Multi-level Traffic-responsive Tilt Camera Surveillance through Predictive Correlated Online Learning

Tao Li[a,*], Zilin Bian[b,**], Haozhe Lei[a], Fan Zuo[b], Ya-Ting Yang[a], Quanyan Zhu[a], Zhenning Li[c], Kaan Ozbay[b]

[a]*Department of Electrical and Computer Engineering, New York University*
[b]*Department of Civil and Urban Engineering, New York University*
[c]*State Key Laboratory of Internet of Things for Smart City, University of Macau*

Abstract

In the realm of urban traffic management, the primary challenge of dynamically and efficiently monitoring traffic conditions is compounded by the insufficient utilization of thousands of surveillance cameras along the intelligent transportation system. This paper introduces the multi-level Traffic-responsive Tilt Camera surveillance system (TTC-X), a novel framework designed for dynamic and efficient monitoring and managing various traffic objects in urban transportation networks. By leveraging widely deployed pan-tilt-cameras (PTCs), TTC-X overcomes the limitations of a fixed field of view in traditional surveillance systems by providing mobilized and 360-degree coverage. The innovation of TTC-X lies in the integration of advanced machine learning modules, including a detector-predictor-controller structure, with a novel Predictive Correlated Online Learning (PiCOL) methodology and the Spatial-Temporal Graph Predictor (STGP) for real-time traffic estimation and PTC control. The TTC-X is tested and evaluated under three experimental scenarios (e.g., maximum traffic flow capture, dynamic route planning, traffic state estimation) based on a simulation environment calibrated using real-world traffic data in Brooklyn, New York. The experimental results showed that TTC-X captured over 60% total number of vehicles at the network level, dynamically adjusted its route recommendation in reaction to unexpected full-lane closure events, and reconstructed link-level traffic states with best MAE less than 1.25 vehiclehour. Demonstrating scalability, cost-efficiency, and adaptability, TTC-X emerges as a potent solution for urban traffic management in both cyber-physical and real-world environments.

Keywords: Real-time traffic surveillance, online learning control, spatial-temporal forecasting, traffic state estimation, dynamic route planning

1. Introduction

The intelligent transportation system (ITS) in urban areas has significantly expanded traffic surveillance cameras along road segments. These cameras are strategically placed at key locations such as road intersections, aiming to help traffic management agencies by providing real-time traffic information such as traffic congestion, traffic incidents, special events[1], and so on. Unlike traditional surveillance cameras with fixed fields of view (FOVs) that offer limited, single-direction [2] traffic monitoring, pan-tilt-cameras (PTCs) offer a more dynamic solution with their 360-degree mobilized coverage [3], enabling versatile monitoring

---

[*]Authors contributed equally
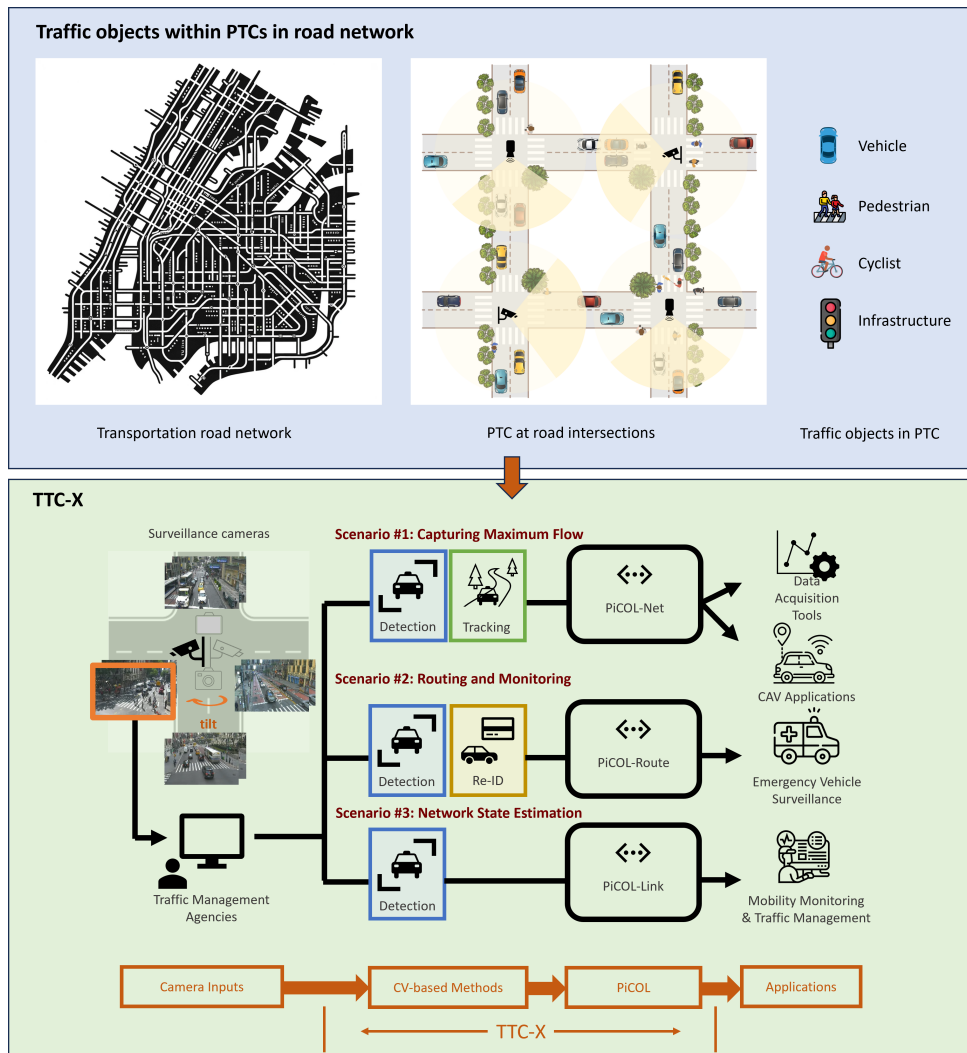[**]Corresponding author

Figure 1: TTC-X conceptualization and formulation

across different areas. Although the PTCs allow for more mobility and coverage, the widespread camera network (with PTCs) remains underutilized. PTCs operate in a static mode, oriented in a single direction at one time, and these PTCs within the network still function in isolation without inter-camera communication [4]. This leads to prolonged inactivity periods, during which traffic management agencies must manually reposition the cameras for specific monitoring tasks. Consequently, when traffic situations (e.g., congestion, accidents) arise along the road, the PTCs still require manual intervention to adjust their orientation appropriately for monitoring and management purposes. One way to avoid the manual intervention of PTCs and labor-intensive costs from transportation management agencies is to design an automated and responsive traffic surveillance system. Seizing the opportunities of the current broadly deployed PTC system along intersections in urban areas, the idea is to extend from a single-camera-single-view to a multi-level surveil-

2

lance strategy encompassing link, route, and network-level information. As shown in Figure 1, each PTC covers four oriented directions, and each orientation covers one roadway segment with traffic from at most two directions. Therefore, the workflow of designing such a PTC-based traffic surveillance system includes 1) detecting and tracking multiple traffic objects (e.g., vehicle, pedestrian, cyclist, incident) from PTCs, 2) learning and estimating the mobility nature of traffic objects, and 3) collaboratively controlling multiple PTCs to achieve multi-level monitoring tasks for specific traffic object. However, this approach presents several challenges:

- **Complexity and dynamics in urban transportation network:** Urban transportation networks are inherently complex and dynamic, characterized by intricate road connections and diverse network structures coupled with time-variant traffic demand and frequent traffic incidents. A challenging task is to seize traffic dynamics in both spatial and temporal dimensions in constraints to road network features to help the multi-camera surveillance system.

- **Cost-efficient monitoring of large-scale transportation networks:** While PTCs offer extended coverage and mobility, effectively utilizing them to monitor large-scale networks with minimal equipment poses a significant challenge. For example, as shown in Figure 1, on a theoretical basis, only four PTCs (one PTC at each intersection) are needed to monitor traffic along 24 road segments. However, the practical implementation faces the hurdle of coordinating these cameras to cover multiple directions simultaneously without losing focus on critical areas. This necessitates the development of intelligent algorithms that can dynamically prioritize camera orientations based on real-time traffic conditions and historical data trends.

- **Rapid and automatic response to traffic situations:** Given the complicated and dynamic nature of traffic patterns across a large-scale transportation network, it is essential for a surveillance system to proactively and responsively adapt to various traffic situations, such as congestion or incidents. For example, when a non-recurrent traffic incident occurs on a roadway segment, disrupting traffic flow, the disturbance evolves dynamically in both temporal and spatial dimensions. Detecting such changes in real-time and responding appropriately is a critical challenge, requiring advanced algorithms and responsive system architecture.

This study proposes a novel Traffic-responsive Tilt Camera surveillance system for everything (X) in a large-scale transportation network (TTC-X) to overcome the aforementioned challenges and hurdles. The TTC-X framework is innovatively designed based on a detector-predictor-controller (DPC) structure, integrating various technologies for comprehensive traffic monitoring. Specifically, the detector component utilizes established video extraction methods, including multiple-object-detection (MOD)[5], tracking[6, 7, 8], and re-identification (Re-ID)[9, 10]. The appropriate use of the mentioned video extraction methods enables TTC-X to handle multiple objects captured by PTCs in large-scale transportation road networks. Theoretically, anything within the frame of PTC can be tracked and extracted into digital information, as shown in Fig.1. Multiple traffic objects, including vehicles, pedestrians, cyclists, and infrastructure, can be extracted using appropriate video extraction methods.

To comprehensively demonstrate the proposed TTC-X, we limit our research scope and focus on vehicles as primary traffic objects. As shown in Figure 1, this study presents three distinct scenarios captured by PTCs within the transportation network, each illustrating different levels of traffic information: network, route, and link level. Due to space constraints, these scenarios are selectively demonstrated to encompass the comprehensive capabilities of TTC-X.

At the network level, TTC-X is designed to capture and collect maximum traffic flow data across the entire transportation network in real-time. This scenario focuses on the overarching view of traffic dynamics,

aiming to optimize the overall traffic flow and reduce congestion. With the help of appropriate MOD and tracking methods, TTC-X ensures the capture of the maximum number of vehicle trajectories, supporting applications such as data acquisition tools and connected and autonomous vehicle (CAV) simulation. In the route-level scenario, TTC-X facilitates real-time route planning. It identifies paths with the least travel time, considering unexpected traffic disruptions. This feature is handy for dynamic route optimization in response to evolving traffic conditions, such as accidents or road closures. Unlike probe-vehicle-based routing methods, this real-time route planning is high resolution and high accuracy thanks to the PTC surveillance cameras, which can provide reliable routing for cases requiring precise routing, such as emergency vehicle routing. The link-level scenario encompasses two sub-tasks: traffic state reconstruction and forecasting. The reconstruction task ensures that TTC-X accurately recovers the most realistic current traffic states across all road segments within the network. Meanwhile, the forecasting task enables TTC-X to provide confident predictions under various traffic situations, including sudden disturbances like non-recurrent traffic incidents.

The TTC-X framework integrates the Predictive Correlated Online Learning (PiCOL) methodology, which bridges the predictor and controller to provide optimal and responsive PTC tilting strategies. Within PiCOL, the Spatial-Temporal Graph Predictor (STGP) plays a crucial role. STGP is tasked with estimating and forecasting traffic states on each road segment within the transportation network. It operates with a plug-and-play online learning controller that distributedly calibrates each PTC's tilting strategy in correlation with the rest online. This correlated online learning leads to scalable, responsive, and coordinated PTC coverage over the road network in dynamic traffic patterns. The predictor and the controller function at two complementary levels. At the first level, STGP monitors and supplements traffic state information for road segments not directly observed by the PTCs whose tilting policy is dedicated by the controller. The controller, in turn, combines these estimates with its observations to enhance the accuracy and reliability of STGP's predictions. At the second level, STGP's estimations consider the nature of traffic flow and network structural constraints. This enables STGP to confidently predict traffic states for segments the controller has not observed directly, particularly by considering observations from neighboring segments. Through this approach, STGP leverages network topology and local spatial dependencies to learn and infer traffic patterns, thereby enhancing the overall effectiveness of the TTC-X system in managing and responding to dynamic traffic conditions.

The performance of the TTC-X framework is evaluated in each of three scenarios using a simulation dataset calibrated using real-world data from Brooklyn, New York. At the network level, it is found that TTC-X can capture over 60% number of vehicles within the road network throughout the day. With the help of STGP, TTC-X can estimate the traffic demand within the road network with MAPE (Mean Absolute Percentage Error) less than 10%. At the route level, TTC-X is assessed using a case study with a full-lane closure event in the road network. TTC-X identified the optimal route within 5min and provided accurate trip travel time with an error of less than 1.6%. At the link level, TTC-X achieved the best MAE (Mean Absolute Error) as 12.36 vehicles per hour for the forecasting task and 1.25 for the reconstruction task.

The contributions of this study are summarized below.

- We propose a novel framework for the Traffic-responsive Tilt Camera surveillance (TTC-X) system, a versatile PTC-based surveillance system designed for network-wide, real-time traffic monitoring and management. TTC-X integrates advanced machine learning modules across its detector, predictor, and controller components, enabling real-time multi-camera coordination. The proposed TTC-X offers responsive intelligence that rapidly adapts PTC tilting on the fly to a variety of traffic situations.

- A cutting-edge distributed PTC control framework, PiCOL is proposed for forecasting and reconstructing multi-level traffic information by integrating an offline learning-based predictive model named STGP into an online learning control paradigm. With the customization of loss functions, PiCOL is

eligible to extend into applications driven by multiple tasks.

- We design three case scenarios at the network, route, and link levels, targeting different real-world application usage. Findings reveal that TTC-X can forecast and reconstruct multi-level (both network and link level) traffic information using a minimal set of PTCs with low information loss. Moreover, abrupt disturbances in the traffic network can be swiftly handled by TTC-X at both route- and link-level. The optimal route can be updated, and link-level traffic states can be estimated in reaction to abrupt changes with rapid responsiveness.

- The light-weighted, scalable, cost-efficient features of the TTC-X framework enables its connection with wide-range applications, in both cyber-physical world and real world environment. In the cyber-physical realm, TTC-X can be integrated within a virtual setup such as CARLA-SUMO co-simulation, enabling rigorous testing and refinement in a digital twin of actual traffic conditions, avoiding the high costs and safety risks of direct real-world deployment. For real-world applicability, TTC-X functions as a versatile plug-and-play system, easily adaptable to existing traffic management infrastructures, enhancing traffic monitoring and management with minimal disruption and expense.

2. Literature Review

2.1. Visual surveillance camera systems in ITS

Intelligent transportation systems (ITS) [11] have been revolutionized by various applications, including traffic surveillance [12], data acquisition [13, 14], information estimation [15, 16], congestion recognition [17], and sustainably traffic management [18]. A diverse range of objects that can be captured through visual surveillance camera systems, including vehicles [19, 20], pedestrians[21, 22], cyclists [23], obstacles [24], road infrastructures, etc. Within the realm of traffic surveillance and monitoring, conventional methods can be found in [25, 26, 27, 28]. [25] reviews both appearance and motion-based feature methods for vision-based vehicle detection, where the former uses the appearance features (e.g., shape, color, and texture of the vehicle) to detect the vehicle or separates it from the background and the latter one utilizes moving characteristic to distinguish vehicles from the stationary background image. They also mention popular approaches (different vehicle representations [29], Kalman filter [30, 31], etc.) in vehicle tracking that are often used to extract vehicles' dynamic attributes, including velocity, the direction of movement, and vehicle trajectories. [26, 27] give reviews on possible applications (vehicle counting, vehicle speed measurement, identification of traffic accidents, traffic flow prediction, etc.) of video-based ITS under varying challenging environments such as illumination change and different weather conditions. [28] specifically focuses on the issues related to vehicle monitoring with cameras at road intersections. Recent innovations in computer vision (CV) and deep learning (DL) research for visual traffic surveillance systems can be found in [32, 33]. [32] gives a systematic review of the detection and tracking CV-based methods, while [33] conducts a comprehensive survey on multi-object multi-camera tracking (MOMCT) frameworks from multiple-object single-camera detection and tracking to vehicle re-identification approaches across cameras.

2.2. Traffic state estimation methods

Graph Neural Networks (GNNs) have recently gained prominence among the various methods available for traffic state estimation. By transforming large-scale road networks into graph structures, GNNs facilitate the exchange of local information between neighboring road segments. Several conventional GNN-based methods, such as Graph Convolutional Networks (GCNs) [34], Graph Attention Networks (GATs) [35], and Graph Recurrent Networks (GRNs) [36], have been effectively employed to forecast traffic states within

large-scale road networks. While GNN-based methods can capture the spatial dependencies of traffic states in the road network, transformer-based methods evolve as power methods of capturing the dynamic flux as traffic states are time-series data. Originating from the nature language processing (NLP) method, the transformer model [37] has demonstrated exceptional performance in capturing temporal dependencies within sequential data. This success has led to its adoption in addressing transportation-related challenges. Huang et al., [38] combined a convolutional neural network (CNN) with a modified transformer to predict network-wide traffic demand. Zhang et al.,[39] proposed a spatial-temporal graph attention transformer to predict autonomous vehicle trajectories.

2.3. Online learning and control approaches

The proposed PiCOL methodology falls within the realm of online learning and optimization study [40], also known as the multi-arm bandit problem in the literature [41]. The backbone of the PiCOL subscribes to exponential weights [42], a seminal online learning algorithm, and its variant: EXP3 [43]. Similarly to PiCOL, [44, 45] considers online learning with predictions of future loss functions. However, unlike our investigation into PiCOL, these prior works focus on the influence of prediction accuracy on online learning control performance without touching the interplay and mutual influence between prediction and control. In contrast to the aforementioned prior works studying single-agent cases, PiCOL is a multi-agent distributed online learning scheme, also about the line of research works on multi-agent learning [46, 47]. In particular, PiCOL bears a spirit of coordinating each agent's online learning through the shared loss function, similar to learning for correlated equilibria [48, 49]. Finally, online learning applications in urban traffic networks remain largely uncharted. Even though a handful of prior works have applied online learning and control methods to traffic signal control [50] and traffic forecasting [51], our work is among the first endeavors to explore the role of online learning in PTC controls.

3. Methodology

3.1. An Overview of TTC-X

The comprehensive architecture of the TTC-X system encapsulates several integral components, as shown in Figure 2. The first component is the detector, which serves as the nexus for information extraction. It adeptly processes the visual information acquired through PTCs, translating into link-level (individual road segment) traffic states, such as vehicular volume, travel time, and traffic speed per road segment. The converted information is subsequently fed into the PiCOL component, which includes both the predictor and controller. Within this framework, the predictor's Spatial-Temporal Graph Prediction (STGP) module employs an offline learning paradigm, having been trained on historical data to enable real-time, online predictions. In contrast, the controller adopts an online learning approach, continuously calibrating and optimizing the PTC tilting policy in response to live traffic states. By integrating the extracted road network topology and traffic states into the STGP, the predictor acquires a spatial and temporal understanding of traffic dynamics specific to each road segment in the network. After a warm-up phase involving PTC observation, a dynamic information feedback loop is established between the predictor's STGP and the controller, as shown in Figure 4. The STGP provides the predictions of traffic states for several upcoming time steps. These predictions are then utilized by the controller to formulate bespoke tilting strategies, addressing diverse operational scenarios at the network, route, and link levels. Subsequent to the implementation of these strategies, the controller assimilates observations and overlays the forecasted traffic states with actual observed values. This sequence of updated traffic states is then re-introduced into the STGP for ensuing prediction cycles. This continuous feedback mechanism allows for successive refinements in the STGP's future traffic state estimations as well as the controller's online calibration, thereby significantly augmenting the overall efficiency and precision of the TTC-X system.
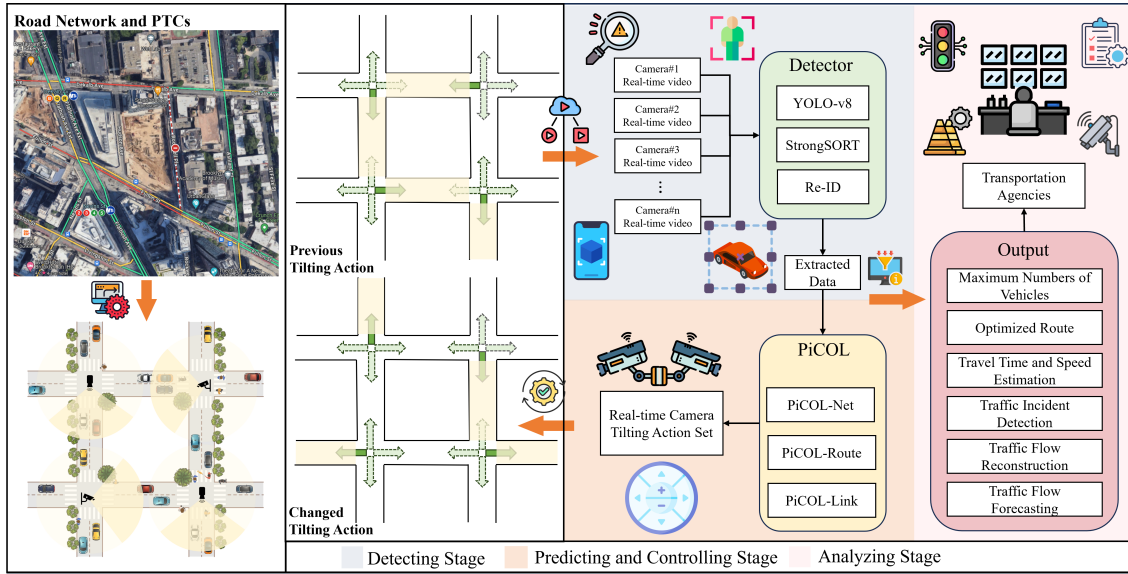
Figure 2: The overall framework architecture of TTC-X system.

3.2. Detector

In the proposed TTC-X system, traffic surveillance and analysis are primarily powered by deep learning models, specifically YOLOv8 for object detection and StrongSORT for object tracking. These models are fed with frames from CCTV camera data strategically positioned across the transportation network ([52, 53]). This section delineates the functioning of these models and their integration within the TTC-X system.

YOLOv8 (You Only Look Once, version 8) is an advanced iteration in the YOLO series[54], known for real-time object detection with high accuracy. YOLOv8 operates on the principle of dividing the input image (CCTV frames) into a grid, and each grid cell predicts bounding boxes and class probabilities. The model inputs continuing video feed from CCTV cameras and then processes each frame to detect various traffic objects (vehicles, pedestrians, cyclists). YOLOv8 is designed to identify objects swiftly and accurately, which is essential in dynamic traffic conditions. Following detection, StrongSORT [55] is employed for tracking. It is an enhanced version of the SORT (Simple Online and Realtime Tracking)[56] algorithm, augmented with deep learning features for improved tracking accuracy. StrongSORT receives the identified objects from YOLOv8 and tracks them across successive frames. This is crucial in monitoring traffic flow, calculating speeds, and identifying traffic incidents.

The integrated YOLOv8 and StrongSORT models generate multiple essential traffic data as output: vehicle counts (volume data), speeds (speed data), and behavioral patterns (safety data). This data becomes the backbone for various traffic management applications. The output is classified and channeled into three different scenarios, aligning with the multi-level surveillance strategy of TTC-X:

The Traffic-responsive Tilt Camera system (TTC-X) is engineered to optimize traffic flow, monitor routes, and forecast traffic states at different levels of urban traffic management. Figure 3 shows the data flow from the detector to the PiCOL modules. At the network level, PiCOL-Net integrates traffic flow data to refine signal timings and ease congestion, particularly during peak periods. Route-level management via PiCOL-Route leverages this data for real-time routing, significantly improving emergency vehicle response
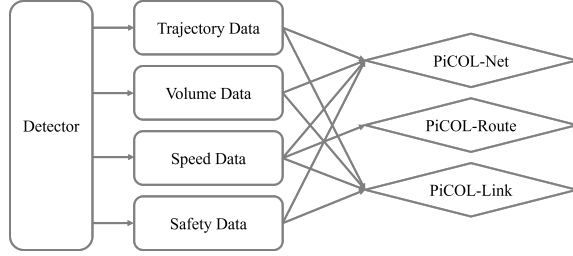
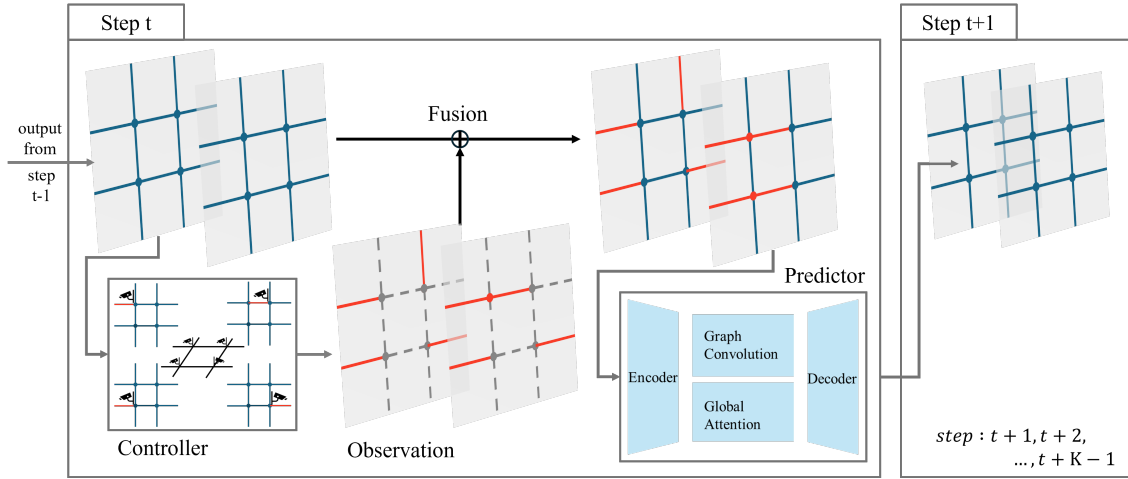Figure 3: The data flow from the detector to the PiCOL modules.

Figure 4: An overview of PiCOL module in TTC-X. The controller's tilting policy is calibrated online and distributed using the predictor's forecast of traffic states. These predictions, fused with actual traffic state observations, are fed to the predictor for the future predictions that guide the controller's calibration.

by adapting to live traffic conditions. PiCOL-Link, at the link level, focuses on accurate estimation and forecasting of traffic states, aiding in swift incident response and maintaining traffic network efficiency.

3.3. Predictive Correlated Online Learning

The predictive correlated online learning (PiCOL) module follows the detector, which is directly responsible for the PTCs' tilting control. As shown in Figure 4, PiCOL includes two essential pillars: the controller, the PTCs' joint online tilting policy, and the predictor, an offline-trained traffic forecasting model. Despite different data input and task objectives across the network, route, and link levels, PiCOL treats the three scenarios from a unified online learning perspective, leading to a generic formulation of these tasks as later presented in Section 3.3.1 and a shared online learning control approach. The resulting online learning algorithm, detailed in Section 3.3.2, is a plug-and-play control scheme without any domain knowledge of the network topology and the underlying traffic evolution, readily applicable to a variety of traffic situations and road networks. Even though the online learning control is traffic and network-agnostic, it relies on the STGP to extract the spatial and temporal dependencies of traffic states (see Section 3.3.3), which creates a cooperative and responsive PTC coverage of the road network.

Notation(s)	Description
$\mathcal{G}, \mathcal{N}, \mathcal{E}$	The traffic network, the junction (nodes) set, and roads (edges) set
$(i, j) \in \mathcal{E}, i, j \in \mathcal{N}$	The directed edge from node i to j
$T, [T] := \{1, 2, \ldots, T\}$	The horizon length and the set of discrete time indices
$\mathcal{N}^c$	The subset of nodes where cameras are deployed
$\mathcal{E}^i$	The set of incident edges of node i
$\mathcal{A}^i, a_t^i$	The action set of camera i and the implemented action at time t
$s_t = (s_t^{ij}), (i, j) \in \mathcal{E}$	The network-level traffic state information at time t, an aggregation of local information on each edge
$\delta_t = (\delta_t^{ij}), (i, j) \in \mathcal{E}$	The network-level relative traffic state difference at time t
s_t^c	The joint observation of the TTC-X
δ_t^c	The observed relative state difference
$\hat{s}_t$	The predicted traffic state
$\hat{s}_t^f$	The traffic state fusion, a combination of $\hat{s}_t$ and s_t^c
$L(a_t; s_t)$	The loss function of joint action and network state
$\mathcal{P}_{(i_o, i_d)}$	The set of all paths connecting the OD pair (i_o, i_d)
$\mathcal{G}_{(i_o, i_d)}$	The smallest subgraph containing $\mathcal{P}_{(i_o, i_d)}$
$\otimes, \vee$	The entry-wise product and the boolean operator "or"

Table 1: A summary of frequently used notations.

3.3.1. Multi-level Real-time Traffic Information Collection and Processing

Consider a traffic network represented by a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where the node set $\mathcal{N} := \{1, 2, \ldots, N\}$ denote the intersections, and the terms intersections and nodes are used interchangeably in this paper. Each element in the edge set $\mathcal{E} = \{(i, j) | i, j \in \mathcal{N}\}$ represents the road segment connecting two intersections. We use link, edge, and road segments interchangeably throughout the paper. We model the traffic evolution within the network during a given period of time using a discrete-time sequence $\{s_t\}_{t \in [T]}$, where s_t denotes the real-time traffic information of interest at time t. The traffic state variable s_t denotes the aggregated local information from each link s_t^{ij}: $s_t = [s_t^{ij}]_{(i,j) \in \mathcal{E}}$.

PTCs are allocated to a subset of intersections for surveillance purposes. Let $\mathcal{N}^c \subset \mathcal{N}$ be the set of the nodes equipped with a PTC. With a slight abuse of notation, we denote by $i \in \mathcal{N}^c$ the PTC deployed at the i-th intersection. The PTC i can be tilted to monitor the inbound and outbound links (i, j) and (j, i) from a neighboring node j at a time. We define the PTC's action as a $|\mathcal{E}|$-dimensional binary vector $a^i \in \mathcal{A}^i := \{0, 1\}^{|\mathcal{E}|}$, and the one entries correspond to the monitored edges: $a^i(i, j) = 1$, if (i, j) is covered by the PTC. Let $a_t = \vee_{i \in \mathcal{N}^c} a_t^i \in \{0, 1\}^{|\mathcal{E}|}$ be the joint action of all cameras at time t, where $\vee$ denotes the entry-wise boolean operator "or". In plain words, $a_t(i, j) = 1$ if the edge (i, j) is covered by at least one of the PTCs. Employing the entry-wise product (Hadamard product), denoted by $\otimes$, the joint observation of the TTC system is given by $s_t^c = s_t \otimes a_t$, where the local traffic states without any camera monitoring are masked with zero.

Real-time traffic information can be captured and processed on multiple levels depending on the transportation agencies' needs. At the network level, the objective of the TTC-X is to monitor the edges with significant volume to collect maximum flow data at each time step. At the route level, the task is to utilize cameras to update the traffic information in real-time for dynamic route recommendation. At the link level, to fulfill the traffic state reconstruction and forecasting, the TTC-X needs to be sensitive to the traffic incidents on each edge, leading to dramatic changes in traffic patterns within a short period. The cameras are

tasked to capture these incidents as soon as they occur. The following formally defines the three scenarios in mathematical terms and the corresponding objective functions specifically. The key observation is that all three scenarios admit a shared optimization formulation (up to a constant).

Network Level. In the first scenario, the agency aims to capture the maximum flow in real time using the TTC-X. The traffic state s_t^{ij} represents the traffic volume on the road segment (i, j) at time t, while the aggregation s_t presents the network-level volume distribution. Suppose the joint action at time t is a_t, and then the performance metric is given by the absolute percentage error $L(a_t; s_t) := \|s_t^c - s_t\|_1 / \|s_t\|_1$, which evaluates the amount of traffic volume not captured by the cameras. The TTC-X aims to minimize the mean absolute percentage error (MAPE) across the horizon.

$$\min_{\{a_t\}_{t \in [T]}} \frac{1}{T} \sum_{t \in [T]} L(a_t; s_t), \quad L(a_t; s_t) := \frac{\|s_t^c - s_t\|_1}{\|s_t\|_1}. \tag{1}$$

Route Level. The objective in the second scenario is real-time route planning. The cameras collectively identify the path with the least travel time in dynamic traffic patterns with unexpected traffic incidents. Such route planning admits a dynamic nature: the route recommendation shall be updated in real-time if traffic incidents render the current plan no longer optimal. In this scenario, the traffic state in information s_t^{ij} denotes the travel time estimate of (i, j), which can be obtained by the speed estimate returned by the camera.

Consider trips traversing from o_t to d_t at time t, and the origin-destination pair $(o_t, d_t) \in \mathcal{N} \times \mathcal{N}$ admits multiple possible paths. Denote by $\mathcal{P}_{(o,d)}$ (or simply $\mathcal{P}$ if the OD pair is clear from the context) the set of all paths connecting the o and d, with its typical element $p \in \mathcal{P}_{(i_o, i_d)}$ represented by a sequence of nodes in which each node is connected by an edge to the next. Let $\mathcal{G}_{(o,d)}$ be the smallest subgraph containing the nodes and edges on the paths included in $\mathcal{P}_{(o,d)}$. For simplicity, we assume that those paths in $\mathcal{P}_{(o,d)}$ for any arbitrary OD pair can be fully covered by the camera nodes within $\mathcal{G}_{(o,d)}$. The travel time of the chosen path (represented by the joint action a_t) is given by the inner product $\langle a_t, s_t \rangle$ and hence, the TTC-X aims to return a sequence of route planning $\{a_t\}_{t \in [T]}$ such that the total travel time is minimized, as presented in Eq. (2).

$$\min_{\{a_t\}_{t \in [T]}} \sum_{t \in [T]} L(a_t; s_t), \quad L(a_t; s_t) := \langle a_t, s_t \rangle, \tag{2}$$

Link Level. The last scenario concerns the real-time network state estimation that requires link-level change-point detection. The TTC-X is tasked to monitor the edges with significant fluctuations in traffic patterns. In this case, the quantity of interest is the relative difference in traffic volume on each edge. Let s_t^{ij} be the volume number defined in the network-level case, and then the relative difference in volume at time t is as $\delta_t^{ij} = (s_t^{ij} - s_{t-1}^{ij}) / s_{t-1}^{ij}$. The network-level aggregation is then denoted by $\delta_t = [\delta_t^{ij}]_{(i,j) \in \mathcal{E}}$. Similar to the network-level case, the sensitivity of the TTC-X is evaluated through the error $\|\delta_t^c - \delta_t\|_1 / \|\delta_t\|_1$. Therefore, the objective function over the whole horizon is given by the time average in Eq. (3).

$$\min_{\{a_t\}_{t \in [T]}} \frac{1}{T} \sum_{t \in [T]} L(a_t; \delta_t), \quad L(a_t; \delta_t) := \frac{\|\delta_t^c - \delta_t\|_1}{\|\delta_t\|_1}, \delta_t^c = \delta_t \otimes a_t \tag{3}$$

3.3.2. PiCOL Online-Learning Controller

Due to urban transportation networks' complex and highly dynamic nature, modeling the network traffic state evolution using closed-form dynamic systems is prohibitive. Even though the three distinct scenarios share a sequential decision-making formulation, the current control design methodologies, such as optimal control [57], rely on a well-structured dynamic model to capture the system evolution. A model-free

learning-based control can better equip the cameras with responsive intelligence and online adaptability in complex and dynamic traffic patterns, enabling them to collect real-time traffic information cooperatively.

With rich historical traffic data and advancement in traffic simulation and digital twins [58], reinforcement learning (RL) stands out as a strong candidate for learning intelligent TTC-X control policies. Model-free RL, such as value-based [59] and policy-based algorithms [60], offers a data-driven approach to tackling large-scale network control and management problems [61] without domain knowledge of underlying system dynamics. However, within the offline learning paradigm, model-free RL suffers from the distribution shift between the training and testing environments [62]. In the urban transportation, the distribution shift is embodied by non-recurrent traffic incidents unobserved in the historical traffic data. An RL control policy trained using such historical data typically does not generalize well to the real-world traffic environment, which displays different traffic patterns due to incidents. Even though many research efforts, such as online meta-learning [63, 64], have been dedicated to the generalization and adaptability of RL, the resulting machinery suffers from exhaustive offline training and online computation [65].

This work explores the online learning paradigm [43, 66] to address the TTC-X challenges without domain knowledge of the underlying traffic and the offline training on the control policy. The notable features of the proposed PiCOL methodology, as a synthesis of an online-learning-based control algorithm (controller) and an offline-trained forecasting model (predictor), include that

- the controller employs an online learning algorithm derived from exponential weight (EW) [42], a seminal multi-arm bandit algorithm, whose implementation is independent of the network topology and only requires the controller's online observations and predictor's forecast;

- the learning-based control policy is executed and updated at each camera node in parallel, achieving distributed control;

- the predictor learns network-structure nature and the sequential characteristics of traffic states in offline training, and its predictions help coordinate each camera's tilting.

The following revisits the three representative scenarios and casts them as a multi-arm bandit problem, a well-established setup for online learning. Using the network-level maximum flow capturing as an example, we illustrate the operation and insights of the PiCOL.

In the multi-arm bandit problem, initially proposed by [67], a gambler must choose one of the finitely many slot machines to play. By pulling the arm of one of the machines, he receives a reward at each time step. Mathematically, a multi-armed bandit problem is specified by a finite action set $\mathcal{A}$ from which the decision-maker selects an action a_t at each time step. After the action implementation, the environment returns a loss incurred by such action $L_t(a_t)$. Note that the loss function $L_t(\cdot)$ is time-varying and unknown to the decision-maker before time t. This online decision-making process can be viewed as a repeated game between the decision-maker and nature, where the game's payoff is jointly determined by the actions and the loss functions chosen by the two players, respectively, at each time step. The decision-maker aims to minimize the cumulative loss $\sum_{t \in [T]} L_t(a_t)$ by learning to play actions based on its previous online observations $\{a_1, a_2, \ldots, a_{t-1}; L_1, L_2, \ldots, L_{t-1}\}$.

Comparing the aforementioned multi-arm bandit problem setup and the three scenarios in Section 3.3.1, we note that the latter can be viewed as a special instance of multi-arm bandit problem despite the difference in loss function definitions across the three scenarios. The joint camera action gives the finite action set: $\mathcal{A} = \prod_{i \in \mathcal{N}^c} \mathcal{A}^i$, and the time-varying loss function is specified by the traffic state: $L_t(\cdot) = L(\cdot; s_t)$. Recasting the multi-level real-time traffic information collection and processing as a multi-arm bandit (online learning) problem brings several advantages over the control and RL approaches. First, the time-varying loss sequences picked by nature vividly capture the complex and dynamic urban traffic and, particularly, the

11

unforeseen traffic incidents. Second, as presented in the later discussion, multi-arm bandit problems admit lightweight and elegant algorithmic solutions that are more intuitive and explainable than black-box deep RL policy networks.

In the sequel, we unfold our PiCOL method using the network-level flow capturing as an example. The resulting learning scheme is referred to as PiCOL-Net. Our treatment below directly lends itself to the route and link-level tasks, and we refer to the corresponding learning-based control algorithms as PiCOL-Route and PiCOL-Link, respectively. If no confusion arises from the context, we simply use PiCOL for reference. To facilitate our discussion, we begin with the assumption that the true traffic state s_t is revealed to the controller *after* the joint action a_t such that the incurred loss defined in Eq. (1) at each time step can be computed. This unrealistic assumption will be removed later when we use the traffic state prediction as a surrogate. Suppose the controller's joint policy at time t is $\pi_t \in \Delta(\mathcal{A})$, from which the realized joint action a_t is sampled and executed. The resulting loss is the absolute percentage error defined in Eq. (1): $L(a_t, s_t) = \|s_t \otimes a_t - s_t\|_1 / \|s_t\|$. Beginning with a uniform distribution over the joint action set $\pi_1 = \text{Unif}(\mathcal{A})$, one can calibrate the policy using the exponential weight algorithm [42] presented in Eq. (4) with $\gamma_t \in (0, 1)$ being the discounting factor.

$$\pi_{t+1}(a) = \frac{\pi_t(a) \exp\{-\gamma_t L(a; s_t)\}}{\sum_{a \in \mathcal{A}} \pi_t(a) \exp\{-\gamma_t L(a; s_t)\}}, a \in \mathcal{A}. \tag{4}$$

EW is named after the calibration of the probability $\pi_{t+1}(a)$ using exponential weights $\exp\{-\gamma_t L(a; s_t)\}$. For each action a, the larger the loss (error) it incurs, the smaller the exponential becomes, and consequently, the smaller its weight becomes at the next step. In plain words, the EW algorithm begins with equal probability (weights) for every possible joint camera titling action and then reduces the weights of less desired joint actions by multiplying their previous weights by the exponential.

Simple and intuitive, the EW algorithm has proved to be an effective online learning scheme in the face of non-stationary loss sequences [43]. The exponential weight calibration enables the TTC-X to adapt rapidly to dynamic traffic situations. However, the calibration defined in Eq. (4) is performed centralized, which suffers from the curse of dimensionality as the number of camera nodes and associated incident edges grow. One natural remedy is to distribute the policy update and perform the calibration at each node. However, the challenge of this parallel calibration is that the joint action of all cameras determines the loss, and hence, assigning proper weights to each camera's action is the key to a distributed EW suitable for large-scale TTC-X. Inspired by game-theoretic learning methods[61], we introduce the *correlated* exponential weight (CEW) to evaluate the contribution of each individual action to the network-level loss conditional on others' actions. Mathematically, the CEW for the node i's action a^i at time t is $\exp\{-\gamma_t L(a^i, a_t^{-i}; s_t)\}$, given that the other nodes' joint action is a_t^{-i}. Unlike the weight in centralized EW adapting the joint action to the changing traffic state for smaller loss, CEW adapts individual actions to other nodes' actions for better coordination (greater flow captured) across TTC when facing non-stationary state evolution. Define the control policy at each camera node as $\pi_t^i \in \Delta(\mathcal{A}^i)$ in Eq. (5), and the correlated exponential weight algorithm is presented in Eq. (5).

$$\pi_{t+1}^i(a^i) = \frac{\pi_t^i(a^i) \exp\{-\gamma_t L(a^i, a_t^{-i}; s_t)\}}{\sum_{a^i \in \mathcal{A}^i} \pi_t^i(a^i) \exp\{-\gamma_t L(a^i, a_t^{-i}; s_t)\}}, a^i \in \mathcal{A}^i. \tag{5}$$

In summary, the CEW algorithm above begins with equal probability for every individual action at each node and then increases the weights of actions better coordinated with other nodes' actions by multiplying their previous probability with correlated exponential weights.

Our discussion has developed so far under the assumption that the actual traffic state s_t is revealed to the controller at each time step. Yet, the controller can only receive the partial state information, $s_t^c =$

$s_t \otimes a_t$, captured by the cameras. Consequently, the exponential weights of actions not employed cannot be computed, i.e., $\exp\{-\gamma_t L(a^i, a_t^{-i}; s_t)\}$, for $a^i \neq a_t^i$, are unknown, which prevents CEW algorithm in Eq. (5) from real-world deployment.

The case of missing data in online observation resonates with traffic-sensing failure, where traffic sensors fail to function properly due to short life cycles, unanticipated use, or adversarial attacks. As a well-accepted methodology to deal with traffic-sensing failure, traffic data imputation, referring to the process of estimating and filling in missing or erroneous data points in traffic datasets, naturally lends itself to the problem of online learning control with limited observations. Taking inspiration from spatial-temporal forecasting methods [68], we develop a traffic state forecasting model, referred to as the predictor, by combining graph convolution network [69] and transformer [37]. Section 3.3.3 presents a detailed description, and we here abstract from this machine learning machinery and treat it as a mapping $\mathscr{F}$ to complete our algorithm development on PiCOL.

During the online implementation, the predictor $\mathscr{F}$ takes in a batch of past partial observation $\{s_\tau^c\}_{\tau=t-K}^{t-1}$ and predicts a batch of future traffic states of the same batch size, denoted by $\{\hat{s}_\tau\}_{\tau=t}^{t+K-1}$. In other words, the predictor forecasts future K-step traffic state evolution based on the most recent K-step observations. At the t-th step, we first carry out a data fusion operation that integrates the actual partial observation s_t^c and the prediction $\hat{s}_t$. This fusion operation fills the missing data point in s_t^c with the corresponding entries from $\hat{s}_t$, meaning that the prediction $\hat{s}_t^{ij}$ serves as the surrogate for the missing local state information s_t^{ij}, for (i, j) not covered by any cameras. Recall that the joint action a_t is a binary vector of $|\mathcal{E}|$-dimension, and its non-zero entries indicate the links covered by at least one camera at time t. Let $\mathbb{1}$ be the $|\mathcal{E}|$-dimensional all-one vector, the fusion state, denoted by $\hat{s}_t^f$, is then define as $\hat{s}_t^f = s_t^c + \hat{s}_t \otimes (\mathbb{1} - a_t) = s_t \otimes a_t + \hat{s}_t \otimes (\mathbb{1} - a_t)$, which is a substitute of the true state. Consequently, one can compute the CEW using the fusion, leading to the proposed PiCOL scheme:

$$\hat{s}_t = \mathscr{F}(\{\hat{s}_\tau^f\}_{\tau=t}^{t+K-1}), \tag{6a}$$

$$\hat{s}_t^f = s_t^c + \hat{s}_t \otimes (\mathbb{1} - a_t), \tag{6b}$$

$$\pi_{t+1}^i(a^i) = (1 - \epsilon_t) \frac{\pi_t^i(a^i) \exp\{-\gamma_t L(a^i, a_t^{-i}; \hat{s}_t^f)\}}{\sum_{a^i \in \mathcal{A}^i} \pi_t^i(a^i) \exp\{-\gamma_t L(a^i, a_t^{-i}; \hat{s}_t^f)\}} + \epsilon_t \frac{1}{|\mathcal{A}^i|}, a^i \in \mathcal{A}^i. \tag{6c}$$

One may notice that the PiCOL scheme admits an additional ϵ-exploration term in Eq. (6c), subscribing to the seminal online learning algorithm *exploration and exploitation with exponential weights* (EXP3) [43]. As we later present in Section 3.4, the exploration term is instrumental in PiCOL-Link, encouraging the PTC to probe every edge with positive probabilities. As such, TTC-X is more likely to directly capture the sudden change in link-level traffic conditions or infer the incidents by observing fluctuations on neighboring edges through the predictor. In general, the explorations contribute to accurate predictions, which, in turn, benefits online policy calibration. To fully explain the interconnection between the controller and the predictor, we first articulate the inner workings of the predictor in the subsequent.

3.3.3. Spatial-Temporal Graph Predictor (STGP)

The motivation for introducing the STGP into PiCOL is similar to the data imputation task in traffic forecasting problems. Traffic data imputation, which is often used to deal with the failure of traffic sensing systems, spatial-temporal forecasting methods [68] are preferred due to their effectiveness in learning the network-structure nature as well as the sequential characteristics of traffic states. Similarly, in the context of PiCOL, deploying a limited number of Public Transportation Cameras (PTCs) within an extensive transportation network often results in underrepresentation. This limitation arises because each PTC can monitor only one direction at a time, necessitating confident estimations for unobserved directions.

13

Traffic State Estimation Problem Formulation.. The formulation of STGP is straightforward, adhering to the notations in Table.1. The traffic states of road segments on the traffic network $\mathcal{G}$ at time step t are denoted by traffic state matrix $s_t = (s_t^{ij})$, where $(i, j) \in \mathcal{E}, i, j \in \mathcal{N}$. Given a sequence of recent historical traffic states $\{s_\tau\}_{\tau=t-K}^{t-1}$ over the past K time steps, STGP aims to predict the sequence of future traffic states, denoted by $\{\hat{s}_\tau\}_{\tau=t}^{t+K-1}$, over the next K time steps. It is important to note that STGP is predicting traffic states $\{\hat{s}_\tau\}_{\tau=t}^{t+K-1}$ over each road segment, which is each edge in $\mathcal{G}$, denoted by $(i, j) \in \mathcal{E}$.

The design of STGP is open-ended and flexible; one requirement of STGP is to capture the spatial dependencies among road segments, acknowledging that traffic propagates in short- and long-range within the road network. Another favor of STGP is to capture both stationary and non-stationary temporal trends in the evolution of traffic states. This study takes the encoder-decoder architecture following the design of vanilla transformer [37] and integrates the graph convolution kernel and global attention kernel inspired by [69].

Encoder-decoder architecture. The encoder transforms the input sequence $\{s_\tau^{t-1}_{t-K}\}$ into a latent representation $\{\tilde{s}_\tau\}_{\tau=t-K}^{t-1}$, which the decoder then uses to predict $\{\hat{s}_\tau\}_{\tau=t}^{t+K-1}$ with masked attention. This process employs a moving window of length K in an auto-regressive manner, where each time step's prediction builds upon the last. As shown in Figure 5, both the encoder and decoder comprise a multi-head self-attention mechanism, with the decoder incorporating an additional cross-attention sublayer to engage with the encoder's output.

Graph convolution kernel. The original design of the graph convolution kernel follows the well-established layer-wise propagation rule proposed in [70], which can be formally defined as follows for directed graph:

$$H^{(l+1)} = \sigma(\tilde{D}^{-1}\tilde{A}H^{(l)}W^{(l)}). \tag{7}$$

where $H^{(l)} \in \mathbb{R}^{E \times d_{\text{model}}}$ is the graph signal representation in the lth layer. It is important to note that in this study, matrix A describes the connectivity between edges $(i, j) \in withinset\mathcal{E}$, rather than between nodes, as the node set $\mathcal{N}$ represents each intersection. E is the number of edges in $\mathcal{E}$. $\tilde{A} = A + I_E$ is the adjacency matrix of the directed graph G with added self-connections. $I_E \in \mathbb{R}^{E \times E}$ is the identity matrix. D is degree matrix and $W^{(l)}$ is layer-specific trainable weights. $\sigma(\cdot)$ is the nonlinear activation function.

Given the dynamic nature of real-world transportation networks, this study adapts the spatial correlation weight matrix to learn spatial dependencies dynamically [69]. The spatial weight matrix SA_t is thus defined to modify A through an element-wise dot-product operation:

$$\mathbf{S} = \text{softmax}\Big(\frac{H^{(l)}H^{(l)\top}}{\sqrt{d_{\text{model}}}}\Big) \in \mathbb{R}^{E \times E}. \tag{8}$$

$$\text{GCN}(H^{(l)}, \tilde{A}) = \sigma\Big(((\tilde{D}^{-1}\tilde{A}) \odot \mathbf{S})H^{(l)}W^{(l)}\Big). \tag{9}$$

Global attention kernel. Although the design of STGP is open-ended, it is not arbitrary. Given the constraints in PiCOL with a limited number of cameras as well as partial observations, it is favorable for STGP to estimate traffic states of unobserved road segments based on observed information from other road segments. While graph convolution captures the local (i.e., short-range) spatial dependencies residing in the traffic propagation, the long-range patterns also exist in traffic states along the road network. The multi-head attention mechanism allows each position in the traffic state sequence to attend to every other position, enabling the transformer to represent long-range patterns in traffic states [37]. Furthermore, acknowledging the complexity of traffic patterns often subject to non-stationary disturbances, the STGP needs to extend beyond
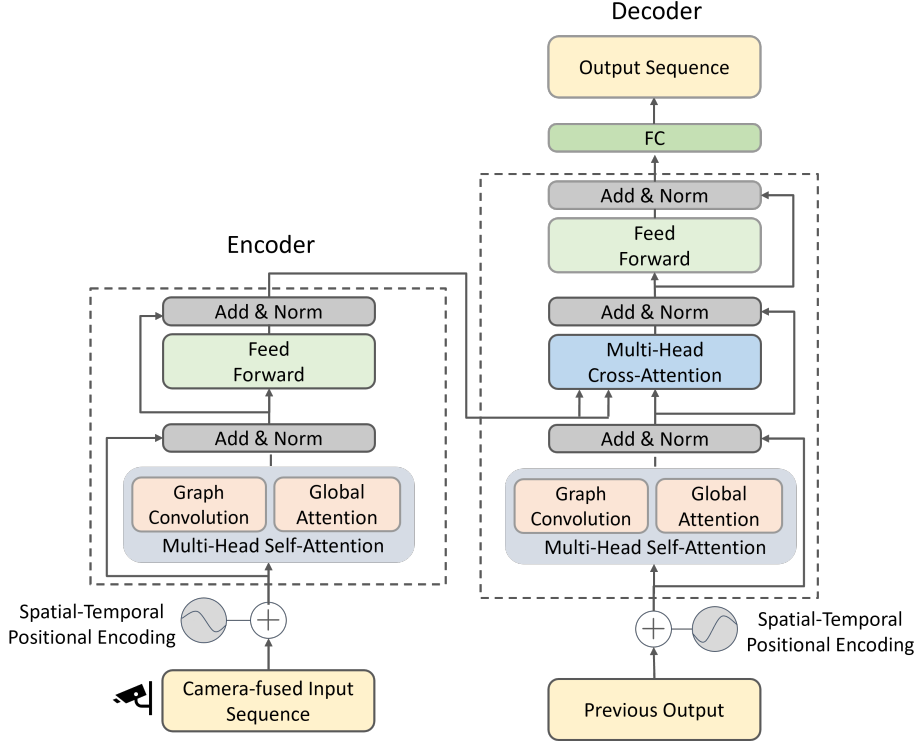
Figure 5: The architecture of STGP in TTC-X.

the standard multi-head attention. It incorporates 2D convolutional layers instead of the linear projection in the original multi-head attention mechanism. This change aims to extract the local trend of temporal patterns [69], as shown in Eq.10.

$$
\begin{aligned}
\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) &= \oplus(\text{head}_1, \cdots, \text{head}_h) W^O, \\
\text{head}_i &= \text{Attention}(\Phi_i^Q * \mathbf{Q}, \Phi_i^K * \mathbf{K}, \mathbf{V} W_i^V),
\end{aligned}
\tag{10}
$$

where Φ_i^Q and Φ_i^K are the parameters for the 2D convolutional layers, the query $Q \in \mathbb{R}^{L_Q \times d_{\text{model}}}$, key $K \in \mathbb{R}^{L_K \times d_{\text{model}}}$ and value $V \in \mathbb{R}^{L_V \times d_{\text{model}}}$ are both transformed from the input token sequence $X \in \mathbb{R}^{L \times C}$, and L_Q, L_K, L_V are the corresponding length. Linear projection $W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_k}$ maps dimension of d_{model} into head h_i.

3.4. The Interplay between Controller and Predictor

The spatial and temporal dependencies captured by STGP are the cornerstone of a successful PiCOL implementation, contributing to network-wide camera coordination and rapid response. Recall that the online calibration relies on a proper evaluation of CEW in Eq. (6c), which further depends on accurate predictions of traffic states on unobserved road segments. Thanks to the graph convolution in STGP, the predictor can infer one edge's missing local traffic state with high accuracy using observations from its neighboring edges. In other words, the fusion state $\hat{s}_t^f$ gives a decent substitute to the true state s_t and, consequently, effective exponential weights in online learning.

Figure 6: The simulation network building process. Left to right: Open Street Map, Network in SUMO, Abstracted Network.

In addition to the spatial aspect, STGP also learns offline the temporal correlation among the time series of traffic states through the global attention mechanism. The combination of multi-head attention with 2D convolutional layers enables STGP to capture local temporal trends and long-range (global) temporal dependencies among traffic states. When some cameras observe unanticipated traffic incidents, the learned local temporal trend helps STGP accurately forecast the future network traffic evolution within a short window under these abnormal conditions. Receiving the corresponding predictions, PiCOL quickly adjusts its policy calibration and responds rapidly to these unexpected traffic conditions by tilting neighboring cameras to those affected edges. One important note is that certain exploration in PiCOL (see Eq. (6c)) is necessary when dealing with unexpected traffic conditions. Incidents may likely take place on lesser-monitored edges with relatively small flow. Hence, PiCOL needs to assign non-zero weights to every edge for link-level change-point detection. In contrast to the local trend, the long-range temporal dependencies in STGP unveil the long-term evolution pattern caused by recurrent traffic conditions (e.g., rush hours) to the controller. Therefore, PiCOL prepares for these recurrent incidents by tilting cameras in relevant directions without worrying about losing track of important information from unobserved edges.

4. Results and discussion

4.1. Dataset summary

Simulation Environment. The dataset pivotal to the training and experimental phases of our TTC-X is generated using the Simulation of Urban Mobility (SUMO) [71]. SUMO is a versatile, open-source traffic simulation package designed to handle large networks and provide detailed representations of vehicular movements. This approach models each vehicle individually, allowing us to capture intricate interactions and behaviors in real-world traffic. This level of detail is essential for understanding and predicting traffic dynamics, especially in response to disruptions or changes in typical traffic patterns. The simulated environment for our study is meticulously crafted to mirror the dynamics of the Flatbush Avenue corridor, spanning from Dekalb Avenue to Fulton Street. This corridor is a microcosm of urban traffic complexity, making it an ideal candidate for our simulation needs. The process of network generation, depicted in Figure 6, involves a meticulous recreation of this urban corridor within the SUMO environment.

Calibration of the Simulation. The realism and accuracy of the simulation are paramount. The simulation is calibrated using real-world data from multiple sources to achieve this. This data encompasses various aspects of urban traffic, such as travel times, volumes, speeds, and environmental factors like weather conditions. Additionally, surrogate safety measurements are included to provide a more comprehensive understanding
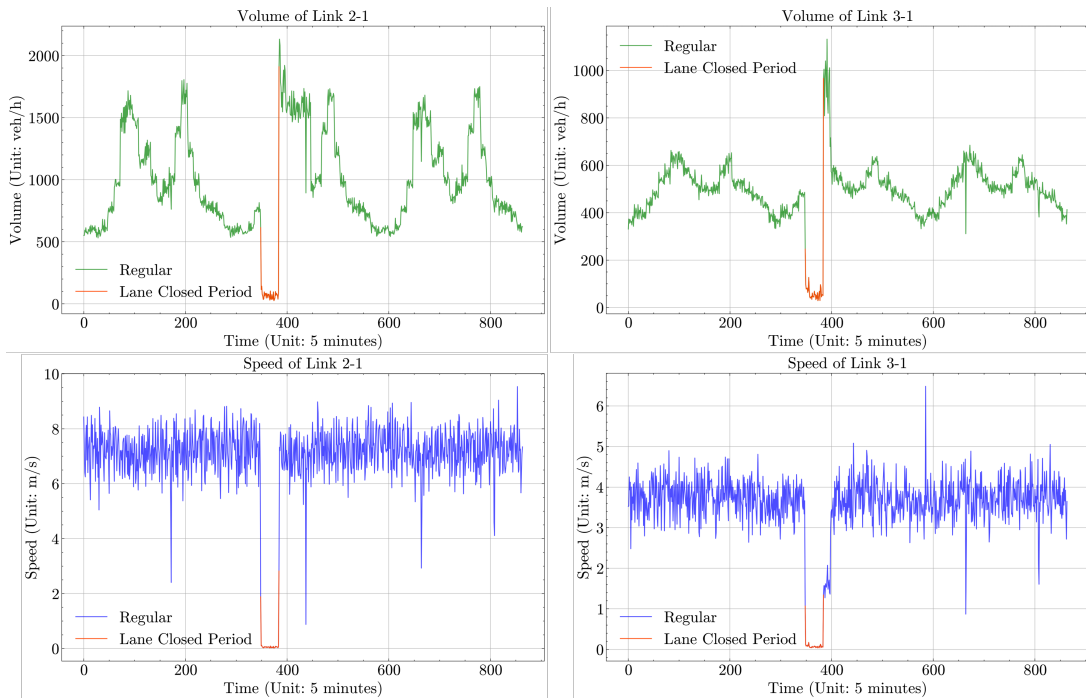
Figure 7: The volume and speed data sample for different links.

of traffic dynamics and safety scenarios. This multi-faceted calibration ensures that our simulation closely mirrors actual traffic conditions and behaviors observed on Flatbush Avenue.

Dataset Generation. For the foundation of our dataset, we generated data representing 200 days of base traffic conditions. Each day in the simulation is detailed, encompassing 24 hours with a granularity of one-second step length. This high-resolution data provides an intricate view of daily traffic patterns and fluctuations.

We introduced change point data into the simulation to simulate traffic disruptions and study their impact. This involved randomly closing lanes for durations varying from one to five hours, thus creating 38 unique sets of change point scenarios. These disruptions are crucial for testing the robustness and responsiveness of the TTC-X under varying traffic conditions. The output from these simulations is comprehensive. For each edge in the network, we generated data that includes average volume and average speed, recorded on a minute-by-minute basis. This granularity allows for a nuanced understanding of traffic flow and speed patterns across different segments of the road network.

4.2. Experiment Setup

Predictor Configuration. The predictor in the TTC-X, based on the Spatial-Temporal Graph Prediction (STGP) model (Section.3.1), follows an offline learning approach. Data is divided into training, validation, and test sets in a 6:2:2 ratio and normalized to the range $[-1, 1]$ using min-max normalization. STGP implementation uses the Pytorch library [72] on a Windows-11 desktop with an AMD Ryzen 9 7900X, 64.0 GB RAM, and an NVidia RTX 3090Ti. Optimal hyperparameters determined from validation tests include

17

eight attention heads ($h = 8$), a learning rate of 0.001, and a model dimension of $d_{\text{model}} = 64$. Traffic state data is aggregated into 5-minute intervals, with both input and prediction sequences spanning one hour (12-time steps), balancing detail and computational efficiency for effective traffic state prediction.

Controller Configuration. In accordance with the minute-by-minute data generation, the controller also operates on a minute-by-minute basis in the simulation network, and the horizon length is $T = 1440$ (a day). Since each prediction instance denotes the average traffic state forecast for a 5-minute interval, the controller uses the one prediction instance repeatedly five times within the interval until a new prediction is generated for the next 5-minute interval. The motivation behind this 5-minute predictor operation cycle is that calling the predictor usually consumes 0.8-1 seconds. Hence, we choose a 5-minute cycle instead of a minute-by-minute cycle to lower the time complexity in experiments. As mentioned above, the predictor's batch size is $K = 12$, so the input sequence spans one hour. Consequently, the first hour of the day is the warm-up period, where each PTC uniformly picks one edge to monitor at each time step. The starting time is 1 am, and the proposed PiCOL algorithm takes over afterward.

For the network-level maximum flow capture task, we set the exploration rate of PiCOL-Net to zero, whereas the rate becomes 0.3 in the route-level and link-level tasks. In particular, we compare the PiCOL-Link's performance under 0-exploration and 0.3 exploration in the link-level forecasting and reconstruction tasks to showcase the role of exploration. On a separate note, we consider purely predictor-based forecasting and reconstruction, in addition to PiCOL-Link with 0/0.3-exploration. After the warm-up, the predictor-based method discards the online observations and only uses past predictions to forecast and reconstruct the link-level traffic state. All PiCOL variants under different tasks share the same discounting factor 1. Unless otherwise specified, all results on MAE and MAPE are obtained from 20 repeated experiments with different traffic simulation data and random seeds. All experiments are carried out on the same desktop mentioned above.

4.3. Network-level: Maximum Flow Capture

To evaluate PiCOL-Net's capability in network-level flow capturing, we consider the maximum number of vehicles captured by PTCs in TTC-X. Towards this end, we employ two metrics: mean absolute error (MAE) and mean absolute percentage error (MAPE) of the observed volume over the true volume within the k-th hour. Using notations defined in Table 1, the two metrics are defined in Eq.11, where k stands for the hour index.

$$ObsMAE_k = 1/60 \sum_{t=60k}^{60(k+1)-1} \|\boldsymbol{s}_t^c - \boldsymbol{s}_t\|_1$$

$$ObsMAPE_k = 1/60 \sum_{t=60k}^{60(k+1)-1} \|\boldsymbol{s}_t^c - \boldsymbol{s}_t\|_1 / \|\boldsymbol{s}_t\|_1 \tag{11}$$

$$FusionMAE_k = 1/60 \sum_{t=60k}^{60(k+1)-1} \|\hat{\boldsymbol{s}}_t^f - \boldsymbol{s}_t\|_1$$

$$FusionMAPE_k = 1/60 \sum_{t=60k}^{60(k+1)-1} \|\hat{\boldsymbol{s}}_t^f - \boldsymbol{s}_t\|_1 / \|\boldsymbol{s}_t\|_1 \tag{12}$$

In addition to observation data, we also consider the maximum traffic flow estimated by the TTC-X, reflecting the total travel demand within the road network, which can also be split into inbound and outbound travel demand. With the knowledge of network-wide travel demand, traffic managers can better understand

the current traffic situations and make corresponding operation or planning strategies. Same as the argument above, we use MAE and MAPE of the fusion data over the true data as evaluation metrics, and the definitions are shown in Eq.12, respectively. The lower the MAPE is, the less information is lost regarding estimated traffic demand within the road network.

Table 2 summarizes the evaluation of the four metrics. The lower part of Figure 8 indicates the distribution of network-wide total traffic volume throughout the day; it is found that the total network-wide traffic volume ranges between 6000 vehicle/hour and 20000 vehicle/hour with clear temporal patterns. By a close look at *ObsMAPE*, one can see that TTC-X can capture over 60% of the total number of vehicles within the road network every time of day, as visualized in Figure 8. Despite the variant of temporal patterns in traffic volume evolution shown in Figure 8, TTC-X, powered by PiCOL, always keeps track of road segments with a significant number of vehicles, maintaining *ObsMAPE* and *ObsMAE* around a stable level. This stable performance is credited to the PTCs' coordination achieved by the correlated policy calibration in PiCOL, where PTCs cooperatively monitor the road network and adjust their policies to the dynamic traffic patterns in coordination. Furthermore, PTCs' online observations of those significant road segments under PiCOL policies serve as informative inputs to the predictor, which gives accurate inferences on unobserved edges with relatively small flows using the spatial and temporal dependencies. The inferences, in turn, help the controller to update the tilting strategies online. The interplay between the predictor and the controller enables TTC-X to derive an accurate estimation of total travel demand for every time of day, as evidenced by the low *FusionMAPE* across the whole day (less than 10%) in Table 2. The results presented in this section emphasize the cost-efficiency of the TTC-X. Using just six PTCs, TTC-X successfully detected over 60% of the total vehicle count within the road network. Moreover, it estimated the network-wide traffic volume with a Mean Absolute Percentage Error (MAPE) of less than 10%. This demonstrates the effectiveness of TTC-X in providing comprehensive traffic monitoring and estimation with a minimal camera setup.

4.4. Route-level: Dynamic Route Planning

To evaluate the performance of TTC-X at the route level, we examine PiCOL-Route's capability of dynamic route planning in a special testing case where an unexpected traffic incident takes place within the route planning time window. Specifically, we consider trips that need to traverse from origin Node 12 to destination Node 8 in the road network (Figure 9) starting from the 290th min, and an entire lane closure event takes place 10 min later. When vehicles arrive at a new intersection (a new node), we update the route recommendation with a new route using the predictor (i.e., derive the optimal path with the least travel time using prediction data), the proposed PiCOL (using the fusion data with online observations included), and the actual travel time information. We report the recommended route sequence and associated estimated time of arrival (ETA) provided by the three approaches above in Table 3, and a visualization of the route sequence is presented in Figure 9.

In Figure 9, each route recommendation is indicated by colored arrows, with darker shades representing more recent recommendations. For example, the lightest purple arrows in the PiCOL plot indicate the route recommended to the vehicle when arriving at Node 12. Comparing the routes and time estimates under the three methods, we notice that PiCOL's initial planning (12-6-4-2-1-8) is the same as the predictor's, which differs from the true optimal path. This is because PiCOL largely relies on the prediction to figure out each path's travel time when there is insufficient information during the initial exploration. As PiCOL begins to calibrate its policy using online observations and explore different paths, the time estimate gets closer to the truth (see the 2nd row in Table 3). Thanks to PiCOL's rapid response to real-time traffic fluctuations, it timely captures the lane closure on $3 - 1$ and swiftly replans the route that bypasses the closed road segment when the vehicle arrives at node 4. In contrast, the predictor fails to respond to the incident and continues to recommend 4-3-1-8, and the estimate deviates much further from the truth.
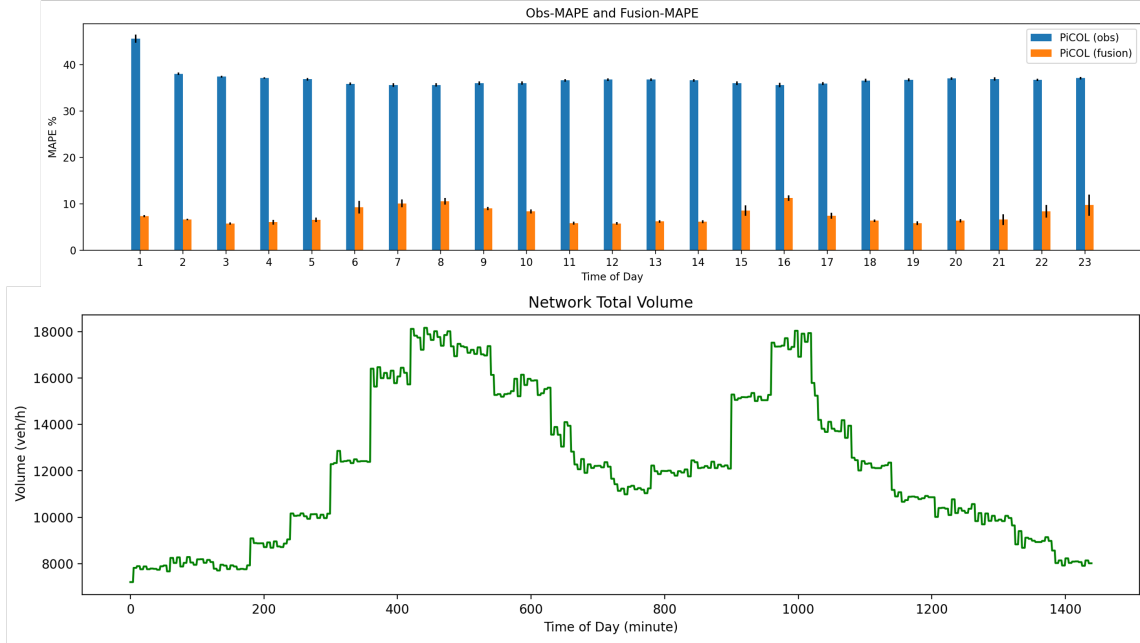
Figure 8: The bar plot of the hourly MAPE for OBS and Fusion. Employing PiCOL-Net, TTC-X can capture over 60% of the total number of vehicles every time of day when facing dynamic traffic patterns. Moreover, the fusion data offer highly accurate estimates of the underlying traffic volumes.

This case study underscores TTC-X's adaptability and real-time response ability in real-time dynamic route planning, which is crucial for managing unexpected traffic situations like major lane closure events. PiCOL's online policy calibration allows TTC-X to swiftly adjust to such disruptions, providing real-time optimal routing with accurate travel time estimates. This responsiveness helps vehicles avoid traffic incidents effectively. In comparison, GPS-based navigation methods rely heavily on the density of probe vehicles to aggregate travel time for each road segment. This navigation is slow in response to sudden changes compared to TTC-X, which feeds real-time high-resolution information to PiCOL and makes fast and accurate decisions on route planning. This capability is not only beneficial for routine traffic management but is also critical in life-saving scenarios involving emergency vehicles. In such situations, the high accuracy and real-time operation provided by TTC-X are essential for efficient traffic management, ensuring that emergency responders can navigate through traffic with minimal delay.

4.5. Link-level: Traffic State Forecasting and Reconstruction

In this section, we assess the direct output of PiCOL-Link, which estimates traffic flow for individual road segments within the network. As outlined in Section 3.3.1, PiCOL-Link's loss function emphasizes the relative differences in traffic volume in two consecutive time steps, particularly focusing on segments with significant traffic fluctuations (e.g., abrupt increases or decreases in volume). At the link level, we define two key sub-tasks: 1) the forecasting task and 2) the reconstruction task. The forecasting task leverages the predictive capabilities of the predictor to forecast future traffic volumes, utilizing input data refined by the controller's observations. This task empowers traffic managers with foresight, allowing them to anticipate potential congestion or identify segments prone to traffic build-up, thus enabling early response

		1 am	2 am	3 am	4 am	5 am
Obs	MAPE	45.60 ± 0.87	38.01 ± 0.27	37.40 ± 0.19	37.11 ± 0.15	36.83 ± 0.28
Obs	MAE	3700.29 ± 68.57	2985.74 ± 20.99	3315.45 ± 16.98	3739.35 ± 15.01	4577.32 ± 35.17
Fusion	MAPE	7.38 ± 0.22	6.66 ± 0.15	5.77 ± 0.25	6.08 ± 0.50	6.59 ± 0.51
Fusion	MAE	596.92 ± 17.40	521.24 ± 11.61	512.31 ± 22.07	614.03 ± 50.01	821.63 ± 66.57
		6am	7 am	8 am	9 am	10 am
Obs	MAPE	35.86 ± 0.28	35.58 ± 0.40	35.62 ± 0.38	36.00 ± 0.38	36.00 ± 0.37
Obs	MAE	5777.06 ± 45.28	6334.23 ± 72.11	6129.33 ± 65.31	5608.93 ± 58.16	5274.64 ± 54.10
Fusion	MAPE	9.28 ± 1.37	10.12 ± 0.84	10.57 ± 0.73	9.03 ± 0.38	8.40 ± 0.44
Fusion	MAE	1512.86 ± 220.45	1819.67 ± 150.10	1834.54 ± 125.20	1425.07 ± 59.34	1247.56 ± 66.60
		11 pm	12 pm	13 pm	14 pm	15 pm
Obs	MAPE	36.59 ± 0.28	36.79 ± 0.28	36.75 ± 0.26	36.62 ± 0.28	35.98 ± 0.40
Obs	MAE	4485.29 ± 33.91	4140.98 ± 31.13	4397.28 ± 30.92	4477.97 ± 33.75	5452.79 ± 60.79
Fusion	MAPE	5.90 ± 0.29	5.80 ± 0.29	6.23 ± 0.27	6.18 ± 0.35	8.58 ± 1.12
Fusion	MAE	727.78 ± 40.41	653.53 ± 34.55	748.26 ± 33.83	756.16 ± 42.32	1310.91 ± 170.30
		16 pm	17 pm	18 pm	19 pm	20 pm
Obs	MAPE	35.60 ± 0.46	35.91 ± 0.34	36.57 ± 0.36	36.68 ± 0.32	37.02 ± 0.27
Obs	MAE	6238.18 ± 80.32	5076.98 ± 48.38	4488.97 ± 43.12	3993.01 ± 35.14	3834.64 ± 28.10
Fusion	MAPE	11.28 ± 0.60	7.47 ± 0.66	6.40 ± 0.27	5.86 ± 0.43	6.40 ± 0.37
Fusion	MAE	1987.12 ± 104.12	1068.34 ± 93.14	788.01 ± 34.88	636.69 ± 45.42	656.43 ± 36.00
		21 pm	22 pm	23 pm		
Obs	MAPE	36.90 ± 0.33	36.72 ± 0.26	37.07 ± 0.25		
Obs	MAE	3688.67 ± 33.27	3327.74 ± 23.33	2003.48 ± 20.73		
Fusion	MAPE	6.65 ± 1.15	8.43 ± 1.38	9.75 ± 2.26		
Fusion	MAE	659.71 ± 111.69	753.52 ± 119.97	779.91 ± 181.60		

Table 2: The MAPE and MAE of Obs and fusion data per hour. Despite the time-varying traffic pattern within a day, PiCOL-Net always maintains a stable ObsMAPE around 40% and FusionMAPE of less than 10%. Given that the total network-wide traffic volumes range from 6000 veh/h and 20000 veh/h, the ObsMAE and FusionMAE indicate that PiCOL-Net successfully captures a significant share of total vehicle numbers for every time of day.

and resource allocation to mitigate congestion. The reconstruction task, on the other hand, aims to depict real-time traffic evolution across each road segment. Its goal is to reconstruct traffic flow to match the actual state closely, demonstrating TTC-X's capacity to utilize a limited number of PTCs for comprehensive network coverage, underscoring its cost efficiency. We evaluated PiCOL-Link using a one-day traffic dataset, including a full-lane closure on Edge 2-1 starting at 300 minutes and lasting 2 hours. As illustrated in the top right of Figure 10, this incident led to a sharp traffic volume decrease at 300 minutes, with a rebound around 420 minutes post-event. Table 4 displays the edge-wise Mean Absolute Error (MAE), defined in Eq. 13, for both forecasting and reconstruction tasks across various road segments, revealing accurate results with MAEs ranging from 12.36 to 371.96 vehicles/hour for forecasting and 1.25 to 344.54 vehicles/hour for

		Predictor		PiCOL		Truth	
		Route	ETA	Route	ETA	Route	ETA
←—time	Before	12-6-4-2-1-8	25 min 34.44 s	12-6-4-2-1-8	25 min 29.33 s	12-6-4-3-1-8	25 min 62.14 s
		6-4-3-1-8	20 min 28.56 s	6-4-3-1-8	20 min 40.68 s	6-4-3-1-8	20 min 58.31 s
	After	4-3-1-8	15 min 20.98 s	4-2-1-8	15 min 100.95 s	4-2-1-8	15 min 99.29 s
		3-1-8	10min 9.80 s	2-1-8	10min 23.63 s	2-1-8	10min 23.63 s
		1-8	5 min	1-8	5 min	1-8	5 min

Table 3: Sequence of route recommendations and corresponding ETAs under Predictor, PiCOL, and the true data. As time proceeds, PiCOL-Route acquires enough online observations, leading to more accurate ETAs starting from Node 6. When the lane closure suddenly happens on 3-1, PiCOL-Route is able to rapidly respond to this incident by re-planning the route, as shown in the 3rd row.
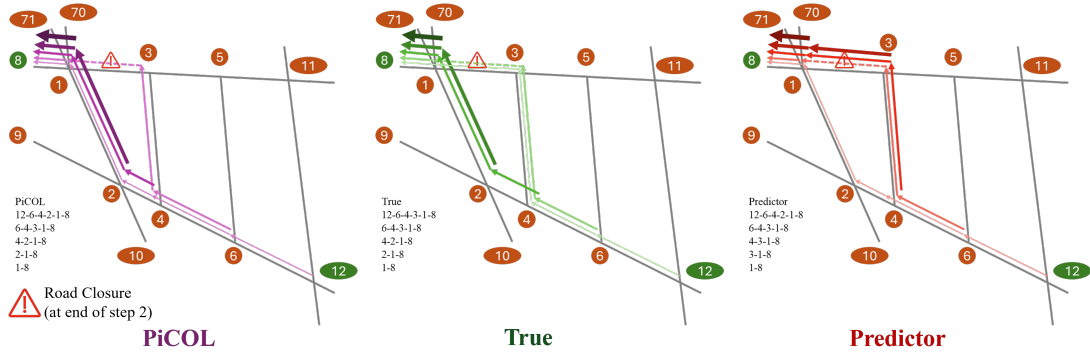

Figure 9: Dynamic route planning using PiCOL, Predictor, and True data. In this case, real-time dynamic routes are planned for trips traversing from origin Node 12 to destination Node 8, shown in green color.

reconstruction.

$$ForecastingMAE^{ij} = 1/1440 \sum_{t=0}^{1439} |\hat{s}_t^{ij} - s_t^{ij}|,$$

$$ReconstructionMAE^{ij} = 1/1440 \sum_{t=0}^{1439} |\hat{s}_t^{f,ij} - s_t^{ij}|, \text{ for} (i, j) \in \mathcal{E}.$$

(13)

Comparing two PiCOL settings (refer to Section 4.2), PiCOL-3 generally outperforms PiCOL-0 in both tasks, particularly at Edge 2-1 and neighboring edges affected by the lane closure. This outcome highlights PiCOL-Link's design strategy of incorporating exploration to enhance coverage and responsiveness to sudden disturbances.

Visualizations in Figure 10 and Figure 11 display selected edge results for both tasks. The green line represents the ground truth traffic volume, orange for PiCOL-0, red for PiCOL-3 (0.3 exploration), and blue for the predictor. The vertical dashed line indicates the start time index for the PiCOL-Link module after the warm-up period from 0 AM to 1 AM, as detailed in Section 4.2.

PiCOL accurately captured the lane closure event on Edge 2-1 and the subsequent queue buildup on its upstream Edge 4-2, demonstrating PiCOL-Link's rapid response capability within TTC-X to sudden traffic changes. This is further corroborated in Figure 12, which shows PTC coverage changes at Node 1. The
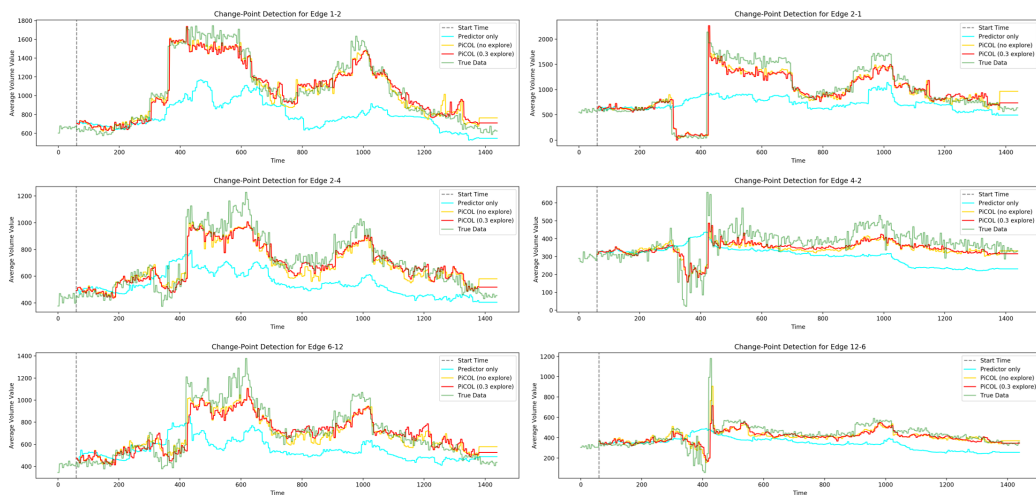
Figure 10: Selected edge results for link-level forecasting task. PiCOL-3 outperforms PiCO-0 in general, particularly regarding the forecast on Edge 2-1 and its neighboring Edge 2-4 and 4-2.

green line depicts real traffic volume on each edge, and the light blue vertical bars indicate PTC coverage at Intersection 1, which is capable of monitoring six edges. This finding confirms TTC-X's ability to detect and quickly respond to traffic disturbances, with its focus on monitoring Edge 2-1 around the time of the lane closure event.

5. Implication

Connection with the Cyber-Physical World. The deployment of the Traffic-responsive Tilt Camera surveillance system (TTC-X) within a cyber-physical framework presents a strategic alternative to direct real-world implementation, primarily due to challenges in safety and cost. Implementing such advanced systems directly in real-world settings often involves high costs and potential safety risks, especially during the initial deployment phases. For example, adjusting traffic control in real-time based on system recommendations without thorough testing could lead to unpredictable scenarios, potentially endangering public safety and requiring significant financial investment for widespread installation and maintenance.

To mitigate these challenges, the cyber-physical world serves as a bridge, allowing for the collecting and processing of real-world data in a controlled, virtual environment. This approach provides decision-supporting information or direct guidance to transportation agencies without the immediate risks and costs associated with physical deployment. The CARLA-SUMO co-simulation system is a prime example of this implementation, which creates a digital twin of the real-world traffic environment [73]. This system allows for the testing and refining of the TTC-X in a virtual setup that closely mirrors actual traffic conditions. Figure 13 shows a sample of the cyber-physical world and the connection to the proposed framework (Detector + PiCOL).

In the CARLA-SUMO co-simulation, various scenarios, such as network-level traffic flow optimization, route-level dynamic planning, and link-level traffic state forecasting, can be replicated and analyzed. This digital twin setup provides a safe and cost-effective platform for rigorous testing and optimization of the TTC-X framework, ensuring its readiness and reliability for eventual real-world deployment.
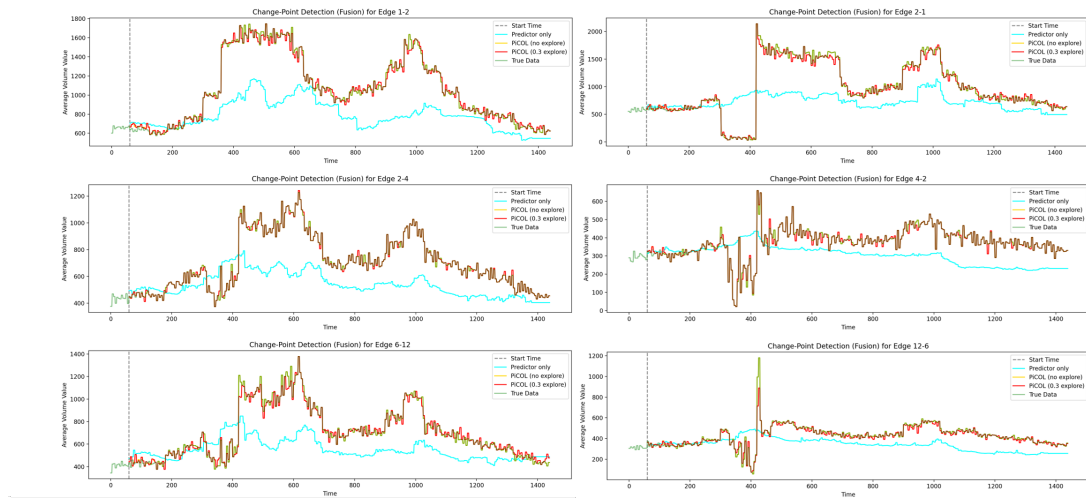
23

Figure 11: Selected edge results for link-level reconstruction task. Both PiCOL instances achieve impressive reconstruction accuracy. PiCOL-3 slightly outperforms on Edge 2-1 and its neighboring edges.
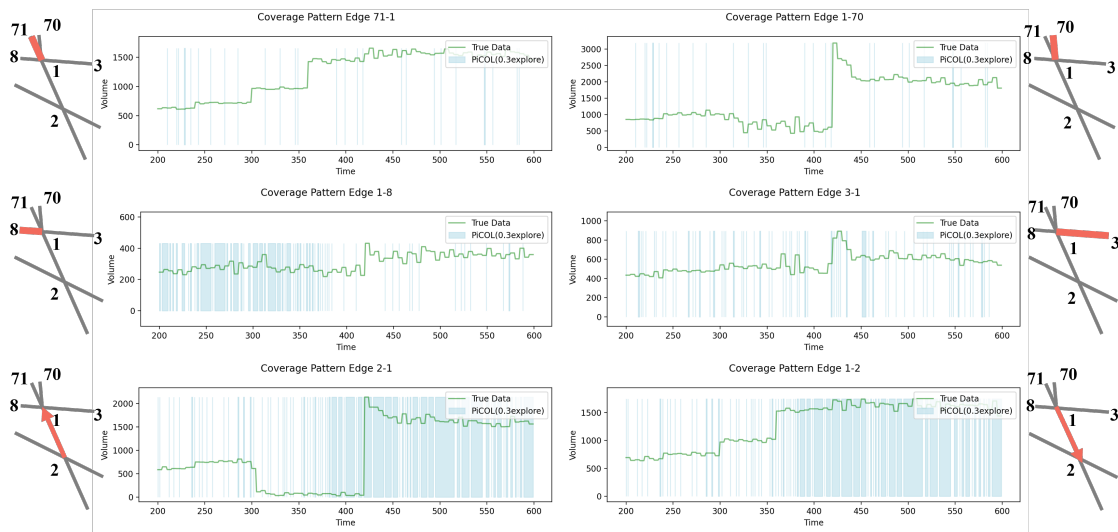

Figure 12: PTC coverage patterns at Intersection 1 (Node 1) under PiCOL-3. The light blue bars indicate the PTC coverage at each time step, showing that PiCOL successfully captured the lane closure on Edge 2-1 and the subsequent queue buildup on its upstream Edge 4-2.

		1-2	1-70	1-8	2-1	2-4
Forecasting	PiCOL-0	206.43 ± 75.45	414.19 ± 69.96	74.22 ± 5.00	287.19 ± 90.57	**125.61 ± 7.86**
	PiCOL-3	**156.31 ± 26.39**	**371.96 ± 18.14**	**73.78 ± 2.21**	**223.14 ± 33.31**	126.75 ± 4.29
Reconstruction	PiCOL-0	146.32 ± 131.95	393.43 ± 125.17	**45.28 ± 34.62**	195.47 ± 180.14	**17.64 ± 31.29**
	PiCOL-3	**75.87 ± 50.60**	**344.54 ± 17.64**	53.55 ± 21.19	**99.72 ± 71.73**	30.71 ± 21.15
		2-10	3-1	4-2	4-3	4-6
Forecasting	PiCOL-0	294.23 ± 77.81	101.93 ± 14.97	**107.22 ± 2.34**	22.56 ± 1.62	**134.92 ± 19.86**
	PiCOL-3	**264.95 ± 38.72**	**86.86 ± 4.95**	107.84 ± 1.91	**22.47 ± 1.08**	135.90 ± 12.99
Reconstruction	PiCOL-0	186.50 ± 173.74	100.57 ± 14.99	**12.39 ± 22.03**	**2.78 ± 4.56**	**48.09 ± 66.87**
	PiCOL-3	**177.82 ± 103.09**	**72.40 ± 4.79**	24.53 ± 16.70	5.47 ± 0.42	62.68 ± 44.74
		5-3	5-6	6-4	6-12	71-1
Forecasting	PiCOL-0	86.23 ± 15.66	**12.36 ± 1.44**	**86.59 ± 8.51**	154.00 ± 27.34	256.22 ± 66.21
	PiCOL-3	**70.21 ± 4.82**	12.37 ± 1.43	89.11 ± 8.98	**148.82 ± 12.16**	**208.92 ± 17.79**
Reconstruction	PiCOL-0	80.80 ± 24.13	**1.25 ± 0.15**	**30.15 ± 38.50**	82.26 ± 82.85	248.06 ± 85.57
	PiCOL-3	**57.33 ± 4.49**	3.16 ± 0.30	42.40 ± 29.14	192.16 ± 52.29	**194.29 ± 16.67**
		9-2	10-2	11-5	12-6	
Forecasting	PiCOL-0	105.35 ± 8.48	287.25 ± 85.46	102.13 ± 17.36	83.07 ± 11.23	
	PiCOL-3	**102.01 ± 4.36**	**272.61 ± 56.19**	**84.82 ± 6.34**	**82.19 ± 7.69**	
Reconstruction	PiCOL-0	101.39 ± 20.88	**182.21 ± 178.35**	101.60 ± 17.45	**44.11 ± 41.27**	
	PiCOL-3	**94.88 ± 4.07**	185.79 ± 118.02	**76.94 ± 5.62**	52.13 ± 28.57	

Table 4: The MAE of forecasting and reconstructed data of each edge. PiCOL-3 achieves smaller MAEs on Edge 2-1 (where incident happens), and its neighboring edges 1-2, 1-70, 2-10, 3-1, 71-1, and 9-2.

Figure 13: Cyber-Physical world implementation using CARLA-SUMO Co-simulation framework based on Flatbush corridor's network.

Connection with the Real World. The TTC-X is designed as a plug-and-play system for real-world application, emphasizing ease of deployment and cost-effectiveness. This approach ensures the system can be integrated into existing traffic management infrastructures with minimal disruption and additional expense. The system's flexibility allows it to be tailored to specific urban traffic scenarios, making it a versatile tool for a wide range of traffic management tasks.

For instance, the TTC-X can be rapidly deployed at key traffic junctions using existing camera infrastruc-

ture in a real-world scenario. Its adaptive nature allows for immediate improvements in traffic monitoring and management, such as adjusting signal timings in real-time based on traffic flow data, enhancing route planning during emergencies by providing accurate and updated traffic information, and rapidly responding to traffic incidents by forecasting and analyzing traffic state changes. This real-world applicability demonstrates the system's potential to significantly improve urban traffic management, reduce congestion, and enhance overall traffic safety while maintaining a cost-effective and efficient deployment strategy.

6. Conclusion

This study presents the Traffic-responsive Tilt Camera surveillance system (TTC-X), a comprehensive framework utilizing advanced machine learning for urban traffic management. The proposed system incorporates a novel Predictive Correlated Online Learning (PiCOL) module featuring the integration of an online learning controller and Spatial-Temporal Graph Predictor (STGP), enhancing real-time traffic estimation and camera control. Experiments were conducted in a simulated environment, which was calibrated using real-world traffic data from Brooklyn, New York, focusing on scenarios like maximum traffic flow capture, dynamic route planning, and link-level traffic state reconstruction and forecasting. TTC-X demonstrated significant improvements in capturing vehicles, adjusting optimal routes during incidents, and reconstructing traffic states. These scenarios demonstrated the capability of TTC-X in terms of its cost-effectiveness, rapid responsiveness to variant traffic situations as well as accurate capture of traffic states, showcasing its potential for effective traffic management in diverse urban settings.

Future work should consider expanding the system's adaptability by refining PiCOL for more real-world dynamic traffic scenarios and improving the PTC control with consideration of decentralization approaches, such as game-theoretic learning algorithms. Additionally, given the good scalability of TTC-X, we will consider adding more traffic objects such as pedestrians, cyclists, and infrastructure to showcase more interactions between different traffic objects such as vehicle-to-pedestrian, and vehicle-to-infrastructure applications.

7. Acknowledgements

8. Author Contributions

The authors confirm their contribution to the paper as follows. Conceptualization: Zilin Bian, Tao Li, Fan Zuo; Data curation: Fan Zuo and Haozhe Lei; Formal analysis: Tao Li, Zilin Bian, Haozhe Lei; Methodology: Tao Li, Zilin Bian, Haozhe Lei, Fan Zuo, Ya-Ting Yang; Validation: Tao Li, Zilin Bian, Haozhe Lei; Visualization: Ya-Ting Yang, Fan Zuo, Tao Li, Zilin Bian; Writing - original draft: Tao Li, Zilin Bian, Fan Zuo, Ya-Ting Yang; and Writing - review & editing: Tao Li, Zilin Bian, Fan Zuo, Kaan Ozbay, Quanyan Zhu, Zhenning Li. All authors reviewed the results and approved the final version of the manuscript.

References

[1] Zilin Bian, Fan Zuo, Jingqin Gao, Yanyan Chen, Sai Sarath Chandra Pavuluri Venkata, Suzana Duran Bernardes, Kaan Ozbay, Xuegang Jeff Ban, and Jingxing Wang. Time lag effects of covid-19 policies on transportation systems: A comparative study of new york city and seattle. *Transportation Research Part A: Policy and Practice*, 145:269–283, 2021.

[2] Thangarajah Akilan, QM Jonathan Wu, and Wandong Zhang. Video foreground extraction using multi-view receptive field and encoder–decoder dcnn for traffic and surveillance applications. *IEEE Transactions on Vehicular Technology*, 68(10):9478–9493, 2019.

[3] Arya Haghighat and Anuj Sharma. A computer vision-based deep learning model to detect wrong-way driving using pan–tilt–zoom traffic cameras. *Computer-Aided Civil and Infrastructure Engineering*, 38(1):119–132, 2023.

[4] Hao Frank Yang, Jiarui Cai, Chenxi Liu, Ruimin Ke, and Yinhai Wang. Cooperative multi-camera vehicle tracking and traffic surveillance with edge artificial intelligence and representation learning. *Transportation research part C: emerging technologies*, 148:103982, 2023.

[5] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection, 2016.

[6] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric, 2017.

[7] Zheng Tang and Jenq-Neng Hwang. Moana: An online learned adaptive appearance model for robust multiple object tracking in 3d. *IEEE Access*, 7:31934–31945, 2019.

[8] Bipul Neupane, Teerayut Horanont, and Jagannath Aryal. Real-time vehicle classification and tracking using a transfer learning-improved deep learning network. *Sensors*, 22(10):3813, 2022.

[9] Fei Shen, Xiaoyu Du, Liyan Zhang, and Jinhui Tang. Triplet contrastive learning for unsupervised vehicle re-identification. *arXiv preprint arXiv:2301.09498*, 2023.

[10] Zhili Zhou, Yujiang Li, Jin Li, Keping Yu, Guang Kou, Meimin Wang, and Brij Bhooshan Gupta. Gan-siamese network for cross-domain vehicle re-identification in intelligent transport systems. *IEEE Transactions on Network Science and Engineering*, 2022.

[11] Damilola Oladimeji, Khushi Gupta, Nuri Alperen Kose, Kubra Gundogan, Linqiang Ge, and Fan Liang. Smart transportation: an overview of technologies and applications. *Sensors*, 23(8):3880, 2023.

[12] Raik Aissaoui, Hamid Menouar, Amine Dhraief, Fethi Filali, Abdelfettah Belghith, and Adnan Abu-Dayya. Advanced real-time traffic monitoring system based on v2x communications. In *2014 IEEE International Conference on Communications (ICC)*, pages 2713–2718. IEEE, 2014.

[13] Kishwer Abdul Khaliq, Omer Chughtai, Abdullah Shahwani, Amir Qayyum, and Jürgen Pannek. Road accidents detection, data collection and data analysis using v2x communication and edge/cloud computing. *Electronics*, 8(8):896, 2019.

[14] Li Li, Rui Jiang, Zhengbing He, Xiqun Michael Chen, and Xuesong Zhou. Trajectory data-based traffic flow studies: A revisit. *Transportation Research Part C: Emerging Technologies*, 114:225–240, 2020.

[15] Julio A Sanguesa, Javier Barrachina, Manuel Fogue, Piedad Garrido, Francisco J Martinez, Juan-Carlos Cano, Carlos T Calafate, and Pietro Manzoni. Sensing traffic density combining v2v and v2i wireless communications. *Sensors*, 15(12):31794–31810, 2015.

[16] Heesub Rim, Cheol Oh, Kyungpyo Kang, and Seongho Kim. Estimation of lane-level travel times in vehicle-to-vehicle and vehicle-to-infrastructure–based traffic information system. *Transportation research record*, 2243(1):9–16, 2011.

[17] Xiao Han, Guojiang Shen, Xi Yang, and Xiangjie Kong. Congestion recognition for hybrid urban road systems via digraph convolutional network. *Transportation Research Part C: Emerging Technologies*, 121:102877, 2020.

[18] Tong Wang, Jiahua Cao, and Azhar Hussain. Adaptive traffic signal control for large-scale scenario with cooperative group-based multi-agent reinforcement learning. *Transportation research part C: emerging technologies*, 125:103046, 2021.

[19] Huansheng Song, Haoxiang Liang, Huaiyu Li, Zhe Dai, and Xu Yun. Vision-based vehicle detection and counting system using deep learning in highway scenes. *European Transport Research Review*, 11(1):1–16, 2019.

[20] Fukai Zhang, Ce Li, and Feng Yang. Vehicle detection in urban traffic surveillance images based on convolutional neural networks with feature concatenation. *Sensors*, 19(3):594, 2019.

[21] Anouar Ben Khalifa, Ihsen Alouani, Mohamed Ali Mahjoub, and Atika Rivenq. A novel multi-view pedestrian detection database for collaborative intelligent transportation systems. *Future generation computer systems*, 113:506–527, 2020.

[22] Sara El Hamdani, Nabil Benamar, and Mohamed Younis. Pedestrian support in intelligent transportation systems: challenges, solutions and open issues. *Transportation research part C: emerging technologies*, 121:102856, 2020.

[23] Marta Campos Ferreira, Paulo Dias Costa, Diogo Abrantes, Joana Hora, Soraia Felício, Miguel Coimbra, and Teresa Galvão Dias. Identifying the determinants and understanding their effect on the perception of safety, security, and comfort by pedestrians and cyclists: A systematic review. *Transportation research part F: traffic psychology and behaviour*, 91:136–163, 2022.

[24] Xiaoyan Yu and Marin Marinov. A study on recent developments and issues with obstacle detection systems for automated vehicles. *Sustainability*, 12(8):3281, 2020.

[25] Bin Tian, Brendan Tran Morris, Ming Tang, Yuqiang Liu, Yanjie Yao, Chao Gou, Dayong Shen, and Shaohu Tang. Hierarchical and networked vehicle surveillance in its: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 16(2):557–580, 2015.

[26] Michael Bommes, Adrian Fazekas, Tobias Volkenhoff, and Markus Oeser. Video based intelligent transportation systems–state of the art and future development. *Transportation Research Procedia*, 14:4495–4504, 2016.

[27] Zi Yang and Lilian SC Pun-Cheng. Vehicle detection in intelligent transportation systems and its applications under varying environments: A review. *Image and Vision Computing*, 69:143–154, 2018.

[28] Sokèmi René Emmanuel Datondji, Yohan Dupuis, Peggy Subirats, and Pascal Vasseur. A survey of vision-based traffic monitoring of road intersections. *IEEE Transactions on Intelligent Transportation Systems*, 17(10):2681–2698, 2016.

[29] Dieter Koller, Kostas Daniilidis, and Hans-Hellmut Nagel. Model-based object tracking in monocular image sequences of road traffic scenes. *International Journal of Computer 11263on*, 10:257–281, 1993.

[30] Luis Unzueta, Marcos Nieto, Andoni Cortés, Javier Barandiaran, Oihana Otaegui, and Pedro Sánchez. Adaptive multicue background subtraction for robust vehicle counting and classification. *IEEE Transactions on Intelligent Transportation Systems*, 13(2):527–540, 2011.

[31] Brendan Tran Morris and Mohan Manubhai Trivedi. Learning, modeling, and classification of vehicle track patterns from live video. *IEEE transactions on intelligent transportation systems*, 9(3):425–437, 2008.

[32] Xingchen Zhang, Yuxiang Feng, Panagiotis Angeloudis, and Yiannis Demiris. Monocular visual traffic surveillance: A review. *IEEE Transactions on Intelligent Transportation Systems*, 23(9):14148–14165, 2022.

[33] Lunlin Fei and Bing Han. Multi-object multi-camera tracking based on deep learning for intelligent transportation: a review. *Sensors*, 23(8):3852, 2023.

[34] Byeonghyeop Yu, Yongjin Lee, and Keemin Sohn. Forecasting road traffic speeds by considering area-wide spatio-temporal dependencies based on a graph convolutional neural network (gcn). *Transportation research part C: emerging technologies*, 114:189–204, 2020.

[35] Chenhan Zhang, JQ James, and Yi Liu. Spatial-temporal graph attention networks: A deep learning approach for traffic forecasting. *IEEE Access*, 7:166246–166256, 2019.

[36] Loan NN Do, Hai L Vu, Bao Q Vo, Zhiyuan Liu, and Dinh Phung. An effective spatial-temporal attention based neural network for traffic flow prediction. *Transportation research part C: emerging technologies*, 108:12–28, 2019.

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[38] Feihu Huang, Peiyu Yi, Jince Wang, Mengshi Li, Jian Peng, and Xi Xiong. A dynamical spatial-temporal graph neural network for traffic demand prediction. *Information Sciences*, 594:286–304, 2022.

[39] Kunpeng Zhang, Xiaoliang Feng, Lan Wu, and Zhengbing He. Trajectory prediction for autonomous driving using spatial-temporal graph attention transformer. *IEEE Transactions on Intelligent Transportation Systems*, 23(11):22343–22353, 2022.

[40] Shai Shalev-Shwartz. Online Learning and Online Convex Optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2011.

[41] Aleksandrs Slivkins. Introduction to Multi-Armed Bandits. *Foundations and Trends® in Machine Learning*, 12(1-2):1–286, 2019.

[42] Yoav Freund and Robert E. Schapire. Adaptive Game Playing Using Multiplicative Weights. *Games and Economic Behavior*, 29(1-2):79–103, 1999.

[43] Peter Auer, Nicol Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The Nonstochastic Multiarmed Bandit Problem. *SIAM Journal on Computing*, 32(1):48–77, 2002.

[44] Alexander Rakhlin and Karthik Sridharan. Online Learning with Predictable Sequences. 30:993–1019, 2013.

[45] Ali Jadbabaie, Alexander Rakhlin, Shahin Shahrampour, and Karthik Sridharan. Online Optimization : Competing with Dynamic Comparators. In Guy Lebanon and S. V. N. Vishwanathan, editors, *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, volume 38 of *Proceedings of Machine Learning Research*, pages 398–406, San Diego, California, USA, 09–12 May 2015. PMLR.

[46] Tao Li, Yuhan Zhao, and Quanyan Zhu. The role of information structures in game-theoretic multi-agent learning. *Annual Reviews in Control*, 53:296–314, 2022.

[47] Nicolo Cesa-Bianchi and Gabor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

[48] Sergiu Hart and Andreu Mas-Colell. A Simple Adaptive Procedure Leading to Correlated Equilibrium. *Econometrica*, 68(5):1127–1150, 2000. regret matching.

[49] Vianney Perchet. Approachability, regret and calibration: Implications and equivalences. *Journal of Dynamics & Games*, 1(2):181–254, 2014.

[50] Wanshi Hong, Gang Tao, Hong Wang, and Chieh Wang. Traffic Signal Control With Adaptive Online-Learning Scheme Using Multiple-Model Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 34(10):7838–7850, 2023.

[51] Zhengang Guo, Yingfeng Zhang, Jingxiang Lv, Yang Liu, and Ying Liu. An Online Learning Collaborative Method for Traffic Forecasting and Routing Optimization. *IEEE Transactions on Intelligent Transportation Systems*, 22(10):6634–6645, 2021.

[52] Fan Zuo, Jingxing Wang, Jingqin Gao, Kaan Ozbay, Xuegang Jeff Ban, Yubin Shen, Hong Yang, and Shri Iyer. An interactive data visualization and analytics tool to evaluate mobility and sociability trends during covid-19. *arXiv preprint arXiv:2006.14882*, 2020.

[53] Fan Zuo, Jingqin Gao, Abdullah Kurkcu, Hong Yang, Kaan Ozbay, and Qingyu Ma. Reference-free video-to-real distance approximation-based urban social distancing analytics amid covid-19 pandemic. *Journal of Transport & Health*, 21:101032, 2021.

[54] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[55] Yunhao Du, Zhicheng Zhao, Yang Song, Yanyun Zhao, Fei Su, Tao Gong, and Hongying Meng. Strongsort: Make deepsort great again. *IEEE Transactions on Multimedia*, 2023.

[56] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE international conference on image processing (ICIP)*, pages 3464–3468. IEEE, 2016.

[57] Dimitri P Bertsekas et al. Dynamic programming and optimal control 3rd edition, volume ii. *Belmont, MA: Athena Scientific*, 1, 2011.

[58] F Zuo, K Ozbay, A Kurkcu, J Gao, Huajie Yang, and K Xie. Microscopic simulation based study of pedestrian safety applications at signalized urban crossings in a connected-automated vehicle environment and reinforcement learning based optimization of vehicle decisions. *Advances in transportation studies*, 2020.

[59] Tao Li and Quanyan Zhu. On Convergence Rate of Adaptive Multiscale Value Function Approximation for Reinforcement Learning. *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, 2019.

[60] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems 12*, pages 1057—1063. MIT press, 2000.

[61] Tao Li, Guanze Peng, Quanyan Zhu, and Tamer Baar. The Confluence of Networks, Games, and Learning a Game-Theoretic Framework for Multiagent Decision Making Over Networks. *IEEE Control Systems*, 42(4):35–67, 2022.

[62] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

[63] Chelsea Finn, Aravind Rajeswaran, Sham Kakade, and Sergey Levine. Online meta-learning. In *International Conference on Machine Learning*, pages 1920–1930. PMLR, 2019.

[64] Tao Li, Haozhe Lei, and Quanyan Zhu. Self-Adaptive Driving in Nonstationary Environments through Conjectural Online Lookahead Adaptation. *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 00:7205–7211, 2023.

[65] Yunian Pan, Tao Li, Henger Li, Tianyi Xu, Zizhan Zheng, and Quanyan Zhu. A first order meta stackelberg method for robust federated learning. *The 2nd Workshop on New Frontiers In Adversarial Machine Learning at 40th International Conference on Machine Learning*, 2023.

[66] Tao Li, Guanze Peng, and Quanyan Zhu. Blackwell Online Learning for Markov Decision Processes. *2021 55th Annual Conference on Information Sciences and Systems (CISS)*, 00:1–6, 2021.

[67] Herbert Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535, 1952.

[68] Kunpeng Zhang, Feng Zhou, Lan Wu, Na Xie, and Zhengbing He. Semantic understanding and prompt engineering for large-scale traffic data imputation. *Information Fusion*, 102:102038, 2024.

[69] Shengnan Guo, Youfang Lin, Huaiyu Wan, Xiucheng Li, and Gao Cong. Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 34(11):5415–5428, 2021.

[70] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[71] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. Microscopic traffic simulation using sumo. In *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018.

[72] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[73] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017.