ASCE

Cell-Trans: A Traffic Prediction Method for Motion Planning of Autonomous Vehicles at Signalized Intersections

Anran Li[1]; Zhenlin Xu[2]; Yuyan (Annie) Pan[3]; Bolin Gao[4]; Jian Zhang[5]; Yanyan Chen[6]; and Yongxing Li[7]

Abstract: Intelligent vehicle cyberphysical systems can integrate real-time traffic scene perception with built-in high-definition maps to construct digital twins of real-world signalized intersections. Based on digital twins, this paper presents a traffic prediction method named cell transformer (cell-trans), comprising vehicle-, cell-, and road-level encoders and a decoder. The vehicle-level encoder first converts vehicle features into vehicle encodings, which the cell-level encoder then fuses with lane segment features to generate cell encodings. Next, the road-level encoder treats the connectivity between lane segments and the phase information at signalized intersections as a dynamic directed graph, extracting spatial-temporal evolution patterns to improve traffic predictions. The cell-trans is compared with baseline models on pNEUMA and CitySim data sets, and the performance comparison validates its optimal predictive accuracy. Moreover, the outstanding performance of the cell-trans is confirmed by ablation study, parameter analysis, and computational efficiency analysis. Finally, this paper develops a cell-trans-based motion planner for autonomous vehicles (AV) in a joint simulation platform combined CARLA and SUMO to indicate its contributions to AVs. **DOI: 10.1061/JTEPBS.TEENG-9105.**

Author keywords: Autonomous vehicle; Signalized intersection; Traffic prediction; Transformer; Vehicle motion planning.

Introduction

The advancement in communication, computing, and sensing technologies has significantly accelerated the development of intelligent vehicle cyberphysical systems (IVCPS) (Gao et al. 2023). IVCPS integrates real-time traffic scene perception from on-board and roadside sensors with built-in high-definition (HD) maps and supplemental traffic information to construct digital twins (Dong et al. 2023). As illustrated in Fig. 1, IVCPS employs a three-layer architecture to represent traffic scenarios through physical, relational, and topological layers (Bao et al. 2022). This system partitions continuous road space into discrete lane segments characterized by precise geometric attributes and rich semantic information (Feng et al. 2023). Through this framework, IVCPS localizes real-time vehicle perception to specific lane segments while dynamically updating segment states via high-frequency data streams, thereby mirroring evolving traffic conditions (Kiran et al. 2022). The system further establishes lane segments as nodal elements, constructing dynamic directed graphs based on lane topology and signalized intersection phase patterns. By synthesizing multimodal traffic data through these mechanisms, IVCPS generates digital twins of signalized intersections that deliver real-time, accurate operational data for traffic prediction algorithms and autonomous vehicle (AV) motion planning.

As a classical traffic model, the cell transmission model (CTM) divides road space into lane segments called cells and constructs the fundamental traffic diagram to predict their states (Daganzo 1995). Derived from the Lighthill–Whitham–Richards (LWR) model, which employed fluid dynamics equations to characterize traffic flow, CTM simplifies the solution process by discretizing the LWR model across spatial-temporal dimensions. This discretization strategy models vehicle motions as sequential transmissions between adjacent cells, effectively reducing computational complexity (Daganzo 1995). Owing to its simplicity and efficiency, CTM has been widely advanced to address traffic prediction demands in diverse scenarios (Adacher and Tiriolo 2018; Mayakuntla and Verma 2019; Liu and Chang 2011). Nevertheless, such model-driven approaches often rely on idealized assumptions that may conflict with real-world traffic dynamics. In contrast, data-driven methods have gained prominence by leveraging real-world data to capture traffic evolution patterns. Early research primarily employed statistical approaches for traffic predictions, though their effectiveness was constrained by sensitivity to outliers (Hamed et al. 1995; Williams and Hoel 2003; Ji et al. 2023). With advancements in computational power, machine-learning algorithms were subsequently developed to extract spatial-temporal patterns from

[1]Ph.D. Candidate, College of Metropolitan Transportation, Beijing Univ. of Technology, Beijing 100124, China. ORCID: https://orcid.org/0009-0000-0483-7287. Email: lianran@emails.bjut.edu.cn

[2]Ph.D. Candidate, Dept. of Transport and Planning, Faculty of Civil Engineering and Geosciences, Delft Univ. of Technology, Delft 2628 CN, Netherlands. ORCID: https://orcid.org/0000-0001-5280-9803. Email: z.xu-28@student.tudelft.nl

[3]Ph.D. Candidate, Dept. of Civil and Environmental Engineering, Pennsylvania State Univ., University Park, PA 16802. Email: yypan@psu.edu

[4]Professor, School of Vehicle and Mobility, Tsinghua Univ., Beijing 100084, China. Email: gaobolin@tsinghua.edu.cn

[5]Professor, College of Metropolitan Transportation, Beijing Univ. of Technology, Beijing 100124, China. Email: jian.zhang@bjut.edu.cn

[6]Professor, College of Metropolitan Transportation, Beijing Univ. of Technology, Beijing 100124, China. Email: cdyan@bjut.edu.cn

[7]Professor, College of Metropolitan Transportation, Beijing Univ. of Technology, Beijing 100124, China (corresponding author). Email: liyx@bjut.edu.cn

Note. This manuscript was submitted on December 26, 2024; approved on June 18, 2025; published online on September 29, 2025. Discussion period open until February 28, 2026; separate discussions must be submitted for individual papers. This paper is part of the ***Journal of Transportation Engineering, Part A: Systems***, © ASCE, ISSN 2473-2907.

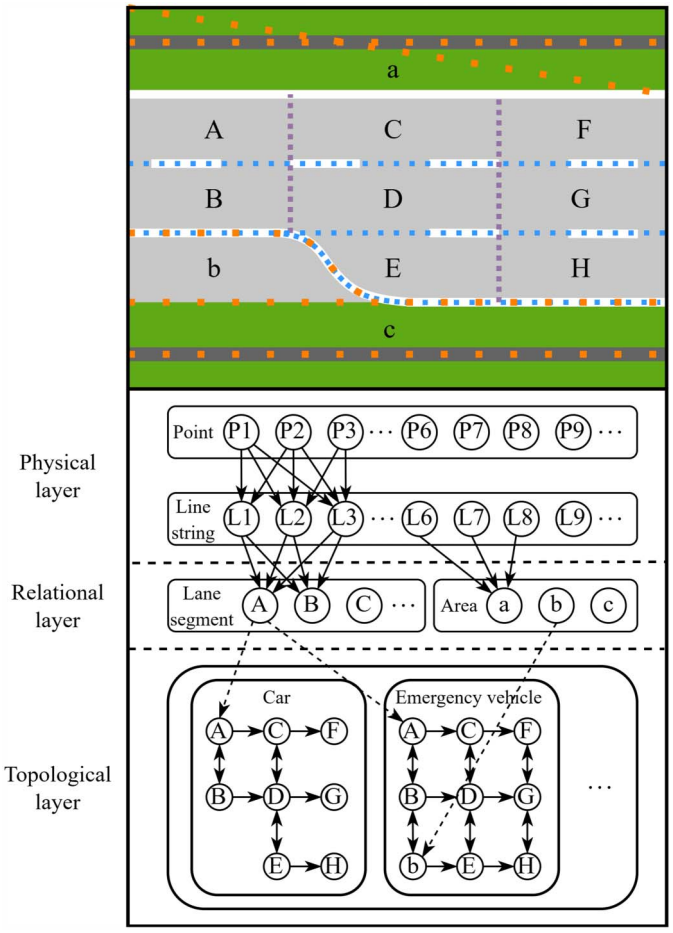

Fig. 1. Tripartite layer structure of HD maps, including the physical, relational, and topological layers.

historical traffic data (Zhang and Liu 2009; Ye et al. 2012). Recently, deep-learning architectures have revolutionized traffic prediction research due to their structural flexibility and superior nonlinear modeling capabilities (Hu et al. 2018; Lu et al. 2023). Meanwhile, hybrid frameworks integrating traffic theory with data-driven methods have demonstrated exceptional performance by combining domain knowledge with learning-based representations (Pan et al. 2022; Guo et al. 2021; Liu et al. 2023). Notably, some advanced hybrids have successfully embedded classical traffic models or statistical methodologies within deep neural networks, achieving simultaneous improvements in prediction accuracy and interpretability (He et al. 2023; Shao and Sun 2021). Despite CTM's structural compatibility with digital twin architectures, its inability to directly process multimodal traffic data severely limits adaptability in complex traffic environments. Thus, a critical challenge lies in developing multimodal data integration mechanisms that synergize with CTM's framework for improved traffic forecasting.

This paper proposes a novel traffic prediction method named cell transformer (cell-trans), which comprises a three-level encoder: vehicle-level; cell-level; and road-level; and a decoder. The cell-trans utilizes the discrete spatial-temporal structure of CTM without employing the fundamental traffic diagram. Instead, it focuses on vehicle transmission patterns between adjacent cells, utilizing the transformer architecture for time-series prediction. In prediction, the vehicle-level encoder first converts vehicle motion and type features into vehicle encodings. Subsequently, the cell-level encoder integrates geometric properties and cell-type features of lane segments into lane segment embeddings, which are combined with vehicle encodings at each time step to produce cell encodings. These dynamic encodings are further fused to form cell encodings at each time-step. The road-level encoder treats different signal phases as dynamic connections of nodes corresponding to specific cells in the road topology graph, which is eigen-decomposed into spatial embeddings of cells at each time-step. Meanwhile, the road-level encoder converts temporal tags into the temporal embedding sequence, which is fused with the spatial embedding sequence to create a spatial-temporal embedding sequence. This sequence is transformed into spatial-temporal embedding matrices to guide multihead attention on cell encodings, yielding the historical encoding sequence (Ye et al. 2022). The decoder utilizes the traffic state from the last time-step as the initial input, encoding it with cell embeddings to generate the advanced decoder input. Similar to the road-level encoder, the decoder generates spatial-temporal embedding matrices to facilitate the translation of the decoder input into the traffic prediction sequence. Compared to existing methods, the cell-trans can utilize a three-level encoder to extract vehicle-level, lane-level, and road-level dimension features from multimodal data and sequentially fuse them to generate predictions of future traffic conditions, achieving accurate traffic forecasting. Moreover, the cell-trans treats cells representing lane segments as nodes in a dynamic topological graph and models phase information as dynamic connections between specific nodes within signalized intersections. Therefore, the cell-trans can comprehend the signal control information in multimodal traffic data and further improve prediction performance. In this way, the cell-trans can sufficiently mine spatial-temporal correlations and dependencies from multimodal traffic data to achieve an accurate traffic prediction at signalized intersections.

This paper extracts real-world traffic data from five signalized intersections in the pNEUMA data set (Barmpounakis and Geroliminis 2020), combined with Lanelet2 maps derived from OpenStreetMap (OSM) (Althoff et al. 2017), to create two dedicated data sets: pNEUMA-128; and pNEUMA-256. Similarly, traffic data from three signalized intersections in the CitySim data set (Zheng et al. 2024) are selected to build the dedicated CitySim-128 data set. The cell-trans is trained and evaluated on these dedicated data sets and benchmarked against baseline models to demonstrate its superior predictive accuracy. Meanwhile, this paper conducts ablation studies to evaluate the contributions of cell-trans's components. Additionally, comprehensive analyses of parameter and computational efficiency are performed to provide a thorough assessment of the cell-trans. Further, the integration of the cell-trans with the Baidu Apollo EM planner is executed to develop a predictive motion planning method on a joint simulation platform based on SUMO and CARLA (Fan et al. 2018), highlighting the potential of the cell-trans for predictive motion planning of AVs at signalized intersections. The contributions of this paper are summarized as follows:

- This paper proposes the cell-trans, a traffic prediction model that extracts vehicle-level motion patterns, cell-level physical configurations, and road-level topological features from multimodal data provided by IVCPS. Through multilevel feature fusion, the cell-trans enables accurate traffic state prediction at signalized intersections.
- The cell-trans treats lane segments as nodes in a dynamic topological graph and models phase information as dynamic connections between nodes representing lane segments within signalized intersections. In this way, the cell-trans can generate spatial-temporal matrices to improve traffic prediction accuracy at signalized intersections.
- The traffic prediction results of the cell-trans can assist in constructing spatial-temporal corridors for AVs at signalized

intersections. These spatial-temporal corridors serve as constraints for motion planning solutions of AVs, enhancing both safety and computational efficiency.

The remainder of this paper is structured as follows. Section "Method" introduces the premise, foundation, and architecture of Cell-Trans. Section "Experiment" describes the experiments and predictive performance evaluation methods. Section "Results and Analysis" presents the experimental results and corresponding analysis. Section "Case Study" presents a case study of a Cell-Trans-based motion planner. Finally, the section "Conclusions" summarizes this paper and looks forward to further research.

Method

Multimodal Traffic Data

In this study, the multimodal traffic data used in our proposed traffic prediction model are provided by IVCPS, which leverages edge cloud infrastructures to integrate perception data from connected vehicles and intelligent roadside units and matches this information with built-in HD maps to generate digital twins of real-time traffic scenarios (Gao et al. 2023). Based on digital twins, IVCPS acquires abundant data for utilization in our traffic prediction model, including vehicular motion data, vehicular semantic labels, inherent characteristics of lane segments, roadway topology structure, and phase control information at signalized intersections (Dong et al. 2023).

IVCPS incorporates HD maps featuring centimeter-level road geometry characterization and semantically annotated traffic elements, with cartographic data primarily acquired through survey vehicles equipped with LiDARs, cameras, inertial measurement units, and global navigation satellite system receivers (Bao et al. 2022). The raw data undergo cleaning, synchronization, multisource fusion, coordinate unification, feature extraction, 3D modeling, semantic annotation, and accuracy correction, ultimately being utilized to construct HD maps. Fig. 1 demonstrates the hierarchical architecture of HD maps, featuring a tripartite layer structure composed of physical, relational, and topological dimensions. The physical layer consists of points, polylines, and polygons, where points store actual location information, polylines describe the shape of map elements, and polygons add custom information. The relational layer includes lane segments and areas, with lane segments defining traffic rules and topological relationships, and areas representing nondirectional or immovable parts, such as road signs and parking spaces. The topological layer describes the dynamic topological relationships between lane segments and areas, influenced by traffic markings, road priorities, and traffic lights.

Meanwhile, IVCPS utilizes edge cloud infrastructures to aggregate and calibrate multisource sensory inputs from connected vehicles and intelligent roadside units equipped with LiDAR and vision sensors, achieving millisecond-latency generation of traffic environment perception (Kiran et al. 2022). Subsequently, IVCPS synthesizes traffic environment perception with built-in HD maps, constructing subsecond updated digital twins that dynamically mirror real-time traffic scenarios. Based on digital twins, IVCPS can package multimodal data in vehicle-, lane-, and road-level dimensions for traffic prediction models to capture spatial-temporal correlations and dependencies for conducting traffic forecasting.

While IVCPS enables multimodal traffic data provision via digital twins, existing prediction methods exhibit limitations in applying these traffic data. Therefore, we propose a traffic prediction model based on the discrete spatial-temporal architecture of digital twins, which sufficiently leverages IVPCS-provided multimodal traffic data to enable accurate forecasting.

Cell Transmission Model

Similar to the spatial-temporal of digital twins, the classical CTM discretizes time into successive steps and partitions road space into lane segments called cells. Each cell's longitudinal dimension exceeds the maximum travel distance of standard vehicles within a time step to prevent them from skipping over adjacent cells. According to traffic flow theory, CTM can progressively evaluate cell states as follows:

$$N_i^{t+1} = N_i^t + Q_i^t - Q_{i+1}^t \tag{1}$$

where N_i^t = vehicle count in cell i at time-step t; and Q_i^t = cumulative inflow to cell i during $[t-1, t]$. The inflow dynamics follow

$$Q_i^t = \min\{N_{i-1}^t, Q_{\text{cap}}, \alpha(N_{\max} - N_i^t)\} \tag{2}$$

$$\alpha = \begin{cases} 1, & N_{i-1}^t \le Q_{\text{cap}} \\ \dfrac{w}{\nu}, & N_{i-1}^t > Q_{\text{cap}} \end{cases} \tag{3}$$

where Q_{cap} = cellular inflow capacity; and $N_{\max}$ = maximum vehicle occupancy; and w and ν = congestion wave speed and free-flow velocity, respectively.

This study retains CTM's spatial-temporal discretization framework while shifting focus from modeling continuous traffic flow to learning traffic evolution patterns using data-driven methods in this discrete spatial-temporal system. To this end, we encode historical vehicular motion data, vehicular semantic information, and inherent characteristics of lane segments into a temporal sequence. Moreover, we convert spatial connectivity of lane segments and phase information at signalized intersections into a dynamic directed graph and embed it into the temporal sequence. On this basis, we apply the transformer architecture to extract spatial-temporal correlations and dependencies from the temporal sequence to predict future cell states. In this way, we synthesize CTM and transformer into the cell-trans, enabling traffic prediction in a discrete spatial-temporal system using multimodal traffic data.

Problem Formulation

The cell-trans defines the historical traffic sequence $X = \{X^1, X^2, \ldots, X^{T_h}\}$, dynamic road graph $G = \{G^1, G^2, \ldots, G^{T_h+T_p}\}$, and temporal tags $T = \{T^1, T^2, \ldots, T^{T_h+T_p}\}$, where T_h and T_p denote historical and predictive time-steps, respectively. In X, a historical state is defined as $X^t = \{X_1^t, X_2^t, \ldots, X_{N_c}^t\}$, where $X_\zeta^t \in X^t$ denotes the state of cell ζ at time-step t, and N_c denotes the number of cells. For cell ζ, X_ζ^t contains the internal vehicle state sequence $V_\zeta^t = \{V_{\zeta,1}^t, V_{\zeta,2}^t \ldots, V_{\zeta,N_\zeta^t}^t\}$, cell direction vector $\theta_\zeta \in [0, 2\pi]$, and semantic tag $s_\zeta \in \mathbb{R}^{d_c}$, where $V_{\zeta,i}^t \in V_\zeta^t$ denotes the state of vehicle i in cell ζ at time-step t, N_ζ^t denotes the internal vehicle count of cell ζ at time-step t, s_ζ denotes the one-hot encoding to classify cell ζ, and d_c denotes the dimension to count different cell types. For vehicle i, $V_{\zeta,i}^t$ is defined as $V_{\zeta,i}^t = \{p_i^t, \nu_i^t, \theta_i^t, s_i^t\}$, where $p_i^t \in \mathbb{R}^2$ denotes the position coordinate of vehicle i at time-step t, $\nu_i^t \in \mathbb{R}$ denotes instantaneous velocity, $\theta_i^t \in [0, 2\pi]$ denotes instantaneous azimuth, and $s_i^t \in \mathbb{R}^{d_\nu}$ denotes the one-hot encoding to classify vehicle i among d_ν different vehicle types. Meanwhile, the map service also defines the tth road topology as $G^t = \{C, E^t, A^t\}$, where C denotes the cell set on the road, E^t denotes the directed edge set for vehicle transmission between adjacent cells at time step t, and $A^t \in \mathbb{R}^{N_c \times N_c}$ denotes the adjacency matrix of E^t:

$$A^t(\zeta,\xi)=\begin{cases}1, & \text{if } (c_\zeta,c_\xi)\in E^t,\\ 0, & \text{otherwise,}\end{cases}\quad \text{for } \zeta,\xi=1,2,\ldots,N_c \tag{4}$$

where c_ζ = cell ζ in C. Moreover, the map service provides the one-hot encoding $T^t\in\mathbb{R}^{d_t}$ for the cell-trans to identify the position of time-step t in a week, where d_t denotes the dimension to count different temporal tags.

The objective of the cell-trans is to predict the future state sequence $Y\in\{Y^{T_h+1},Y^{T_h+2},\ldots,Y^{T_h+T_p}\}$, where $Y^{t'}\in\mathbb{R}^{N_c}$ denotes the number of vehicles in each cell at time-step t'. In prediction, the cell-trans creates the initial predictive state sequence as $\hat{Y}_{(0)}=\{Y^{T_h}\}$ and predicts the next state $\hat{Y}^{T_h+1}\in\mathbb{R}^{N_c}$. Then, the cell-trans adds $\hat{Y}^{T_h+1}$ into $\hat{Y}_{(0)}$ and updates $\hat{Y}_{(0)}\in\mathbb{R}^{1\times N_c}$ to $\hat{Y}_{(1)}\in\mathbb{R}^{2\times N_c}$. After T_p predicting rounds, the cell-trans generates the T_pth predictive state sequence $\hat{Y}_{(T_p)}\in\mathbb{R}^{(T_p+1)\times N_c}$ and deletes Y^{T_h} from $\hat{Y}_{(T_p)}$ to acquire the final predictive state sequence $\hat{Y}\in\mathbb{R}^{T_p\times N_c}$.

Prediction Model

Model Overview

Fig. 2(a) presents the architecture of the cell-trans, which comprises a three-level encoder system, i.e., vehicle-, cell-, and road-level encoders, along with a decoder. Once receiving input, the vehicle-level encoder processes the motion and vehicle-type features, generating vehicle encodings. Subsequently, the cell-level encoder transforms direction and cell-type features into cell embeddings. It employs multihead attention to integrating cell embeddings and vehicle encodings at each time-step, resulting in dynamic cell features that are encoded with the cell embeddings to produce a cell encoding sequence. The road-level encoder then conducts eigen-decomposition on the dynamic road graph to derive eigenmaps at each time step, which serve as spatial embeddings. Additionally, it converts temporal tags, augmented with position encodings, into temporal embeddings. The road-level encoder concatenates the spatial and temporal embeddings to form spatial-temporal embedding matrices, which guide the multihead attention mechanism applied to the cell encoding sequence, thereby generating a historical encoding sequence. The decoder initiates with the traffic state from the previous time-step as its initial input, which is encoded with cell embeddings to create an advanced decoder sequence. Similar to the road-level encoder, the decoder generates spatial-temporal embedding matrices to direct masked multihead attention on the advanced decoder sequence, resulting in a predictive encoding sequence. Ultimately, the decoder utilizes multihead attention to translate the predictive and historical encoding sequences into the next state prediction, which is incorporated into the decoder input for subsequent decoding iterations until the prediction process is complete. The following sections will provide a comprehensive overview of the key functional modules of the cell-trans.

Vehicle-Level Encoder

Fig. 2(b) illustrates the vehicle-level encoder that encodes the state of vehicle i in cell ζ at time step t. The encoder first calculates the angle $\Delta\theta^t_{\zeta,i}$ between θ^t_i and θ_ζ then decomposes the velocity ν^t_i into a velocity vector $\nu^t_{\zeta,i}\in\mathbb{R}^2$, consisting of the radial component $\nu^t_i\cdot\sin\Delta\theta^t_{\zeta,i}\in\mathbb{R}$ and lateral velocity component $\nu^t_i\cdot\cos\Delta\theta^t_{\zeta,i}\in\mathbb{R}$. Next, linear transformations are applied to $\nu^t_{\zeta,i}$ and s^t_i to acquire motion feature $\hat{\nu}^t_{\zeta,i}\in\mathbb{R}^{d_h}$ and vehicle-type feature $\hat{s}^t_i\in\mathbb{R}^{d_h}$, where d_h denotes the hidden dimension. Next, the vehicle-level encoder employs a multilayer perceptron (MLP) to encode $\hat{\nu}^t_{\zeta,i}$ and $\hat{s}^t_i$

$$\epsilon^t_{\zeta,i}=\text{LayerNorm}(ReLU(W^{\nu e}_2\cdot ReLU(W^{\nu e}_1\cdot[\hat{\nu}^t_{\zeta,i},\hat{s}^t_i]+b^{\nu e}_1)+b^{\nu e}_2)) \tag{5}$$

where $\epsilon^t_{\zeta,i}\in\mathbb{R}^{d_h}$ = the encoding of vehicle i at time-step t; and $W^{\nu e}_1\in\mathbb{R}^{d_h\times d_h}$, $W^{\nu e}_2\in\mathbb{R}^{d_h\times d_h}$, $b^{\nu e}_1\in\mathbb{R}^{d_h}$, and $b^{\nu e}_2\in\mathbb{R}^{d_h}$ = learnable parameters. This process transforms vehicle states into encodings, enabling the cell-trans to learn the precursors of vehicle lane changes and differentiate between vehicle types.

Cell-Level Encoder

Fig. 2(c) illustrates the cell-level encoder that encodes the state of cell ζ at time-step t. Similar to the vehicle-level encoder, the cell-level encoder first performs decomposition and linear transformation on θ_ζ to produce the geometric feature $\hat{\theta}_\zeta\in\mathbb{R}^{d_h}$. It then converts s_ζ into the cell-type feature $\hat{s}_\zeta\in\mathbb{R}^{d_h}$ and uses an MLP to encode $\hat{\theta}_\zeta$ and $\hat{s}_\zeta$

$$\varepsilon_\zeta=\text{LayerNorm}(ReLU(W^{se}_2\cdot ReLU(W^{se}_1\cdot[\hat{\theta}_\zeta,\hat{s}_\zeta]+b^{se}_1)+b^{se}_2)) \tag{6}$$

where $\varepsilon_\zeta\in\mathbb{R}^{d_h}$ = the embedding of cell ζ; and $W^{se}_1\in\mathbb{R}^{d_h\times d_h}$, $W^{se}_2\in\mathbb{R}^{d_h\times d_h}$, $b^{se}_1\in\mathbb{R}^{d_h}$, and $b^{se}_2\in\mathbb{R}^{d_h}$ = learnable parameters.

Additionally, the cell-level encoder integrates the internal vehicle encodings of cell ζ at time-step t. To capture the sequential order of internal vehicles, the cell-level encoder first incorporates the position encoding into $\epsilon^t_{\zeta,i}$

$$\text{PE}(\text{pos},j)=\begin{cases}\sin\left(\dfrac{\text{pos}}{10{,}000^{\frac{j}{d_h}}}\right), & \text{if } j \text{ is even}\\ \cos\left(\dfrac{\text{pos}}{10{,}000^{\frac{j-1}{d_h}}}\right), & \text{if } j \text{ is odd}\end{cases} \tag{7}$$

$$\hat{\epsilon}^t_{\zeta,i}=\epsilon^t_{\zeta,i}\oplus\text{PE} \tag{8}$$

where $\text{PE}\in\mathbb{R}^{d_h}$ represents the position encoding; pos = the sequential order of vehicle i in the queue within cell ζ at time-step t, and $\oplus$ = elementwise tensor addition. The cell-level encoder transforms ε_ζ and $\hat{\epsilon}^t_{\zeta,i}$ into the query vector $q_\zeta\in\mathbb{R}^{d_h}$, the key vector $k^t_{\zeta(i)}\in\mathbb{R}^{d_h}$, and the value vector $v^t_{\zeta(i)}\in\mathbb{R}^{N_h\times d_k}$. It then performs multihead attention using q_ζ, $k^t_{\zeta(i)}$, and $v^t_{\zeta(i)}$

$$\text{head}_h=\text{Softmax}\left(\frac{q_\zeta k^{t\intercal}_{\zeta(i)}}{\sqrt{d_k}}\right)v^t_{\zeta(i)} \tag{9}$$

$$\varepsilon^t_{\zeta(i)}=\text{Concat}(\text{head}_1,\text{head}_2,\ldots,\text{head}_{N_h})W^0 \tag{10}$$

where $\varepsilon^t_{\zeta(i)}\in\mathbb{R}^{d_h}$ = the ith component of the dynamic feature of cell ζ at time-step t; N_h = the number of attention heads; d_k = the attention dimension; and W^0 = a learnable parameter. Finally, the cell-level encoder combines all dynamic feature components and employs an MLP to generate the cell encoding

$$\varepsilon^t_\zeta=\sum_{i=1}^{N^t_\zeta}\varepsilon^t_{\zeta(i)} \tag{11}$$

$$\hat{\varepsilon}^t_\zeta=\text{LayerNorm}(ReLU(W^{ce}_2\cdot ReLU(W^{ce}_1\cdot[\varepsilon_\zeta,\varepsilon^t_\zeta]+b^{ce}_1)+b^{ce}_2)) \tag{12}$$

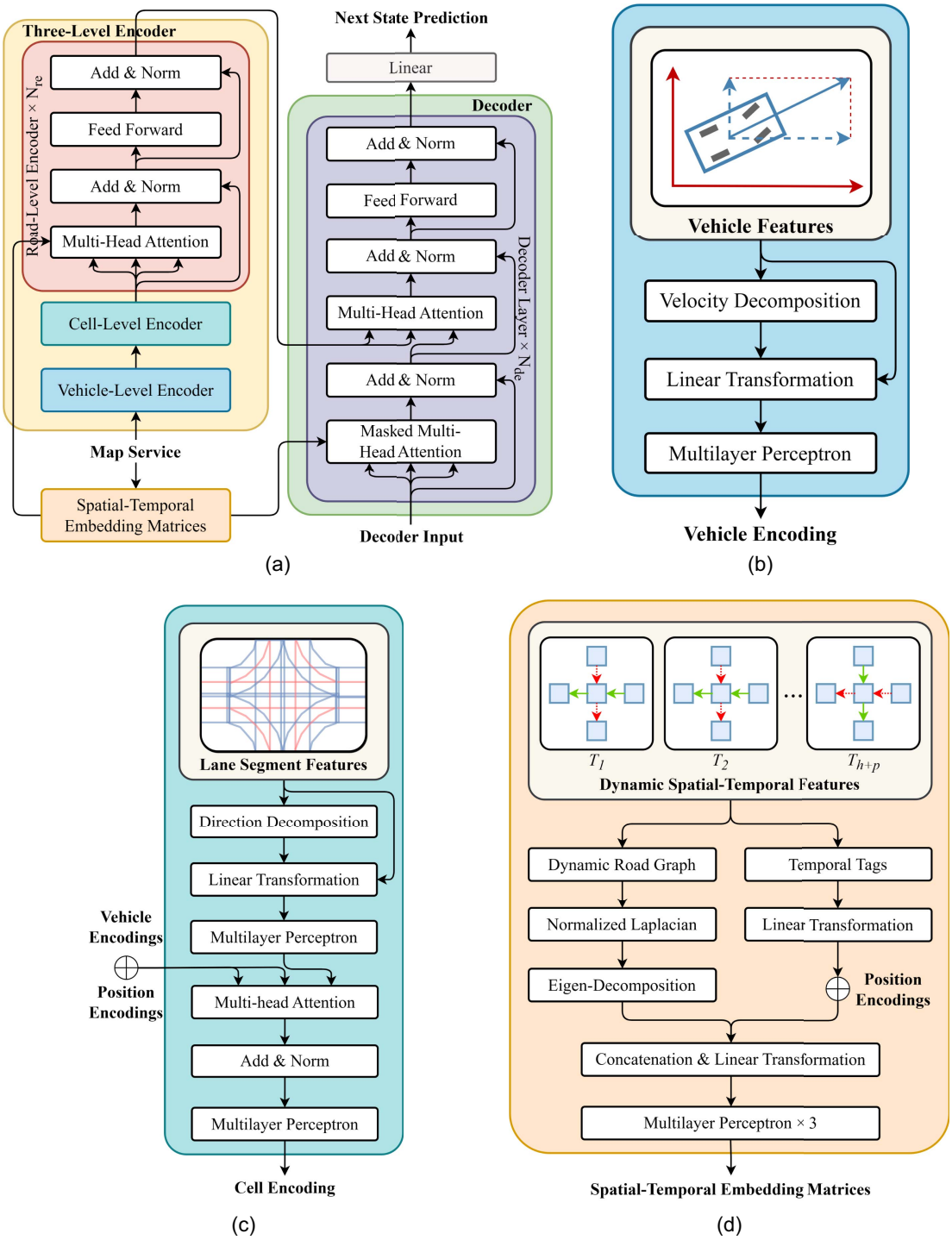

Fig. 2. Architecture of the cell-trans and its key functional components, including the vehicle-level encoder, the cell-level encoder, and the spatial-temporal embedding matrices: (a) the overall architecture of cell-trans; (b) the vehicle-level encoder; (c) the cell-level encoder; and (d) the spatial-temporal information embedding.

where $\hat{\varepsilon}_\zeta^t \in \mathbb{R}^{d_h}$ = the encoding of cell ζ at time-step t; and $W_1^{ce} \in \mathbb{R}^{d_h \times d_h}$, $W_2^{ce} \in \mathbb{R}^{d_h \times d_h}$, $b_1^{ce} \in \mathbb{R}^{d_h}$, and $b_2^{ce} \in \mathbb{R}^{d_h}$ = learnable parameters.

Road-Level Encoder

As illustrated in Fig. 2(d), the road-level encoder uses $\hat{\varepsilon}^t = \{\hat{\varepsilon}_1^t, \hat{\varepsilon}_2^t, \ldots, \hat{\varepsilon}_{N_c}^t\}$ to represent the encoding of all cells at time-step t and incorporates the tth spatial-temporal information into $\hat{\varepsilon}^t$. Specifically, the road-level encoder first obtains A^t from $G^t \in G$ and symmetrizes A^t as follows:

$$\hat{A}^t(\zeta, \xi) = \max(A^t(\zeta, \xi), A^t(\xi, \zeta)) \tag{13}$$

where $\hat{A}^t \in \mathbb{R}^{N_c \times N_c}$ = the symmetric matrix of A^t. Based on $\hat{A}^t$, the road-level encoder constructs the normalized Laplacian matrix $L^t \in \mathbb{R}^{N_c \times N_c}$ and performs eigen-decomposition on L^t

$$L^t = I - D^t - \frac{1}{2}\hat{A}^t D^t - \frac{1}{2} \tag{14}$$

$$L^t = U^t \Lambda^t U^{t\intercal} \tag{15}$$

where $I \in \mathbb{R}^{N_c \times N_c}$ = the identity matrix; $D^t \in \mathbb{R}^{N_c \times N_c}$ = the diagonal degree matrix of $\hat{A}^t$; and $\Lambda^t \in \mathbb{R}^{N_c \times N_c}$ and $U^t \in \mathbb{R}^{N_c \times N_c}$ = the eigenvalue matrix and eigenvector matrix, respectively. From U^t, the road-level encoder extracts K principal components of each eigenvector as $u^t \in \mathbb{R}^{N_c \times K}$, which is then linearly transformed into the tth eigenmap $\hat{u}^t \in \mathbb{R}^{N_c \times d_h}$. The eigenmaps capture the directed channel for vehicle transmission between adjacent cells and the signal phase of the intersection at time-step t.

Meanwhile, the road-level encoder expands and transforms T^t to generate the tth temporal encoding $\tau^t \in \mathbb{R}^{N_c \times d_h}$. Following Eqs. (7) and (8), the road-level encoder combines τ^t with a position encoding $PE \in \mathbb{R}^{N_c \times d_h}$ to obtain $\hat{\tau}^t \in \mathbb{R}^{N_c \times d_h}$. Then, $\hat{u}^t$ and $\hat{\tau}^t$ are concatenated and linearly transformed into the tth spatial-temporal embedding $w^t \in \mathbb{R}^{N_c \times d_h}$. Next, the road-level encoder incorporates the spatial-temporal embedding sequence $w = \{w^1, w^t, \ldots, w^{T_h}\}$, which is converted into spatial-temporal embedding matrices $W_q \in \mathbb{R}^{T_h \times N_c \times d_{st}}$, $W_k \in \mathbb{R}^{T_h \times N_c \times d_{st}}$, and $W_v \in \mathbb{R}^{T_h \times N_c \times d_{st}}$ by three independent MLPs. Afterward, the road-level encoder linearly transforms $W_k \cdot \hat{\varepsilon}$, $W_q \cdot \hat{\varepsilon}$, and $W_v \cdot \hat{\varepsilon}$ into the query vector $q \in \mathbb{R}^{T_h \times N_c \times d_h}$, key vector $k \in \mathbb{R}^{T_h \times N_c \times d_h}$, and value vector $v \in \mathbb{R}^{T_h \times N_c \times d_h}$, where $\hat{\varepsilon} = \{\hat{\varepsilon}^1, \hat{\varepsilon}^2, \ldots, \hat{\varepsilon}^{T_h}\}$ represents the cell encoding sequence. Following Eqs. (19) and (20), the road-level encoder utilizes multihead attention to encode q, k, and v into the historical encoding sequence $h \in \mathbb{R}^{T_h \times N_c \times d_h}$.

Decoder

In t'th decoding round, the decoder takes $\hat{Y}_{(t')} \in \mathbb{R}^{(t'+1) \times N_c}$ as the decoder input and conducts expansion and linear transformation on $\hat{Y}_{(t')}$ to generate $\delta \in \mathbb{R}^{(t'+1) \times N_c \times d_h}$. Then, the decoder employs an MLP to integrate ε_ζ and $\delta_\zeta^{t'} \in \delta$

$$\hat{\delta}_\zeta^{t'} = \text{LayerNorm}(ReLU(W_2^{de} \cdot ReLU(W_1^{de} \cdot [\varepsilon_\zeta, \delta_\zeta^{t'}] + b_1^{de}) + b_2^{de})) \tag{16}$$

where $\hat{\delta}_\zeta^{t'} \in \mathbb{R}^{d_h}$ = the state feature of cell ζ at time-step t'; and $W_1^{de} \in \mathbb{R}^{d_h \times d_h}$, $W_2^{de} \in \mathbb{R}^{d_h \times d_h}$, $b_1^{de} \in \mathbb{R}^{d_h}$, and $b_2^{de} \in \mathbb{R}^{d_h}$ = learnable parameters. Following Eqs. (13)–(15), the decoder generates the spatial-temporal embedding matrices $W_q \in \mathbb{R}^{(t'+1) \times N_c \times d_h}$, $W_k \in \mathbb{R}^{(t'+1) \times N_c \times d_h}$, and $W_v \in \mathbb{R}^{(t'+1) \times N_c \times d_h}$. Next, the decoder converts $W_q \cdot \hat{\delta}$, $W_k \cdot \hat{\delta}$, and $W_v \cdot \hat{\delta}$ into the query vector $q \in \mathbb{R}^{(t'+1) \times N_c \times d_h}$, key vector $k \in \mathbb{R}^{(t'+1) \times N_c \times d_h}$, and $v \in \mathbb{R}^{(t'+1) \times N_c \times d_h}$ and utilizes masked multihead attention to encode q, k, and v

$$\text{head}_h = \text{Softmax}\left(\frac{qk^\intercal}{\sqrt{d_k}} + M\right)v \tag{17}$$

$$\sigma = \text{Concat}(\text{head}_1, \text{head}_2, \ldots, \text{head}_{N_h})W^1 \tag{18}$$

where $\sigma \in \mathbb{R}^{(t'+1) \times N_c \times d_h}$ = the predictive encoding sequence; M = a masked matrix to prevent future information leakage; and W^1 = a learnable parameter. Afterward, the decoder conducts linear transformation on σ and h to generate the query vector $q' \in \mathbb{R}^{t' \times N_c \times d_h}$, key vector $k' \in \mathbb{R}^{T_h \times N_c \times d_h}$, and value vector $v' \in \mathbb{R}^{T_h \times N_c \times d_h}$; further, it utilizes multihead attention to decode q', k', and v' to generate the decoding sequence $\hat{\sigma} \in \mathbb{R}^{(t'+1) \times N_c \times d_h}$ by Eqs. (19) and (20). Finally, $\hat{\sigma}$ is linearly transformed into the next state prediction $\hat{Y}^{t'+1} \in \mathbb{R}^{N_c}$, which is incorporated into $\hat{Y}_{(t')}$ and inputted into the $t'+1$ decoding round if t' is less than T_p.

Experiment

Data Sets

To evaluate the performance of the proposed traffic prediction model, we develop three dedicated data sets (pNEUMA-128, pNEUMA-256, and CitySim-128) derived from two large-scale open-source traffic data sets: pNEUMA and CitySim (Barmpounakis and Geroliminis 2020; Zheng et al. 2024). The following text provides a detailed explanation of these dedicated data sets.

- The pNEUMA data set comprises over 500,000 ground-truth trajectories recorded at 25 FPS in the central business district of Athens during morning peak hours. In this study, we extract traffic data from the pNEUMA data set, comprising three two-phase signalized intersections and two four-phase signalized intersections, and standardize their update frequency to 1 FPS to construct dedicated data sets. Since the pNEUMA data set cannot provide official maps, we acquire corresponding map data from OSM and utilize CommonRoad's map converter to convert them into Lanelet2 format (Althoff et al. 2017; Feng et al. 2023). In this study, we configure the segments of the lane representing the Lanelet2 maps of cells with a standardized length of 15 ± 2.5 m and a width corresponding to the dimensions of a single lane so that one cell can contain a bus or heavy vehicle. On this basis, we apply an unscented Kalman filter to smooth traffic data and perform spatial-temporal alignment with the Lanelet2 maps (Siebinga 2021). This processing pipeline generates two dedicated data sets: pNEUMA-128 featuring 128 cells per scenario; and pNEUMA-256 containing 256 cells per scenario. Both dedicated data sets encompass various traffic conditions such as low-density, high-density, and congestion.
- The CitySim data set includes data from five signalized intersections in Orlando, with 120 min of trajectory data per intersection recorded by drones at 30 FPS. From the CitySim data set, we select three four-phase signalized intersections and standardize their update frequency to 1 FPS for constructing dedicated data sets. Similar to pNEUMA-128, we construct Lanelet2 maps and conduct spatial-temporal alignment to match them with smoothed traffic data to create a dedicated data set named CitySim-128 with traffic scenes containing 128 cells.

Implementations

In this study, we define the cell-type dimension d_c as 4 to categorize upstream, straight, left-turn, and right-turn cells, and the vehicle-type dimension d_v as 6 to classify buses, cars, heavy vehicles, medium-sized vehicles, motorcycles, and taxis. The temporal dimension d_t is set to 48 to differentiate between weekdays, weekends, and various day-hours. The historical time steps T_h and predictive time-steps T_p are set to 7 and 20, respectively. The hidden dimension d_h, number of attention heads N_h, and head dimension d_k are set to 64, 8, and d_h/N_h, respectively.

Cell-Trans is deployed on an NVIDIA DGX-1 server equipped with an Intel Xeon Gold 5218 CPU (Intel, Santa Clara, California) and an NVIDIA Tesla V100S GPU (NVIDIA, Santa Clara, California), running on the Ubuntu 22.04.2 LTS operating system. It is implemented using the PyTorch 1.5 framework in a Python 3.8 environment. The Adam optimizer is used for training the cell-trans, with a learning rate of 10^{-3}, batch size of 32, β_1 set to 0.9, β_2 set to 0.99, and epsilon set to 10^{-8} (Liu et al. 2020).

Baseline Models

To evaluate prediction performance, the cell-trans is compared against nine baseline models, which include one statistical method, two machine-learning algorithms, and six deep-learning models:

- Autoregressive integrated moving average (ARIMA) (Hamed et al. 1995): This statistical model uses historical traffic data to determine the autoregressive, differencing, and moving average orders, fitting a regression model by minimizing the Akaike information criterion.
- Support vector regression (SVR) (Drucker et al. 1996): Implemented with Scikit-learn, this model constructs a hyperplane in the kernel space based on historical traffic states and learns a linear regression model using the radial basis function kernel.
- Random forest (RF) (Athey et al. 2019): Also implemented with Scikit-learn, this meta-regressor consists of 10 decision trees and uses averaging to enhance predictive accuracy and control overfitting.
- Feedforward neural network (FNN) (Ye et al. 2012): This multilayer perceptron (MLP) has one hidden layer with 256 neurons and predicts future traffic states using the cell space states from the previous time steps.
- Adaptive graph convolutional recurrent network (AGCRN) (Xu et al. 2023): This model integrates cell-adaptive parameters into a graph convolutional network (GCN) to learn spatial dependencies and uses a recurrent neural network (RNN) to capture temporal correlations, based on open-source codes.
- Graph WaveNet (GWN) (Wu et al. 2019): Utilizing stacked dilated convolutions, this model captures long-term temporal correlations and constructs an adaptive dependency matrix through learnable cell encoding to capture spatial dependencies, based on open-source codes.
- Propagation delay-aware dynamic long-range transformer (PDFormer) (Jiang et al. 2023): This model utilizes attention mechanisms to construct semantic spatial attention, geographic spatial attention, and temporal self-attention to capture spatial-temporal dependencies, based on open-source codes.
- Spatio-temporal memory augmented multilevel attention network (ST-MAN) (Liu et al. 2024): This model combines attention mechanisms and convolutional long-short memory (convLSTM) networks to encode spatial-temporal sequence and semantic information to conduct traffic predictions, based on open-source codes.
- FlashST (Li et al. 2024): This model utilizes graph neural networks (GNNs) to create a spatial dependence encoder and applies gating mechanisms to construct a temporal dependence encoder. Concatenating spatial and temporal dependence encoders, this model extracts spatial-temporal correlations and dependencies to predict traffic states, based on open-source codes.

Metrics

This paper employs the mean absolute error (MAE) and the root mean square error (RMSE) to evaluate the cell-trans against baseline models

$$\mathrm{MAE} = \frac{1}{W \times H} \sum_{g=1}^{W \times H} |Y_g - \hat{Y}_g| \tag{19}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{W \times H} \sum_{g=1}^{W \times H} (Y_g - \hat{Y}_g)^2} \tag{20}$$

MAE measures the average absolute difference between predicted and actual values, indicating the model's overall prediction error while exhibiting low sensitivity to outliers. In contrast, RMSE calculates the square root of the average of the squared differences between predicted and actual values, which places greater emphasis on larger errors and demonstrates a higher sensitivity to outliers. This distinction underscores that MAE provides a more robust evaluation when anomalies are present, whereas RMSE is more sensitive to significant deviations. The model with the lowest MAE and RMSE indicates superior prediction performance.

Table 1. Performance comparison between the cell-trans and baseline models on pNEUMA-128 and CitySim-128

	pNEUMA-128			CitySim-128		
Model	MAE	RMSE	ACC (%)	MAE	RMSE	ACC (%)
ARIMA	0.509	1.007	69.91	0.506	1.253	68.93
RF	0.417	0.815	78.95	0.423	0.960	77.11
SVR	0.434	0.849	77.81	0.448	0.936	75.45
FNN	0.334	0.656	84.89	0.360	0.764	83.76
AGCRN	0.319	0.631	86.71	0.355	0.759	85.42
GWN	0.305	0.597	88.34	0.336	0.717	87.02
PDFormer	0.289	0.558	89.57	0.321	0.682	87.89
ST-MAN	0.277	0.542	90.14	0.315	0.661	88.63
FlashST	*0.274*	*0.537*	*90.84*	*0.308*	*0.657*	*88.90*
Cell-Trans	**0.268**	**0.524**	**91.90**	**0.304**	**0.646**	**89.81**

Note: The best result is highlighted in bold; the second-best result is labeled with an italic.

Results and Analysis

Prediction Comparison

Table 1 compares the predictive performance of the cell-trans and baseline models on pNEUMA-128 and CitySim-128. The results demonstrate that the cell-trans consistently surpasses the best baseline model across all metrics, including MAE, RMSE, and ACC for both data sets. Compared to the best baseline model, The cell-trans achieves improvements of 2.19%, 2.42%, and 1.19% in MAE, RMSE, and ACC in pNEUMA-128. Similarly, on CitySim-128, the cell-trans achieves enhancements of 1.29%, 1.67%, and 1.02% in MAE, RMSE, and ACC, respectively. The comparison underscores the effectiveness of the cell-trans in traffic prediction, as well as its robustness, demonstrated by its consistent superior performance across diverse data sets.

ARIMA identifies periodic traffic patterns from historical data to predict traffic states at signalized intersections, but it struggles to detect sudden changes, diminishing its accuracy for long-term forecasts. In contrast, SVR and RF use historical data to extract traffic evolution patterns, yet they fail to comprehend spatial relationships between different lane segments, limiting their predictive capabilities. FNN employs a fully connected network with one hidden layer to extract features from historical data. AGCRN enhances this by integrating GCN and RNN to capture spatial-temporal features to improve prediction performance across data sets. Similarly, GWN utilizes GCN with an adaptive dependency matrix and dilated causal convolutions to enhance accuracy and manage long sequences. PDFormer leverages attention mechanisms to capture temporal, geographic, and semantic correlations, enhancing its ability to capture complex spatial-temporal dependencies for improved prediction accuracy. ST-MAN integrates attention mechanisms for spatial feature extraction and ConvLSTM for temporal modeling, then fuses them to achieve accurate traffic forecasting. FlashST constructs a spatial-temporal encoder using GNNs and gating mechanisms, enabling it to effectively extract sequential patterns and semantic relationships from input data, thereby generating

future traffic states. However, these methods inadequately account for the influence of vehicular motion features and signal changes at intersections on lane segment interactions, which limits their accuracy improvements.

Compared to baseline models, the cell-trans fuses vehicle- and cell-type information into historical traffic sequences, enabling it to differentiate between various vehicles and lane segments and understand their specific impacts on traffic. Additionally, it adapts to signal changes using dynamic road graphs and learns distinct traffic patterns based on signal phases, thereby enhancing prediction accuracy. By recognizing temporal changes in traffic patterns, the cell-trans further refines its predictions.

Ablation Studies

This study develops several variants to evaluate the key components of the cell-trans by alternatively removing each component from the complete model. Due to the lack of vehicle-type information in CitySim-128, the ablation studies comparing the cell-trans and its variants are conducted using pNEUMA-128. Except for the removed components, the variants employ the same experimental setups as the complete model. These components contain the vehicle-type component (VTC), vehicle-azimuth component (VAC), cell-type component (CTC), cell-direction component (CDC), spatial embedding component (SEC), and temporal embedding component (TEC).

Table 2 evaluates the contributions of key components to improving traffic predictions, indicating the robustness of the cell-trans model under partial inaccessibility of input. Performance degradation caused by the removal of VTC and VAC reflects scenarios in which vehicle motion and semantic information become compromised due to vehicle sensor failures. Specifically, the exclusion of VTC leads to a 7.46% MAE increase, 7.63% RMSE rise, and 1.45% ACC drop, while the removal of VAC results in 2.20% MAE deterioration, 2.89% RMSE growth, and 0.82% ACC decline, demonstrating the significant impact of vehicle sensing information on the predictive performance of the cell-trans. Ablation of CTC triggers the deterioration of 4.48% MAE deterioration, an increase in 4.39% RMSE, and a reduction of 1.28% ACC, while removal of CDC causes an increase in 1.49% MAE, a rise of 1.72% RMSE and a decrease of 0.55% ACC. These results quantify the adaptive capacity of the cell-trans under incomplete geometric and semantic features of lane segments. SEC exhibits the strongest individual impact, with its exclusion inducing an increase of 10.44% MAE, a growth of 10.87% RMSE, and a deterioration of 3.18% ACC, corresponding to the lack of connectivity of lane segments and phase information at signalized intersections. The removal of TEC reveals the influence of temporal encoding through 5.22% MAE degradation, 5.35% increase in RMSE, and 1.45% decrease in ACC, reflecting the decline in the prediction performance of the cell-trans ns due to insufficient extraction of temporal information. It is worth noting that the combination of key components has a synergistic effect, such as the removal of SEC and TEC, which achieves a 2.03% MAE increase, 1.75% RMSE rise, and 0.56% ACC drop over standalone SEC. This synergistic effect is also reflected in the removal of multiple component combinations, indicating that the complete model can comprehensively utilize multimodal input data to further improve prediction performance. The complete model outperforms the weakest variant with 17.53% MAE reduction, 17.86% RMSE decline, and 6.45% ACC enhancement, confirming that its hierarchical fusion mechanism effectively extracts spatial-temporal correlations and dependencies from input information to improve prediction accuracy. These ablation studies not only verify that the key components of the

Table 2. Ablation studies on key components on pNEUMA-128

Ablated components								
VTC	VAC	CTC	CDC	SEC	TEC	MAE	RMSE	ACC (%)
N	N	N	N	N	N	0.325	0.638	85.98
Y	N	N	N	N	N	0.319	0.625	86.67
N	Y	N	N	N	N	0.324	0.636	86.09
N	N	Y	N	N	N	0.320	0.627	86.72
N	N	N	Y	N	N	0.323	0.633	86.23
N	N	N	N	Y	N	0.314	0.617	87.14
N	N	N	N	N	Y	0.321	0.629	86.53
Y	Y	N	N	N	N	0.318	0.624	86.79
Y	N	Y	N	N	N	0.316	0.620	86.98
Y	N	N	Y	N	N	0.317	0.622	86.87
Y	N	N	N	Y	N	0.310	0.609	87.56
Y	N	N	N	N	Y	0.316	0.619	87.03
N	Y	Y	N	N	N	0.320	0.625	86.69
N	Y	N	Y	N	N	0.323	0.632	86.29
N	Y	N	N	Y	N	0.309	0.604	87.90
N	Y	N	N	N	Y	0.315	0.617	86.51
N	N	Y	Y	N	N	0.321	0.630	86.47
N	N	Y	N	Y	N	0.312	0.612	87.35
N	N	Y	N	N	Y	0.318	0.625	86.72
N	N	N	Y	Y	N	0.309	0.606	87.65
N	N	N	Y	N	Y	0.318	0.623	86.88
N	N	N	N	Y	Y	0.304	0.598	88.14
Y	Y	Y	N	N	N	0.314	0.617	87.15
Y	Y	N	Y	N	N	0.317	0.621	86.93
Y	Y	N	N	Y	N	0.302	0.594	88.62
Y	Y	N	N	N	Y	0.312	0.612	87.41
Y	N	Y	Y	N	N	0.315	0.618	87.08
Y	N	Y	N	Y	N	0.300	0.589	88.58
Y	N	Y	N	N	Y	0.309	0.605	87.75
Y	N	N	Y	Y	N	0.298	0.585	88.74
Y	N	N	Y	N	Y	0.307	0.602	87.83
Y	N	N	N	Y	Y	0.296	0.579	89.03
N	Y	Y	Y	N	N	0.312	0.609	87.57
N	Y	Y	N	Y	N	0.297	0.582	88.90
N	Y	Y	N	N	Y	0.306	0.599	88.06
N	Y	N	Y	Y	N	0.301	0.590	88.57
N	Y	N	Y	N	Y	0.310	0.607	87.61
N	Y	N	N	Y	Y	0.299	0.586	88.72
N	N	Y	Y	Y	N	0.295	0.577	89.18
N	N	Y	Y	N	Y	0.308	0.604	87.75
N	N	Y	N	Y	Y	0.293	0.574	89.37
N	N	N	Y	Y	Y	0.297	0.581	88.91
Y	Y	Y	Y	N	N	0.302	0.591	88.49
Y	Y	Y	N	Y	N	0.285	0.558	90.06
Y	Y	Y	N	N	Y	0.300	0.588	88.62
Y	Y	N	Y	Y	N	0.287	0.562	89.91
Y	Y	N	Y	N	Y	0.301	0.589	88.63
Y	Y	N	N	Y	Y	0.282	0.553	90.38
Y	N	Y	Y	Y	N	0.285	0.558	90.15
Y	N	Y	Y	N	Y	0.299	0.586	90.29
Y	N	Y	N	Y	Y	0.274	0.537	91.23
Y	N	N	Y	Y	Y	0.281	0.550	90.53
N	Y	Y	Y	Y	N	0.294	0.576	89.29
N	Y	Y	Y	N	Y	0.303	0.594	88.32
N	Y	Y	N	Y	Y	0.289	0.565	89.84
N	Y	N	Y	Y	Y	0.295	0.577	89.10
N	N	Y	Y	Y	Y	0.291	0.571	90.45
Y	Y	Y	Y	Y	N	0.282	0.551	90.58
Y	Y	Y	Y	N	Y	0.296	0.581	88.99
Y	Y	Y	N	Y	Y	0.272	0.533	91.42
Y	Y	N	Y	Y	Y	0.280	0.547	90.73
Y	N	Y	Y	Y	Y	0.276	0.539	91.16
N	Y	Y	Y	Y	Y	0.288	0.564	89.89
Y	Y	Y	Y	Y	Y	**0.268**	**0.524**	**91.90**

Note: The best result is highlighted in bold. Y/N indicates whether the component is included/excluded in the model.

cell-trans contribute to the improvement of model prediction performance but also demonstrate that the combination of these components can ensure the robustness of the cell-trans in the absence of partial input data.

Table 3. Analysis of various historical time steps on cell-trans's prediction performance on pNEUMA-128 and CitySim-128

	pNEUMA-128			CitySim-128		
Time-steps	MAE	RMSE	ACC (%)	MAE	RMSE	ACC (%)
$T_h = 2$	0.347	0.693	84.66	0.398	0.806	83.21
$T_h = 3$	0.304	0.604	88.29	0.341	0.762	86.76
$T_h = 4$	0.288	0.572	89.74	0.325	0.696	88.62
$T_h = 5$	0.279	0.559	90.20	0.318	0.673	89.17
$T_h = 6$	0.272	0.540	90.73	0.311	0.659	89.64
$T_h = 7$	*0.268*	*0.524*	*91.90*	*0.306*	*0.652*	*89.81*
$T_h = 8$	**0.265**	**0.518**	**92.13**	**0.304**	**0.649**	**89.96**

Note: The minimum value of historical time steps is set to 2 to enable the cell-trans to recognize signal changes at intersections. The best result is highlighted in bold; the second-best result is labeled with an italic.

Table 4. Analysis of various hidden dimensions on cell-trans's prediction performance on pNEUMA-128 and CitySim-128

	pNEUMA-128			CitySim-128		
Hidden dimension	MAE	RMSE	ACC (%)	MAE	RMSE	ACC (%)
$d_h = 8$	0.273	0.546	90.69	0.311	0.659	89.63
$d_h = 16$	0.271	0.535	90.94	0.310	0.658	89.66
$d_h = 32$	0.270	0.530	91.36	0.307	0.655	89.76
$d_h = 64$	**0.268**	**0.524**	**91.90**	*0.306*	*0.652*	*89.81*
$d_h = 128$	*0.268*	*0.527*	*91.74*	**0.305**	**0.650**	**89.89**

Note: The best result is highlighted in bold; the second-best result is labeled with an italic.

Table 5. Analysis of various attention heads on the cell-trans's prediction performance on pNEUMA-128 and CitySim-128

	pNEUMA-128			CitySim-128		
Attention heads	MAE	RMSE	ACC (%)	MAE	RMSE	ACC (%)
$N_h = 2$	0.272	0.539	90.85	0.309	0.657	89.70
$N_h = 4$	0.270	0.533	91.24	0.307	0.654	89.75
$N_h = 8$	**0.268**	**0.524**	**91.90**	**0.306**	**0.652**	**89.81**
$N_h = 16$	*0.270*	*0.530*	*91.39*	*0.306*	*0.654*	*89.77*

Note: The best result is highlighted in bold; the second-best result is labeled with an italic.

Performance Analysis

This analysis indicates the influence of various parameter settings on the prediction performance of the cell-trans on pNEUMA-128 and CitySim-128. Specifically, Table 3 indicates that the increase in historical time-steps T_h from 2 to 8 results in a marked enhancement in prediction performance across all metrics (MAE, RMSE, and ACC) for both data sets, with the most pronounced improvements observed between $T_h = 2$ and $T_h = 5$. When T_h exceeds 7, the cell-trans's prediction performance plateaus, prompting the selection of $T_h = 7$ as the optimal value to balance predictive accuracy and computational efficiency. In terms of the hidden dimension d_h, Table 4 demonstrates that cell-trans's prediction performance exhibits stability across a range of values (8 to 128), with optimal results for pNEUMA-128 at $d_h = 64$ and for CitySim-128 at $d_h = 128$, underscoring the model's robustness to variations in d_h. Similarly, Table 5 shows that the number of attention heads N_h demonstrates consistent performance across different configurations (2 to 16), with $N_h = 8$ generally yielding the best outcomes for both data sets. In summary, while increasing historical time-steps significantly enhances performance up to a certain threshold, the cell-trans exhibits robustness to changes in hidden dimensions and attention heads, thereby maintaining strong predictive accuracy across diverse parameter settings and offering flexibility in model configuration.

Computational Efficiency Analysis

Table 6 compares the predictive accuracy and computational efficiency of the cell-trans and baseline models on pNEUMA-128 and pNEUMA-256. The results demonstrate that all models perform worse on pNEUMA-256 compared to pNEUMA-128, indicating a decline in predictive accuracy in larger-scale scenarios. However, the cell-trans outperforms the second-best model, achieving improvements of 2.33%, 2.67%, and 1.32% in MAE, RMSE, and ACC on pNEUMA-256. These improvements are greater than the increases of 2.19%, 2.42%, and 1.19% observed on pNEUMA-128. These findings further highlight the cell-trans's advantage over baseline models in traffic prediction for large-scale environments. Additionally, the cell-trans takes 0.025 s to predict a sample on pNEUMA-128, making it the second-fastest model, only behind FNN. However, the prediction time for the cell-trans on pNEUMA-256 rises sharply to 0.089 s, exceeding that of GWN and FlashST. This suggests that cell-trans's computational efficiency decreases as the number of cells increases.

To address this challenge, it is advisable to partition large-scale scenes into subscenes for traffic state prediction and leverage cloud computing resources to execute these tasks in parallel, ensuring timely predictions. Moreover, model compression techniques, including low-rank factorization and knowledge distillation, are proposed

Table 6. Performance comparison between the cell-trans and baseline models on pNEUMA-128 and pNEUMA-256

	pNEUMA-128				pNEUMA-256			
Model	MAE	RMSE	ACC (%)	Speed (s)	MAE	RMSE	ACC (%)	Speed (s)
FNN	0.334	0.656	84.89	**0.014**	0.804	1.644	79.91	**0.016**
AGCRN	0.319	0.631	86.71	0.035	0.659	1.297	81.53	0.124
GWN	0.305	0.597	88.34	0.028	0.598	1.181	83.20	*0.061*
PDFormer	0.289	0.558	89.57	0.043	0.534	1.027	85.25	0.183
ST-MAN	0.277	0.542	90.14	0.079	0.511	0.998	85.79	0.334
FlashST	*0.274*	*0.537*	*90.84*	0.031	*0.506*	*0.991*	*86.43*	0.081
Cell-Trans	**0.268**	**0.524**	**91.90**	*0.025*	**0.494**	**0.965**	**87.47**	0.089

Note: The best result is highlighted in bold; the second-best result is labeled with an italic.

J. Transp. Eng., Part A: Systems, 2025, 151(12): 04025102

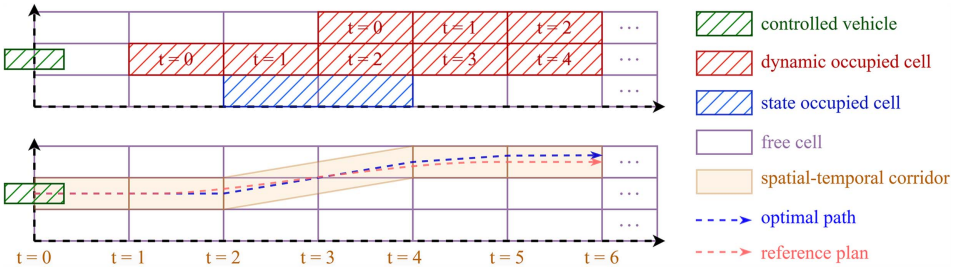

Fig. 3. Illustration of path planning for the AV based on the prediction result of the cell-trans in the Frenet frame coordinate system.

to decrease the computational complexity of the cell-trans (Cheng et al. 2018).

Case Study

This paper employs the cell-trans to enhance the Baidu EM planner and develop a predictive motion planner that divides the motion planning task into path planning and speed trajectory planning (Fan et al. 2018). In path planning, the predictive motion planner maps the lane segments ahead of the AV onto Frenet coordinates and marks their states based on cell-trans predictions, as shown in Fig. 3. It then constructs a spatial-temporal corridor within the free lane segments of each frame, avoiding occupied segments and maintaining continuity without skipping adjacent segments. The spatial-temporal corridor serves as the solution space for path planning, reducing collision risks and expediting the solving process. Following the Baidu Apollo EM planner, the total cost function combines smoothness, obstacle avoidance, and lane costs

$$C_{\text{total}}(f(s)) = C_{\text{smooth}}(f(s)) + C_{\text{obs}}(d(s)) + C_{\text{guidance}}(f(s)) \quad (21)$$

where $f(s)$ = the path planning; $d(s)$ = the distance between the AV and the nearest obstacle; and C_{total}, C_{smooth}, C_{obs}, and C_{guidance} = the total, smoothness, obstacle avoidance, and lane cost functions, respectively. The smoothness cost function C_{smooth} is defined as follows:

$$C_{\text{smooth}}(f(s)) = w_1 \cdot \int (f'(s))^2 dt + w_2 \cdot \int (f''(s))^2 dt + w_3 \cdot \int (f'''(s))^2 dt \quad (22)$$

where $f(s)$ = the path planning of the AV; $f''(s)$ = the curvature of the path planning; and $f'''(s)$ = the derivative of $f''(s)$. The obstacle avoidance cost function C_{obs} is defined as follows:

$$C_{\text{obs}}(d(s)) = \begin{cases} 0, & d(s) > d_n \\ C_{\text{nudge}}(d(s) - d_c), & d_c \le d(s) \le d_n \\ C_{\text{collosion}}, & d(s) < d_c \end{cases} \quad (23)$$

where C_{nudge} = a monotonically decreasing cost function; d_c = the minimum safe time interval of the AV; d_n = the nudge range determined by traffic scenes; and $C_{\text{collision}}$ = a collision cost with a large value to exclude infeasible path plannings. The lane cost function C_{guidance} is defined as follows:

$$C_{\text{guidance}}(f(s)) = \int (f(s) - g(s))^2 ds \quad (24)$$

where $g(s)$ = the guidance line of the AV, which is defined as the centerline of the path planning. The predictive motion planner employs dynamic and quadratic programming algorithms to find a smooth path within the spatial-temporal corridor that minimizes the cost function, serving as the final path planning result.

In the speed planning phase, the predictive motion planner constructs a spatial-temporal graph along the final path and maps lane segment states onto it, as shown in Fig. 4(a). It then maps the trajectory predictions of surrounding agents onto the spatial-temporal graph as dynamic obstacles, as depicted in Fig. 4(b). On this basis, the predictive motion planner derives a speed curve from obstacle-free points within the spatial-temporal corridor, expediting the solution while ensuring safety. It employs dynamic programming to

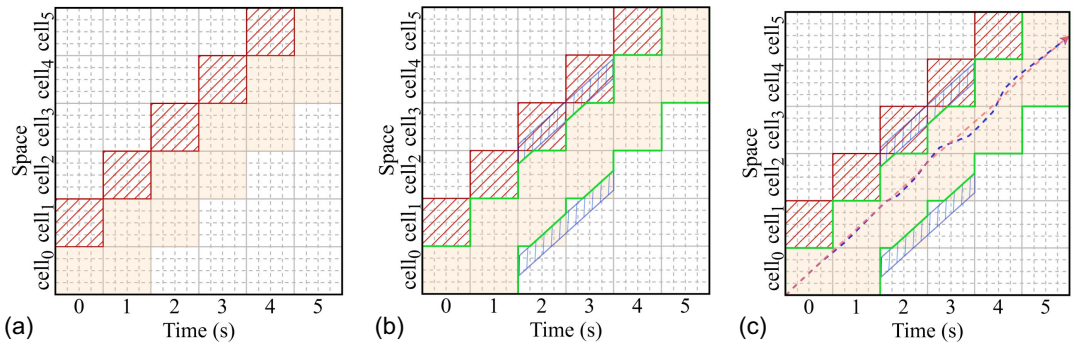

Fig. 4. Illustration of speed planning for the controlled vehicle based on the spatial-temporal corridor and the trajectory predictions of surrounding agents in the spatial-temporal graph: (a) the occupied cells and spatial-temporal corridor are mapped onto the spatial-temporal graph to reduce the solution space for speed planning; (b) the trajectory predictions of surrounding agents are mapped onto the spatial-temporal graph to reduce the solution space for speed planning; and (c) the dynamic programming is used to solve for the optimal speed planning in the solution space, which is refined to derive a followable speed planning.

J. Transp. Eng., Part A: Systems, 2025, 151(12): 04025102

Table 7. Comparison between EM planner and our predictive motion planner

Metric	Representative Case 1		Representative Case 2	
	EM planner	Ours	EM planner	Ours
Travel time (s)	11.59	12.03	21.15	20.62
Risky driving time (s)	0	0	8.20	4.74
Computational speed (s)	0.067	0.044	0.085	0.038

Note: Travel time: duration from arrival at intersections until passage. Risky driving time: when collision time <2.6 s. Computational speed: time for one motion planning operation.

determine an optimal speed plan with minimal cost, utilizing the cost function of the EM planner, expressed as follows:

$$C_{\text{total}}(s(t)) = w_1 \cdot \int_{t_0}^{t_n} g(s'(t) - \nu_{\text{ref}})dt + w_2 \cdot \int_{t_0}^{t_n} (s''(t))^2 dt + w_3 \cdot \int_{t_0}^{t_n} (s(t))^2 dt + C_{\text{obs}}(d(t)) \quad (25)$$

where $s(t)$ = a piecewise speed profile function; V_{ref} = the reference speed; g = a function that imposes different penalties for values below or above V_{ref}; and t_0 and t_n = the start and end times of the speed planning horizon, respectively. Since the time-space diagram is discrete, the derivative of $s(t)$ is approximated by the finite difference method

$$s'(t) \approx \frac{s(t) - s(t-1)}{dt} \quad (26)$$

$$s''(t) \approx \frac{s(t) - 2 \cdot s(t-1) + s(t-2)}{dt^2} \quad (27)$$

$$s'''(t) \approx \frac{s(t) - 3 \cdot s(t-1) + 3 \cdot s(t-2) - s(t-3)}{dt^3} \quad (28)$$

Subsequently, the predictive motion planner uses quadratic programming to refine the speed curve into a smooth one, as shown in Fig. 4(c). Finally, the predictive motion planner delivers the path and speed curve as the motion planning results to the AV for execution, initiating the next round of planning.

This paper integrates SUMO and CARLA to create a joint simulation platform and reconstructs a signalized intersection from the pNEUMA data set. The virtual intersection features a one-way, two-lane main road intersecting with a single-lane side road, with a two-phase signal cycle of 25 s for the main road and 20 s for the side road. The platform uses TraCI to access real-world traffic data from the pNEUMA data set and inputs it into the virtual scene, designating a low-traffic scenario as representative Case 1 and a high-traffic scenario as representative Case 2. It then employs CARLA to generate a car model positioned in the right lane, 150 m upstream of the signalized intersection on the main road, at 2 s into the simulation. This car model acts as the AV to execute the motion planning from the predictive motion planner. Simulation results in Table 7 show that the cell-trans-based predictive motion planner achieves faster solving speeds and better motion planning performance compared to the EM planner.

Conclusions

This paper presents a traffic prediction method named cell-trans, which utilizes multimodal traffic data from IVCPS to conduct traffic prediction at signalized intersections. The cell-trans comprises the vehicle-, cell-, and road-level encoders, along with a decoder. Once multimodal traffic data enters cell-trans, the vehicle-level encoder converts vehicle features into vehicle encodings. Then, the cell-level encoder converts lane segment features derived from HD maps into cell embeddings and fuses them with vehicle encodings to produce cell encodings. Next, the road-level encoder decomposes the dynamic directed graph, which represents the topological relationships between lane segments and the signal intersection control information, into spatial-temporal embedding matrices to facilitate the extraction of traffic evolution patterns from the cell encodings. Similarly, the decoder constructs spatial-temporal embedding matrices to guide the translation of decoder inputs and predicts the occupancy states of lane segments.

The cell-trans is compared with baseline models on pNEUMA and CitySim. The performance comparison demonstrates that the cell-trans improves MAE, RMSE, and ACC by 2.19%, 2.42%, and 1.19% on pNEUMA and by 1.29%, 1.67%, and 1.02% on CitySim, respectively. Ablation studies validate the contributions of the functional components in the cell-trans to enhance predictive accuracy. Next, parameter analysis is performed to determine the optimal parameter combination for cell-trans. Moreover, the computational efficiency analysis reveals that, while the cell-trans and baseline models lose predictive accuracy in large-scale scenarios, the cell-trans's accuracy declines at a much slower rate than baseline models. However, the computation time for the cell-trans increases significantly as the traffic scene expands, underscoring the need for enhanced computational efficiency.

As a multimodal traffic prediction method, the cell-trans can capture traffic evolution patterns from vehicle, lane segment, and dynamic road topology features to predict the occupancy states of lane segments in the predictive horizon. On this basis, this paper develops a predictive motion planner that constructs the spatial-temporal corridor from unoccupied lane segments as the solution space for motion planning of AVs. The predictive motion planner is evaluated within a signalized intersection scenario on a joint simulation platform that combines CARLA and SUMO, and its performance is compared against the Baidu EM planner. The simulation results demonstrate that the predictive motion planner based on the cell-trans exhibits superior safety, driving efficiency, and single-instance computational speed compared to the EM planner, highlighting the potential of the cell-trans in motion planning for AVs.

Future research will focus on three primary directions. First, the architecture of the cell-trans should be refined to enhance predictive accuracy and computational efficiency. Second, the predictive motion planner based on the cell-trans will undergo further optimization, with its performance evaluated across a broader range of traffic scenarios. Third, the cell-trans will be integrated as a functional module and deployed within IVCPS.

Data Availability Statement

All data, models, and code generated or used during the study appear in the published article.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant no. 52402375) and the Beijing Municipal Education Commission Science and Technology Program General Project (KM202410005002).

Author Contributions

Anran Li: Conceptualization; Investigation; Methodology; Software; Validation; Visualization; Writing – original draft; Writing – review and editing. Zhenlin Xu: Conceptualization; Investigation; Methodology; Software. Yuyan (Annie) Pan: Conceptualization; Supervision; Validation; Writing – review and editing. Bolin Gao: Conceptualization; Data curation; Funding acquisition; Resources. Jian Zhang: Conceptualization; Project administration; Supervision. Yanyan Chen: Conceptualization; Data curation; Funding acquisition; Resources; Supervision. Yongxing Li: Conceptualization; Funding acquisition; Project administration; Resources; Supervision; Writing – review and editing.

References

Adacher, L., and M. Tiriolo. 2018. "A macroscopic model with the advantages of microscopic model: A review of cell transmission model's extensions for urban traffic networks." *Simul. Modell. Pract. Theory* 86 (Aug): 102–119. https://doi.org/10.1016/j.simpat.2018.05.003.

Althoff, M., M. Koschi, and S. Manzinger. 2017. "CommonRoad: Composable benchmarks for motion planning on roads." In *Proc., 2017 IEEE Intelligent Vehicles Symp. (IV)*, 719–726. New York: IEEE.

Athey, S., J. Tibshirani, and S. Wager. 2019. "Generalized random forests." *Ann. Stat.* 47 (2): 1148–1178. https://doi.org/10.1214/18-AOS1709.

Bao, Z., S. Hossain, H. Lang, and X. Lin. 2022. "High-definition map generation technologies for autonomous driving." Preprint, submitted June 11, 2022. https://arxiv.org/abs/2206.05400.

Barmpounakis, E., and N. Geroliminis. 2020. "On the new era of urban traffic monitoring with massive drone data: The pNEUMA large-scale field experiment." *Transp. Res. Part C Emerging Technol.* 111 (Feb): 50–71. https://doi.org/10.1016/j.trc.2019.11.023.

Cheng, Y., D. Wang, P. Zhou, and T. Zhang. 2018. "Model compression and acceleration for deep neural networks: The principles, progress, and challenges." *IEEE Signal Process Mag.* 35 (1): 126–136. https://doi.org/10.1109/MSP.2017.2765695.

Daganzo, C. F. 1995. "The cell transmission model. Part II: Network traffic." *Transp. Res. Part B Methodol.* 29 (2): 79–93. https://doi.org/10.1016/0191-2615(94)00022-R.

Dong, J., Q. Xu, J. Wang, C. Yang, M. Cai, C. Chen, Y. Liu, J. Wang, and K. Li. 2023. "Mixed cloud control testbed: Validating vehicle-road-cloud integration via mixed digital twin." *IEEE Trans. Intell. Veh.* 8 (4): 2723–2736. https://doi.org/10.1109/TIV.2023.3243096.

Drucker, H., C. J. Burges, L. Kaufman, A. Smola, and V. Vapnik. 1996. "Support vector regression machines." In Vol. 7 of *Proc., 10th Int. Conf. on Neural Information Processing Systems*, 155–161. New York: Association for Computing Machinery.

Fan, H., F. Zhu, C. Liu, L. Zhang, L. Zhuang, D. Li, W. Zhu, J. Hu, H. Li, and Q. Kong. 2018. "Baidu Apollo EM motion planner." Preprint, submitted July 20, 2018. https://arxiv.org/abs/1807.08048.

Feng, C., Q. Du, X. Fan, F. Ren, L. Xiong, Y. Tian, W. Liu, T. Wang, Y. Wang, and B. Zhang. 2023. "A crowdsourcing update technology route of HD dynamic map basic platform." *J. Geomatics* 48 (1): 10–15.

Gao, B., K. Wan, Q. Chen, Z. Wang, R. Li, Y. Jiang, R. Mei, Y. Luo, and K. Li. 2023. "A review and outlook on predictive cruise control of vehicles and typical applications under cloud control system." *Mach. Intell. Res.* 20 (5): 614–639. https://doi.org/10.1007/s11633-022-1395-3.

Guo, J., Y. Liu, Q. Yang, Y. Wang, and S. Fang. 2021. "GPS-based citywide traffic congestion forecasting using CNN-RNN and C3D hybrid model." *Transportmetrica A: Transport Sci.* 17 (2): 190–211. https://doi.org/10.1080/23249935.2020.1745927.

Hamed, M. M., H. R. Al-Masaeid, and Z. M. B. Said. 1995. "Short-term prediction of traffic volume in urban arterials." *J. Transp. Eng.* 121 (3): 249–254. https://doi.org/10.1061/(ASCE)0733-947X(1995)121:3(249).

He, S., S. Wang, Y. Shao, Z. Sun, and M. W. Levin. 2023. "Real-time traffic prediction considering lane changing maneuvers with application to eco-driving control of electric vehicles." In *Proc., 2023 IEEE Intelligent Vehicles Symp. (IV)*, 1–7. New York: IEEE.

Hu, Y., W. Zhan, and M. Tomizuka. 2018. "Probabilistic prediction of vehicle semantic intention and motion." In *Proc., 2018 IEEE Intelligent Vehicles Symposium (IV)*, 307–313. New York: IEEE.

Ji, X., P. Mao, and Y. Han. 2023. "Traffic state prediction for urban networks: A spatial–temporal transformer network model." *J. Transp. Eng. Part A. Syst.* 149 (11): 04023105. https://doi.org/10.1061/JTEPBS.TEENG-7860.

Jiang, J., C. Han, W. X. Zhao, and J. Wang. 2023. "PDFormer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction." *Proc. AAAI Conf. Artif. Int.* 37 (4): 4365–4373. https://doi.org/10.1609/aaai.v37i4.25556.

Kiran, B. R., I. Sobh, V. Talpaert, P. Mannion, A. A. A. Sallab, S. Yogamani, and P. Pérez. 2022. "Deep reinforcement learning for autonomous driving: A survey." *IEEE Trans. Intell. Transp. Syst.* 23 (6): 4909–4926. https://doi.org/10.1109/TITS.2021.3054625.

Li, Z., L. Xia, Y. Xu, and C. Huang. 2024. "FlashST: A simple and universal prompt-tuning framework for traffic prediction." In *Proc., 41st Int. Conf. on Machine Learning (ICML 2024)*, 1–11. New York: Association for Computing Machinery.

Liu, L., Z. Cui, R. Ke, and Y. Wang. 2023. "Lane-level short-term freeway traffic volume prediction based on graph convolutional recurrent network." *J. Transp. Eng. Part A. Syst.* 149 (10): 04023102. https://doi.org/10.1061/JTEPBS.TEENG-7868.

Liu, M., W. Zhang, F. Orabona, and T. Yang. 2020. "Adam$^+$: A stochastic method with adaptive variance reduction." Preprint, submitted November 24, 2020. https://arxiv.org/abs/2011.11985.

Liu, Y., and G.-L. Chang. 2011. "An arterial signal optimization model for intersections experiencing queue spillback and lane blockage." *Transp. Res. Part C Emerging Technol.* 19 (1): 130–144. https://doi.org/10.1016/j.trc.2010.04.005.

Liu, Y., B. Guo, J. Meng, D. Zhang, and Z. Yu. 2024. "Spatio-temporal memory augmented multi-level attention network for traffic prediction." *IEEE Trans. Knowl. Data Eng.* 36 (6): 2643–2658. https://doi.org/10.1109/TKDE.2023.3322405.

Lu, Y., W. Wang, X. Hu, P. Xu, S. Zhou, and M. Cai. 2023. "Vehicle trajectory prediction in connected environments via heterogeneous context-aware graph convolutional networks." *IEEE Trans. Intell. Transp. Syst.* 24 (8): 8452–8464. https://doi.org/10.1109/TITS.2022.3173944.

Mayakuntla, S. K., and A. Verma. 2019. "Cell transmission modeling of heterogeneous disordered traffic." *J. Transp. Eng. Part A. Syst.* 145 (7): 04019027. https://doi.org/10.1061/JTEPBS.0000248.

Pan, Y. A., J. Guo, Y. Chen, S. Li, and W. Li. 2022. "Incorporating traffic flow model into a deep learning method for traffic state estimation: A hybrid stepwise modeling framework." *J. Adv. Transp.* 2022 (1): 5926663. https://doi.org/10.1155/2022/5926663.

Shao, Y., and Z. Sun. 2021. "Eco-approach with traffic prediction and experimental validation for connected and autonomous vehicles." *IEEE Trans. Intell. Transp. Syst.* 22 (3): 1562–1572. https://doi.org/10.1109/TITS.2020.2972198.

Siebinga, O. 2021. "TraViA: A traffic data visualization and annotation tool in Python." *J. Open Source Software* 6 (65): 3607. https://doi.org/10.21105/joss.03607.

Williams, B. M., and L. A. Hoel. 2003. "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results." *J. Transp. Eng.* 129 (6): 664–672. https://doi.org/10.1061/(ASCE)0733-947X(2003)129:6(664).

Wu, Z., S. Pan, G. Long, J. Jiang, and C. Zhang. 2019. "Graph WaveNet for deep spatial-temporal graph modeling." Preprint, submitted May 31, 2019. https://arxiv.org/abs/1906.00121.

Xu, Y., Y. Lu, C. Ji, and Q. Zhang. 2023. "Adaptive graph fusion convolutional recurrent network for traffic forecasting." *IEEE Internet Things J.* 10 (13): 11465–11475. https://doi.org/10.1109/JIOT.2023.3244182.

Ye, Q., W. Y. Szeto, and S. C. Wong. 2012. "Short-term traffic speed forecasting based on data recorded at irregular intervals." *IEEE Trans. Intell. Transp. Syst.* 13 (4): 1727–1737. https://doi.org/10.1109/TITS.2012.2203122.

Ye, X., S. Fang, F. Sun, C. Zhang, and S. Xiang. 2022. "Meta graph transformer: A novel framework for spatial–temporal traffic prediction."

Neurocomputing 491 (Jun): 544–563. https://doi.org/10.1016/j.neucom.2021.12.033.

Zhang, Y., and Y. Liu. 2009. "Traffic forecasting using least squares support vector machines." *Transportmetrica* 5 (3): 193–213. https://doi.org/10.1080/18128600902823216.

Zheng, O., M. Abdel-Aty, L. Yue, A. Abdelraouf, Z. Wang, and N. Mahmoud. 2024. "CitySim: A drone-based vehicle trajectory dataset for safety-oriented research and digital twins." *Transp. Res. Rec.* 2678 (4): 606–621. https://doi.org/10.1177/03611981231185768.