Deep Koopman Traffic Modeling for Freeway Ramp Metering

Chuanye Gu, Tao Zhou, and Changzhi Wu

***Abstract*— Ramp metering has been considered as one of the most effective approaches of dealing with the traffic congestion on the freeways. The modelling of the freeway traffic flow dynamics is challenging because of its non-linearity and uncertainty. Recently, Koopman operator, which transfers a non-linear system to a linear system in an infinite-dimensional space, has been studied for modelling complex dynamics. In this paper, we propose a data-driven modelling approach based on neural networks, denoted by deep Koopman model, to learn a finite-dimensional approximation of the Koopman operator. To consider the sequential relations of the ramps and main roads on the freeway, a long short-term memory network is applied. Furthermore, a model predictive controller with the trained deep Koopman model is proposed for the real-time control of the ramp metering on the freeway. To validate the performance of the proposed approach, experiments based on the simulation in the traffic simulation software Simulation of Urban MObility (SUMO) environment are conducted. The results demonstrate the effectiveness of the proposed approach on both the dynamics prediction and the real-time control of the ramp metering.**

***Index Terms*— Freeway traffic modeling, neural network, deep Koopman model, ramp metering, model predictive control.**

I. Introduction

SINCE the rapid development of urbanization and the significant increase of private vehicle ownership, traffic congestion has been an obstinate and long-standing problem not only for the urban traffic in big cities but also for freeways. Traffic congestion on freeways could trigger serious consequences, such as the extensive time delays, opportunity cost, and fuel consumption. Ramp metering is one of the effective strategies to mitigate traffic jams on freeways through controlling the inflows of vehicles from on-ramps to the freeway mainline. Over the past decades, many studies have been conducted in the field of traffic flow modelling and ramp metering control.

Traditionally, the control strategies of ramp metering control are designed as fixed or traffic responsive. The fixed control approach is easy to implement, but the main disadvantage is that the fixed parameters do not respond to the changing traffic states, which greatly reduces the performance of approach [1]. The traffic responsive strategies can be divided into as feedback-based and model-based methods. The most widely used feedback-based approaches, the ALINEA [2] and its variants [3], [4], [5], aim at regulating the occupancy approaching a predefined target value. However, these methods control the metering rate according to traffic conditions passively, which greatly limits the performance of approaches, especially in a rapidly changing traffic environment. Furthermore, the control parameters in the controllers depend on human prior knowledge [1]. On the other hand, the model-based approaches optimize the control strategies based on the conventional data collected by loop detectors [6] (such as flow, density, speed). The model predictive control (MPC) [7] framework considering the interaction between ramp control and future traffic conditions is often used to predict the traffic evolution and realize the active traffic control [8]. However, the models used to describe traffic dynamics may lead to nonlinear control problems, which brings difficulties in finding the optimal metering rate [9], [10], [11], [12]. Some studies [6], [13], [14], [15] propose the cell transmission model (CTM)-based approaches, which simplify the macroscopic fundamental diagrams with a piecewise-point linear model. Because of reformulating as a linear framework, these approaches could obtain optimal solutions. However, such a simplification could result in inaccurate modeling of non-linear dynamics. Therefore, it is still challenging to model the non-linear dynamics system in ramp metering problem.

With the development of big data technologies in recent years, real-time traffic data can be obtained through the sensors deployed on vehicles and road infrastructures. Compared with the conventional loop detector data, real-time sensory data has the ability to better describe the underlying traffic dynamics [18], [19], [20]. Machine learning techniques, such as deep learning (DL) and reinforcement learning (RL), have been applied to learn the freeway traffic flow dynamics and ramp metering controlling strategies [21], [22], [23], [24]. However, the deep neural network-based traffic models in these studies suffer from a lack of interpretability [25]. Therefore, they might have unknown sensitive modes and be influenced by the model uncertainty. More importantly, due to the non-linearity of the activation functions in the deep neural networks, it is hard to integrate the learned dynamical models in the design of well-adjusted controllers, such as MPC and Linear Quadratic Regulator (LQR) [27]. Thus, massive data is required to train a

Manuscript received 15 January 2022; revised 21 July 2022, 14 November 2022, and 5 February 2023; accepted 20 February 2023. This work was supported in part by NSFC under Grant 11991020; in part by the Natural Science Foundation of Guangdong under Grant 2022A1515011761; in part by the Australia Research Council; in part by WA Mainroads; and in part by NSW Roads and Maritime Service, Australia. The Associate Editor for this article was I. Papamichail. *(Corresponding author: Changzhi Wu.)*

Chuanye Gu and Changzhi Wu are with the School of Management, Guangzhou University, Guangzhou 510006, China (e-mail: chuanye.gu@gzhu.edu.cn; changzhiwu@gzhu.edu.cn).

Tao Zhou is with the Institute for Intelligent Systems Research and Innovation (IISRI), Deakin University, Geelong, VIC 3217, Australia (e-mail: tao.zhou@deakin.edu.au).

Digital Object Identifier 10.1109/TITS.2023.3248649

deep neural network in reinforcement learning. Usually, such massive data can only be accessed through simulator which might result the control policy to be trapped into an inaccurate environment. To address this challenge, a physics-informed reinforcement learning method is developed in [26] through combination of historic data with synthetic data. However, a direct non-linear model is still absent from determination of the control policy.

Recently, the Koopman operator theory, which provides a data-driven framework for learning non-linear dynamical systems with the theoretical guarantee, has drawn a lot of attentions from researchers in the field of control [25], [28]. The Koopman operator is an invariant linear operator acting on the infinite dimensional lifted observable space of the state of the dynamical system [29], [30], [31]. The spectral properties of Koopman operator are helpful to describe the intrinsic properties of complex system dynamics by decomposing it into spatial-temporal coherent structures [31]. The Koopman operator could formulate a linear dynamical system instead of a black-box mapping of deep neural networks, which is more suitable to be used with controllers such as the linear MPC [25]. Therefore, the idea of the Koopman operator is theoretically widely applicable and useful for modelling dynamical systems that can not be accurately approximated by local linearized models [31].

Various approaches have been proposed to approximate the properties of the Koopman operator including the Koopman eigenvalues, eigenfunctions and modes. Three main classic algorithms that approximate these properties with finite-dimensional matrix have been widely studied: Ulam Galerkin Method [32], [33], Generalized Laplace Analysis (GLA) [34], [35], [36], and Dynamic mode Decomposition (DMD) [37], [38], [39]. All these algorithms are data-driven and require no explicit formulas, while none of them can approximate all properties of the Koopman operator. For example, the Ulam Galerkin Method can be applied to approximate directly both the Koopman eigenfunctions and eigenvalues [33], while DMD is used to approximate the Koopman modes and eigenvalues [37], [39]. Despite this disadvantage, they have been successfully applied for the analysis of non-linear dynamics, such as fluid flows [38], [40], buildings and power systems [41], [42], and traffic management [43], [44]. Based on the DMD method, an extended dynamic mode decomposition algorithm (EDMD) [31] has been proposed to approximate all three properties of Koopman operator with a dataset of snapshot pairs and a dictionary of observable functions. In EDMD, the observable functions that map the system state into a lifted space are normally manually selected. In studies [31], [45], the standard procedure of function selection is to choose a commonly used observable function, such as the radial basis function or the polynomial function. However, such approach does not take any empirical information or data of dynamics into consideration.

Deep neural networks have been demonstrated as an alternative approximator of the Koopman operator, which directly learn the observable functions [25]. Inspired by these works, we propose a Deep Koopman method, a variant of the EDMD, which uses neural networks to approximate the Koopman operator with finite-dimensional feature space. This is different from the aforementioned deep learning based traffic models. In [21], [23], and [24], neural networks are adopted to model and predict the traffic dynamics directly. More specifically, an encoder-decoder network is designed to learn the representation of the system state in the lifted space as well as the mapping process of converting features from the lifted space back to its original space. The long short-term memory (LSTM) unit is utilized to process the sequential data of the traffic flow on the freeway. To realize a reliable real-time controlling of the ramp metering on the freeway, we design a linear MPC controller to produce an online optimal control with the objective of maximizing the total throughput of leaving vehicles within a time frame. A few studies of combining the Koopman operator and the MPC have been conducted [25], [46]. In this paper, the simulation study is conducted in SUMO (Simulation of Urban MObility) environment due to its real-time visualization, the inclusion of various vehicle behaviour models, and the supportive built-on computational frameworks for DL.

The contributions of our paper are presented as follows:

i) A data-driven Deep Koopman model is designed to approximate the freeway traffic flow dynamics on the mainline and ramps. Compared to the EDMD, the Deep Koopman model formulates and learns appropriate observable functions and the reconstruction of the features of the traffic dynamics through an encoder-decoder neural network. Moreover, the Deep Koopman model keeps the linear structure of the dynamics system.

ii) A real-time ramp metering controller for the freeway traffic is developed by integrating the MPC method with the pre-trained Deep Koopman model for optimizing the total throughput of vehicles, denoted by MPC-Deep-Koopman. In the MPC-Deep-Koopman model, the encoder works as a non-linear function mapping the original system state into the lifted space at each decision time step. A linear framework of dynamics is then formulated based on the lifted system state and the controls over the MPC prediction horizon. Finally, the decoder reconstructs the system state from the lifted space to the primal system state.

iii) A series of simulation studies are carried out to validate the effectiveness of the proposed approach for the ramp metering control problem. The Deep Koopman model shows a competitive performance on the learning of dynamical systems as well as the controlling of ramp meters.

The rest of the paper is organized as follows: Section II presents the modelling of the dynamics of the freeway traffic flow and the ramp metering control problem. Section III develops the Deep Koopman model for the freeway traffic flow dynamics and Section IV describes the MPC controller with the dynamics system approximated by the Deep Koopman model. Numerical studies and conclusions are presented in Section V and Section VI, respectively.

II. The Freeway Traffic Model and Ramp Metering Controls

We consider a multi-lane freeway road consisting of M main road segments $\mathcal{M} = \{m_1 \ \dots \ m_M\}$, N on-ramps $\mathcal{N} = \{n_1 \ \dots \ n_N\}$, and L off-ramps $\mathcal{L} = \{l_1 \ \dots \ l_L\}$. Fig. 1(a) illustrates the freeway network, where each on-ramp has a

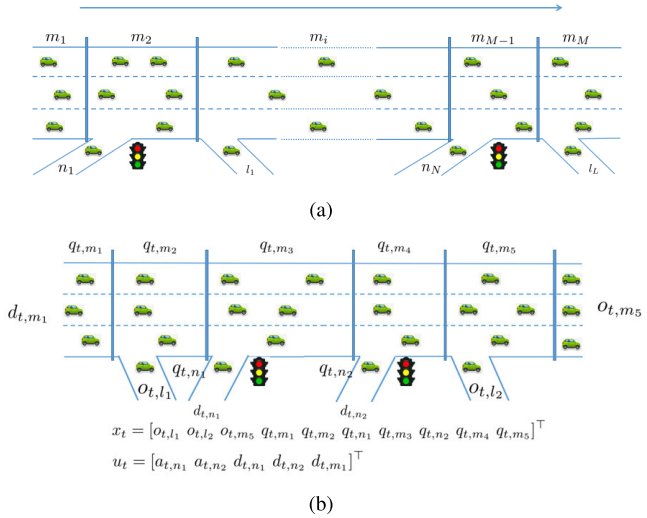


Fig. 1. Schematic diagram of ramp metering for a freeway (a) and an example of describing the state x_t and control u_t with ramp metering (b).

ramp metering controller adjusting the traffic flow merging into the main road. At each time step $t \in \{0\ 1\ \dots\ T\}$, the state $x_t \in \mathcal{X} \subseteq \mathbb{R}^n$ of the freeway traffic is defined by the vehicle out-flow through each off-ramp $o_{t,i}, i \in \mathcal{L}$ and the main road $o_{t,i}, i = m_M$, the vehicle queue length at each on-ramp $q_{t,i}, i \in \mathcal{N}$, and the number of vehicles on each main road segment $q_{t,i}, i \in \mathcal{M}$, where $n = M + N + L + 1$. The sequence of state features should be consistent with the flow direction of the main road segments and ramps, such as Fig. 1(b). We define the sequence of main road segments and on-ramps as $\mathcal{J} = \{j_1\ \dots\ j_J\}$, where $j_j \in \mathcal{M} \cup \mathcal{N}$. The decision variables of the ramp metering control are defined as the proportion of green light duration over the decision interval t for each on-ramp, denoted by $a_{t,i}, i \in \mathcal{N}$. Except for the features in the traffic state x_t, the estimated in-flow demand of vehicles for each on-ramp would also affect the control variables $a_{t,i}$. Therefore, we also include the demand of each on-ramp, $d_{t,i}, i \in \mathcal{N}$ and in-flow of main road $d_{t,i}, i = m_1$ as a fixed variable in the system control input $u_t \in \mathcal{U} \subseteq \mathbb{R}^m$, where $m = 2N + 1$. In this section, we describe the dynamics system of the freeway traffic as a discrete control model:

$$\begin{aligned} x_{t+1} &= f(x_t, u_t), \\ x_t &= [o_{t,l_1}\ \dots\ o_{t,l_L}\ o_{t,m_M}\ q_{t,j_1}\ \dots\ q_{t,j_J}]^\top, \\ u_t &= [a_{t,n_1}\ \dots\ a_{t,n_N}\ d_{t,n_1}\ \dots\ d_{t,n_N}\ d_{t,m_1}]^\top, \end{aligned} \tag{1}$$

Based on the nonlinear system (1), we define an optimal control problem as follows:

$$\max_{u_t, t=0,\dots,T-1} \mathrm{T} = \sum_{t=0}^{T} v_t^\top x_t - \lambda \sum_{t=0}^{T-2} ||u_{t+1} - u_t||^2, \tag{2}$$

$$\text{s.t. } x_{t+1} = f(x_t, u_t), t = 0, \dots, T-1, \tag{3}$$

$$x_{min} \le x_t \le x_{max}, t = 0, \dots, T, \tag{4}$$

$$u_{min} \le u_t \le u_{max}, t = 0, \dots, T-1, \tag{5}$$

where the first linear part in (2) denotes the throughput of the freeway road including vehicles leaving through the main road and the off-ramps, where $v_t \in \mathbb{R}^n$. Let $v_{t,i} = 1, i = 1, \dots, L+1$, others are 0. In our design of the ramp metering control, we anticipate the control variables to be as stable as possible. This is due to that a stable control strategy is expected to be applied to various scenarios rather than changing significantly for different demands. Based on this considerations, travelers waiting on on-ramps would be less confused. Therefore, the second term in the objective function (2) is a penalty term that penalizes the variation of control variables within two consecutive time steps, and the coefficient λ is a pre-defined hyperparameter adjusting the weight of penalty. (4) and (5) define the upper and lower bounds of state variables x and control inputs u, respectively.

One of the main challenges of resolving the problem (2) - (5) is the modelling of the non-linear dynamics system presented in (3). Thus, we introduce a Deep Koopman model to learn a linear approximation with convergence guarantees of the non-linear dynamics system of the freeway traffic based on the Koopman operator.

III. Deep Koopman Model for the Freeway Traffic Flow Dynamics

In this section, we first introduce the forced system described using the Koopman operator. Then, we outline the extended dynamic mode decomposition (EDMD) method developed based on the Koopman operator theory. Finally, a data-driven Deep Koopman model is presented to construct an approximation of the appropriate observable functions.

A. The Koopman Operator With Forced System

The Koopman operator is originally developed to describe the inherent properties of an uncontrolled non-linear dynamical system through a linear dynamical evolution. With a slight variation, we can employ the Koopman operator for the controlled dynamical system. In this paper, we consider a non-linear dynamical system for the freeway traffic flow:

$$x_{t+1} = f(x_t, u_t). \tag{6}$$

Equation (6) is a discrete-time freeway traffic flow system that evolves in the light of an unknown nonlinear law, where $x_t \in \mathcal{X}$ defined as Equation (1) is the state variable at time step t; $u_t \in \mathcal{U}$ defined as Equation (1) is the control input variable at time step t. Let $\mathbf{u}_t = (u_l)_{l=t}^{\infty} \in \overline{\mathcal{U}}$ and $\mathcal{G}\mathbf{u}_t = \mathbf{u}_{t+1}$ with $\mathcal{G}$ being the left shift operator, where $\overline{\mathcal{U}}$ denotes the space of all sequences $(u_l)_{l=0}^{\infty}$ with $u_l \in \mathcal{U}$. Then, we define the Koopman operator of the dynamics on the extended state $[x, \mathbf{u}]$ as follows:

$$\mathcal{K}g(x_t, \mathbf{u}_t) = g(x_{t+1}, \mathbf{u}_{t+1}) = g(f(x_t, u_t), \mathcal{G}\mathbf{u}_t), \tag{7}$$

where $\mathcal{K} : \mathcal{H} \to \mathcal{H}$ denotes the Koopman operator, $g : \mathcal{X} \times \overline{\mathcal{U}} \to \mathbb{R}^{\mathrm{N}}$, $g \in \mathcal{H}$ is an observable function, $\mathcal{H}$ is a Banach space. The Koopman operator $\mathcal{K}$ is a composition map of g and f. Note that, $\mathcal{K}$ is infinite-dimensional and linear even though the dynamics $f(\cdot)$ is non-linear. Due to the linear characteristics of $\mathcal{K}$, we can have the eigendecomposition of $\mathcal{K}$ in the standard form:

$$\mathcal{K}\varphi_j(x_t, \mathbf{u}_t) = \lambda_j \varphi_j(x_t, \mathbf{u}_t), j = 1, 2, \dots \tag{8}$$

where $\varphi_j(x_t, \mathbf{u}_t)$ and λ_j are the eigenfunction and the associated eigenvalue of $\mathcal{K}$, respectively. Obviously, the operator is spanned by eigenfunctions that are governed by the state

and the control input. A vector-valued observable g can be rewritten in terms of these eigenfunctions φ_j as:

$$g(x_t, \mathbf{u}_t) = (\mathcal{K}^t g)(x_0, \mathbf{u}_0) = \sum_{j=1}^{\infty} \lambda_j^t \varphi_j(x_0, \mathbf{u}_0) v_j, \tag{9}$$

where $\mathcal{K}^t$ denotes t times the Koopman operator and v_j is the Koopman mode [37] corresponding to the eigenvalue λ_j and x_0 is the initial state. Noted that not all vector-valued observables can be formulated as Equation (9). For some systematic observables, additional parts may be required to explain the rest of the spectrum of Koopman operators, as shown in [47], and [48].

B. Extended Dynamic Mode Decomposition

The extended dynamic mode decomposition (EDMD) [31] is an approach of finding the finite-dimensional approximation of the Koopman operator. The EDMD uses a variety of basis functions, such as radial basis functions (RBF) with different kernel centers and widths, to represent the observable functions and the least square method to calculate a finite-dimensional approximation $\mathbf{K} \in \mathbb{R}^{\mathrm{N}\times\mathrm{N}}$ of $\mathcal{K}$. Note it is impossible if we use the least square method to compute an approximation of the Koopman operator in a finite time because $[x, \mathbf{u}]$ is in general infinite-dimensional. To obtain the finite-dimensional approximation $\mathbf{K}$, we choose the lifting function g_i in a special way for any $i = \{1, 2, \ldots, \mathrm{N}\}$, which is written of the form

$$g_i(x_t, \mathbf{u}_t) = \psi_i(x_t) + h_i(\mathbf{u}_t), \tag{10}$$

where $\psi_i : \mathcal{X} \to \mathbb{R}$ is in general nonlinear but $h_i : \overline{\mathcal{U}} \to \mathbb{R}$ is linear. Without loss of generality, we can assume that the vector-valued observable function $g = [g_1 \ \ldots \ g_{\mathrm{N}}]^\top$ is of the form

$$g(x_t, \mathbf{u}_t) = [\psi^\top(x_t) \ \widetilde{u}_t^\top]^\top, \tag{11}$$

where $\psi(x_t) = [\psi_1(x_t) \ \psi_2(x_t) \ \ldots \ \psi_l(x_t)]^\top$ with $l = \mathrm{N} - m$, and $\widetilde{u}_t \in \mathbb{R}^m$ is the first component of the sequence $\mathbf{u}_t$, In order not to abuse the notation, we use u_t to denote $\widetilde{u}_t$ throughout the paper. Based on Equations (7) and (11), it gives rise

$$\mathcal{K}g(x_t, \mathbf{u}_t) = \mathbf{K}[\psi^\top(x_t) \ u_t^\top]^\top + r(x_t, u_t), \tag{12}$$

where the function $r(x_t, u_t)$ is a residual term that describes the gap between the N−dimensional approximation of the observable space and the actual lifted space of the Koopman operator. We optimize the cost function $||r(x_t, u_t)||_2^2$ to determine $\mathbf{K}$. Because we do not care about the predictive future value of control sequence, we can disregard the last m components of $[\psi^\top(x_{t+1}) \ u_{t+1}^\top]^\top$. Let $\Psi(x_t) = [\psi^\top(x_t) \ u_t^\top]^\top$, we rewrite the observables dynamics with the Koopman operator as:

$$\begin{aligned} \psi(x_{t+1}) &= \mathrm{A}\Psi(x_t), \\ \widetilde{x}_t &= \mathcal{C}\psi(x_t), \end{aligned} \tag{13}$$

where $\mathrm{A} = [\mathcal{A} \ \mathcal{B}] \in \mathbb{R}^{l\times\mathrm{N}}$, $\mathcal{A} \in \mathbb{R}^{l\times l}$, $\mathcal{B} \in \mathbb{R}^{l\times m}$. A is the former l rows of $\mathbf{K}$, and $\mathcal{C} \in \mathbb{R}^{n\times l}$ is the mapping from the lifted space to the original space, and $\widetilde{x}_t \in \mathbb{R}^n$ is the reconstruction state of x_t. Hence, we can obtain the solution of $\mathcal{A}, \mathcal{B}$ by optimizing the cost function

$$\min_{\mathcal{A},\mathcal{B}} ||r||_2^2 = \min_{\mathcal{A},\mathcal{B}} \sum_{t=0}^{T} ||\psi(x_{t+1}) - \mathcal{A}\psi(x_t) - \ \mathcal{B}u_t||_2^2. \tag{14}$$

Furthermore, its solution can be obtained as:

$$[\mathcal{A} \ \mathcal{B}] = \mathcal{S}\mathcal{V}^\top(\mathcal{V}\mathcal{V}^\top)^\dagger, \tag{15}$$

where $\mathcal{S} = [\psi(x_1) \ \psi(x_2) \ \ldots \ \psi(x_{T+1})]$, $\mathcal{V} = [\Psi(x_0) \ \Psi(x_1) \ \ldots \ \Psi(x_T)]$, and $\dagger$ is the Moore Penrose pseudoinverse. The matrix $\mathcal{C}$ can be computed by minimizing the following problem:

$$\min_{\mathcal{C}} \sum_{t=0}^{T} ||x_t - \mathcal{C}\psi(x_t)||_2^2, \tag{16}$$

with the solution:

$$\mathcal{C} = \mathcal{R}\mathfrak{W}^\top(\mathfrak{W}\mathfrak{W}^\top)^\dagger, \tag{17}$$

where $\mathcal{R} = [x_0 \ x_1 \ \ldots \ x_T]$, and $\mathfrak{W} = [\psi(x_0) \ \psi(x_1) \ \ldots \ \psi(x_T)]$. In particular, if the set of lifting functions $\{\psi_1 \ \ldots \ \psi_l\}$ includes the state variable x, i.e., after possible reordering, $\psi_i(x) = x_i$ for any $i \in \{1, \ldots, n\}$. In this case, we have that $\mathcal{C} = [\mathbf{I} \ \mathbf{0}]$, where $\mathbf{I}$ is the identity matrix of size n. More details regarding to the calculation of A, $\mathcal{C}$ can be referred to literatures [31], [46].

C. The Deep Koopman Method

Compared to the classic EDMD, the Deep Koopman model uses a deep neural network to generate the observable subspace of the Koopman operator. The proposed Deep Koopman model adopts an encoder-decoder structure, where the encoder ψ^e transforms the system inputs to a lifted space and the decoder ψ^d converts the lifted system features back to the original space. We can represent the dynamical system as:

$$\begin{aligned} \psi^e(x_{t+1}, w^e) &= \widetilde{\mathrm{A}}\Phi(x_t, w^e, u_t), \\ \widetilde{x}_t &= \psi^d(\psi^e(x_t, w^e), w^d), \end{aligned} \tag{18}$$

where $\widetilde{\mathrm{A}} = [\widetilde{\mathcal{A}} \ \widetilde{\mathcal{B}}] \in \mathbb{R}^{l\times\mathrm{N}}$, $\Phi(x_t, w^e, u_t) = [\psi^e(x_t, w^e)^\top \ u_t^\top]^\top \in \mathbb{R}^{\mathrm{N}}$, ψ^e is the encoder with parameter w^e, and ψ^d is the decoder with parameter w^d. Fig. 2 shows the detail process of the Deep Koopman model.

For the ramp metering control problem on the freeway, there exists spatial dependencies for elements of state x. In this case, a long short-term memory (LSTM) is used to process the sequential data in the encoder, i.e., $\psi^e(x, w^e)$, which lifts the primal state x to the higher-dimension vector-valued observables. Fig. 3 shows the structure of the LSTM. The LSTM is an architecture of recurrent neural network (RNN) consisting of feedback connections. A LSTM has a forget gate $\mathfrak{f}_k$, an input gate $\mathfrak{i}_k$ and an output gate $\mathfrak{o}_k$. The forget gate utilizes the output $\mathfrak{h}_{k-1}$ in the last step and the current state x_k to compute the part of the cell state $\mathfrak{c}_{k-1}$ retaining in the current evaluation. Similarly, the input gate determines the update of the cell state. Finally, the output gate uses the newly updated cell state to determine the output. Algorithm 1 shows the process of LSTM, where $\otimes$ stands for the Hadamard

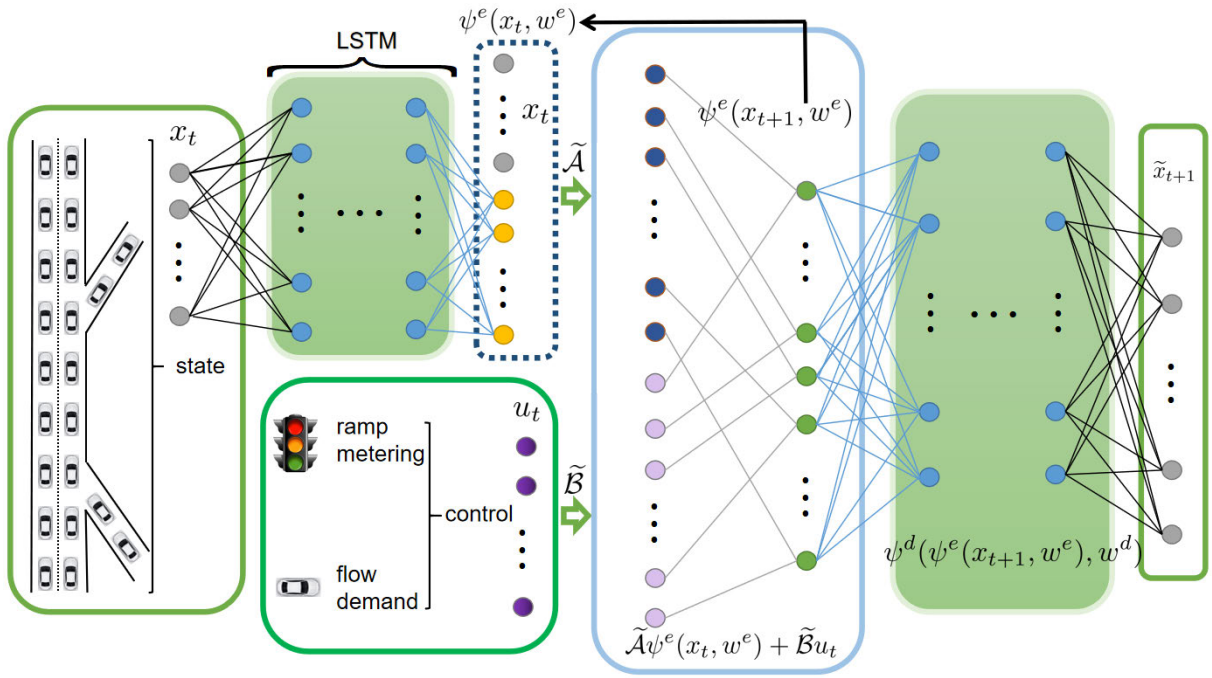


Fig. 2. The diagram of the proposed DKM. The original state x_t is lifted with the encoder, i.e.,$\psi^e(x_t, w^e)$. Then $\psi^e(x_t, w^e)$, u_t form the lifted state for constructing the linear evolution in the vector-valued observables. The freeway traffic flow dynamics can be recovered via a decoder $\psi^d(\psi^e(x_{t+1}, w^e), w^d)$ from the vector-valued observables.

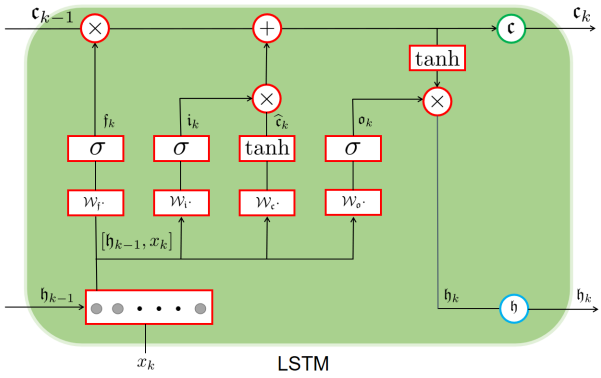


Fig. 3. The diagram of the LSTM.

Algorithm 1 Compute the Output Sequence of a LSTM Network

Input: Sequence $x_1, x_2, \ldots, x_K$,
Output: Sequence $\mathfrak{h}_1, \mathfrak{h}_2, \ldots, \mathfrak{h}_K$,
Set $\mathfrak{h}_0 = 0$
Set $\mathfrak{c}_0 = 0$
for $k \leftarrow k$ to K do **do**
  $\mathfrak{f}_k = \sigma(\mathcal{W}_\mathfrak{f}[\mathfrak{h}_{k-1}\ x_k] + \mathfrak{b}_\mathfrak{f})$
  $\mathfrak{i}_k = \sigma(\mathcal{W}_\mathfrak{i}[\mathfrak{h}_{k-1}\ x_k] + \mathfrak{b}_\mathfrak{i})$
  $\widehat{\mathfrak{c}}_k = \tanh(\mathcal{W}_\mathfrak{c}[\mathfrak{h}_{k-1}\ x_k] + \mathfrak{b}_\mathfrak{c})$
  $\mathfrak{c}_k = \mathfrak{f}_k \otimes \mathfrak{c}_{k-1} + \mathfrak{i}_k \otimes \widehat{\mathfrak{c}}_k$
  $\mathfrak{o}_k = \sigma(\mathcal{W}_\mathfrak{o}[\mathfrak{h}_{k-1}\ x_k] + \mathfrak{b}_\mathfrak{o})$
  $\mathfrak{h}_k = \mathfrak{o}_k \otimes \tanh(\mathfrak{c}_{k-1})$
end for

product and σ represents the logistic sigmoid function. $\mathcal{W}$ and $\mathfrak{b}$ denote the weight and bias.

Assuming K is the sequence length of the state inputs in the encoder, the vector-valued observables can be estimated by the last layer hidden state:

$$\psi^e(x, w^e) = [x^\top\ \mathfrak{h}_K^\top]^\top. \tag{19}$$

We use a feed-forward neural network to construct the decoder, i.e., ψ^d, which includes K' fully connected layers to recover the original state from the higher-dimensional observables. In particular, the output at any hidden layer $k \in \{1, \ldots, K'\}$ can be written as

$$\mathfrak{h}_k^d = \sigma_k^d(\mathcal{W}_k^d \mathfrak{h}_{k-1}^d + \mathfrak{b}_k^d). \tag{20}$$

The predicted state can be estimated by the output of the last layer of the decoder:

$$\widetilde{x}_t = \psi^d(\psi^e(x_t, w^e), w^d) = \sigma(\mathcal{W}_{K'}^d \mathfrak{h}_{K'-1}^d + \mathfrak{b}_{K'}^d). \tag{21}$$

Based on Equations (19) and (21), we can have the expressions of the encoder and decoder for dynamics system in (18). Moreover, parameters $\mathcal{W}$ and $\mathfrak{b}$ can be determined by optimizing a fixed loss function during training.

To approximate the freeway traffic flow dynamics in a longer time window, a multi-step prediction error rather than the single-step prediction error is considered. To formulate the multi-step loss function, we first present the state prediction in an M time-step interval:

$$x_{t+M} = \psi^d(\widetilde{\mathrm{A}}^M \Phi(x_t, w^e, u_t), w^d) + r(x, M), \tag{22}$$

where $\widetilde{\mathrm{A}}^M \Phi(x_t, w^e, u_t)$ is the M-step ahead state beginning from x_t with the expression of:

$$\begin{aligned} \widetilde{\mathrm{A}}^M \Phi(x_t, w^e, u_t) &= \psi^e(x_{t+M}, w^e) \\ &= \widetilde{\mathcal{A}}\psi^e(x_{t+M-1}, w^e) + \widetilde{\mathcal{B}}u_{t+M-1} \\ &= \widetilde{\mathcal{A}}^M \psi^e(x_t, w^e) + \sum_{s=1}^{M} \widetilde{\mathcal{A}}^{s-1} \widetilde{\mathcal{B}} u_{t+M-s}. \end{aligned} \tag{23}$$

Algorithm 2 The Deep Koopman Method (DKM)

1: Initialization: w^e, w^d, $\widetilde{\mathcal{A}}$, $\widetilde{\mathcal{B}}$, M. $Epoch = 0$, $Epoch_{max}$, $\alpha_i, i = 1, \ldots, 5$, batch size b_s, a small constant $\epsilon > 0$;
2: Train: trained w^e, w^d, $\widetilde{\mathcal{A}}$ and $\widetilde{\mathcal{B}}$;
3: **while** $Epoch > Epoch_{max}$ or $|\mathfrak{L}| \leq \epsilon$ **do**
4: Reset the training episodes;
5: **while** $Epoch$ is not Terminated **do**
6: Sample a batch data sequence of state x and control u;
7: Compute the vector-valued observables $\psi^e(x, w^e)$ with (19) and reconstruction states $\widetilde{x} = \psi^d(\psi^e(x, w^e), w^d)$ with (21);
8: Calculate the multi-step vector-valued observables $\widetilde{\mathrm{A}}^s\Phi(x_0, w^e, u_0)$ with (23) and predicted states $\widetilde{x}_s = \psi^d(\widetilde{\mathrm{A}}^s\Phi(x_0, w^e, u_0), w^d)$, where $s = 1, 2, \ldots, M$;
9: Compute the loss function $\mathfrak{L}$ with (27);
10: Update w^e, w^d, $\widetilde{\mathcal{A}}$ and $\widetilde{\mathcal{B}}$ via solving problem (27);
11: **end while**
12: $Epoch = Epoch + 1$
13: **end while**

Now, we define the main components of the loss function for the M time steps:

$$\mathfrak{L}_1 = \frac{1}{M}\sum_{s=1}^{M} ||x_{t+s} - \psi^d(\widetilde{\mathrm{A}}^s\Phi(x_t, w^e, u_t), w^d)||_2^2, \quad (24)$$

$$\mathfrak{L}_2 = \frac{1}{M}\sum_{s=1}^{M} ||\psi^e(x_{t+s}, w^e) - \widetilde{\mathrm{A}}^s\Phi(x_t, w^e, u_t)||_2^2, \quad (25)$$

$$\mathfrak{L}_3 = \frac{1}{M}\sum_{s=1}^{M} ||x_s - \psi^d(\psi^e(x_s, w^e), w^d)||_2^2, \quad (26)$$

where Equation (24) describes the prediction error, Equation (25) describes the predicted error in the lifted vector-valued observable space, and Equation (26) shows the reconstruction error. Based on the definitions of above loss functions, we can write the optimization problem as follows:

$$\min_{w^e, w^d, \widetilde{\mathcal{A}}, \widetilde{\mathcal{B}}} \mathfrak{L} = \alpha_1\mathfrak{L}_1 + \alpha_2\mathfrak{L}_2 + \alpha_3\mathfrak{L}_3 + \alpha_4||w^e||_2^2 + \alpha_5||w^d||_2^2, \quad (27)$$

where $\alpha_i, i = 1, 2, \ldots, 5$, are the weight parameters, and the last two terms of $\mathfrak{L}$ are regularization terms for preventing from over-fitting. Algorithm 2 shows the training process of the Deep Koopman model.

With the Deep Koopman model trained in Algorithm 2, and the approximated dynamics for freeway traffic flow can be written as:

$$\psi^e(x_{t+1}, w^e) = \widetilde{\mathcal{A}}\psi^e(x_t, w^e) + \widetilde{\mathcal{B}}u_t, \quad (28)$$

$$\widetilde{x}_t = \psi^d(\psi^e(x_t, w^e), w^d). \quad (29)$$

IV. MPC-Deep-Koopman for Freeway Traffic Flow Systems With Ramp Metering

The main idea of MPC controller is to optimize a specified optimization problem of the control input at each time step of the closed-loop operator. In this section, we design a MPC controller based on the linear system approximated by the Deep Koopman model (28) to solve the non-linear system in (6). Its computational complexity is comparable to that of a MPC controller for a linear system with original system state space. We formulate the optimization problem at each time step t of the closed loop as:

$$\max_{u_{t,p}} \mathrm{T}_t = \sum_{p=0}^{N_p} \widetilde{v}_p^\top z_{t,p} - \lambda \sum_{p=0}^{N_p-2} ||u_{t,p+1} - u_{t,p}||^2, \quad (30)$$

$$\text{s.t. } z_{p+1} = \widetilde{\mathcal{A}}z_{t,p} + \widetilde{\mathcal{B}}u_{t,p},\ p = 0, \ldots, N_p - 1, \quad (31)$$

$$b_{min,p} \leq E_p z_{t,p} \leq b_{max,p},\ p = 0, \ldots, N_p, \quad (32)$$

$$u_{min,p} \leq u_{t,p} \leq u_{max,p},\ p = 0, \ldots, N_p - 1, \quad (33)$$

$$z_{t,0} = \psi^e(x_t, w^e), \quad (34)$$

where N_p is the predictive horizon, $\widetilde{v}_p \in \mathbb{R}^l$, $E_p \in \mathbb{R}^{d\times l}$, and $u_{min,p}$ and $u_{max,p}$ are the lower and upper bounds of control input $u_{t,p}$, respectively, and $z_{t,0}$ is the lifted state obtained using the Deep Koopman method at time step t. Note that the optimization problem (30) is parametrized by the current state x_t of the non-linear system (34). We can obtain a feedback controller by solving the problem T_t. We denote u_t^* as the optimal solution and reformulate the problem T_t in a so-called dense form:

$$\begin{aligned} \max_{u_t} \mathrm{T}_t &= u_t^\top R u_t + h^\top u_t + c^\top z_{t,0}, \\ \text{s.t. } & b_{min} \leq E z_{t,0} + D u_t \leq b_{max}, \\ & u_{min} \leq u_t \leq u_{max}, \\ & z_{t,0} = \psi^e(x_t, w^e), \end{aligned} \quad (35)$$

where $u_t = [u_{t,0}^\top \ldots u_{t,N_p-1}^\top]^\top \in \mathbb{R}^{mN_p}$; $R \in \mathbb{R}^{mN_p\times mN_p}$; $h \in \mathbb{R}^{mN_p}$; $c = \sum_{p=0}^{N_p} \widetilde{v}_p^\top \widetilde{\mathcal{A}}^p \in \mathbb{R}^l$; $E \in \mathbb{R}^{d(N_p+1)\times l}$; $D \in \mathbb{R}^{d(N_p+1)\times mN_p}$; $b_{min} = [b_{min,0}^\top \ \ldots \ b_{min,N_p}^\top]^\top \in \mathbb{R}^{d(N_p+1)}$; $b_{max} = [b_{max,0}^\top \ \ldots \ b_{max,N_p}^\top]^\top \in \mathbb{R}^{d(N_p+1)}$; $u_{min} = [u_{min,0}^\top \ \ldots \ u_{min,N_p-1}^\top]^\top \in \mathbb{R}^{mN_p}$; and $u_{max} = [u_{max,0}^\top \ \ldots \ u_{max,N_p-1}^\top]^\top \in \mathbb{R}^{mN_p}$. In particular,

$$R = \begin{bmatrix} -\lambda * e & \lambda * e & \ldots & \mathbf{0} & \mathbf{0} \\ \lambda * e & -2\lambda * e & \ldots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ldots & -2\lambda * e & \lambda * e \\ \mathbf{0} & \mathbf{0} & \ldots & \lambda * e & -\lambda * e \end{bmatrix},$$

where $e = \mathbf{I} \in \mathbb{R}^{m\times m}$ is a unit matrix,

$$h = \begin{bmatrix} \sum_{p=1}^{N_p} \widetilde{v}_p^\top \widetilde{\mathcal{A}}^{p-1}\widetilde{\mathcal{B}} & \sum_{p=2}^{N_p} \widetilde{v}_p^\top \widetilde{\mathcal{A}}^{p-2}\widetilde{\mathcal{B}} & \ldots & \widetilde{v}_{N_p}\widetilde{\mathcal{B}} \end{bmatrix}^\top,$$

$$E = \begin{bmatrix} E_0 \\ E_1\widetilde{\mathcal{A}} \\ \vdots \\ E_{N_p}\widetilde{\mathcal{A}}^{N_p} \end{bmatrix},$$

and

$$D = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} \\ E_1\widetilde{\mathcal{B}} & \mathbf{0} & \ldots & \mathbf{0} \\ \vdots & \vdots & \ldots & \mathbf{0} \\ E_{N_p}\widetilde{\mathcal{A}}^{N_p-1}\widetilde{\mathcal{B}} & E_{N_p}\widetilde{\mathcal{A}}^{N_p-2}\widetilde{\mathcal{B}} & \ldots & E_{N_p}\widetilde{\mathcal{B}} \end{bmatrix}.$$

Algorithm 3 MPC-Deep-Koopman to Compute the Control Sequence of Problem (35)

1: **for** $t = 0, 1, \ldots, T$ **do**
2: Let $z_{t,0} = \psi^e(x_t, w^e)$;
3: Minimize Problem T_t to obtain an optimal solution u_t^*;
4: Let $u_t = u_t^*(1:m)$;
5: Update $\widetilde{x}_t$ with system (29).
6: Obtain the practical x_{t+1} with SUMO by implementing the control u_t.
7: **end for**

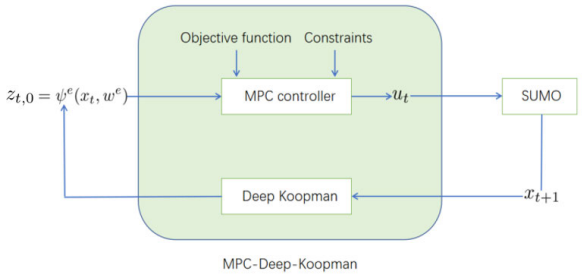


Fig. 4. The diagram of MPC-Deep-Koopman to compute the control sequence.

Note that the dimension of matrix R and the number of constraints d do not rely on the dimension of vector-valued observables z. We can solve the linear MPC problem on the same predictive horizon if we obtain $z_{t,0} = \psi^e(x_t, w^e)$. Fig. 4 shows the diagram of MPC-Deep-Koopman to compute the control sequence. Thus, we summarized the algorithm of the closed-loop operation of vector-valued observables as follows:

V. Numerical Studies

In this section, the proposed Deep Koopman model is verified in the microscopic simulated environment SUMO for various demand scenarios. We consider the comparisons with baselines.

A. Experiment Design

To validate the performance of our proposed Deep Koopman model, we conduct a case study in the SUMO simulation environment. SUMO is an open-source program of traffic flow simulation. We extract a freeway road network in the Yubei vicinity, which is one of the most congested freeway roads in the city of Chongqing, China. Fig. 5 shows the actual map of the selected freeway road including its nearby layout. The selected freeway road has a total length of around 27km, starting from Jinxing road and ending at Modern road with 11 on-ramps and 8 off-ramps. At every on-ramp, there is a ramp meter controlling the in-flow of vehicles. Fig. 6 illustrates the freeway traffic model in the SUMO simulation.

The training data for the proposed Deep Koopman model is sampled through the freeway traffic simulation with randomized controls on different scenarios. We generate 100 scenarios with varied traffic flows. The duration of each simulation is set as 4 hours (i.e. 14400 seconds) and the control interval is set as 60s. The simulation period is divided into peak and off-peak hours. For off-peak hours, the total in-flows for the main road

Fig. 5. Road layout.

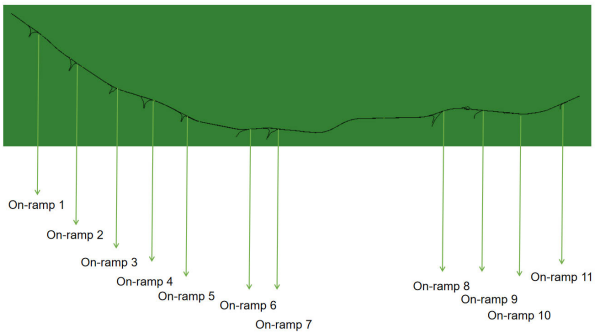


Fig. 6. Illustration of the SUMO simulation model for the investigated freeway traffic network.

and the on-ramps are generated randomly within the range of $[1000, 4000]$ and $[1, 400]$, respectively. For peak hours, the in-flows for the main road and the on-ramps would vary within the range of $[3000, 8000]$ and $[200, 800]$, respectively. 80 scenarios are used for the training data collection resulting in a total of 19200 data triplets, i.e., (x_t, u_t, x_{t+1}), while the other 20 scenarios are used for validation. Random controls are sampled uniformly within the range of $[u_{min}, u_{max}]$. During the simulation, we convert the control variable $a_{t,i} \in u_t, i \in \mathcal{N}$ for each on-ramp i to the duration of green light: $\Delta t = 60 * a_{t,i}$. In our experiment, all models are complied in Python 3.7 running on a computer with Intel Xeon E5-1650 v4, 3.6GHz CPU.

B. Evaluation Metrics and Baselines

To comprehensively evaluate the performance of the proposed Deep Koopman model, baseline approaches are selected for comparison. Firstly, we assess the performances of predicting the dynamics system between the proposed Deep Koopman model and extended dynamic mode decomposition (EDMD) [31]. For EDMD, we adopt the Gaussian radial basis function (RBF) [49] as the basis function $\psi(x)$:

$$\psi(x) = e^{(\epsilon||x-C||)^2}, \tag{36}$$

where ϵ denotes the shape parameter and $C \in \mathbb{R}^l$ is the center points. In our experiment, the kernel centers are selected using K-means clustering [51]. Then, we estimate the performance of the classic traffic control models including ALINEA regulator [2] and CTM-based deterministic optimization (CTM-DO) [13]. Finally, we validate the performance of different approaches on ramp metering control. Besides the EDMD, we design a linear control method [39] which approximates the dynamics system with a linear model on the original state. To illustrate the effectiveness of our proposed Deep Koopman model, we also evaluate the performances of traffic simulation with a fixed control and with no ramp meters. The

TABLE I
MODEL PARAMETERS FOR DEEP KOOPMAN

Name of parameter	Value	Name of parameter	Value
$\alpha_1, \alpha_2, \alpha_3$	1	α_4, α_5	10^{-9}
$\widetilde{v}_{i,p}, i = 1, \ldots, 9$	1	$\widetilde{v}_{i,p}, i = 10, \ldots, 43$	0
$b_{min,ip}, i = 1, \ldots, 9$	0veh/m	$b_{max,ip}, i = 1, \ldots, 8$	33veh/m
$b_{max,ip}, i = 9$	66veh/m	$u_{min,ip}, i = 1 \ldots, 11$	20%
$u_{max,ip}, i = 1 \ldots, 11$	70%	$\triangle T$	60s
λ	10^{-4}	N_p	17m
T	240	N_c	1m

TABLE II
MODEL PARAMETERS FOR ALINEA REGULATOR

Name of parameter	Value	Name of parameter	Value
$o_{cr,i}, i = 1, \ldots, 11, 14, 15$	10.35veh/m	$o_{cr,i}, i = 12, 13, 16, \ldots, 23$	6.9veh/m
r_{max}	0.5veh/s	K_p	0.02veh/s

TABLE III
MODEL PARAMETERS FOR CTM

Name of parameter	Value	Name of parameter	Value
$Q_i, i = 1, \ldots, 11, 14, 15$	1.92veh/s	$Q_i, i = 12, 13, 16, \ldots, 23$	1.28veh/s
$\rho_{max,i}, i = 1, \ldots, 11, 14, 15$	0.4veh/m	$\rho_{max,i}, i = 12, 13, 16, \ldots, 23$	0.27veh/m
w	5.81m/s	v_f	27.78m/s
ΔT	5s	I	23
N_p	324	N_c	12
r_{max}	0.5veh/s	q_{max}	60veh

fixed control means that each ramp meter would follow a fixed duration for each phase. In our experiment, we set the durations of green light, yellow light and red light as 39s, 3s and 18s, respectively. Note that models (CTM-DO, EDMD, Linear and Deep Koopman) are tested in MPC framework.

Apart from the above-mentioned model-based algorithms, we also compare the performance of our proposed Deep Koopman approach with the model-free deep reinforcement learning (DRL)-based framework. DRL has drawn significant attention due to its remarkable performance on many real-time decision making problems [21], [22], [23], [24], [50]. In our experiment, the iterative DRL framework with DQN algorithm proposed by [23] is adopted for comparison. The action of the DRL approach is defined as the proportion of green light in a control cycle and the ultimate goal is to maximize the total throughput of vehicles over the simulation time. The episode horizon of the simulation is set as 4 hours and the simulation environment will be reset according to the configurations described in Section V-A after each episode of training.

As indicated in Section V-A, we generate 4800 testing data from 20 testing scenarios and the mean squared error (MSE) of prediction are utilized to evaluate the performance of traffic state prediction. The performance of traffic control is evaluated based on four metrics including the number of leaving vehicles per decision interval, the average travel time of vehicles, the total throughput of traffic out-flow, and the deviation of control variables.

C. Parameters Setting

In our experiment, we set the dimension of the lifted space as 1000. While training the Deep Koopman model, we run the mini-batch training with a batch size of 32 for 100 epochs. Xavier initialization [52] is adopted for initializing the weights of neural networks in the proposed Deep Koopman model. The details of the associated parameters are shown in Table I. In particular, we define the matrix $E_p \in \mathbb{R}^{9\times l}$ as follows:

$$E_p = \begin{bmatrix} 1 & 0 & \ldots & 0 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \ldots & 0 & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 1 & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 & 1 & \ldots & 0 \end{bmatrix}, \ p = 0, \ldots, N_p.$$

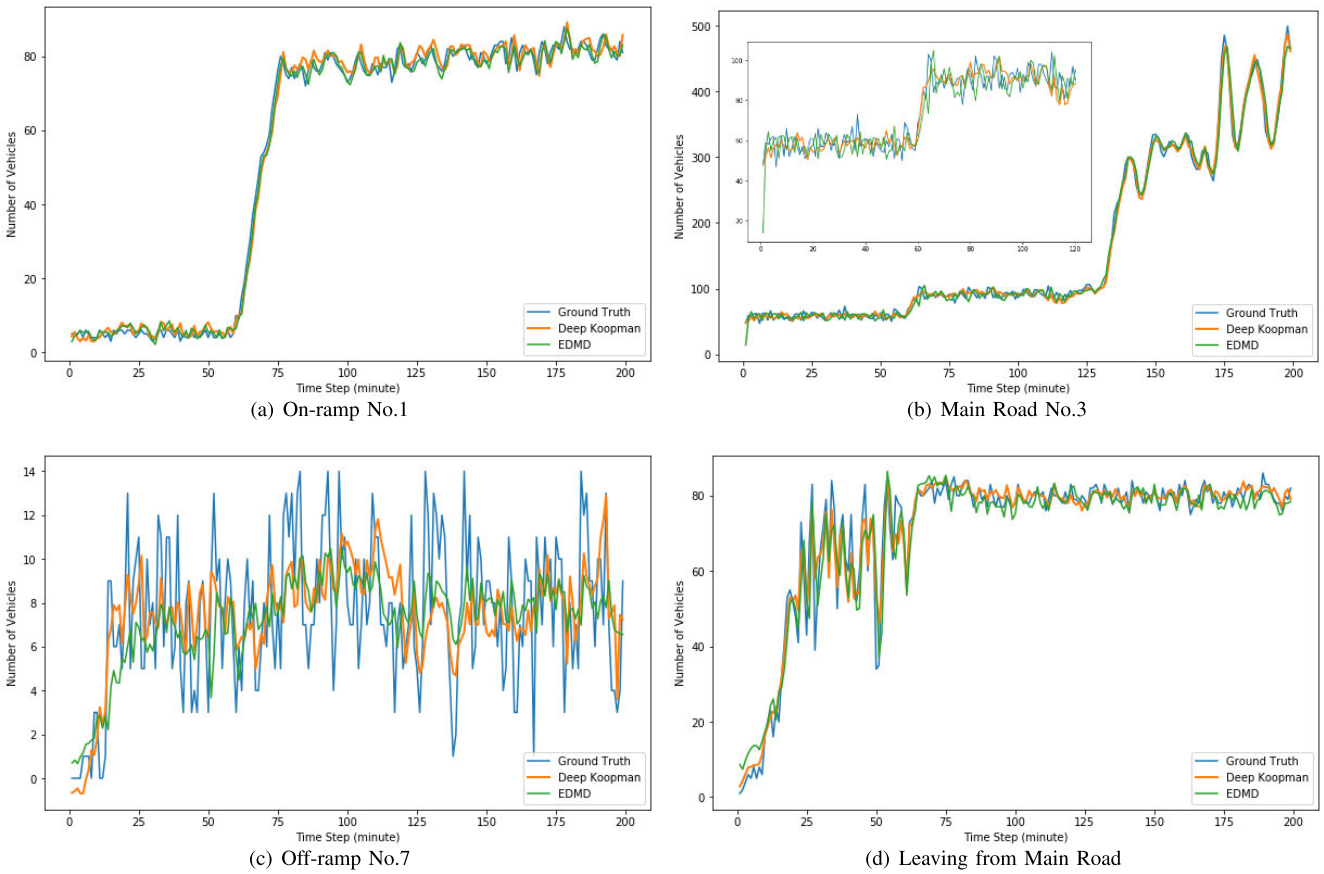
For implementing the ALINEA regulator and the CTM model, we set their corresponding parameters as shown in Tables II and III:

Note that N_p and N_c are the prediction horizon and control horizon, respectively; ΔT is the simulation duration at each step; T represents total simulation step; o_{cr} is the critical occupancy; r_{max} is the maximum ramp metering rate; Q denotes the capacity; ρ_{cr} is the critical density; ρ_{max} is the jam density; w is the backward wave speed; v_f is the free flow speed; I denotes the number of cells and q_{max} represents the maximum queue length.

D. Results and Analysis

1) Traffic Prediction: We first calculate the mean squared error of traffic state prediction: MSE $= \sum_{t=1}^{240}\sum_{i=1}^{43}(x_{t,i} - \widetilde{x}_{t,i})^2$. The MSE values of EDMD and Deep Koopman are 59.062 and 47.293, respectively. Obviously, the Deep Koopman model outperforms the EDMD with 24.885% improvement. Fig. 7 shows the comparison of traffic state prediction between EDMD and the proposed Deep Koopman for the scenario 1. The blue line is the actual traffic state. More specifically, Figs. 7(a)-7(d) show the predictions of the real-time number of vehicles on the on-ramp 1 and the main road segment 3, and the number of the leaving vehicles from the off-ramp 7 and the main road. We can observe that the Deep Koopman model has a higher accuracy compared to the EDMD. This implies that the proposed Deep Koopman model could better predict future states of the freeway traffic flow and have a more precise reconstruction of the state from the lifted space.

2) Traffic Control Performance Analysis: To assess the performance of the controller, we present the average travel time (ATT) of the vehicles, the total throughput of vehicles (TTV) and the deviation of the control variables (DCV) for four testing scenarios, as shown in Table IV, where ATT can be retrieved from the SUMO directly, TTV $= \sum_{t=0}^{240} v_t^\top x_t$ and DCV $= \sum_{t=0}^{238} \lambda||u_{t+1} - u_t||^2$. It is obvious that all the baseline control methods could reduce the ATT and improve the TTV compared to No control strategy. For the traditional control methods, the ALINEA does not show a significant improve compared to the Fixed Control strategy, while the MPC-CTM-DO model is better performed on the ATT for all scenarios. However, their results on ATT are no better than that of MPC-EDMD and MPC-Deep-koopman. For example, the proposed MPC-Deep-Koopman model achieves a 18.55% lower ATT than the MPC-CTM-DO for the scenario 1, and reduces the ATT by 23.956% for the scenario 3, and by at least 21% for



(a) On-ramp No.1 (b) Main Road No.3

(c) Off-ramp No.7 (d) Leaving from Main Road

Fig. 7. The comparison of state prediction between EDMD and the proposed deep Koopman for the scenario 1.

the other 3 cases compared to No Control strategy. The metric of TTV shows the performance of the controllers on optimizing the first part of the objective function in (30), while DCV represents the second part. While analyzing the results of TTV, the control strategy produced by the MPC-Deep-Koopman model allows 6391 more vehicles completing their trips during the simulation compared to the No Control approach for the scenario 2. There are 5000 vehicles more completing the travel for other 3 instances. It is worth mentioning that ALINEA and MPC-CTM-DO outperform the MPC-linear in scenario 3. The DCV of the MPC-EDMD is better than the MPC-Linear model for all instances, while the proposed MPC-Deep-Koopman model has the best performance and ALINEA performs the worst on DCV. This is reasonable because ALINEA computes the control strategy based on the current state without considering future changes. Moreover, the controls of MPC-Deep-Koopman model are more stable. We can conclude that our proposed MPC-Deep-Koopman model is more efficient compared to the baselines. The controller integrated with the MPC-Deep-Koopman is capable of mitigating traffic jams so that improving the freeway management.

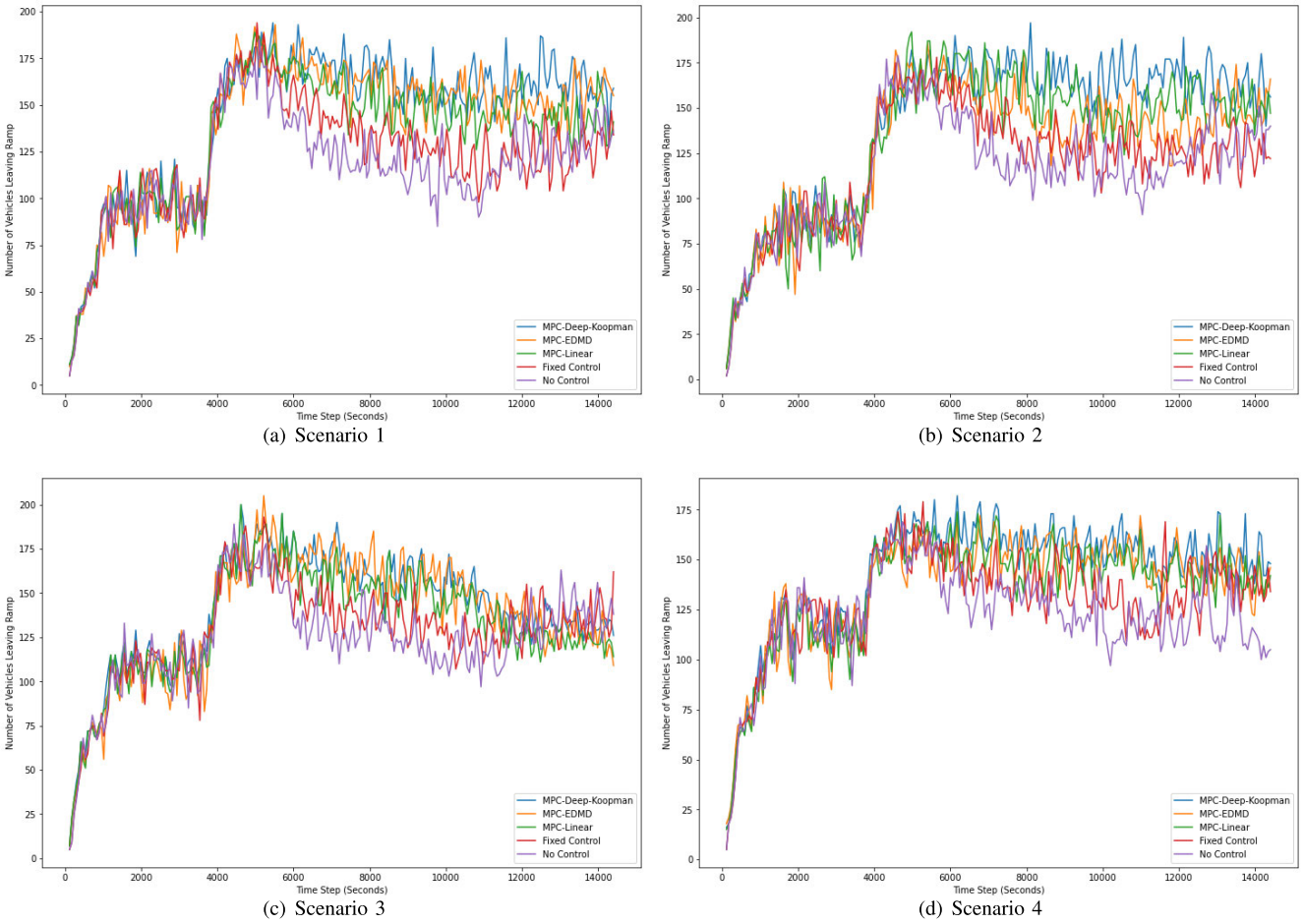
3) Number of Leaving Vehicles: To further visualize and validate the effectiveness of the MPC-Deep-Koopm, we monitor the real-time number of leaving vehicles from the freeway to evaluate the performance of controllers on improving the traffic throughput. We present the results of 4 different scenarios obtained by the proposed MPC-Deep-Koopman model and other baselines, as shown in Fig. 8. As we can observe, the performance of No Control is the worst for all 4 scenarios. This is because the increase of traffic flow on the freeway could lead to serious traffic jams, especially when no control actions are applied. It is obvious that the number of leaving vehicles increases after 450s in simulation when a controller is adopted. Compared to the MPC-Linear model, MPC-EDMD and the Fixed Control, the proposed MPC-Deep-Koopman model has the most significant improvement on the real-time throughput, especially in the scenarios 1, 2 and 4. The results indicate that the proposed MPC-Deep-Koopman model is more effective in improving the traffic operation.

4) Ramp Control Variation: We show the results of ramp metering control for approaches including the MPC-Linear model, the MPC-EDMD and the MPC-Deep-Koopman in scenarios 1 and 3. Figs. 9(a)-9(b) show the control variables for the on-ramp 1. We find the variation of controls showing an opposite trend with the changing of the traffic demand flow. As shown in Fig. 9(a), we can see that the higher demand results in shorter duration of green light. This is consistent with the actual situation, because reducing the green phase duration of traffic light of on-ramps can stop vehicles from merging to the main road from on-ramps so that mitigating the traffic jams on the main road. Fig. 10 shows the comparison of the fluctuations of the ramp control for the on-ramp 1. Red lines in Figs. 10(a)-10(b) represent the median of ramp metering control variables. The bottom and top lines are the minimum and maximum values, respectively. The bottom and top edges of quadrangle refer to the first quarter and the third quarter of the quantile, respectively. By comparison, we can observe that the MPC-EDMD produces the controls with a

TABLE IV

PERFORMANCE COMPARISON OF AVERAGE OBJECTIVES (ATT: AVERAGE TRAVEL TIME, TTV: TOTAL THROUGHPUT, DCV: DEVIATION OF CONTROL)

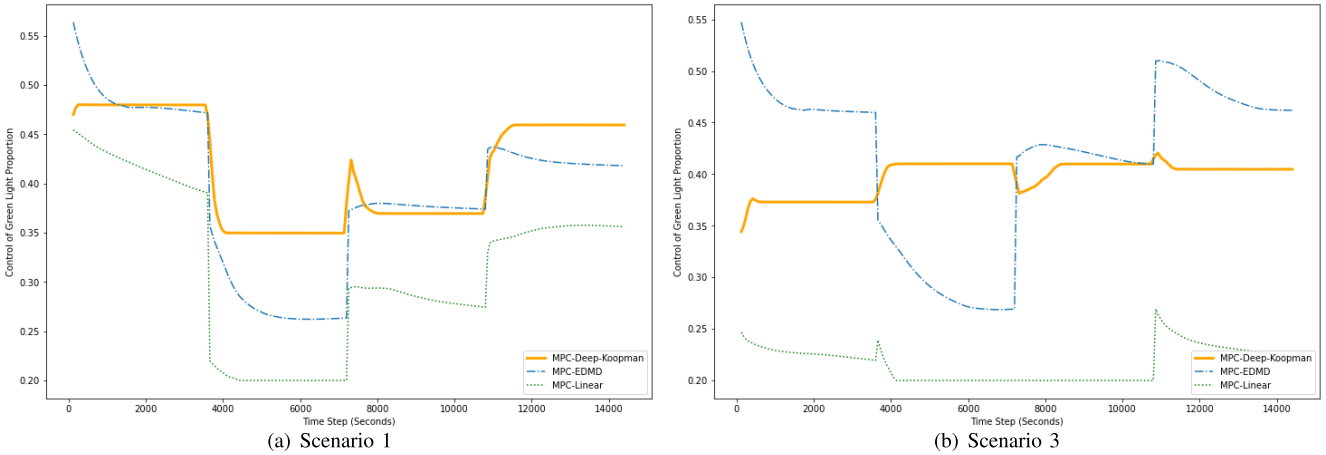
Approach	Scenario 1			Scenario 2			Scenario 3			Scenario 4		
	ATT	TTV	DCV	ATT	TTV	DCV	ATT	TTV	DCV	ATT	TTV	DCV
ALINEA	1833.23	28951	0.36130	1774.16	27643	0.33570	1688.34	31383	0.37520	1566.15	31122	0.37640
MPC-CTM-DO	1790.06	31328	0.31730	1739.92	27968	0.28430	1578.85	31355	0.29720	1496.82	32613	0.27830
MPC-EDMD	1571.74	32613	0.20779	1514.45	30948	0.20336	1459.89	32674	0.22930	1370.05	32723	0.15006
MPC-Linear	1626.27	32290	0.28700	1554.33	32195	0.21649	1549.06	31057	0.32817	1433.83	31608	0.32998
Fixed Control	1810.53	29403	0	1774.38	28662	0	1763.53	30582	0	1640.80	31029	0
No Control	1893.01	27998	-	1829.19	27560	-	1792.48	28903	-	1696.09	29106	-
MPC-Deep-Koopman	**1457.96**	**34333**	**0.09384**	**1433.62**	**33951**	**0.13973**	**1363.08**	**34267**	**0.15235**	**1295.33**	**34615**	**0.11362**



(a) Scenario 1 (b) Scenario 2

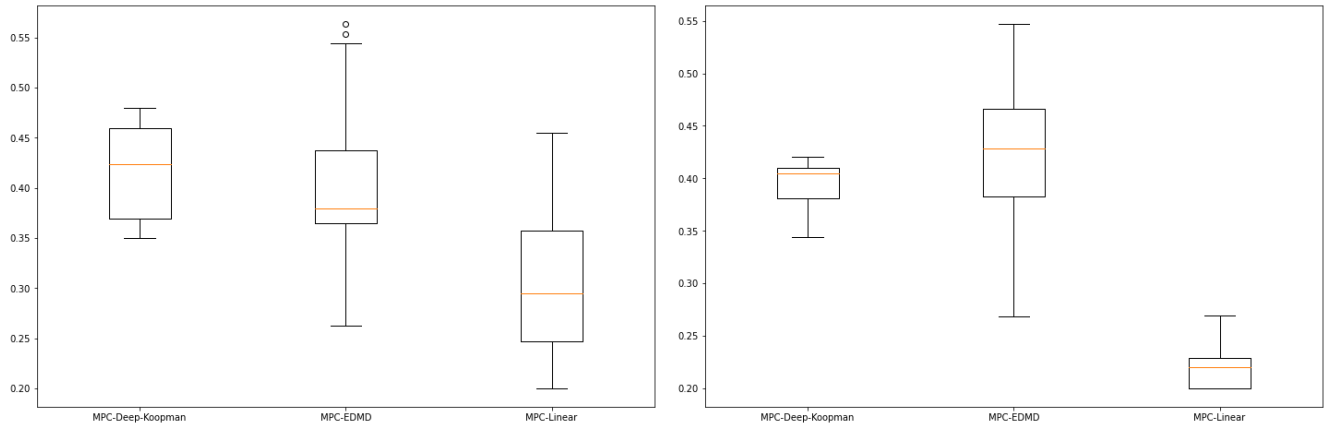
(c) Scenario 3 (d) Scenario 4

Fig. 8. The comparison of number of leaving vehicles between baseline approaches and the proposed MPC-Deep-Koopman model.



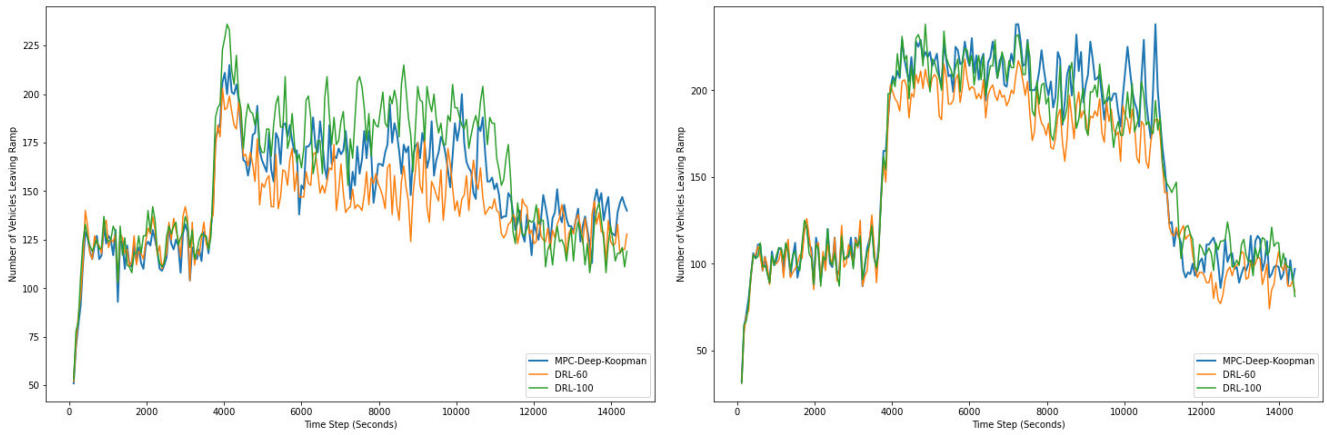
(a) Scenario 1 (b) Scenario 3

Fig. 9. The comparison of ramp control for on-ramp No.1.



(a) Scenario 1 (b) Scenario 3

Fig. 10. The comparison of ramp control fluctuation for on ramp No.1.



(a) Scenario 5 (b) Scenario 6

Fig. 11. The comparison of the number of leaving vehicles per time step between DRL and MPC-Deep-Koopman.

TABLE V

PERFORMANCE COMPARISON OF AVERAGE OBJECTIVES (ATT: AVERAGE TRAVEL TIME, TTV: TOTAL THROUGHPUT, DCV: DEVIATION OF CONTROL)

Approach	Scenario 5			Scenario 6		
	ATT	TTV	DCV	ATT	TTV	DCV
DRL-60	1475.91	33671	0.20130	1532.59	34751	0.17269
DRL-100	1268.48	37599	0.18226	1427.27	36967	0.15135
MPC-Deep-Koopman	1365.36	35743	**0.17521**	**1396.05**	**37383**	0.15365

median value around 0.37 and the MPC-Linear model has the controls with a median value around 0.28, and the median value of our controller is around 0.42 in the scenario 1. The MPC-Deep-Koopman and MPC-EDMD achieve similar values on instance 3, but the MPC-Linear model has a lower median value. In addition, the MPC-EDMD and the MPC-Linear model have greater fluctuations than the MPC-Deep-Koopman model, which implies that the MPC-Deep-Koopman model has a better performance on optimizing the deviation of controls.

5) Comparison With DRL Approach: In this section, we train the proposed MPC-Deep-Koopman with data sampled from 60 episodes and compare it with the DRL approach on two testing scenarios. In our experiment, we show the performance of the DRL approach trained for 60 and 100 episodes, denoted by DRL-60 and DRL-100, respectively. Fig 11 indicates the number of leaving vehicles per time step. It is obvious that the MPC-Deep-Koopman outperforms the DRL approach (DRL-60) when they are trained with the same amount of data samples. The increment of the training samples could result in a great improvement on the performance of the DRL approach (DRL-100). Table V shows the results of the overall metrics for these three methods. For scenario 5, the ATT of the MPC-Deep-Koopman is 8.097% higher than that of the DRL-60 while 7.096% lower than the DRL-100. With 40 more episodes of training, the ATT of the DRL on scenario 5 is improved by 5.193%. These results demonstrate the fact that the model-free DRL algorithms suffer from the low sample efficiency. On the contrary, the proposed MPC-Deep-Koopman method is a model-based algorithm which has a higher sample efficiency. However, the limitation of the proposed approach comes from the error of dynamics approximation. Furthermore, the ATT and TTV of the MPC-Deep-Koopman are better than that of DRL-60 and DRL-100 for Scenario 6, which implies that the performance of the MPC-Deep-Koopman method is competitive to the baseline DRL method.

TABLE VI
COMPUTATION TIME FOR BASELINES AND MPC-DEEP-KOOPMAN FOR EACH MPC COMPUTATION WITH $N_p = 17$M, $N_c = 1$M

Approaches	Computation Time (s)
ALINEA	0.2388
MPC-CTM-DO	0.6992
MPC-Linear	0.3581
MPC-EDMD	2.7641
DRL	0.2518
MPC-Deep-Koopman	0.8852

6) Computational Time: In this section, we present the actual computation time for each MPC closed loop for the ALINEA, MPC-CTM-DO, MPC-Linear model, the MPC-EDMD, DRL and our proposed MPC-Deep Koopman, as shown in Table VI. As we can observe, the ALINEA and DRL are the fastest approaches as they have simpler modeling of dynamics, while the MPC-CTM-DO outperforms the MPC-EDMD and MPC-Deep Koopman. Moreover, the computation time of MPC-Deep-Koopman is 0.8852s, which means that the running time is less than 1s at each time step when the optimal control strategy is obtained. This is acceptable for our problem and can satisfy the practical requirements. The computation time for the DRL approach per decision step is 0.2518, which is the second fastest among all algorithms. However, it is worth mentioning that the training time of the DRL-60 is 7 hours compared to 5 mins for the training of the encoder-decoder network of the proposed Deep Koopman model.

VI. CONCLUSION

In this paper, we study a ramp metering control problem on the freeway traffic. Due to the high non-linearity and complexity of the freeway traffic flow dynamics, we use the Koopman operator to model the evolution of the dynamical system due to its interpretability and linearity. To learn a finite-dimensional approximation of the Koopman operator, we propose a Deep Koopman model based on the neural networks with an encoder-decoder structure. LSTM units are used to process the sequential traffic data in the encoder. Compared to the EDMD, the Deep Koopman model is a data-driven approach of learning the lifted state of dynamics without any prior information of the basis function selection. To provide a real-time control of the ramp metering, we design a model predictive control with pre-trained Deep Koopman model to maximize the total throughput of vehicles on the freeway. SUMO environment is used for the experimental studies to evaluate the performance of the proposed approach. Compared to the EDMD with RBF basis functions, the proposed Deep Koopman model has shown a better accuracy in the predictions of the dynamical states. We also compare the performance of the real-time control of ramp metering between the MPC-Deep-Koopman, MPC-EDMD, and classic traffic control models. When compared to the model-free DRL based method, the proposed approach also shows a competitive result.

However, as most of the data-based methods, our proposed method is dependent on the data to learn traffic dynamics. To train the model accurately, our method requires both traffic flow data and traffic control data. However, to collect diverse traffic control data is not an easy job in practice. Our future work will focus on the model-based reinforcement learning with Koopman operators through combination of historical data with synthetic data to promote it into real implementations.

REFERENCES

[1] M. Yang, Z. Li, Z. Ke, and M. Li, "A deep reinforcement learning-based ramp metering control framework for improving traffic operation at freeway weaving sections," in *Proc. Transp. Res. Board 98th Annu. Meeting*, Washington, DC, USA, pp. 13–17, Jan. 2019.

[2] M. Papageorgiou, H. Hadj-Salem, and J.-M. Blosseville, "ALINEA: A local feedback control law for on-ramp metering," *Transp. Res. Rec.*, vol. 1320, no. 1, pp. 58–67, 1991.

[3] E. Smaragdis and M. Papageorgiou, "Series of new local ramp metering strategies," *Transp. Res. Rec.*, vol. 1856, pp. 74–86, Jan. 2003.

[4] I. Papamichail and M. Papageorgiou, "Traffic-responsive linked ramp-metering control," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 1, pp. 111–121, Mar. 2008.

[5] Y. Wang, E. B. Kosmatopoulos, M. Papageorgiou, and I. Papamichail, "Local ramp metering in the presence of a distant downstream bottleneck: Theoretical analysis and simulation study," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 2024–2039, Oct. 2014.

[6] A. H. F. Chow and Y. Li, "Robust optimization of dynamic motorway traffic via ramp metering," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 3, pp. 1374–1380, Jun. 2014.

[7] J. Chen, W. Zhan, and M. Tomizuka, "Autonomous driving motion planning with constrained iterative LQR," *IEEE Trans. Intell. Veh.*, vol. 4, no. 2, pp. 244–254, Jun. 2019.

[8] B. Liu, Y. Tang, Y. Ji, Y. Shen, and Y. Du, "A deep reinforcement learning approach for ramp metering based on traffic video data," *J. Adv. Transp.*, vol. 2021, pp. 1–13, Oct. 2021.

[9] T. Bellemans, B. De Schutter, and B. De Moor, "Model predictive control for ramp metering of motorway traffic: A case study," *Control Eng. Pract.*, vol. 14, no. 7, pp. 757–767, 2006.

[10] I. Papamichail, A. Kotsialos, I. Margonis, and M. Papageorgiou, "Coordinated ramp metering for freeway networks—A model-predictive hierarchical control approach," *Transp. Res. C, Emerg. Technol.*, vol. 18, no. 3, pp. 311–331, Jun. 2010.

[11] C. Pasquale, S. Sacone, S. Siri, and B. De Schutter, "A multi-class model-based control scheme for reducing congestion and emissions in freeway networks by combining ramp metering and route guidance," *Transp. Res. C, Emerg. Technol.*, vol. 80, pp. 384–408, Jul. 2017.

[12] A. Kotsialos, "A varying parameter multi-class second-order macroscopic traffic flow model for coordinated ramp metering with global and local environmental objectives," *Transp. Res. C, Emerg. Technol.*, vol. 128, Jul. 2021, Art. no. 103106.

[13] G. Gomes and R. Horowitz, "Optimal freeway ramp metering using the asymmetric cell transmission model," *Transp. Res. C, Emerg. Technol.*, vol. 14, no. 4, pp. 244–262, 2006.

[14] C. Roncoli, M. Papageorgiou, and I. Papamichail, "Traffic flow optimisation in presence of vehicle automation and communication systems—Part I: A first-order multi-lane model for motorway traffic," *Transp. Res. C, Emerg. Technol.*, vol. 57, pp. 241–259, Aug. 2015.

[15] C. Roncoli, M. Papageorgiou, and I. Papamichail, "Traffic flow optimisation in presence of vehicle automation and communication systems—Part II: Optimal control for multi-lane motorways," *Transp. Res. C, Emerg. Technol.*, vol. 57, pp. 260–275, Aug. 2015.

[16] C. F. Daganzo, "The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory," *Transp. Res. B, Methodol.*, vol. 28, no. 4, pp. 269–287, Aug. 1994.

[17] C. F. Daganzo, "The cell transmission model, Part II: Network traffic," *Transp. Res. B, Methodol.*, vol. 29, no. 2, pp. 79–93, Apr. 1995.

[18] B. Ran, P. J. Jin, D. Boyce, T. Z. Qiu, and Y. Cheng, "Perspectives on future transportation research: Impact of intelligent transportation system technologies on next-generation transportation modeling," *J. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 226–242, Aug. 2013.

[19] I. Yaqoob et al., "Big data: From beginning to future," *Int. J. Inf. Manag.*, vol. 36, no. 6, pp. 1231–1247, Dec. 2016.

[20] P. Zhao and H. Hu, "Geographical patterns of traffic congestion in growing megacities: Big data analytics from Beijing," *Cities*, vol. 92, pp. 164–174, Sep. 2019.

[21] F. Deng, J. Jin, Y. Shen, and Y. Du, "Advanced self-improving ramp metering algorithm based on multi-agent deep reinforcement learning," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 3804–3809.

[22] S. Ghanbartehrani, A. Sanandaji, Z. Mokhtari, and K. Tajik, "A novel ramp metering approach based on machine learning and historical data," *Mach. Learn. Knowl. Extraction*, vol. 2, no. 4, pp. 379–396, Sep. 2020.

[23] Y. Zhou, K. Ozbay, P. Kachroo, and F. Zuo, "Ramp metering for a distant downstream bottleneck using reinforcement learning with value function approximation," *J. Adv. Transp.*, vol. 2020, pp. 1–13, Oct. 2020.

[24] T. Alexakis, N. Peppes, E. Adamopoulou, and K. Demestichas, "An artificial intelligence-based approach for the controlled access ramp metering problem," *Vehicles*, vol. 3, no. 1, pp. 63–83, Jan. 2021.

[25] Y. Xiao, X. Zhang, X. Xu, X. Liu, and J. Liu, "Deep neural networks with Koopman operators for modeling and control of autonomous vehicles," 2020, *arXiv:2007.02219*.

[26] Y. Han, M. Wang, L. Li, C. Roncoli, J. Gao, and P. Liu, "A physics-informed reinforcement learning-based strategy for local and coordinated ramp metering," *Transp. Res. C, Emerg. Technol.*, vol. 137, Apr. 2022, Art. no. 103584.

[27] D. Moser, R. Schmied, H. Waschl, and L. Del Re, "Flexible spacing adaptive cruise control using stochastic model predictive control," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 1, pp. 114–127, Jan. 2018.

[28] L. C. Iacob, G. I. Beintema, M. Schoukens, and R. Tóth, "Deep identification of nonlinear systems in Koopman form," 2021, *arXiv:2110.02583*.

[29] B. O. Koopman, "Hamiltonian systems and transformation in Hilbert space," *Proc. Nat. Acad. Sci. USA*, vol. 17, no. 5, p. 315, 1931.

[30] B. O. Koopman and J. V. Neumann, "Dynamical systems of continuous spectra," *Proc. Nat. Acad. Sci. USA*, vol. 18, no. 3, p. 255, 1932.

[31] M. Williams, I. Kevrekidis, and C. Rowley, "A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition," *J. Nonlinear Sci.*, vol. 25, pp. 1307–1346, Dec. 2015.

[32] E. M. Bollt and N. Santitissadeekorn, *Applied and Computational Measurable Dynamics*. Philadelphia, PA, USA: SIAM, 2013.

[33] G. Froyland, G. A. Gottwald, and A. Hammerlindl, "A computational method to extract macroscopic variables and their dynamics in multiscale systems," *SIAM J. Appl. Dyn. Syst.*, vol. 13, no. 4, pp. 1816–1846, Jan. 2014.

[34] M. Budisic, R. Mohr, and I. Mezic, "Applied koopmanism," *Chaos, Interdiscipl. J. Nonlinear Sci.*, vol. 22, no. 4, Dec. 2012, Art. no. 047510.

[35] A. Mauroy and I. Mezic, "On the use of Fourier averages to compute the global isochrons of (quasi)periodic dynamics," *Chaos, Interdiscipl. J. Nonlinear Sci.*, vol. 22, no. 3, Sep. 2012, Art. no. 033112.

[36] A. Mauroy, I. Mezic, and J. Moehlis, "Isostables, isochrons, and Koopman spectrum for the action–angle representation of stable fixed point dynamics," *Phys. D, Nonlinear Phenomena*, vol. 261, pp. 19–30, Oct. 2013.

[37] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson, "Spectral analysis of nonlinear flows," *J. Fluid Mech.*, vol. 641, pp. 115–127, Dec. 2009.

[38] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," *J. Fluid Mech.*, vol. 656, pp. 5–28, Aug. 2010.

[39] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz, "On dynamic mode decomposition: Theory and applications," *J. Comput. Dyn.*, vol. 1, no. 2, pp. 391–421, 2014.

[40] T. W. Muld, G. Efraimsson, and D. S. Henningson, "Flow structures around a high-speed train extracted using proper orthogonal decomposition and dynamic mode decomposition," *Comput. Fluids*, vol. 57, pp. 87–97, Mar. 2012.

[41] B. Eisenhower, T. Maile, M. Fischer, and I. Mezic, "Decomposing building system data for model validation and analysis using the Koopman operator," in *Proc. Nat. IBPSAUSA Conf.*, New York, NY, USA, 2010, pp. 1–9.

[42] Y. Susuki and I. Mezic, "Nonlinear Koopman modes and coherency identification of coupled swing dynamics," *IEEE Trans. Power Syst.*, vol. 26, no. 4, pp. 1894–1904, Nov. 2011.

[43] E. Ling, L. Zheng, L. J. Ratliff, and S. Coogan, "Koopman operator applications in signalized traffic systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 4, pp. 3214–3225, Apr. 2022, doi: 10.1109/TITS.2020.3032880.

[44] K. R. Shabab, S. Mustavee, S. Agarwal, M. H. Zaki, and S. Das, "Exploring DMD-type algorithms for modeling signalised intersections," 2021, *arXiv:2107.06369*.

[45] V. Cibulka, T. Hanis, and M. Hromcik, "Data-driven identification of vehicle dynamics using Koopman operator," in *Proc. 22nd Int. Conf. Process Control (PC)*, Jun. 2019, pp. 167–172.

[46] M. Korda and I. Mezić, "Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control," *Automatica*, vol. 93, pp. 149–160, Jul. 2018.

[47] I. Mezić, "Spectral properties of dynamical systems, model reduction and decompositions," *Nonlinear Dyn.*, vol. 41, pp. 309–325, Aug. 2005.

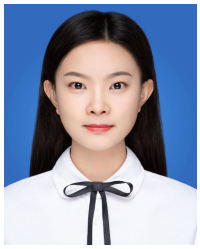
[48] I. Mezic, "Analysis of fluid flows via spectral properties of the Koopman operator," *Annu. Rev. Fluid Mech.*, vol. 45, no. 1, pp. 357–378, Jan. 2013.

[49] B. Fornberg, E. Larsson, and N. Flyer, "Stable computations with Gaussian radial basis functions," *SIAM J. Sci. Comput.*, vol. 33, no. 2, pp. 869–892, Jan. 2011.

[50] T. Zhou, M. L. Kris, D. Creighton, and C. Wu, "GMIX: Graph-based spatial–temporal multi-agent reinforcement learning for dynamic electric vehicle dispatching system," *Transp. Res. C, Emerg. Technol.*, vol. 144, Nov. 2022, Art. no. 103886.

[51] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A K-means clustering algorithm," *Appl. Statist.*, vol. 28, no. 1, pp. 100–108, 1979.

[52] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, 2010, pp. 249–256.



Chuanye Gu received the Ph.D. degree in mathematics and statistics from Curtin University, Australia, in 2022. She is currently a Post-Doctoral Fellow with the School of Management, Guangzhou University. Her current research interests include freeway optimization and its application in the transportation areas.

Tao Zhou received the Ph.D. degree from the School of Design and Built Environment, Curtin University, Australia, in 2019. He is currently a Researcher with the Institute for Intelligent Systems Research and Innovation, Deakin University, Australia. His current research interests include machine learning, deep reinforcement learning, operations research, and intelligent transportation systems.

Changzhi Wu received the Ph.D. degree in optimization and control from Sun Yat-Sen University, China, in 2006. He is currently a Professor with the School of Management, Guangzhou University. He has published more than 100 articles. His current research interests include optimization, intelligent transportation systems, and data analytic.