v^\star)$. We have established the relation between the equilibrium speed and density using the fundamental diagram

$$v^\star = V(\rho^\star), \tag{4}$$

Considering the traffic conditions on the freeway, we set the inlet boundary $x = 0$ as a constant traffic flow $q^\star = \rho^\star v^\star$, thus we get inlet boundary condition as:

$$\rho(0,t) = \frac{q^\star}{v(0,t)}. \tag{5}$$

At the outlet of the road section, we set the traffic density as $\rho^\star$ to obtain the following boundary condition for traffic speed

$$v(L,t) = \frac{q(L,t)}{\rho^\star} + U(t), \tag{6}$$

where $U(t)$ is the control input, actuating the flow of vehicles that leaving of the road section from the outlet. It can be implemented by ramp metering to regulate the traffic flow at the outlet of the mainline road. In this paper, we deal with the congested traffic condition, meaning that the traffic density $\rho(x,t)$ is larger than the critical density for the traffic system. The control objective is to regulate the traffic density and speed at its equilibrium points $(\rho^\star, v^\star)$ within finite time. We consider the congested traffic where stop-and-go traffic oscillations arise. The congested equilibrium density $\rho^\star$ is chosen such that $\rho^\star > \rho_c$ where ρ_c is the critical density that satisfies $Q'(\rho)|_{\rho=\rho_c} = 0$ and defines the congested and free traffic.

The boundary control input is designed to mitigate the traffic oscillations,

$$\begin{aligned} U(t) = &-(\rho(L,t)v(L,t) - q^\star) + \left(\rho^\star + \frac{v^\star}{V'(\rho^\star)}\right)(v(L,t) - v^\star) + \int_0^L \mathrm{e}^{\frac{\xi}{\tau v^\star}} K^w(L,\xi)(q(\xi,t) - q^\star)d\xi \\ &- \int_0^L \left(\frac{v^\star}{V'(\rho^\star)} K^v(L,\xi) + \left(\rho^\star + \frac{v^\star}{V'(\rho^\star)}\right)\mathrm{e}^{\frac{\xi}{\tau v^\star}} K^w(L,\xi)\right)(v(\xi,t) - v^\star)d\xi \end{aligned} \tag{7}$$

where $K^w(L,\xi)$, $K^v(L,\xi)$ are kernel gains of the controller. They are obtained from the backstepping control design. The computation of the control gains will be given in the next section. The following lemma is stated regarding the closed-loop system.

Lemma 1 (*Yu and Krstic, 2019*)**.** *The system (1)–(2) with boundary conditions (5)–(6) and initial conditions $\rho(x,0), v(x,0) \in L^2[0,L]$ under the control law (7) whose kernels are solved by (19)–(22) is locally exponentially stable in L_2-sense at finite time $t_f = \frac{L}{v^\star} + \frac{L}{-\rho^\star V'(\rho^\star) - v^\star}$, and the traffic density $\rho(x,t)$ and speed $v(x,t)$ converge to their equilibrium points at finite time.*

$$\|\rho(x,t) - \rho^\star\| \to 0,$$
$$\|v(x,t) - v^\star\| \to 0.$$

2.1. Backstepping controller design

We firstly linearize the PDE system (1)–(2) around its equilibrium point $(\rho^\star, v^\star)$. The small deviation of the equilibrium point are defined as

$$\bar{\rho}(x,t) = \rho(x,t) - \rho^\star, \tag{8}$$

$$\bar{v}(x,t) = v(x,t) - v^\star. \tag{9}$$

We also have the small deviation of traffic flow $\bar{q}(x,t) = q(x,t) - q^\star$. The original system is a nonlinear hyperbolic PDE where the two PDE states are coupled in domain. We transform the original model into the boundary control model for the follow-up backstepping design. Then we can define the following spatial transformation for the linearzied PDE states $(\tilde{\rho}, \tilde{v})$,

$$\tilde{w}(x,t) = \mathrm{e}^{\frac{x}{\tau v^\star}}\left(\bar{q}(x,t) - \left(\rho^\star + \frac{v^\star}{V'(\rho^\star)}\right)\bar{v}(x,t)\right), \tilde{v}(x,t) = -\frac{v^\star}{V'(\rho^\star)}\bar{v}(x,t) \tag{10}$$

The system (1)–(2) with boundary conditions (5)–(6) are then transformed into the following linearized boundary control model.

$$\partial_t \tilde{w}(x,t) + \lambda_1 \partial_x \tilde{w}(x,t) = 0, \tag{11}$$

$$\partial_t \tilde{v}(x,t) - \lambda_2 \partial_x \tilde{v}(x,t) = c(x)\tilde{w}(x,t), \tag{12}$$

$$\tilde{w}(0,t) = -r\tilde{v}(0,t), \tag{13}$$

$$\tilde{v}(L,t) = \kappa \tilde{w}(L,t) + U(t), \tag{14}$$

where the coefficients are $c(x) = -\frac{1}{\tau}\mathrm{e}^{-\frac{x}{\tau v^\star}}$, $r = \frac{-\rho^\star V'(\rho^\star) - v^\star}{v^\star}$, $\kappa = \mathrm{e}^{-\frac{L}{\tau v^\star}}$. The characteristic speed λ_1, λ_2 of the traffic PDE represent the propagation direction of traffic waves.

$$\lambda_1 = v^\star, \tag{15}$$

$$\lambda_2 = -\rho^\star V'(\rho^\star) - v^\star. \tag{16}$$

The characteristic speeds of the free-flow and congested traffic are: (a) **free-flow regime**: $\lambda_1 > 0, -\lambda_2 > 0$, the traffic waves transports from upstream to downstream; (b) **congested regime**: When the characteristic speed of traffic speed is negative, such as $\lambda_1 > 0$ and $-\lambda_2 < 0$, the traffic is in the congested regime. The speed oscillation will transport from downstream to upstream while the traffic density still transport from upstream to the downstream, making the traffic become congested.

By applying the backstepping transformation, the plant system (11)–(14) is converted into a target system such that in-domain unstable couplings are transformed to the boundary and then the oscillations are damped out by boundary actuation. The following backstepping transformation is introduced,

$$\alpha(x,t) = \tilde{w}(x,t), \tag{17}$$

$$\beta(x,t) = \tilde{v}(x,t) - \int_0^x K^w(x,\xi)\tilde{w}(\xi,t)d\xi - \int_0^x K^v(x,\xi)\tilde{v}(\xi,t)d\xi, \tag{18}$$

where $\alpha(x,t)$, $\beta(x,t)$ are the transformed PDE states of the target system and $K^w(x,\xi)$, $K^v(x,\xi)$ are the kernels of the transformation. This transformation converts the system (11)–(14) into an exponential stable target system combined with the kernel Eqs. (19)–(22) and the backstepping control law (27). The kernel equations evolve in the triangular domain $\mathcal{T} = \{(x,\xi) : 0 \le \xi \le x < L\}$.

$$\lambda_2 K_x^w(x,\xi) - \lambda_1 K_\xi^w(x,\xi) = c(x)K^v(x,\xi), \tag{19}$$

$$\lambda_2 K_x^v(x,\xi) + \lambda_2 K_\xi^v(x,\xi) = 0, \tag{20}$$

$$K^w(x,x) = -\frac{c(x)}{\lambda_1+\lambda_2}, \tag{21}$$

$$K^v(x,0) = -K^w(x,0). \tag{22}$$

Using the transformation and the kernel equations, we can get the target system as follows:

$$\partial_t\alpha(x,t) + \lambda_1\partial_x\alpha(x,t) = 0, \tag{23}$$

$$\partial_t\beta(x,t) - \lambda_2\partial_x\beta(x,t) = 0, \tag{24}$$

$$\alpha(0,t) = -r\beta(0,t), \tag{25}$$

$$\beta(L,t) = 0. \tag{26}$$

with the backstepping control law chosen as follows,

$$U(t) = -\kappa\tilde{w}(L,t) + \int_0^L K^w(L,\xi)\tilde{w}(\xi,t)d\xi + \int_0^L K^v(L,\xi)\tilde{v}(\xi,t)d\xi, \tag{27}$$

such that $\beta(L,t) = 0$. The target system (23)–(26) is proved to be exponentially stable in L_2-sense following the results in Yu and Krstic (2019). The target system is equivalent to the linearized system (11)–(14) since the aforementioned transformation is invertible. Writing (27) in the original coordinates (ρ, v), we obtain the control input (7) for the original system (1)–(2). Thus, applying the control law to the system, and then the unstable traffic system could be stabilized.

3. Neural operator learning to accelerate computation of backstepping gain kernels

In this section, we propose the operator learning framework to accelerate the computation process of the backstepping method. The boundary control law is constructed using the backstepping gain kernels. Solving the kernel equations and calculating the boundary control law can be time-consuming. The neural operators developed in this section are used to reduce the computational burden. Three operator learning schemes. We will use neural operators to learn the mapping from the characteristic speed to the backstepping gain kernels, and the boundary control law. Finally we develop the physics-informed operator learning framework.

3.1. Neural operator for approximating backstepping gain kernels

The neural operator is employed for approximating the operator mapping of functionals. In the section, we introduce the neural operator using DeepONet to approximate the mapping from the characteristic speed λ_2 to kernels $K^w(x,\xi)$ and $K^v(x,\xi)$. A neural operator (NO) for approximating a nonlinear mapping $\mathcal{G} : \mathcal{U} \mapsto \mathcal{V}$

$$\mathcal{G}_{\mathbb{N}}\left(\mathbf{u}_m\right)(y) = \sum_{k=1}^{p} \underbrace{g^{\mathcal{N}}\left(\mathbf{u}_m;\vartheta^{(k)}\right)}_{\text{branch}} \underbrace{f^{\mathcal{N}}\left(y;\theta^{(k)}\right)}_{\text{trunk}}, \tag{28}$$

where $\mathcal{U}$ and $\mathcal{V}$ are function spaces of continuous functions $u \in \mathcal{U}$, $v \in \mathcal{V}$. And $\mathbf{u}_m$ is the evaluation of function u at points $x_i = x_1, \dots, x_m$. p is the number of basis components in the target space, $y \in Y$ is the location of the output function $v(y)$ evaluations, and $g^{\mathcal{N}}$, $f^{\mathcal{N}}$ are NNs termed branch and trunk networks. $\vartheta^{(k)}, \theta^{(k)}$ denote all trainable weights and bias parameters in the branch and trunk networks.

Lemma 2 (*DeepONet Universal Approximation Theorem Bhan et al., 2024; Chen and Chen, 1995; Deng et al., 2022*). *Let $X \subset \mathbb{R}^{d_x}$, $Y \subset \mathbb{R}^{d_y}$ be compact sets of vectors $x \in X$ and $y \in Y$. Let $\mathcal{U}$: $X \to \mathbb{U} \subset \mathbb{R}^{d_u}$ and $\mathcal{V}$: $Y \to \mathbb{V} \subset \mathbb{R}^{d_v}$ be sets of continuous functions $u(x)$ and $v(y)$, respectively. Assume the operator $\mathcal{G}$: $\mathcal{U} \to \mathcal{V}$ is continuous. Then, for all $\epsilon > 0$, there exists a $m^\star, p^\star \in \mathbb{N}$ such that for each $m \ge m^\star$, $p \ge p^\star$, there exist $\theta^{(k)}$, $\vartheta^{(k)}$, neural networks $f^{\mathcal{N}}\left(\cdot;\theta^{(k)}\right), g^{\mathcal{N}}\left(\cdot;\vartheta^{(k)}\right), k = 1,\dots,p$ and $x_j \in X, j = 1,\dots,m$, with corresponding $\mathbf{u}_m = \left(u\left(x_1\right), u\left(x_2\right), \dots, u\left(x_m\right)\right)^\top$, such that*

$$\sup_{\mathbf{u}\in\mathcal{U}} \sup_{y\in Y} \left|\mathcal{G}(\mathbf{u})(y) - \mathcal{G}_{\mathbb{N}}\left(\mathbf{u}_m\right)(y)\right| < \epsilon, \tag{29}$$

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

Definition 1. The kernel operator $\mathcal{K}$: $\mathbb{R}^+ \to C^1(\mathcal{T}) \times C^1(\mathcal{T})$ is defined by:

$$K^w(x,\xi) := \mathcal{K}^w(\lambda_2)(x,\xi), \tag{30}$$

$$K^v(x,\xi) := \mathcal{K}^v(\lambda_2)(x,\xi). \tag{31}$$

The kernel operator $\mathcal{K}$ denotes the mapping from the characteristic speed to the backstepping transformation kernels. Based on Lemma 2, we have the following lemma on the approximation of the neural operator for the kernel equations:

Lemma 3. *For all $\epsilon > 0$, there exists a neural operator $\hat{\mathcal{K}}$ that for all $(x,\xi) \in \mathcal{T}$,*

$$\sup_{\lambda_2 \in \mathcal{U}} \left\| \mathcal{K}(\lambda_2)(x,\xi) - \hat{\mathcal{K}}(\lambda_2)(x,\xi) \right\| < \epsilon. \tag{32}$$

Proof. The existence, uniqueness of the kernel equations have been proved in Vazquez et al. (2011). So the mapping $\mathcal{K} : \mathbb{R}^+ \to C^1(\mathcal{T}) \times C^1(\mathcal{T})$ from λ_2 to $K^w(x,\xi), K^v(x,\xi)$ indicated by (19)–(22) and the solution of the kernel equations exists. The neural operator $\hat{\mathcal{K}}$ approximates the backstepping kernels for a given λ_2 and their derivatives in the triangular domain $\mathcal{T}$. Using Lemma 2, the maximum approximation error is less than ϵ. □

Remark 1. The error between the neural operator and the kernel operator is less than a given constant ϵ. For the partial derivative of the kernels, we also have the continuous operator $\mathcal{M} : \mathbb{R}^+ \to C^1(\mathcal{T}) \times C^0(\mathcal{T}) \times C^0(\mathcal{T}) \times C^1(\mathcal{T}) \times C^1(\mathcal{T})$

$$\mathcal{M}(\lambda_2)(x,\xi) := (K(x,\xi), \kappa_1(x,\xi), \kappa_2(x,\xi), \kappa_3(x,x), \kappa_4(x)) \tag{33}$$

where $\kappa_1(x,\xi) = \lambda_2 K^w_x(x,\xi) - \lambda_1 K^w_\xi(x,\xi) - c(x)K^v(x,\xi)$, $\kappa_2(x,\xi) = \lambda_2 K^v_x(x,\xi) + \lambda_2 K^v_\xi(x,\xi)$, $\kappa_3(x,x) = K^w(x,x) + \frac{c(x)}{\lambda_1+\lambda_2}$, and $\kappa_4(x) = K^v(x,0) + K^w(x,0)$. And there exists a neural operator $\hat{\mathcal{M}}$ such that for all $(x,\xi) \in \mathcal{T}$

$$\sup_{\lambda_2 \in \mathcal{U}} \left\| \mathcal{M}(\lambda_2)(x,\xi) - \hat{\mathcal{M}}(\lambda_2)(x,\xi) \right\| < \epsilon. \tag{34}$$

So there exists the neural operator $\mathcal{K}(\lambda_2)(x,\xi)$, such that

$$\sup_{\lambda_2 \in \mathcal{U}} \left\| \mathcal{K}(\lambda_2)(x,\xi) - \hat{\mathcal{K}}(\lambda_2)(x,\xi) \right\| + \left\| \partial_x (\mathcal{K}(\lambda_2)(x,\xi) - \hat{\mathcal{K}}(\lambda_2)(x,\xi)) \right\| + \left\| \partial_\xi (\mathcal{K}(\lambda_2)(x,\xi) - \hat{\mathcal{K}}(\lambda_2)(x,\xi)) \right\| < \epsilon \tag{35}$$

We then provide the stability analysis of the ARZ traffic system with the NO-approximated kernels. We first start with the approximated kernels and put them into the ARZ system to get the NO-approximated target system. For a given value of λ_2, defining the output of the neural operator $\hat{\mathcal{K}}(\lambda_2)(x,\xi)$:

$$\hat{K}^w = \hat{\mathcal{K}}^w(\lambda_2)(x,\xi), \tag{36}$$

$$\hat{K}^v = \hat{\mathcal{K}}^v(\lambda_2)(x,\xi). \tag{37}$$

For the NO-approximated kernels $\hat{K}^w, \hat{K}^v$, we have the following result for the NO-approximated system.

Remark 2. The maximum approximation error ϵ gives the error bound of the neural operator approximation and provides the sufficient stability condition to prove the closed-loop system with NO-approximated kernels. It is dependent on the network size and the neural layers. In other words, the proposed method can be generalized given any ϵ-accuracy by increasing the network size. The convergence speed is also related to the approximation error ϵ.

Theorem 1. *The system (1)–(2) with boundary conditions (5)–(6) is locally exponential stable under the control law (41) with initial conditions $\bar{\rho}(x,0)$, $\bar{v}(x,0)$, satisfying*

$$\|(\bar{\rho}(x,t), \bar{v}(x,t))\|^2_{L_2} \le c_1 e^{-\eta t} \|(\bar{\rho}(x,0), \bar{v}(x,0))\|^2_{L^2}, \tag{38}$$

where $c_1 = \frac{m_1 n_2 k_1}{m_2 n_1 k_2}$, $m_1 > 0, m_2 > 0, n_1 > 0, n_2 > 0, k_1 > 0, k_2 > 0$, $\eta = v - \frac{2a\epsilon(2\lambda_1+(1+L)\lambda_2)}{m_1\lambda_2}(1+\frac{1}{k_1}) - \frac{2a\epsilon\lambda_2}{m_1\lambda_2}$, $a > 0$. The kernels are approximated by the neural operator (32) with accuracy ϵ. The traffic system can eventually achieve to its equilibrium.

Proof. First, we define the error for the NO-approximated kernels and backstepping kernels: $\tilde{K}^w(x,\xi) = K^w(x,\xi) - \hat{K}^w(x,\xi)$, $\tilde{K}^v(x,\xi) = K^v(x,\xi) - \hat{K}^v(x,\xi)$. We start from the boundary control model (11)–(14), and the backstepping transformation then is turned into:

$$\hat{\alpha}(x,t) = \tilde{w}(x,t), \tag{39}$$

$$\hat{\beta}(x,t) = \tilde{v}(x,t) - \int_0^x \hat{K}^w(x,\xi)\tilde{w}(\xi,t)d\xi - \int_0^x \hat{K}^v(x,\xi)\tilde{v}(\xi,t)d\xi, \tag{40}$$

the corresponding backstepping control law is

$$U(t) = -\kappa\tilde{w}(L,t) + \int_0^L \hat{K}^w(L,\xi)\tilde{w}(\xi,t)d\xi + \int_0^L \hat{K}^v(L,\xi)\tilde{v}(\xi,t)d\xi. \tag{41}$$

Thus we get the target system with the NO-approximated kernels as

$$\partial_t\hat{\alpha}(x,t) + \lambda_1\partial_x\hat{\alpha}(x,t) = 0, \tag{42}$$

$$\begin{aligned}\partial_t\hat{\beta}(x,t) - \lambda_2\partial_x\hat{\beta}(x,t) = {} & \lambda_2(\tilde{K}^w(x,0) + \tilde{K}^v(x,0))\tilde{v}(0,t) + (\lambda_1+\lambda_2)\tilde{K}^w(x,x)\tilde{w}(x,t) \\ & + \int_0^x (\lambda_2\tilde{K}^w_x(x,\xi) + \lambda_1\tilde{K}^w_\xi(x,\xi))\tilde{w}(\xi,t)d\xi \\ & + \int_0^x (\lambda_2\tilde{K}^v_x(x,\xi) + \lambda_2\tilde{K}^v_\xi(x,\xi))\tilde{v}(\xi,t)d\xi,\end{aligned} \tag{43}$$

$$\hat{\alpha}(0,t) = -r\hat{\beta}(0,t), \tag{44}$$

$$\hat{\beta}(L,t) = 0. \tag{45}$$

For the target system (42)–(45) with the NO-approximated kernels, we define the Lyapunov candidate as

$$V_k(t) = \int_0^L \frac{e^{-\frac{\nu}{\lambda_1}x}}{\lambda_1}\hat{\alpha}^2(x,t) + a\frac{e^{-\frac{\nu}{\lambda_2}x}}{\lambda_2}\hat{\beta}^2(x,t)dx, \tag{46}$$

where the coefficients ν and a are constants and $\nu > 0$, $a > 0$. The states of the NO-approximated backstepping target system $(\hat{\alpha},\hat{\beta})$ and the original states $(\tilde{w},\tilde{v})$ have equivalent L^2 norms

$$k_1\|(\tilde{w}(x,t),\tilde{v}(x,t))\|^2_{L_2} \le \left\|(\hat{\alpha}(x,t),\hat{\beta}(x,t))\right\|^2_{L_2} \le k_2\|(\tilde{w}(x,t),\tilde{v}(x,t))\|^2_{L_2}. \tag{47}$$

In the mean time, the Lyapunov functional $V_k(t)$ is also equivalent to the L^2 norm of the target system, so there exist two constants $m_1 > 0$ and $m_2 > 0$,

$$m_1\left\|(\hat{\alpha}(x,t),\hat{\beta}(x,t))\right\|^2_{L_2} \le V_k(t) \le m_2\left\|(\hat{\alpha}(x,t),\hat{\beta}(x,t))\right\|^2_{L_2}. \tag{48}$$

Taking time derivative along the trajectories of the system, and then we plug the system dynamics. Integrating by parts, thus we get the following result of the Lyapunov candidate. The details of the proof are presented in Appendix A.

$$\dot{V}_k(t) \le -\eta V_k(t) + \left(r^2 - a + 2aL\epsilon\right)\hat{\beta}^2(0,t) - e^{-\frac{\nu}{\lambda_1}L}\hat{\alpha}^2(L,t), \tag{49}$$

where $\eta = \nu - \frac{2a\epsilon(2\lambda_1+(1+L)\lambda_2)}{m_1\lambda_2}(1+\frac{1}{k_1}) - \frac{2a\epsilon\lambda_2}{m_1\lambda_2}$. The coefficients ν,ϵ,a are chosen such that

$$\eta > 0, r^2 - a + 2aL\epsilon < 0. \tag{50}$$

So we get the following result:

$$\dot{V}_k(t) \le -\eta V_k(t) \to V_k(t) \le V(0)e^{-\eta t}. \tag{51}$$

Using the equivalent norm of the Lyapunov functional, we have:

$$\|(\tilde{w}(x,t),\tilde{v}(x,t))\|^2_{L^2} \le e^{-\eta t}\frac{m_1k_1}{m_2k_2}\|(\tilde{w}(x,0),\tilde{v}(x,0))\|^2_{L^2}. \tag{52}$$

Thus, the exponential stability of the NO-approximated PDE system (42)–(45) is proved. The state $\tilde{w}(x,t)$ is obtained from (10), so we have the following equivalent L_2 norm:

$$n_1\|(\bar{\rho}(x,t),\bar{v}(x,t))\|^2_{L_2} \le \|(\tilde{w}(x,t),\tilde{v}(x,t))\|^2_{L^2} \le n_2\|(\bar{\rho}(x,t),\bar{v}(x,t))\|^2_{L_2}, \tag{53}$$

where $n_1 > 0$ and $n_2 > 0$. Therefore, for the original $(\bar{\rho}(x,t),\bar{v}(x,t))$ system, we get

$$\|(\bar{\rho}(x,t),\bar{v}(x,t))\|^2_{L_2} \le c_1e^{-\eta t}\|(\bar{\rho}(x,0),\bar{v}(x,0))\|^2_{L^2}. \tag{54}$$

where $c_1 = \frac{m_1n_2k_1}{m_2n_1k_2}$. Thus, we have proved that the original system (1)–(2) with boundary conditions (5)–(6) is locally exponential stable under the NO-approximated kernels and the system can eventually achieve to its equilibrium. This completes the proof of Theorem 1. □

This neural operator method is similar to the backstepping method. Also, then we take the coordinate transformation of the original unstable PDE system (1)–(2) to get the boundary control model. To determine the key parameter in the boundary control model, we select character speed λ_2 as the input to the neural operator. With the trained neural operator, we get the backstepping kernels directly without solving kernel equations. Applying the NO-based kernels to the control law (41), and then adding the control law to the boundary control model. Thus we get the NO-based target PDE system (42)–(45). We have proved that the NO-based target system is exponentially stable in the spatial–temporal domain through the Lyapunov analysis. For the DeepONet to learn an operator $\hat{\mathcal{K}}(\lambda_2)(x,\xi)$, the inputs are taken as the characteristic speed λ_2. Then the characteristic speed λ_2 goes into the branch net

and the triangular domain coordinate goes into the trunk net to train the model. The output of the brunch net and trunk net then are made dot product to get the final learned operator for the mapping $\lambda_2 \to K^w(x,\xi), K^v(x,\xi)$. The detailed diagram of the neural operator can be found in Fig. 1. The computation time and complexity are highly reduced by using the learned operator to get the backstepping kernels. There is no need to solve the kernel equations online anymore with trained operator, which can be more efficient for real-time implementation and more cheaper for the traffic administration.

In the previous section, the backstepping kernels are approximated by neural operators $\hat{\mathcal{K}}^w(\lambda_2)(x,\xi)$ and $\hat{\mathcal{K}}^v(\lambda_2)(x,\xi)$. Using the NO-approximated backstepping kernels, we have proved that the PDE system is exponentially stable. However, the control law (27) still requires integration of the kernels along the road, resulting real-time implementation on the freeway difficult. Therefore, we extend the neural operator to directly approximate the mapping from the characteristic speed λ_2 to the control law (27). The stability we achieve in the control law mapping is practical. Recalling the control law (27), we define the operator mapping $\mathcal{H}\left(\lambda_2\right) : \mathbb{R}^+ \to \mathbb{R}$ that maps λ_2 to $U(t)$. The expression of backstepping control law (27) shows that there is no explicit form for the mapping from λ_2 to $U(t)$. The relation between λ_2 and $U(t)$ is characterized by the kernel Eqs. (19)–(22). The control law mapping is

$$U(t) = \mathcal{H}\left(\lambda_2\right)(L,t), \tag{55}$$

and the NO-approximated mapping for $\mathcal{H}\left(\lambda_2\right) : \mathbb{R}^+ \to \mathbb{R}$ is defined as $\hat{\mathcal{H}}(\lambda_2) : \mathbb{R}^+ \to \mathbb{R}$. The traffic system is practical stable under the NO-approximated control law $\hat{\mathcal{H}}(\lambda_2)$. The detailed proof of stability results is shown in Appendix B.

3.2. Physics-inform neural operator for kernel approximation

In the previous section, we demonstrated that the NO-approximated control gain kernels ensure exponential stability of the closed system, and that the NO-approximated control law can achieve practical exponential stability. In this section, we extend the neural operator to a physics-informed neural operator (PINO) by incorporating physics constraints into the loss function. This extension aims to reduce the reliance on training data typical of the pure NO method.

Considering a general case for a linear or nonlinear differential operator $\mathcal{Q} : \mathcal{A} \times \mathcal{W} \to \mathcal{F}$, where $(\mathcal{A}, \mathcal{W}, \mathcal{F})$ are function spaces. The differential operator takes the form:

$$\mathcal{Q}(\mathbf{u}, w) = 0, \quad \text{in} \quad D \subset \mathbb{R}^d \tag{56}$$

$$w = g, \quad \text{in} \quad \partial D \tag{57}$$

where $\mathbf{u} \in \mathcal{A}$ denotes the input functions (ie. The characteristic speed in this study) and $w \in \mathcal{W}$ denotes the solutions of the PDE system (i.e., the backstepping kernels). g represents boundary conditions and initial conditions of the PDEs. Additionally, we have the operator mapping $\mathcal{G}(\mathbf{u})(y)$ following Lemma 2 using the formulation of $\mathcal{Q}(\mathbf{u}, w)$.

$$\mathcal{G}(\mathbf{u})(y) = w(\mathbf{u}). \tag{58}$$

Using the previous settings again, the neural operator can be obtained by (28). Then the loss function can be defined as

$$\mathcal{L}(\theta) = \mathcal{L}_{operator}(\theta) + \mathcal{L}_{physics}(\theta), \tag{59}$$

where $\mathcal{L}_{operator}(\theta)$ is the loss for the operator,called data loss, and $\mathcal{L}_{physics}(\theta)$ denotes the physical loss following the definition of loss function in PINN and PINN-type methods. The data loss is given as:

$$\mathcal{L}_{operator}(\theta) = \left\| \mathcal{G}_{\mathbb{N}}\left(\mathbf{u}_m\right)(y) - \mathcal{G}(\mathbf{u})(y) \right\|_{L_2}^2 = \int_D \left| \mathcal{G}_{\mathbb{N}}\left(\mathbf{u}_m\right)(y) - \mathcal{G}(\mathbf{u})(y) \right|^2 dy. \tag{60}$$

The physical loss is defined as:

$$\mathcal{L}_{physics}(\theta) = \|\mathcal{Q}(\mathbf{u}, w)\|_{L_2}^2 + \|w - g\|_{L_2}^2 = \int_D |\mathcal{Q}(\mathbf{u}, w)|^2 dx + \int_D |w - g|^2 dx. \tag{61}$$

More specifically, the physical loss consists of equation loss and boundary loss in this study, $\mathcal{L}_{physics}(\theta) = \mathcal{L}_{equation}(\theta) + \mathcal{L}_{boundary}(\theta)$. The equations loss is defined as

$$\mathcal{L}_{equation}(\theta) = \left\| \lambda_2 K_x^w(x,\xi) - \lambda_1 K_\xi^w(x,\xi) - c(x) K^v(x,\xi) \right\|_{L_2}^2 + \left\| \lambda_2 K_x^v(x,\xi) - \lambda_2 K_\xi^v(x,\xi) \right\|_{L_2}^2. \tag{62}$$

The boundary loss is defined as

$$\mathcal{L}_{boundary}(\theta) = \left\| K^w(x,x) + \frac{c(x)}{\lambda_1 + \lambda_2} \right\|_{L_2}^2 + \left\| K^v(x,0) + K^w(x,0) \right\|_{L_2}^2. \tag{63}$$

The diagram of PINO is shown in Fig. 2. The proposed PINO is an extension of the NO methods. The primary distinction between PINO and NO lies in the design of the loss function within the trained model. PINO seeks to integrate PDE model constraints into the training process by penalizing the loss function. Consequently, the training of PINO may be facilitated by prior knowledge of the PDE system embedded in the model. The weights assigned to the operator loss and the physics loss are hyperparameters that can be defined by the user or tuned, and they are crucial in enhancing the trainability of PINO. Depending on the choice of these weights, the trained PINO model will rely more heavily on either the operator data or the physical kernel equations. In this paper, we consider equal weights for the operator loss and physics loss. Further discussion on the selection and tuning of weights for the physics-informed neural network and PINO can be found in Wang et al. (2021a, 2022a) and Karniadakis et al. (2021).

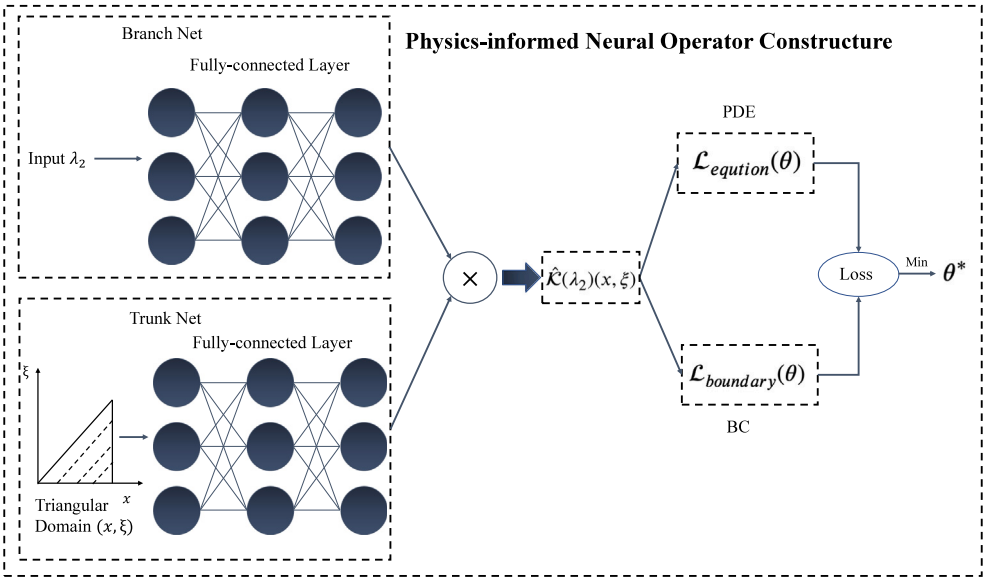


Fig. 2. The diagram of physics-informed neural operator structure for backstepping kernels.

4. Experiments

In this section, we present and analyze the performance of the proposed neural operator controllers for the ARZ traffic PDE system, and also provide comparisons with model-based controllers: (i) backstepping controller (ii) PI controller.

4.1. Simulation and training setups

Simulation setup. To train the neural operator, we use numerical simulations to generate training data. We run the simulation on a road of length $L = 500$ m for a duration of $T = 300$ s. The free-flow speed is $v_f = 144$ km/h, the maximum density is $\rho_m = 160$ veh/km, the equilibrium density is selected as $\rho^\star = 120$ veh/km, $v^\star = 36$ km/h the reaction time for the drives adapting to speed is $\tau = 60$ s, and $\gamma = 1$. For the initial conditions, we use the sinusoidal inputs to mimic the stop-and-go traffic waves.

$$\rho(x,0) = \rho^\star + 0.1\sin\left(\frac{3\pi x}{L}\right)\rho^\star, \tag{64}$$

$$v(x,0) = v^\star - 0.1\sin\left(\frac{3\pi x}{L}\right)v^\star. \tag{65}$$

Using the above parameters of the traffic system, we first run the simulation using numerical solvers to get the evolution of traffic states in the simulation period.

Training setup. To obtain sufficient training data, we use 1000 different values for $\rho^\star \in [90 \text{ veh/km}, 130 \text{ veh/km}]$ to obtain varying values for $\lambda_2 \in [18 \text{ km/h}, 90 \text{ km/h}]$ and different kernels $K^w(x,\xi), K^v(x,\xi)$. We use 1000 samples of λ_2 and its corresponding kernels $K^w(x,\xi), K^v(x,\xi)$ by numerically solving the kernel equations. The samples of λ_2 are randomly sampled from the interval $[18 \text{ km/h}, 90 \text{ km/h}]$. The data is divided into a training set and a test set in a 9:1 ratio, with 900 samples for training and 100 samples for testing. We use the Adam optimizer with a learning rate of 0.0001, decaying every 200 epochs over a total of 1000 epochs. The batch size is set to 20. The training process takes approximately 10 min on a single Nvidia 4090Ti GPU.

Model structure. The DeepONet architecture is employed as the foundational structure of the neural operator. Both branch and trunk networks are designed as fully connected networks. The input to the branch network is selected as the traffic congestion wave speed λ_2, while the input to the trunk network is chosen as the triangular domain. The output of the trained neural operator is then generated by taking the dot product of the outputs from the branch and trunk networks.

4.2. PI controller

To assess the performance of the NO-based controllers, we also include a comparison with the PI controller. Previous research has demonstrated that the PI boundary controller can stabilize the traffic system (Zhang et al., 2019). The PI controller is installed at the outlet of the road section, resulting in the boundary condition of the traffic speed being $\tilde{v}(L,t) = U_{PI}(t)$. The control law is given by:

$$U_{PI}(t) = v^\star + k_p^v(v(0,t) - v^\star) + k_i^v \int_0^t (v(0,t) - v^\star)ds, \tag{66}$$

where k_p^v and k_i^v are tuning gains.

4.3. PINN-based controller

In addition to the PI controller, we further compared NO-based methods with PINN-approximated kernels. The input of PINN is the grid size of the triangular domain x, ξ and the output is the backstepping kernel $K^2(x,\xi), K^v(x,\xi)$. The structure of PINN is the same as Raissi et al. (2019) and Karniadakis et al. (2021). We also have the general form of the kernel equations.

$$\mathcal{N}[u] = 0, \quad x, \xi \in \mathcal{T}, \tag{67}$$

$$u = g_{\text{PINN}}, \quad u \in \partial\mathcal{T}, \tag{68}$$

where $u(x,\xi)$ denotes the latent solution of PDEs. $\mathcal{N}$ is a nonlinear operator which describes the differentiation of PDEs. Then we define $f(x,\xi)$ to denote the residual of PDEs.

$$f(x,\xi) := \mathcal{N}[u]. \tag{69}$$

The loss function is the same as PINO in Section 3.2, consisting of data loss and physics loss.

$$\mathcal{L}_{\text{PINN}} = \mathcal{L}_{\text{data}}(\theta) + \mathcal{L}_{\text{physics}}(\theta), \tag{70}$$

where the $\mathcal{L}_{\text{physics}}(\theta)$ is the same as in Eq. (62)–(63), and the data loss denotes the error between the model output and the ground truth data

$$\mathcal{L}_{\text{data}}(\theta) = \int_{\mathcal{T}} \left\| u(x,\xi) - u_{\text{real}}(x,\xi) \right\|. \tag{71}$$

Besides, the trained PINN model only learns one set of specific backstepping kernels. It cannot output right kernels for the different λ_2.

4.4. Simulation results of closed-loop system

Since the NO-approximated kernels and control law are based on the backstepping method, we select the standard backstepping controller as the baseline for the simulation. The open-loop results for the traffic system are depicted in Fig. 3. Traffic density and speed oscillations persist throughout the simulation period, leading to the occurrence of stop-and-go waves. The density and speed of the closed-loop results using the backstepping method are illustrated in Figs. 4(a) and 5(a). It can be observed that the traffic density and speed all converge to the equilibrium points at the finite time 130 s. The closed-loop results of NO-based controller PI controller, and PINN-based controller are illustrated in Figs. 4 and 5. It is revealed that all control methods effectively stabilize traffic oscillations. Traffic density and speed converge to their equilibrium point, $\rho^\star = 120$ veh/km and $v^\star = 36$ km/h, respectively, despite the sinusoidal initial conditions that initially induce instability throughout the road section. The closed-loop results for the PI controller are depicted in Figs. 4(b) and 5(b). The closed-loop results for the PINN-based controller are shown in Figs. 4(c) and 5(c). The closed-loop results for NO-approximated kernels are shown in Figs. 4(d), 5(d). However, traffic waves are still observable at 150 s due to the small approximation error of the neural operator.

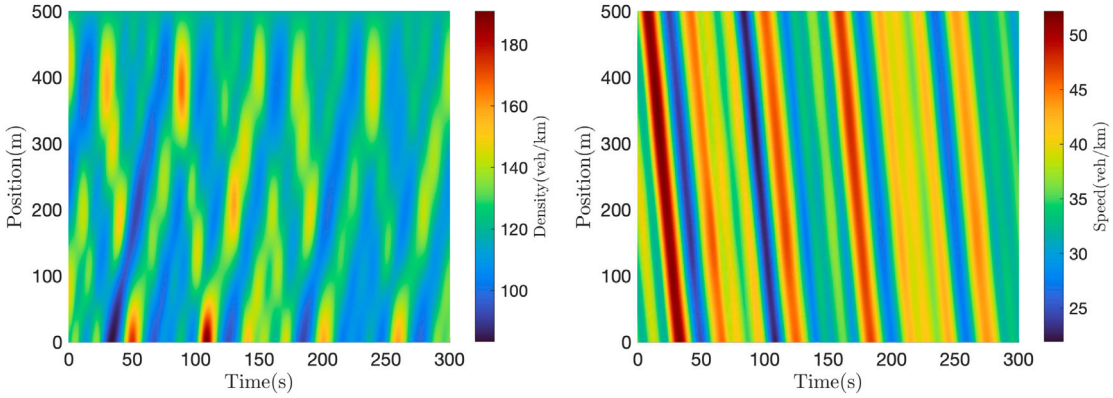
Regarding the results of the NO-approximated control law, the same parameter settings as the previous section were utilized to generate training data, consisting of 900 instances. The results of the approximated control law mapping are illustrated in Fig. 4(e) and Fig. 5(e). It is observed that the NO-approximated control law can practically stabilize the system, as the system does not uniformly converge to the equilibrium point. Small oscillations in both density and speed persist throughout the entire simulation period, preventing uniform convergence to the equilibrium points. This is reasonable because the neural operator only learns the mapping from λ_2 to $U(t)$. As we know, the control law in (27) consists of two parts: the backstepping kernels and the system states at the current time step. The system is practically stable under the condition of the NO-approximated control law.

Subsequently, we present the results for the density and speed errors between other methods and the backstepping method, as depicted in Figs. 6 and 7. The results exclude the PI controller, as it represents a different type of controller compared to the NO-based methods, which all belong to the same backstepping category. The error between the closed-loop result of the backstepping controller and the NO-approximated kernels is illustrated in Fig. 7(a). It is evident that there are some errors at the initial stage of the NO-approximated kernels. The maximum error of the density is approximately 1.5 veh/km at the location of 80 m after 50 s. The maximum speed error is 0.48 km/h. The density and speed error reduce to zero after about 150 s. However, the density and speed errors of the NO-approximated control law persist throughout the entire time period. The density oscillation is smaller than 0.6 veh/km on average and the oscillation of speed is smaller than 0.4 km/h.

For the training procedure of PINO, we utilize half the samples of the training dataset as before to train the model to see whether PINO can still stabilize the traffic system. Using the trained PINO model, we do the prediction for the backstepping kernels. The simulation settings are the same as before, and the results of PINO-approximated kernels are depicted in Figs. 4(d), 5(d). Fig. 4(d) shows the density of the PINO-based result while Fig. 5(d) shows the speed result. It is observed that the maximum density and speed error of PINO-approximated kernels are 2.7 veh/km, 0.8 km/h from Figs. 6(c), 7(c), respectively. The density and speed errors under different NO-based schemes and the PINN method are presented in Table 2. NO-based methods all outperform the PINN-based method.

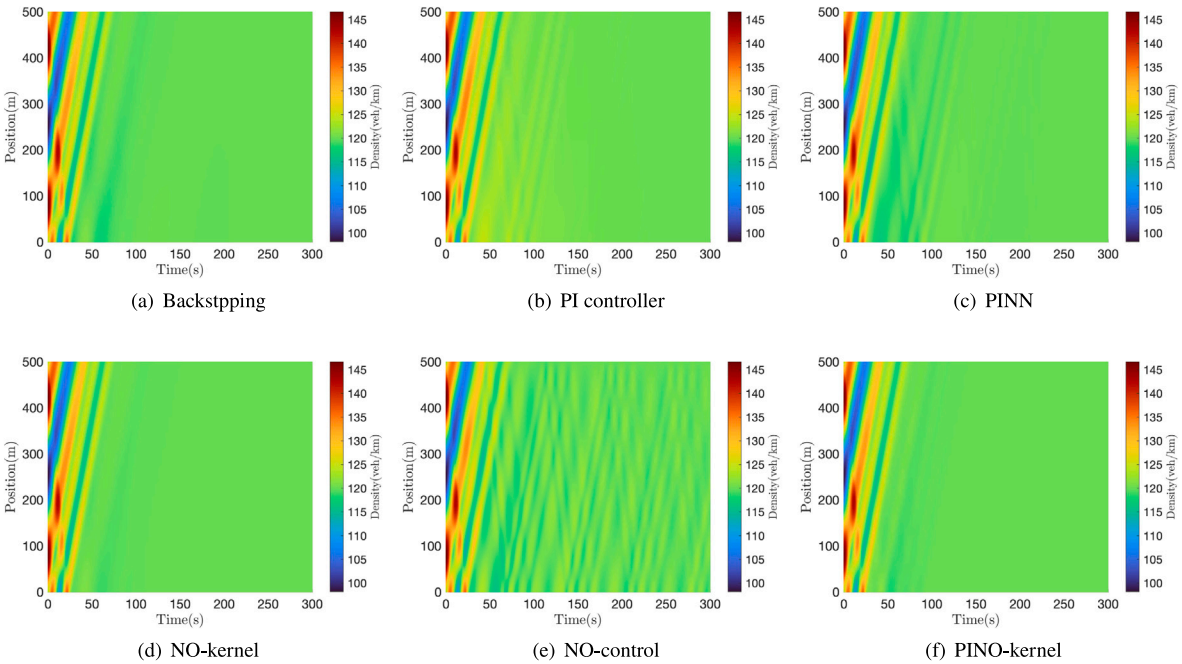
Table 2
The closed-loop density and speed errors under different controllers.

Method	$\rho(x,t)$(veh/km)		$v(x,t)$(km/h)	
	Max absolute error	Mean absolute error	Max absolute error	Mean absolute error
PINN-kernels	3.63%	0.26%	3.76%	0.39%
NO-kernels	1.25%	0.09%	1.34%	0.14%
PINO-kernels	2.23%	0.11%	2.27%	0.16%
NO-control law	3.22%	0.49%	3.45%	0.95%



(a) Density (b) Speed

Fig. 3. Traffic density and speed evolution of the open-loop system.



(a) Backstpping (b) PI controller (c) PINN

(d) NO-kernel (e) NO-control (f) PINO-kernel

Fig. 4. Closed-loop traffic density evolution with different control designs.

4.5. Simulation results of backstepping kernels

Previous section gives the density and speed of the closed-loop system. The NO-based methods learn the backstepping kernels, except for NO-approximated control law. It directly learns the boundary control law for the traffic system. In this section, the results for the NO-approximated kernels are provided. The backstepping kernels, NO-approximated kernels and PINO-approximated kernels

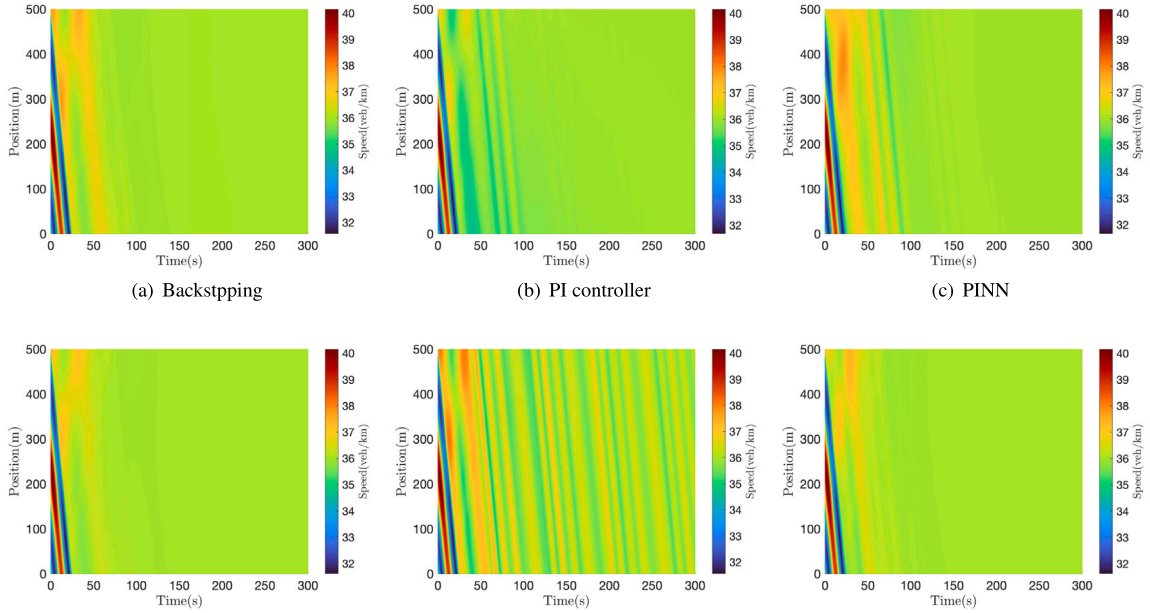


Fig. 5. Closed-loop traffic speed evolution with different control designs.

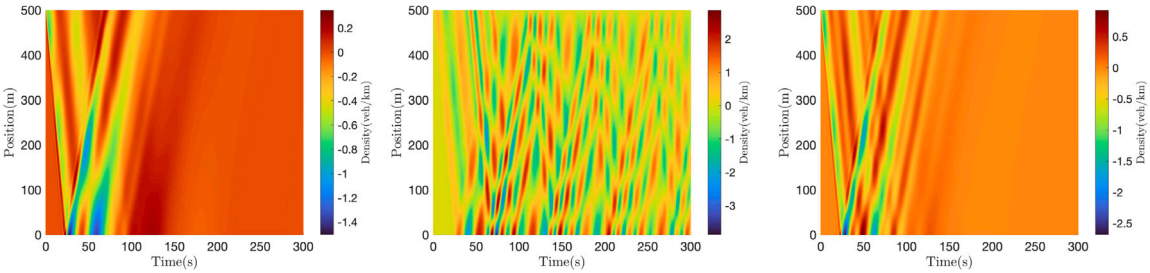


Fig. 6. Traffic density error under different schemes.

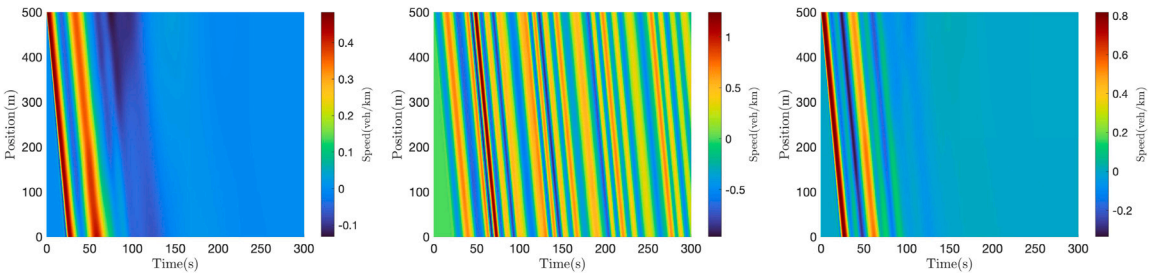


Fig. 7. Traffic speed error under different schemes.

are shown in Fig. 8. To evaluate the performance of the three NO methods, we use the backstepping kernels as the baseline for comparison. The error between the NO methods and the backstepping method is shown in Fig. 9. The first row of Fig. 8 are the results of kernel $\hat{K}^w(x,\xi)$, while the second row shows the results of kernel $\hat{K}^v(x,\xi)$. It is evident that the NO-approximated kernels provided the best approximation to the backstepping kernels. The PINO-approximated kernels exhibit oscillations in the triangular domain, resulting in larger state errors compared to the NO-approximated errors. Therefore, the model does not accurately capture

Table 3
The errors of kernels under different schemes.

Method	$\hat{K}^w(x,\xi)$		$\hat{K}^v(x,\xi)$	
	Max absolute error	Mean absolute error	Max absolute error	Mean absolute error
PINN-kernels	7.833×10^{-4}	1.551×10^{-4}	1.109×10^{-3}	1.525×10^{-4}
NO-kernels	1.452×10^{-4}	1.450×10^{-4}	1.451×10^{-4}	1.071×10^{-4}
PINO-kernels	6.684×10^{-4}	1.232×10^{-4}	2.650×10^{-3}	1.090×10^{-4}

Table 4
The results of different methods with traffic performance indices.

Method	Fuel consumption (↓)	Driving discomfort (↓)	Total travel time (↓)
PINN-kernels	+1.04%	−1.05%	+1.04%
NO-kernels	+1.07%	−23.21%	+1.07%
PINO-kernels	+1.06%	−29.59%	+1.06%
NO-control law	+1.08%	+63.21%	+1.08%

the properties of the kernels. The errors between the NO-based methods and backstepping kernels are depicted in Fig. 9. In the first column, the errors of $K^w(x,\xi)$ using different schemes are presented, while the second column shows the errors of $K^v(x,\xi)$. The maximum error of $K^w(x,\xi)$ and $K^v(x,\xi)$ using NO-approximated kernels is 1.452×10^{-4} and the maximum error of is 1.451×10^{-4}. Compared with NO-approximated kernels, the errors of $K^w(x,\xi)$ and $K^v(x,\xi)$ are higher using PINO-approximated methods. The maximum and mean errors under different schemes are shown in Table 3. The NO-based methods still achieve good performance compared with the PINN method.

In addition to the approximation error of backstepping kernels and traffic states, we also added three traffic performance indices to test the performance of the different methods, including fuel consumption, total travel time (TTT) and comfort value to compare the control performance introduced in Treiber and Kesting (2013). The definition of the performance indices are:

$$J_{\text{fuel}} = \int_0^T \int_0^L \max\{0, b_0 + b_1 v(x,t) + b_2 v(x,t)a(x,t) + b_3 a^2(x,t)\}\rho(x,t)dxdt \tag{72}$$

$$J_{\text{comfort}} = \int_0^T \int_0^L (a^2(x,t) + a_t^2(x,t))\rho(x,t)dxdt \tag{73}$$

$$J_{\text{TTT}} = \int_0^T \int_0^L \rho(x,t)dxdt \tag{74}$$

where the coefficient of fuel consumption model is selected as $b_0 = 2.5\times10^{-3}1/\text{s}$, $b_1 = 2.45\times10^{-7}1/\text{m}$, $b_2 = 1.25\times10^{-8}\ \text{s}^2/\text{m}^2$, $b_3 = 9.5\times10^{-5}\ \text{s}^3/\text{m}^2$ (Ahn, 1998). $a(x,t)$ denotes the local acceleration $a(x,t) = v_t(x,t) + v(x,t)v_x(x,t)$. The results of different NO-based methods are shown in Table 4. The backstepping method is chosen as the baseline for evaluating the indices. The performance of the backstepping method is compared against three other approaches. The analysis reveals that while the NO-approximated and PINO-approximated kernels result in higher fuel consumption and increased total travel time, they significantly enhance driving comfort. Specifically, an improvement of nearly 30% in driving comfort is achieved at the cost of a 1% increase in fuel consumption and total travel time, which is considered acceptable. However, the NO-approximated control law leads to a substantial 60% decrease in driving comfort. This decrease in comfort, characterized by frequent small oscillations across the spatial–temporal domain, results in more abrupt changes in local acceleration, thereby diminishing the driving experience and potentially increasing the risk of traffic accidents.

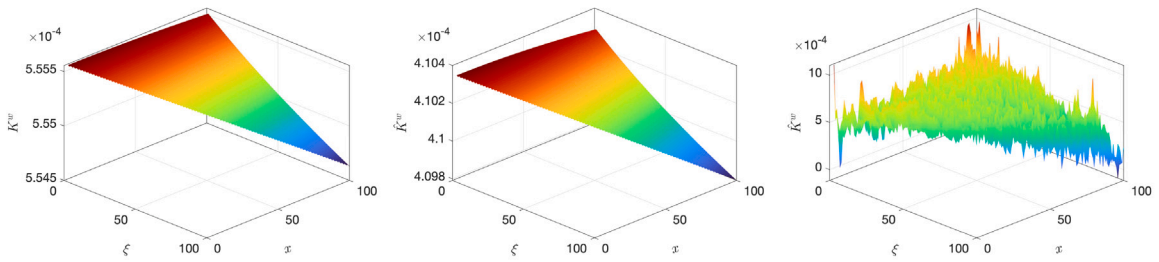
The comparisons of control law and norm of states are shown in Fig. 10. Six methods mentioned in this paper are considered. From the results of $U(t)$, the NO-based methods can approximate the backstepping control law well. All the controllers are eventually stabilizing the system. However, the norm of the states of the NO-approximated control law converges to zero slower than other controllers because we only get the practical stability results for the traffic system. Overall, it can be found that the NO-based and PINO-based methods achieve satisfactory closed-loop results.

The computation time of the neural operator, the backstepping controller, the PI controller and PINN-based controller are shown in Table 5. We set the backstepping control method as the baseline of the system. The Mean Squared Error(MSE) of the traffic system is calculated by

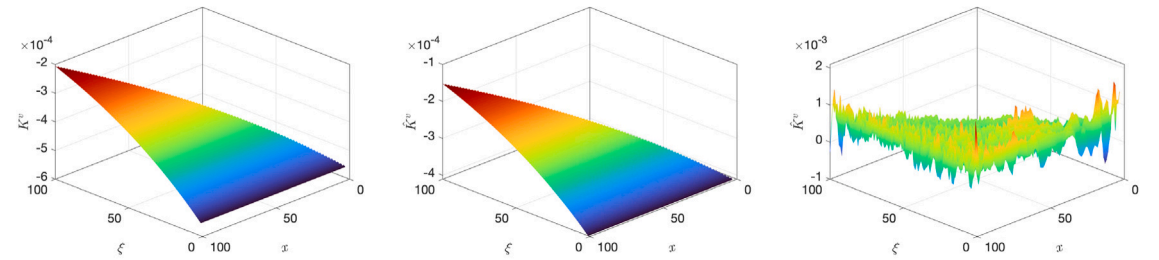
$$\text{MSE}_\rho = \frac{1}{N}\sum_{(x,t)\in D}\left(\frac{\rho(x,t)-\hat{\rho}(x,t)}{\rho^\star}\right)^2 \tag{75}$$

$$\text{MSE}_v = \frac{1}{N}\sum_{(x,t)\in D}\left(\frac{v(x,t)-\hat{v}(x,t)}{v^\star}\right)^2 \tag{76}$$

where $\hat{\rho}(x,t)$ and $\hat{v}(x,t)$ are the density and speed generated by the NO-based methods. $\rho(x,t)$ and $v(x,t)$ are the density and speed of backstepping method. N is the sampled points by numerical methods such as the Godunov scheme (Godunov and Bohachevsky, 1959) in the spatial–temporal domain $D = [0,L]\times[0,T]$. It indicates the average approximation error during the simulation period. It can be observed that the average computation times of the NO-approximated methods are 298 times faster than the backstepping

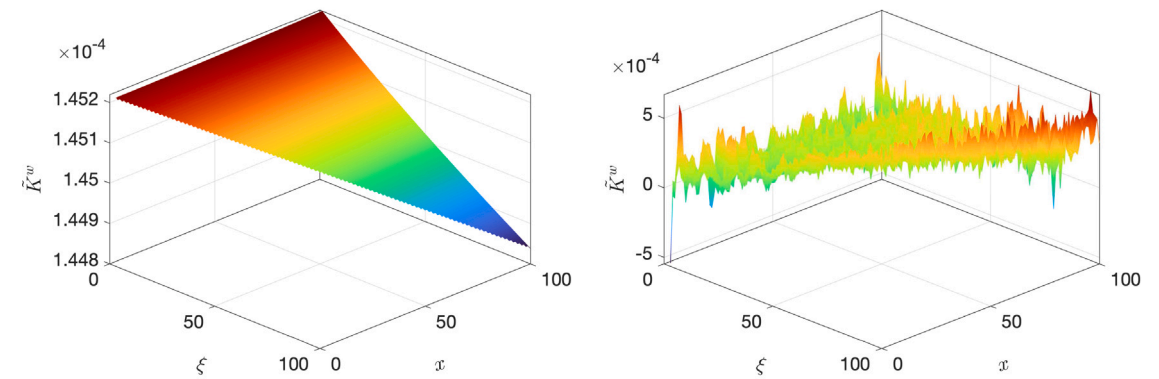


(a) Backstepping (b) NO-kernel (c) PINO-kernel

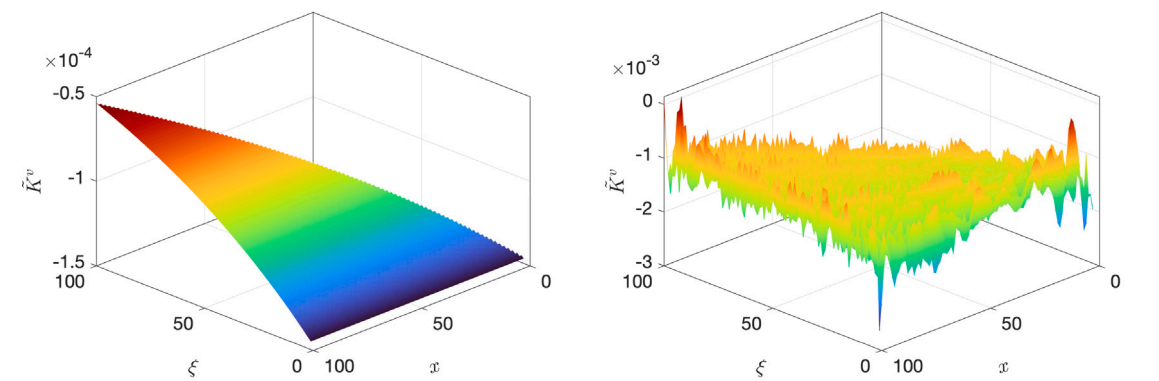


(d) Backstepping (e) NO-kernel (f) PINO-kernel

Fig. 8. Backstepping kernels under different schemes.

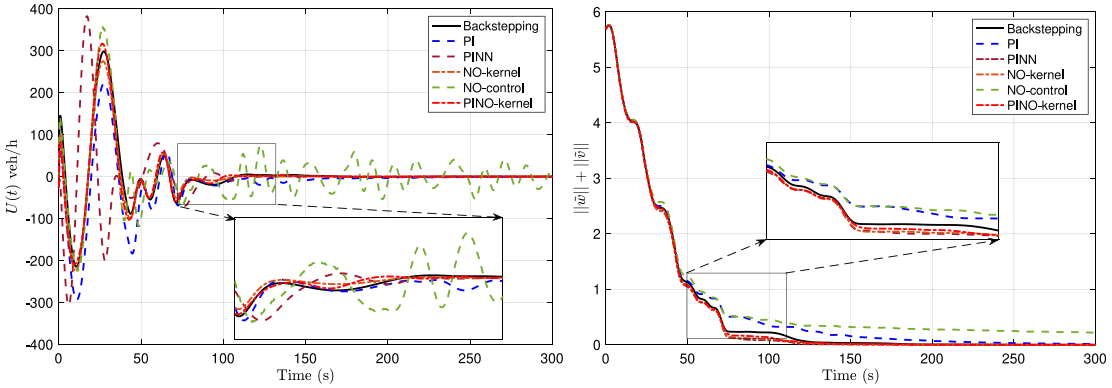


(a) NO-kernel (b) PINO-kernel



(c) NO-kernel (d) PINO-kernel

Fig. 9. Errors of backstepping kernels under different schemes.



(a) Control input $U(t)$ (b) Norm of states

Fig. 10. Comparison of $U(t)$ and the norm of states.

Table 5
The average computation time and MSE of different schemes.

Method	Average computation time (s)	MSE %	
		Density ρ	Velocity v
Backstepping controller	5.960×10^{-2}	/	/
PI controller	1.901×10^{-3} (31×)	6.973×10^{-3}	1.046×10^{-2}
PINN	2.466×10^{-3} (24×)	2.466×10^{-3}	7.523×10^{-3}
NO-approximated kernels	1.997×10^{-4} (298×)	3.783×10^{-4}	8.578×10^{-4}
PINO-approximated kernels	7.854×10^{-4} (75×)	5.861×10^{-4}	1.593×10^{-3}
NO-approximated control law	1.701×10^{-3} (75×)	3.821×10^{-3}	1.274×10^{-3}

controller and 10 times faster than the PI controller with a loss of accuracy less than 1%, giving the possibility of accelerating the online application in a real traffic system.

The computation time of backstepping controller is fast in our simulations. However, this test is limited to a single road segment of 500 meters for 3 min. In real world traffic control problem, such as the cascaded freeway segments (Yu et al., 2022a) described by two sets of ARZ models, multi-class traffic (Burkhardt et al., 2021) or mixed-autonomy traffic (Zhang et al., 2024c), the computation time will increase. Extending the ARZ model to encompass entire traffic networks would further amplify the computational burden and time requirements. Therefore, NO-based methods offer a significant advantage in large-scale traffic scenarios, as their computation time is two orders of magnitude shorter than that of the backstepping method, thereby reducing the overall computational burden.

Summary of simulation results. The simulation results highlight the performance of the backstepping method, PI controller, PINN-based controller, and NO-based methods. All approaches successfully stabilize sinusoidal traffic waves within a finite time, except for the NO-approximated controller, which only achieves practical exponential stability. Among these approaches, the NO-approximated kernels exhibits the smallest density and speed errors during the simulation period and demonstrate the fastest computation speed, averaging just 1.997×10^{-4} seconds per time step. Both the NO-approximated and PINO-approximated kernel methods significantly enhance driving comfort with nearly equivalent cost of fuel consumption and total travel time compared to the PINN. From the perspective of computational efficiency, all NO-based methods outperform the other approaches, with the potential to be up to 298 times faster than the backstepping controller, thus facilitating the acceleration of online control applications in real traffic scenarios.

4.6. Experiments with different demands and conditions

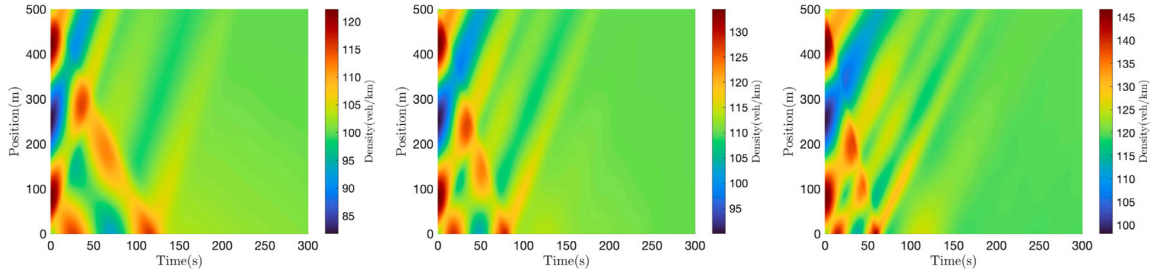
It is revealed that the developed NO-based method performs well in mitigating the stop-and-go traffic in the previous section. It is also needed to mention that whether the proposed NO-based methods could still stabilize traffic for different scenarios. Next, we will test the trained NO in different traffic conditions.

In the data collection and training phases, sinusoidal initial conditions are utilized to simulate stop-and-go traffic waves. However, real-world traffic conditions encompass a variety of scenarios, such as varying traffic demands and non-recurrent traffic conditions. Therefore, it is essential to evaluate the performance of NO-based methods under various initial traffic conditions. To address this, we first assess the performance of the NO method under three distinct traffic demand levels: high, medium, and low. The inflow demand $q^\star$ and the corresponding equilibrium density and speed for each demand level are detailed in Table 6.

By applying different demand levels $q^\star$ to the inlet of the road segment, the results for different demand levels are illustrated in Figs. 11 and 12. It is observed that the method remains effective in stabilizing the traffic system across different demand levels. Both traffic density and speed converge to their equilibrium points within a finite time.

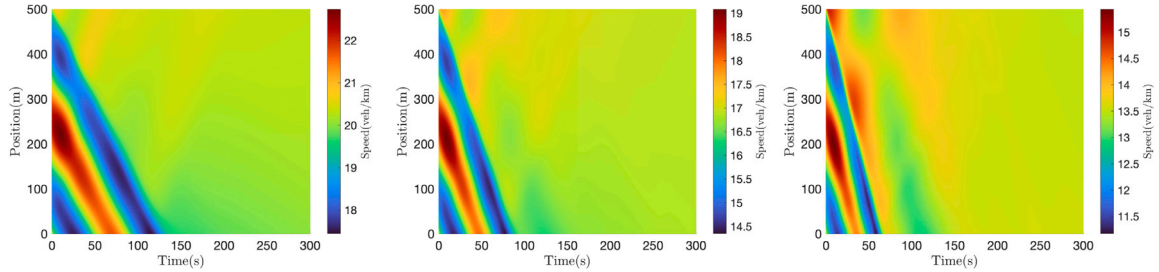
Table 6
The different demand level with equilibrium density and speed.

In-flow demand $q^\star$ (veh/h)	Equilibrium density $\rho^\star$ (veh/km)	Equilibrium speed $v^\star$ (km/h)
2025 (High demand)	100	20.25
1856 (Medium demand)	110	16.87
1620 (Low demand)	120	13.5



(a) $q^\star$=2025 (b) $q^\star$=1856 (c) $q^\star$=1620

Fig. 11. Density evolution of different demand level.



(a) $q^\star$=2025 (b) $q^\star$=1856 (c) $q^\star$=1620

Fig. 12. Speed evolution of different demand level.

Additionally, we evaluate the performance of the developed method under non-recurrent traffic conditions. Specifically, two types of non-recurrent traffic conditions are tested: one with sinusoidal initial conditions at a low frequency, which results in non-recurrent behavior, and another with linear initial conditions. For the first case, the initial conditions are defined as follows:

$$q(x,0) = q^\star + 0.05\sin(\frac{\pi x}{L})q^\star, \tag{77}$$

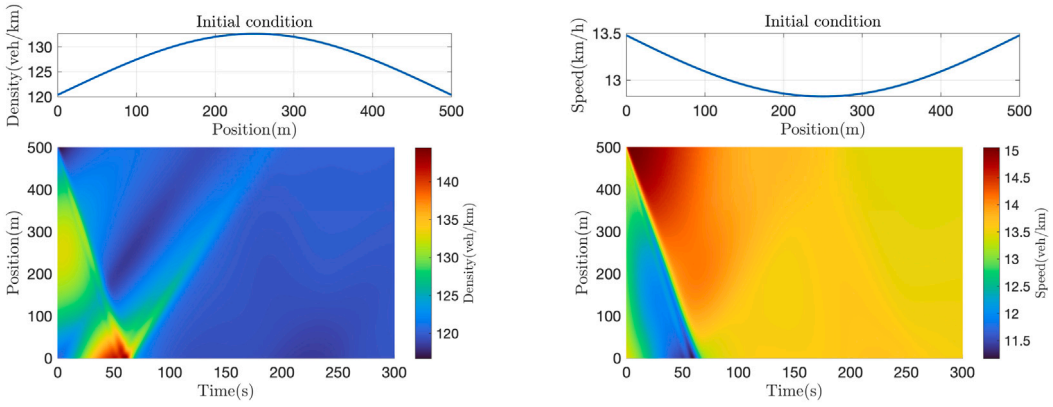
$$v(x,0) = v^\star - 0.05\sin(\frac{\pi x}{L})v^\star. \tag{78}$$

Using the relationship $q = \rho \times v$, we can determine the initial condition of traffic density. This initial condition is designed to replicate a sudden deceleration occurring in the middle of the road segment, resulting in a density wave propagating from upstream to downstream, while a speed wave moves from downstream to upstream. Consequently, the density will initially increase and then decrease, whereas the speed will decrease and subsequently increase. The initial conditions for both density and speed are depicted in Fig. 13. By applying the NO-approximated kernels to these initial conditions of density and speed, the evolution of traffic states is obtained and illustrated in the bottom of Fig. 13. Then, linear initial conditions are applied to simulate a scenario in which vehicles decelerate downstream due to lane closure. This scenario leads to a decrease in traffic speed and an increase in traffic density due to vehicle accumulation. The linear initial conditions are defined as follows:

$$q(x,0) = 9 \times 10^{-5}x, \tag{79}$$

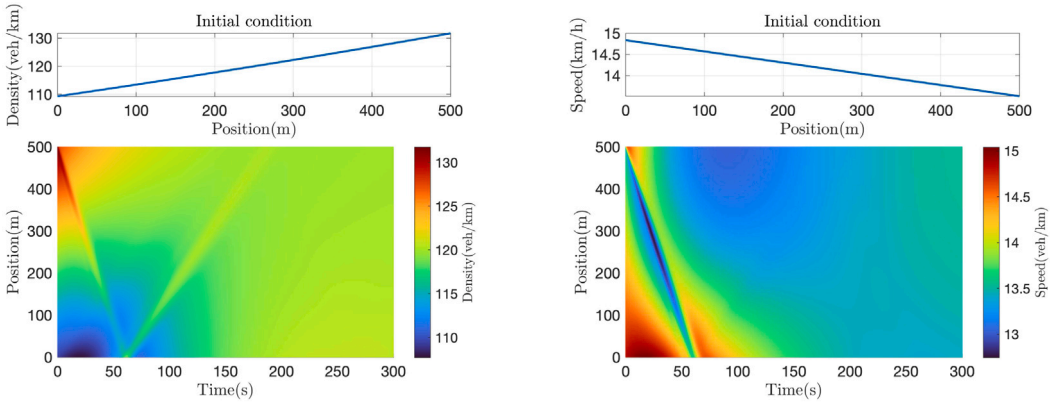
$$v(x,0) = -7 \times 10^{-4}x + 0.375. \tag{80}$$

Using the flow-density relation, we obtain the initial condition of traffic density and speed, as shown in Fig. 14. Applying the NO method to the linear initial conditions, the evolution of traffic states with NO-approximated kernels is depicted in the bottom of Fig. 14. The results demonstrate that, for both types of initial conditions, our method effectively stabilizes the traffic system. Traffic density and speed converge to their equilibrium points, indicating that the developed NO method is robust to variations in traffic conditions.



(a) Density (b) Speed

Fig. 13. Initial conditions and results of density and speed.



(a) Density (b) Speed

Fig. 14. Initial conditions and results of density and speed.

4.7. Experiments with real traffic data

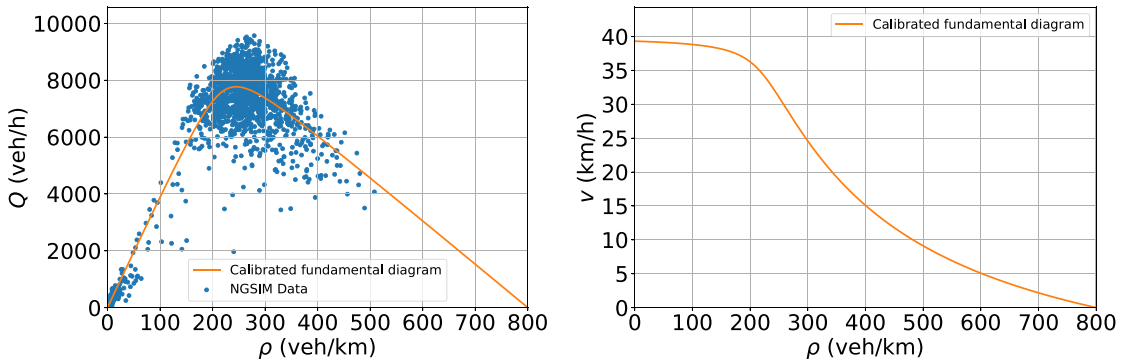
In the previous section, we evaluated the NO-based methods within a simulation environment. We now proceed to apply the trained neural operator to the ARZ system with a calibrated fundamental diagram using real traffic data. Specifically, we utilize the NGSIM dataset, which includes vehicle trajectory data collected in Emeryville, California, USA, on April 13, 2005. The dataset is segmented into 15-minute intervals, and we select data from 4:00 pm to 4:15 pm for calibrating the fundamental diagram. The chosen spatial–temporal domain is L = 500m and T = 700s. To derive the density and flow from the trajectory data, we reconstruct the NGSIM data using Eide's formula (Edie, 1963) and the same method (Zhao and Yu, 2023; Fan and Seibold, 2013; Yu et al., 2021). The spatial–temporal domain is partitioned into small grids with dimensions $\Delta x = 20$ m and $\Delta t = 15$ s. We adopt the three-parameter (ζ, κ, p) fundamental diagram to calibrate the NGSIM dataset, the three-parameter fundamental diagram is denoted by the following form

$$Q(\rho) = \zeta\left(a + \frac{(b-a)\rho}{\rho_m} - \sqrt{1 + \kappa^2\left(\frac{\rho}{\rho_m} - p\right)^2}\right), \tag{81}$$

where $a = \sqrt{1+\kappa^2 p^2}$, $b = \sqrt{1+\kappa^2(1-p)^2}$. The maximum density ρ_m is defined by

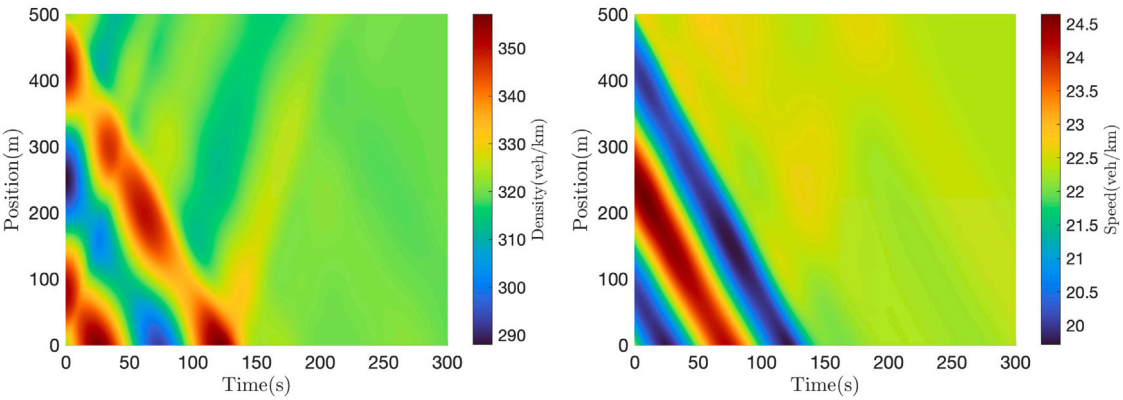
$$\rho_m = \frac{\text{number of lanes}}{\text{vehicle length} \times \text{safety factor}}, \tag{82}$$

where the vehicle length is 5 m and the safety factor is selected as 1.5. There are 6 lanes in the road segment of the NGSIM dataset, therefore, the maximum density is $\rho_m = 800$ veh/km. The calibrated fundamental diagram is shown in Fig. 15(a), the calibrated three parameters are $\zeta = 1339.38$, $\kappa = 16.53$, $p = 0.28$. And the corresponding density–velocity relation is illustrated in Fig. 15(b).



(a) Flow-density relation (b) Density-velocity relation

Fig. 15. The calibrated fundamental diagram from NGSIM dataset.



(a) Density (b) Speed

Fig. 16. NO method for calibrated ARZ model.

The fundamental diagram and relaxation parameter are calibrated for the ARZ model. The trained neural operator for backstepping kernels is then employed to derive the control law required to stabilize the calibrated ARZ system. For the calibrated ARZ system, the equilibrium density is set as 320 veh/km and the corresponding equilibrium speed is 22.3 km/h, as calculated by the three-parameters fundamental diagram. Initial conditions are configured with sinusoidal inputs to simulate stop-and-go traffic, while all other settings remain consistent with previous experiments. The results for the evolution of traffic states are presented in Fig. 16. The experimental results demonstrate that the trained neural operator effectively stabilizes the traffic system with the calibrated fundamental diagram using real traffic data. Traffic states converge to their equilibrium points within a finite time.

5. Conclusion

In this paper, we propose an operator learning framework for the boundary control of traffic systems. First, the ARZ PDE model is adopted to describe the spatial–temporal evolution of traffic density and speed. We first define the operator mapping from the model parameter, i.e., characteristic speed λ_2 to the backstepping control kernels K^w, K^v and then the directly map to a boundary control law $U(t)$. The neural operators using DeepONet are trained to approximate the two operator mappings. Subsequently, the Lyapunov analysis is conducted to derive the theoretical stability for the NO-approximated closed-loop system. To further extend our operator learning framework, we incorporate physical constraints into the neural operator, specifically the equations describing the mapping from the characteristic speed to backstepping kernels, and then the PINO is established.

The performance of the neural operator is assessed using both simulated and real traffic data. The NO-approximated kernels, NO-approximated control law, and PINO-approximated kernels are evaluated under consistent settings. The backstepping method is used as the baseline, with comparisons made to the PI controller and PINN-based controller. The results show that both the NO-approximated and PINO-approximated mappings achieve satisfactory accuracy and provide a significant computational speedup of 298 times compared to the backstepping controller, with only a 1% loss in accuracy. To evaluate the robustness of the proposed NO-based methods, various traffic conditions are tested. The NO-based methods demonstrate strong performance in stabilizing different

traffic scenarios. Additionally, using the NGSIM data to calibrate the fundamental diagram and applying it to real traffic systems shows that the NO method has significant potential for practical applications in freeway traffic control.

Future work should include extending the operator learning framework to explore the performance of NO-based methods in network traffic. Additionally, applying these methods to other traffic-related problems, such as traffic assignment and vehicle routing, would be a valuable area of investigation.

CRediT authorship contribution statement

Yihuai Zhang: Writing – original draft, Visualization, Software, Formal analysis, Conceptualization. **Ruiguo Zhong:** Visualization, Software. **Huan Yu:** Supervision, Methodology, Investigation, Conceptualization.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant 62203131.

Appendix A. Proof of Theorem 1

Taking time derivative along the trajectories of the system, plugging the system dynamics, we have

$$\begin{aligned}\dot{V}_k(t) = &-\nu V_k(t) + (r^2 - a)\hat{\beta}^2(0,t) - \mathrm{e}^{-\frac{\nu}{\lambda_1}L}\hat{\alpha}^2(L,t)\\ &+ \int_0^L 2a\frac{\mathrm{e}^{-\frac{\nu}{\lambda_2}x}}{\lambda_2}\hat{\beta}(x,t)\left(\lambda_2(\tilde{K}^w(x,0) + \tilde{K}^v(x,0))\tilde{v}(0,t) + (\lambda_1+\lambda_2)\tilde{K}^w(x,x)\tilde{w}(x,t)\right.\\ &+ \int_0^x (\lambda_2\tilde{K}^w_x(x,\xi) + \lambda_1\tilde{K}^w_\xi(x,\xi))\tilde{w}(\xi,t)d\xi\\ &+ \left.\int_0^x (\lambda_2\tilde{K}^v_x(x,\xi) + \lambda_2\tilde{K}^v_\xi(x,\xi))\tilde{v}(\xi,t)d\xi\right)dx, \end{aligned} \tag{A.1}$$

For the integral term, we take the norm and use the Young inequality and Cauchy–Schwarz inequality. Then combining the equivalent norm of the Lyapunov candidate, we get:

$$\begin{aligned}\int_0^L \left\|2a\frac{\mathrm{e}^{-\frac{\nu}{\lambda_2}x}}{\lambda_2}\hat{\beta}(x,t)\lambda_2(\tilde{K}^w(x,0) + \tilde{K}^v(x,0))\tilde{v}(0,t)\right\|dx &\le \int_0^L \left\|2a\epsilon\mathrm{e}^{-\frac{\nu}{\lambda_2}x}(\hat{\beta}^2(x,t) + \tilde{v}^2(0,t))\right\|dx\\ &\le \frac{2a\epsilon}{m_1}V_k(t) + 2aL\epsilon\hat{\beta}^2(0,t).\end{aligned} \tag{A.2}$$

Using the same method, we can easily get the results for the other terms of the Lyapunov candidate. For the second term, we have:

$$\begin{aligned}\int_0^L \left\|2a\frac{\mathrm{e}^{-\frac{\nu}{\lambda_2}x}}{\lambda_2}\hat{\beta}(x,t)(\lambda_1+\lambda_2)\tilde{K}^w(x,x)\tilde{w}(x,t)\right\|dx &\le \int_0^L \left\|2a\epsilon\frac{\lambda_1+\lambda_2}{\lambda_2}\mathrm{e}^{-\frac{\nu}{\lambda_2}x}(\hat{\beta}^2(x,t) + \tilde{w}^2(x,t))\right\|dx\\ &\le \frac{2a\epsilon(\lambda_1+\lambda_2)}{m_1\lambda_2}(1+\frac{1}{k_1})V_k(t).\end{aligned} \tag{A.3}$$

For the third term, we can get the following results:

$$\begin{aligned}&\int_0^L \left\|2a\frac{\mathrm{e}^{-\frac{\nu}{\lambda_2}x}}{\lambda_2}\hat{\beta}(x,t) + \int_0^x (\lambda_2\tilde{K}^w_x(x,\xi) + \lambda_1\tilde{K}^w_\xi(x,\xi))\tilde{w}(\xi,t)d\xi\right\|dx\\ &\le \int_0^L \left\|2a\epsilon\frac{\lambda_1+\lambda_2}{\lambda_2}\mathrm{e}^{-\frac{\nu}{\lambda_2}x}L(\hat{\beta}^2(x,t) + \tilde{w}^2(x,t))\right\|dx \le \frac{2a\epsilon(\lambda_1+\lambda_2)}{m_1\lambda_2}(1+\frac{1}{k_1})V_k(t).\end{aligned} \tag{A.4}$$

For the last term, we have:

$$\begin{aligned}\int_0^L \left\|2a\frac{\mathrm{e}^{-\frac{\nu}{\lambda_2}x}}{\lambda_2}\hat{\beta}(x,t) + \int_0^x (\lambda_2\tilde{K}^v_x(x,\xi) + \lambda_2\tilde{K}^v_\xi(x,\xi))\tilde{v}(\xi,t)d\xi\right\|dx &\le \int_0^L \left\|2a\epsilon\mathrm{e}^{-\frac{\nu}{\lambda_2}x}L(\hat{\beta}^2(x,t) + \tilde{v}^2(x,t))\right\|dx\\ &\le \frac{2a\epsilon L}{m_1}(1+\frac{1}{k_1})V_k(t).\end{aligned} \tag{A.5}$$

Using the bound of the four terms, we can derive the estimation of the Lyapunov candidate in (49).

Appendix B. Proof the practical stability of the traffic system

Based on the properties of the neural operators in Lemma 2, it can be found that the NO-approximated mapping $\hat{H}(\lambda_2)$ satisfies the following lemma:

Lemma 4. *For any $\epsilon > 0$, there exists a neural operator $\hat{\mathcal{H}}(\lambda_2)$ that can approximate the control law mapping in the spatial–temporal domain $(x,t) \in [0,L] \times \mathbb{R}^+$:*

$$\max_{\lambda_2 \in U} \left| \mathcal{H}\left(\lambda_2\right)(L) - \hat{\mathcal{H}}(\lambda_2)(L) \right| < \epsilon. \tag{B.1}$$

Proof. We know from Lemma 2 that the neural operator using DeepONet can approximate operator mapping within accuracy ϵ. We can apply Lemma 2 again to approximate the mapping from the characteristic speed λ_2 to $U(t)$. Thus, the lemma is proven. □

Applying the NO-approximated control law to the system (11)–(14), we get the target system as:

$$\partial_t \breve{\alpha}(x,t) + \lambda_1 \partial_x \breve{\alpha}(x,t) = 0, \tag{B.2}$$

$$\partial_t \breve{\beta}(x,t) - \lambda_2 \partial_x \breve{\beta}(x,t) = 0, \tag{B.3}$$

$$\breve{\alpha}(0,t) = -r\breve{\beta}(0,t), \tag{B.4}$$

$$\breve{\beta}(L,t) = \mathcal{H}\left(\lambda_2\right)(L,t) - \hat{\mathcal{H}}(\lambda_2)(L,t). \tag{B.5}$$

Compared with the target system in Yu and Krstic (2019), the system (B.2)–(B.5) is not strictly exponentially stable due to the approximation error of NO-approximated control law.

To prove the stability of the traffic system under the NO-approximated control law, we first give the definition of the local practical exponential stability of the traffic PDE system,

Definition 2 (*Local Practical Exponential Stability Bhan et al., 2024; Teel et al., 1999*)**.** For the ARZ traffic system whose control law is approximated by NO, the traffic system is said to be locally practically exponentially stable if the state satisfy

$$\|(\bar{\rho}(x,t), \bar{v}(x,t))\|_{L^2}^2 \leq \mathrm{e}^{-\mu t} \|(\bar{\rho}(x,0), \bar{v}(x,0))\|_{L^2}^2 + \kappa(\epsilon), \tag{B.6}$$

where $\mu > 0$ and $\kappa : \mathbb{R}^+ \to \mathbb{R}^+$ is of class $\mathcal{K}$ function with strictly increasing and $\kappa(0) = 0$ properties.

Thus, we have the following theorem for the NO-approximated control law:

Theorem 2. *The system (1)–(2) with boundary conditions (5)–(6) is locally practically exponentially stable under the NO-approximated control law $\hat{\mathcal{H}}(\lambda_2)(L,t)$ with initial conditions $\bar{\rho}(x,0), \bar{v}(x,0)$, such that*

$$\|(\bar{\rho}(x,t), \bar{v}(x,t))\|_{L^2}^2 \leq c_2 \mathrm{e}^{-\nu t} \|(\bar{\rho}(x,0), \bar{v}(x,0))\|_{L^2}^2 + \frac{a}{m_4 k_4} \mathrm{e}^{-\frac{\nu}{\lambda_2} L} \epsilon^2. \tag{B.7}$$

where $c_2 = \frac{m_3 n_4 k_3}{m_4 n_3 k_4}$, $m_3 > 0$, $m_4 > 0$, $n_3 > 0$, $n_4 > 0$, $k_3 > 0$, $k_4 > 0$. The control law in (55) is approximated by the neural operator $\hat{\mathcal{H}}(\lambda_2)(L,t)$ with accuracy ϵ in (B.1).

Proof. For the NO-approximated control law, using the Lyapunov candidate again to analyze the stability of the target system (B.2)–(B.5).

$$V_U(t) = \int_0^L \frac{\mathrm{e}^{-\frac{\nu}{\lambda_1} x}}{\lambda_1} \breve{\alpha}^2(x,t) + a \frac{\mathrm{e}^{-\frac{\nu}{\lambda_2} x}}{\lambda_2} \breve{\beta}^2(x,t) dx, \tag{B.8}$$

where the coefficients ν and a are the same as before. The states of the target system with NO-approximated control law $(\breve{\alpha}, \breve{\beta})$ still have equivalent L_2 norm with the system $(\tilde{w}, \tilde{v})$.

$$k_3 \|(\tilde{w}(x,t), \tilde{v}(x,t))\|_{L_2}^2 \leq \left\| (\breve{\alpha}(x,t), \breve{\beta}(x,t)) \right\|_{L_2}^2 \leq k_4 \|(\tilde{w}(x,t), \tilde{v}(x,t))\|_{L_2}^2, \tag{B.9}$$

where $k_3 > 0$ and $k_4 > 0$. The same is true for the Lyapunov functional, it has the following equivalent norm with the NO-approximated target system. There exist $m_3 > 0, m_4 > 0$,

$$m_3 \left\| (\breve{\alpha}(x,t), \breve{\beta}(x,t)) \right\|_{L^2}^2 \leq V_U(t) \leq m_4 \left\| (\breve{\alpha}(x,t), \breve{\beta}(x,t)) \right\|_{L^2}^2. \tag{B.10}$$

Taking time derivative along the trajectories, putting it into the system dynamics, and integrating by parts, we have:

$$\dot{V}_U(t) = -\nu V_U(t) + (r^2 - a)\breve{\beta}^2(0,t) + a\mathrm{e}^{-\frac{\nu}{\lambda_2} L} \breve{\beta}^2(L,t) - \mathrm{e}^{-\frac{\nu}{\lambda_1} L} \breve{\alpha}^2(L,t). \tag{B.11}$$

The term $a\mathrm{e}^{-\frac{\nu}{\lambda_2} L} \breve{\beta}^2(L,t)$ is equal to 0 in the ideal situation when the NO-approximated mapping achieves 100% accuracy of approximation which means that $\mathcal{H}\left(\lambda_2\right)(L) - \hat{\mathcal{H}}(\lambda_2)(L) = 0$ and we can easily get the exponential stability for the system. Here the mapping has the error ϵ. We take the $r^2 - a \leq 0$, and we get

$$V_U(t) \leq V_U(0)\mathrm{e}^{-\nu t} + a\mathrm{e}^{-\frac{\nu}{\lambda_2} L} \sup_{0 \leq \varsigma \leq t} (\mathcal{H}\left(\lambda_2\right)(L) - \hat{\mathcal{H}}(\lambda_2)(L))^2(L,\varsigma). \tag{B.12}$$

Using (B.10), we have

$$\|(\tilde{w}(x,t), \tilde{v}(x,t))\|_{L^2}^2 \leq \frac{m_3 k_3}{m_4 k_4} \mathrm{e}^{-\nu t} \|(\tilde{w}(x,0), \tilde{v}(x,0))\|_{L^2}^2 + \frac{a}{m_4 k_4} \mathrm{e}^{-\frac{\nu}{\lambda_2} L} \epsilon^2. \tag{B.13}$$

Thus, we have proved that the system (11)–(14) is locally practically exponentially stable. Using the equivalent norm

$$n_3\|(\bar{\rho}(x,t),\bar{v}(x,t))\|_{L_2}^2 \le \|(i\tilde{w}(x,t),\tilde{v}(x,t))\|_{L^2}^2 \le n_4\|(\bar{\rho}(x,t),\bar{v}(x,t))\|_{L_2}^2, \tag{B.14}$$

thus, we get

$$\|(\bar{\rho}(x,t),\bar{v}(x,t))\|_{L^2}^2 \le c_2 e^{-\nu t}\|(\bar{\rho}(x,0),\bar{v}(x,0))\|_{L^2}^2 + \frac{a}{m_4 k_4} e^{-\frac{\nu}{\lambda_2}L}\epsilon^2, \tag{B.15}$$

where $c_2 = \frac{m_3 n_4 k_3}{m_4 n_3 k_4}$. Therefore, the original system (1)–(2) with boundary conditions (5) and (6) is locally practically exponentially stable. This finish the proof of Theorem 2. □

References

Ahn, K., 1998. Microscopic Fuel Consumption and Emission Modeling (Ph.D. thesis). Virginia Tech.

Anfinsen, H., Aamo, O.M., 2019. Adaptive Control of Hyperbolic PDEs. In: Communications and Control Engineering, Springer International Publishing, Cham.

Auriol, J., Di Meglio, F., 2020. Robust output feedback stabilization for two heterodirectional linear coupled hyperbolic PDEs. Automatica 115, 108896.

Avedisov, S.S., Bansal, G., Orosz, G., 2020. Impacts of connected automated vehicles on freeway traffic patterns at different penetration levels. IEEE Trans. Intell. Transp. Syst. 23 (5), 4305–4318.

Aw, A., Rascle, M., 2000. Resurrection of "second order" models of traffic flow. SIAM J. Appl. Math. 60 (3), 916–938.

Bastin, G., Coron, J.-M., 2016. Stability and Boundary Stabilization of 1-D Hyperbolic Systems. In: Progress in Nonlinear Differential Equations and Their Applications, Vol. 88, Springer International Publishing, Cham.

Bekiaris-Liberis, N., Delis, A.I., 2021. PDE-based feedback control of freeway traffic flow via time-gap manipulation of ACC-equipped vehicles. IEEE Trans. Control Syst. Technol. 29 (1), 461–469.

Bellemans, T., De Schutter, B., De Moor, B., 2006. Model predictive control for ramp metering of motorway traffic: A case study. Control Eng. Pract. 14 (7), 757–767.

Belletti, F., Haziza, D., Gomes, G., Bayen, A.M., 2018. Expert level control of ramp metering based on multi-task deep reinforcement learning. IEEE Trans. Intell. Transp. Syst. 19 (4), 1198–1207.

Belletti, F., Huo, M., Litrico, X., Bayen, A.M., 2015. Prediction of traffic convective instability with spectral analysis of the Aw–Rascle–Zhang model. Phys. Lett. A 379 (38), 2319–2330.

Bhan, L., Shi, Y., Krstic, M., 2023. Operator learning for nonlinear adaptive control. In: Learning for Dynamics and Control Conference. PMLR, pp. 346–357.

Bhan, L., Shi, Y., Krstic, M., 2024. Neural operators for bypassing gain and control computations in pde backstepping. IEEE Trans. Autom. Control 69 (8), 5310–5325.

Burkhardt, M., Yu, H., Krstic, M., 2021. Stop-and-go suppression in two-class congested traffic. Automatica 125, 109381.

Carey, M., 2021. The cell transmission model with free-flow speeds varying over time or space. Transp. Res. B 147, 245–257.

Carlson, R.C., Papamichail, I., Papageorgiou, M., 2011. Local feedback-based mainstream traffic flow control on motorways using variable speed limits. IEEE Trans. Intell. Transp. Syst. 12 (4), 1261–1276.

Chen, T., Chen, H., 1995. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. IEEE Trans. Neural Netw. 6 (4), 911–917.

Čičić, M., Xiong, X., Jin, L., Johansson, K.H., 2021. Coordinating vehicle platoons for highway bottleneck decongestion and throughput improvement. IEEE Trans. Intell. Transp. Syst. 23 (7), 8959–8971.

Colombo, R.M., Groli, A., 2004. Minimising stop and go waves to optimise traffic flow. Appl. Math. Lett. 17 (6), 697–701.

Daganzo, C.F., 1994. The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. Transp. Res. B 28 (4), 269–287.

De Palma, A., Lindsey, R., 2011. Traffic congestion pricing methodologies and technologies. Transp. Res. C 19 (6), 1377–1399.

Delle Monache, M.L., Piccoli, B., Rossi, F., 2017. Traffic regulation via controlled speed limit. SIAM J. Control Optim. 55 (5), 2936–2958.

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 Netw. 153, 411–426.

Edie, L.C., 1963. Discussion of Traffic Stream Measurements and Definitions. Port of New York Authority.

Fan, S., Seibold, B., 2013. Data-fitted first-order traffic models and their second-order generalizations: Comparison by trajectory and sensor data. Transp. Res. Rec.: J. Transp. Res. Board 2391 (1), 32–43.

Ferrara, A., Sacone, S., Siri, S., 2015. Event-triggered model predictive schemes for freeway traffic control. Transp. Res. C 58, 554–567.

Ferrara, A., Sacone, S., Siri, S., 2016. Design of networked freeway traffic controllers based on event-triggered control concepts. Internat. J. Robust Nonlinear Control 26 (6), 1162–1183.

Flynn, M.R., Kasimov, A.R., Nave, J.-C., Rosales, R.R., Seibold, B., 2009. Self-sustained nonlinear waves in traffic flow. Phys. Rev. E 79 (5), 056113.

Godunov, S.K., Bohachevsky, I., 1959. Finite difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics. Mat. Sb. 47 (3), 271–306.

Gomes, G., Horowitz, R., 2006. Optimal freeway ramp metering using the asymmetric cell transmission model. Transp. Res. C 14 (4), 244–262.

Han, Y., Wang, M., Li, L., Roncoli, C., Gao, J., Liu, P., 2022. A physics-informed reinforcement learning-based strategy for local and coordinated ramp metering. Transp. Res. C 137, 103584.

Hoogendoorn, S., Van Kooten, J., Adams, R., 2016. Lessons learned from field operational test of integrated network management in Amsterdam. Transp. Res. Rec. 2554 (1), 111–119.

Horowitz, R., May, A., Skabardonis, A., Varaiya, P., Zhang, M., Gomes, G., Munoz, L., Sun, X., Sun, D., 2005. Design, Field Implementation and Evaluation of Adaptive Ramp Metering Algorithms. Technical Rep. PATH No.

Karafyllis, I., Krstic, M., 2019. Input-to-State Stability for PDEs. In: Communications and Control Engineering, Springer International Publishing, Cham.

Karafyllis, I., Papageorgiou, M., 2019. Feedback control of scalar conservation laws with application to density control in freeways by means of variable speed limits. Automatica 105, 228–236.

Karniadakis, G.E., Kevrekidis, I.G., Lu, L., Perdikaris, P., Wang, S., Yang, L., 2021. Physics-informed machine learning. Nat. Rev. Phys. 3 (6), 422–440.

Kotsialos, A., Papageorgiou, M., Diakaki, C., Pavlis, Y., Middelham, F., 2002. Traffic flow modeling of large-scale motorway networks using the macroscopic modeling tool METANET. IEEE Trans. Intell. Transp. Syst. 3 (4), 282–292.

Kovachki, N., Li, Z., Liu, B., Azizzadenesheli, K., Bhattacharya, K., Stuart, A., Anandkumar, A., 2023. Neural operator: Learning maps between function spaces. arXiv:2108.08481 [cs, math].

Krstic, M., Smyshlyaev, A., 2008a. Backstepping boundary control for first-order hyperbolic PDEs and application to systems with actuator and sensor delays. Systems Control Lett. 57 (9), 750–758.

Krstic, M., Smyshlyaev, A., 2008b. Boundary control of PDEs: a course on backstepping designs. In: Advances in design and control, Vol. 16, Siam, Soc. for Industrial and Applied Mathematics, Philadelphia, Pa.
Lamare, P.-O., Di Meglio, F., 2016. Adding an integrator to backstepping: Output disturbances rejection for linear hyperbolic systems. In: 2016 American Control Conference (ACC). IEEE, Boston, MA, USA, pp. 3422–3428.
Lee, J.W., Wang, H., Jang, K., Hayat, A., Bunting, M., Alanqary, A., Barbour, W., Fu, Z., Gong, X., Gunter, G., et al., 2024. Traffic control via connected and automated vehicles: An open-road field experiment with 100 CAVs. arXiv preprint arXiv:2402.17043.
Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., Anandkumar, A., 2021. Fourier neural operator for parametric partial differential equations. arXiv:2010.08895 [cs, math].
Lighthill, M.J., Whitham, G.B., 1955. On kinematic waves II. A theory of traffic flow on long crowded roads. Proc. R. Soc. Lond. Ser. A 229 (1178), 317–345.
Lin, C., Li, Z., Lu, L., Cai, S., Maxey, M., Karniadakis, G.E., 2021. Operator learning for predicting multiscale bubble growth dynamics. J. Chem. Phys. 154 (10).
Liu, S., Hellendoorn, H., De Schutter, B., 2016. Model predictive control for freeway networks based on multi-class traffic flow and emission models. IEEE Trans. Intell. Transp. Syst. 18 (2), 306–320.
Lu, L., Jin, P., Pang, G., Zhang, Z., Karniadakis, G.E., 2021a. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. Nat. Mach. Intell. 3 (3), 218–229.
Lu, L., Meng, X., Cai, S., Mao, Z., Goswami, S., Zhang, Z., Karniadakis, G.E., 2022. A comprehensive and fair comparison of two neural operators (with practical extensions) based on fair data. Comput. Methods Appl. Mech. Engrg. 393, 114778.
Lu, L., Meng, X., Mao, Z., Karniadakis, G.E., 2021b. DeepXDE: A deep learning library for solving differential equations. SIAM Rev. 63 (1), 208–228.
Mohan, R., Ramadurai, G., 2013. State-of-the art of macroscopic traffic flow modelling. Int. J. Adv. Eng. Sci. Appl. Math. 5, 158–176.
Müller, E.R., Carlson, R.C., Kraus, W., Papageorgiou, M., 2015. Microsimulation analysis of practical aspects of traffic control with variable speed limits. IEEE Trans. Intell. Transp. Syst. 16 (1), 512–523.
Muralidharan, A., Dervisoglu, G., Horowitz, R., 2009. Freeway traffic flow simulation using the link node cell transmission model. In: 2009 American Control Conference. IEEE, pp. 2916–2921.
Muralidharan, A., Horowitz, R., 2012. Optimal control of freeway networks based on the link node cell transmission model. In: 2012 American Control Conference. ACC, IEEE, pp. 5769–5774.
Muralidharan, A., Horowitz, R., 2015. Computationally efficient model predictive control of freeway networks. Transp. Res. C 58, 532–553.
Pan, T., Guo, R., Lam, W.H., Zhong, R., Wang, W., He, B., 2021. Integrated optimal control strategies for freeway traffic mixed with connected automated vehicles: A model-based reinforcement learning approach. Transp. Res. C 123, 102987.
Papageorgiou, M., Hadj-Salem, H., Blosseville, J.-M., et al., 1991. ALINEA: A local feedback control law for on-ramp metering. Transp. Res. Rec. 1320 (1), 58–67.
Papamichail, I., Papageorgiou, M., Vong, V., Gaffney, J., 2010. Heuristic ramp-metering coordination strategy implemented at monash freeway, australia. Transp. Res. Rec. 2178 (1), 10–20.
Pathak, J., Subramanian, S., Harrington, P., Raja, S., Chattopadhyay, A., Mardani, M., Kurth, T., Hall, D., Li, Z., Azizzadenesheli, K., et al., 2022. Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators. arXiv preprint arXiv:2202.11214.
Qi, J., Mo, S., Krstic, M., 2023. Delay-compensated distributed PDE control of traffic with connected/Automated vehicles. IEEE Trans. Autom. Control 68 (4), 2229–2244.
Raissi, M., Perdikaris, P., Karniadakis, G.E., 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Comput. Phys. 378, 686–707.
Reilly, J., Bayen, A.M., 2015. Distributed optimization for shared state systems: Applications to decentralized freeway control via subnetwork splitting. IEEE Trans. Intell. Transp. Syst. 16 (6), 3465–3472.
Richards, P.I., 1956. Shock waves on the highway. Oper. Res. 4 (1), 42–51.
Schönhof, M., Helbing, D., 2007. Empirical features of congested traffic states and their implications for traffic modeling. Transp. Sci. 41 (2), 135–166.
Shi, Y., Li, Z., Yu, H., Steeves, D., Anandkumar, A., Krstic, M., 2022b. Machine learning accelerated PDE backstepping observers. In: 2022 IEEE 61st Conference on Decision and Control (CDC). IEEE, Cancun, Mexico, pp. 5423–5428.
Shi, R., Mo, Z., Huang, K., Di, X., Du, Q., 2022a. A physics-informed deep learning paradigm for traffic state and fundamental diagram estimation. IEEE Trans. Intell. Transp. Syst. 23 (8), 11688–11698.
Siri, S., Pasquale, C., Sacone, S., Ferrara, A., 2021. Freeway traffic control: A survey. Automatica 130, 109655.
Sun, L., Gao, H., Pan, S., Wang, J.-X., 2020. Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data. Comput. Methods Appl. Mech. Engrg. 361, 112732.
Teel, A.R., Peuteman, J., Aeyels, D., 1999. Semi-global practical asymptotic stability and averaging. Systems Control Lett. 37 (5), 329–334.
Treiber, M., Kesting, A., 2013. Traffic Flow Dynamics: Data, Models and Simulation. Springer Berlin Heidelberg, Berlin, Heidelberg.
Vazquez, R., Krstic, M., Coron, J.-M., 2011. Backstepping boundary stabilization and state estimation of a 2×2 linear hyperbolic system. In: IEEE Conference on Decision and Control and European Control Conference. IEEE, Orlando, FL, USA, pp. 4937–4942.
Wang, Y., Kosmatopoulos, E.B., Papageorgiou, M., Papamichail, I., 2014. Local ramp metering in the presence of a distant downstream bottleneck: Theoretical analysis and simulation study. IEEE Trans. Intell. Transp. Syst. 15 (5), 2024–2039.
Wang, S., Teng, Y., Perdikaris, P., 2021a. Understanding and mitigating gradient flow pathologies in physics-informed neural networks. SIAM J. Sci. Comput. 43 (5), A3055–A3081.
Wang, S., Wang, H., Perdikaris, P., 2021b. Learning the solution operator of parametric partial differential equations with physics-informed DeepONets. Sci. Adv. 7 (40), 1–9.
Wang, Y., Yu, X., Guo, J., Papamichail, I., Papageorgiou, M., Zhang, L., Hu, S., Li, Y., Sun, J., 2022b. Macroscopic traffic flow modelling of large-scale freeway networks with field data verification: State-of-the-art review, benchmarking framework, and case studies using METANET. Transp. Res. C 145, 103904.
Wang, S., Yu, X., Perdikaris, P., 2022a. When and why PINNs fail to train: A neural tangent kernel perspective. J. Comput. Phys. 449, 110768.
Wen, G., Li, Z., Azizzadenesheli, K., Anandkumar, A., Benson, S.M., 2022. U-FNO—An enhanced Fourier neural operator-based deep-learning model for multiphase flow. Adv. Water Resour. 163, 104180.
Whitham, G.B., 2011. Linear and Nonlinear Waves. John Wiley & Sons.
You, H., Zhang, Q., Ross, C.J., Lee, C.-H., Yu, Y., 2022. Learning deep implicit Fourier neural operators (IFNOs) with applications to heterogeneous material modeling. Comput. Methods Appl. Mech. Engrg. 398, 115296.
Yu, H., Auriol, J., Krstic, M., 2022a. Simultaneous downstream and upstream output-feedback stabilization of cascaded freeway traffic. Automatica 136, 110044.
Yu, H., Gan, Q., Bayen, A., Krstic, M., 2021. PDE traffic observer validated on freeway data. IEEE Trans. Control Syst. Technol. 29 (3), 1048–1060.
Yu, H., Krstic, M., 2019. Traffic congestion control for Aw–Rascle–Zhang model. Automatica 100, 38–51.
Yu, H., Krstic, M., 2022. Traffic Congestion Control by PDE Backstepping. In: Systems & Control: Foundations & Applications, Springer International Publishing, Cham.
Yu, H., Park, S., Bayen, A., Moura, S., Krstic, M., 2022b. Reinforcement learning versus PDE backstepping and PI control for congested freeway traffic. IEEE Trans. Control Syst. Technol. 30 (4), 1595–1611.
Zhang, H.M., 2002. A non-equilibrium traffic model devoid of gas-like behavior. Transp. Res. B (36), 275–290.
Zhang, Y., Auriol, J., Yu, H., 2024b. Robust boundary stabilization of stochastic hyperbolic PDEs. In: 2024 American Control Conference. ACC, IEEE, pp. 5333–5338.

Zhang, J., Mao, S., Yang, L., Ma, W., Li, S., Gao, Z., 2024a. Physics-informed deep learning for traffic state estimation based on the traffic flow model and computational graph method. Inf. Fusion 101, 101971.
Zhang, L., Prieur, C., Qiao, J., 2019. PI boundary control of linear hyperbolic balance laws with stabilization of ARZ traffic flow models. Systems Control Lett. 123, 85–91.
Zhang, Z., Wing Tat, L., Schaeffer, H., 2023. Belnet: Basis enhanced learning, a mesh-free neural operator. Proc. R. Soc. A 479 (2276), 20230043.
Zhang, Y., Yu, H., Auriol, J., Pereira, M., 2024c. Mean-square exponential stabilization of mixed-autonomy traffic PDE system. Automatica 170, 111859.
Zhang, Y., Zhong, R., Yu, H., 2024d. Neural operators for boundary stabilization of stop-and-go traffic. In: 6th Annual Learning for Dynamics & Control Conference. PMLR, pp. 554–565.
Zhao, C., Yu, H., 2023. Observer-informed deep learning for traffic state estimation with boundary sensing. IEEE Trans. Intell. Transp. Syst. 25 (2), 1602–1611.
Zhao, C., Yu, H., Molnar, T.G., 2023. Safety-critical traffic control by connected automated vehicles. Transp. Res. C 154, 104230.
Zheng, Y., Wang, J., Li, K., 2020. Smoothing traffic flow via control of autonomous vehicles. IEEE Internet Things J. 7 (5), 3882–3896.