A probabilistic approach for queue length estimation using license plate recognition data Considering overtaking in multi-lane scenarios

Luo, Lyuzhou; Wu, Hao; Liu, Jiahao; Tang, Keshuang; Tan, Chaopeng

DOI
10.1016/j.trc.2025.105029

Publication date
2025

Document Version
Final published version

Published in
Transportation Research Part C: Emerging Technologies

Transportation Research Part C 173 (2025) 105029

Contents lists available at ScienceDirect

Transportation Research Part C

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

A probabilistic approach for queue length estimation using license plate recognition data: Considering overtaking in multi-lane scenarios

Lyuzhou Luo [a], Hao Wu [b], Jiahao Liu [a], Keshuang Tang [a,*], Chaopeng Tan [c]

[a] Key Laboratory of Road and Traffic Engineering of the Ministry of Education, College of Transportation, Tongji University, Cao'an Road 4800, Shanghai 201804, China

[b] Department of Electrical and Electronic Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, China

[c] Department of Transport and Planning, Delft University of Technology, Gebouw 23, Stevinweg 1, 2628CN, Delft, Netherlands

ARTICLE INFO

Keywords:
License plate recognition data
Queue length
Dynamic programming
Markov Chain Monte Carlo
Exact cover

ABSTRACT

Multi-section license plate recognition (LPR) data has emerged as a valuable source for lane-based queue length estimation, providing both input–output information and sampled travel times. However, existing studies often rely on restrictive assumptions such as the first-in-first-out (FIFO) rule and uniform arrival processes, which fail to capture the complexity of multi-lane scenarios, particularly regarding overtaking behaviors and traffic flow variations. To address this issue, we propose a probabilistic approach to derive the stochastic queue length by constructing a conditional probability model of *no-delay arrival time* (NAT), i.e., the arrival time of vehicles without experiencing any delay, based on multi-section LPR data. First, the NAT conditions for all vehicles are established based on upstream and downstream vehicle departure times and sequences. To reduce the computational dimensionality and complexity, a dynamic programming (DP)-based algorithm is developed for vehicle group partitioning based on potential interactions between vehicles. Then, the conditional probability of NATs of each vehicle group is derived and a Markov Chain Monte Carlo (MCMC) sampling method is employed for calculation. Subsequently, the stochastic queue profile and maximum queue length for each cycle can be derived based on the NATs of vehicles. Eventually, we extend our approach to multi-lane scenarios, where the problem can be converted to a weighted general exact coverage problem and solved by a backtracking algorithm with heuristics. Empirical and simulation experiments demonstrate that our approach outperforms the baseline method, demonstrating significant improvements in accuracy and robustness across various traffic conditions, including different V/C ratios, matching rates, miss detection rates, and FIFO violation rates. The estimated queue profiles demonstrate practical value for offset optimization in traffic signal control, achieving a 6.63% delay reduction compared to the conventional method.

1. Introduction

Queue length is a crucial metric reflecting the supply–demand relationship and coordination quality at signalized intersections (Yao et al., 2020; Noaeen et al., 2021). Accurate estimations of lane-based queue lengths could also significantly improve

* Corresponding author.
E-mail addresses: jexxllz@tongji.edu.cn (L. Luo), hao96.wu@polyu.edu.hk (H. Wu), liujiahao@tongji.edu.cn (J. Liu), tang@tongji.edu.cn (K. Tang), tantantan951122@gmail.com (C. Tan).

https://doi.org/10.1016/j.trc.2025.105029
Received 23 July 2024; Received in revised form 15 December 2024; Accepted 27 January 2025
Available online 22 February 2025

signal timings and active queue management at signalized intersections (Ma et al., 2021; Yin et al., 2021; Tan et al., 2024a).

Early research on queue length estimation has depended on traffic detection data such as volume, occupancy, and speed collected by fixed-location detectors (Sharma et al., 2007; Vigos et al., 2008; Skabardonis and Geroliminis, 2008; Liu et al., 2009; Wu et al., 2010). But in reality, these methods often face limitations related to aggregation intervals, malfunctions, and high maintenance costs (Ban et al., 2011). With the development of new data sources like connected vehicles (CV), a considerable amount of real-time trajectory data is becoming available for traffic management at urban road networks (Tan and Yang, 2024). Compared with traditional fixed-location detectors, the connected vehicles can provide continuous position and motion information without any additional expense, thus were used for queue length estimation by deterministic (Cheng et al., 2012; Ramezani and Geroliminis, 2015; Li et al., 2017; Tan et al., 2022b) or stochastic (Comert and Cetin, 2011; Zhang et al., 2020; Tan et al., 2021) approaches. However, since CV trajectory data are sampled observations, these methods typically suffer from low penetration rates or infrequent data uploads.

Vehicle license plate recognition (LPR) data are another type of emerging data source on urban roads. LPR systems are installed near the stop line of signalized intersections and can collect real-time departure information on individual vehicles, including license plate numbers, vehicle types, lane, and appearance times before the stop line. Compared to aggregated data collected from loop detectors and sparsely sampled data from probe vehicles, the LPR data offers high quality, wide coverage, and full-sample detection. Furthermore, by matching license plates collected by LPR systems installed at adjacent intersections, individual vehicles' paths and travel times can be obtained. This feature has been exploited at the network level for OD flow estimation (Rao et al., 2018; Mo et al., 2020; Tang et al., 2021) and vehicle path reconstruction (Yang and Sun, 2015; Cao et al., 2024). While vehicle entry and exit times can be directly obtained at the intersection or link level, vehicle behaviors within the link remain unknown. Thus, recent research has primarily focused on estimating traffic parameters not directly retrievable from LPR data, such as queue length (Wu et al., 2019; Tang et al., 2022; Tan et al., 2022a; Zhan et al., 2020, 2015; Ma et al., 2018; Luo et al., 2019), traffic demand (Ma et al., 2017; An et al., 2021) and speed profile (Mo et al., 2017).

Regarding queue length estimation using LPR data, existing methods can usually be separated into two broad categories: single-section methods and multi-section methods. Single-section methods solely rely on LPR data collected from target lanes and extract patterns from the vehicle discharging sequences for queue length estimation, e.g., critical point analysis (CPA) method (Wu et al., 2019), E-Divisive with Medians (EDM) method (Tang et al., 2022), Gaussian mixture model-based method (Tan et al., 2022a), and Gaussian process model-based method (Zhan et al., 2020). Despite the minimal deployment requirements for LPR systems, the performance of single-section methods drops dramatically when traffic conditions become near-saturated or over-saturated. This limitation arises because single-section LPR systems can only capture link departure information, not arrival information, making these methods unsuitable for scenarios involving residual queues.

In contrast, multi-section methods employ LPR data from both the target lane and its upstream intersection. Zhan et al. (2015) modeled the arrival flow rate by a piecewise linear function and developed a Gaussian process-based interpolation method to reconstruct each lane's equivalent cumulative arrival–departure curve. The arrival times for unmatched vehicles were obtained by presuming the validity of first-in-first-out (FIFO), and then the queue lengths were estimated in a car-following-based simulation scheme. Assuming uniform arrival rates, Ma et al. (2018) introduced two shockwave theory-based models focusing on the leading vehicle's queued position and the cycle maximum queue length and a recursive formula for maximum queue lengths across cycles, respectively. By combining these models, the potential relationship between maximum delay time and maximum queue length in each cycle was achieved. Without the consideration of overtaking, Luo et al. (2019) focused on the intrinsic connections between the travel time of individual vehicles and the queue composition in each cycle, then the queue length in the immediate past cycle was formulated as a prediction problem using regression analysis. Wu et al. (2024) proposed a stochastic queue profile estimation method, considering the scenario where all vehicles entering and exiting the road sections are detected by LPR systems. The arrival and departure curves were obtained using the input–output model, and the pseudo-departure curves were derived by considering the distribution of free-flow travel time. Then the stochastic queue profile was reconstructed by analyzing these three curves.

Current research using multi-section LPR data generally treats multi-lane scenarios by estimating queue lengths for each lane independently, without integrating data across lanes to achieve a comprehensive multi-lane estimation. This approach disregards information provided by upstream unmatched vehicles that may depart from other lanes downstream, thereby overlooking the interactions between lanes that are crucial for accurate queue length estimation. Moreover, these studies typically involve two strict assumptions that influence the estimation of queue lengths. The first assumption is the FIFO rule (Zhan et al., 2015; Luo et al., 2019), which implies that vehicles traveling towards a specific lane do not overtake each other, allowing for the direct extraction of arrival sequences from departure sequences. This assumption often fails in multi-lane scenarios due to frequent overtaking, leading to significant variances in predictions. The second assumption is a specific type of arrival process, either uniform (Ma et al., 2018) or piecewise linear (Zhan et al., 2015), which do not adapt well to the real and dynamic nature of traffic flows. Notably, Wu et al. (2024) attempts to relax the FIFO assumption by considering overtaking in its model for queue length estimation based on LPR data, which involves enumerating various overtaking cases to create a pseudo-departure curve. However, this method requires fully sampled LPR data without any recognition errors and fails to account for hidden overtaking behaviors that are captured in the matched license plates (between the upstream intersection and the target lane).

In response to the identified limitations of current methods in queue length estimation using multi-section LPR data, we propose a novel probabilistic approach that incorporates overtaking behaviors and utilizes data from multiple lanes for enhanced performance. The proposed approach begins with the concept of a vehicle's *no-delay arrival time* (NAT). We adopt this concept proposed by Ban et al. (2011), which indicates the projected time a vehicle would have arrived at the intersection if no delay had been experienced. By establishing conditions that vehicles must satisfy in NATs and applying a dynamic programming (DP)-based partitioning algorithm,

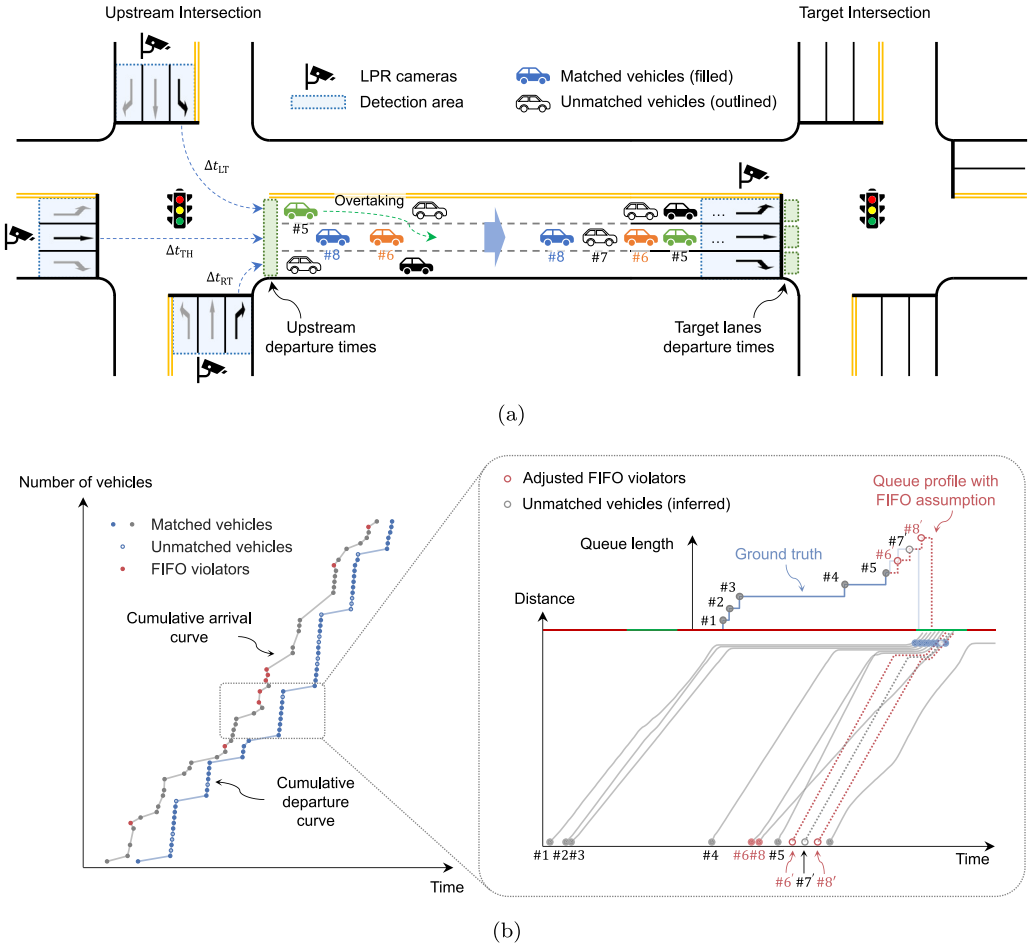

Fig. 1. Problem statement: (a) A typical scenario for collecting LPR data at two adjacent intersections, illustrating overtaking behaviors in multiple lanes; (b) Cumulative arrival and departure curves of the through lane and queue profile estimation errors due to the FIFO assumption.

we efficiently categorize vehicles into constrained and unconstrained groups. Then, a Markov Chain Monte Carlo (MCMC) sampling algorithm is used to estimate the arrival distribution of each vehicle based on the prior running time distribution. After that, the stochastic queue profile and maximum queue length for each cycle can be obtained by accumulating arrival distributions of vehicles. Additionally, we extend this approach from the single-lane estimation to the multi-lane estimation by identifying an optimal matching between unmatched vehicles with a heuristic algorithm, which can reduce the uncertainty in queue length estimation and achieve better estimation results.

The contributions of this paper are summarized as follows:

1. We consider overtaking and its potential effects on other vehicles in queue length estimation based on LPR data, thereby relaxing two commonly used assumptions in existing studies: the FIFO rule and a specific arrival process. This relaxation allows our method to reflect real-world conditions better and apply to complex traffic scenarios.
2. Unlike existing studies, our approach can be extended to a comprehensive multi-lane estimation, rather than treating each lane separately. This allows us to fully exploit LPR data, including those unmatched vehicles that cannot be handled by existing studies.
3. Our proposed probabilistic approach not only enables high-resolution queue profile estimation but also provides confidence intervals. Evaluation results show that our proposed approach performs well in both empirical and simulation case studies, achieving higher accuracy than the baseline method (Zhan et al., 2015) in various data scenarios for both maximum queue length and queue profile estimation.

2. Terminologies and assumptions

In this paper, we define the terminologies corresponding to the queue length as follows:

- *Queue length*: The number of vehicles required to span the distance between the last queued vehicle and the stop line. It remains unchanged until the last queued vehicle starts to depart, even when the queue begins to dissipate.
- *Queue profile*: The queue lengths recorded at each time step, representing the profile of the queue formation wave. While other studies examine both queue formation and dissipation waves, our research focuses primarily on the queue formation process, as this is not directly accessible in LPR data.
- *Maximum queue length*: The maximum queue length observed during a given cycle, from the start of the red light until the queue length begins to decrease. Note that, in cases of high saturation, the queue length may not decrease until the next cycle, but it is still considered the maximum queue length for the current cycle.

Other terminologies used in the methodology are summarized as follows:

- *Running time*: The total time required to traverse the link, excluding the stopped delay (Berry et al., 1951).
- *No-delay arrival time* (NAT): The projected time a vehicle would have arrived at the intersection if no delay had been experienced. The values for each vehicle k can be denoted as t_k^{a}. Unless stated otherwise, ‘‘arrival distribution’’ in the paper refers to the distribution of NATs, and ‘‘NAT conditions’’ refers to conditions that need to be satisfied for each vehicle’s NAT within each group, based on actual observations.
- *Matched vehicle*: Vehicles captured by both upstream and the target lane and passed the license plate matching process.
- *Unmatched vehicle*: Vehicles filtered out in the license plate matching process. Two potential reasons account for this: (1) the LPR camera fails to recognize or incorrectly recognizes the license plate number at either the upstream or the target lane; (2) the vehicles enter from unmonitored lanes, such as the right-turn lane.
- *Constrained group*: For vehicles in a constrained group, both the first and the last vehicles must be matched, then the NATs for other vehicles within the group will be constrained between the two vehicles, including any unmatched vehicles.
- *Unconstrained group*: Unconstrained groups are separated by constrained groups, and vehicles within are all unmatched. Without any upstream information, the NATs of vehicles in an unconstrained group can be determined after calculating the arrival distribution of adjacent constrained groups.
- *Inter-group departure gap*: The minimum time difference between when vehicles depart from the upstream in adjacent constrained groups. An adequate departure gap is necessary to maintain low interaction between adjacent constrained groups.

The main assumptions used in the methodology are defined as follows:

1. The prior running time is assumed to follow a specific distribution during the study period.
2. Miss detections of vehicles are not considered, as the primary source of error in current LPR systems lies in recognition errors (Zhan et al., 2015), leading to decreased matching rates between upstream and target lane vehicle detections, as further discussed in Sections 5.2.2 and 5.3.2.
3. Overtaking within the link is permitted, except in the queuing area, where lane changing is not allowed.

3. Problem statement

As illustrated in Fig. 1(a), a typical scenario for collecting LPR data at two adjacent intersections is depicted, where matched vehicles are represented by solid-filled icons and unmatched vehicles are represented by outlined icons. LPR cameras autonomously record vehicles passing the detection area, which spans 0–20 m from the stop line and typically covers 1–3 vehicles. By combining signal timing information and travel time within the intersection, vehicles’ detected times can be converted to departure times. A key analytical challenge arises from the discrepancy between vehicles’ departure sequence at the upstream intersection and their arrival sequence at a specific target lane. Fig. 1(a) illustrates this phenomenon, where vehicle 5 arrives earlier at the target through lane despite departing later than vehicles 6 and 8 from the upstream intersection. The data also exhibits matching failures, as exemplified by vehicle 7, which appears at the target lane but cannot be matched with any upstream departing vehicle due to errors in license plate recognition.

This paper focuses on analyzing cases where vehicles heading towards the same lane arrive in a different order than they departed from upstream. In traffic engineering, such behavior has traditionally been termed ‘‘overtaking’’, referring to a maneuver where a vehicle passes another vehicle traveling in the same direction by temporarily moving into an adjacent lane or part of the roadway. However, our analysis requires an expanded definition of overtaking to include instances where a vehicle initially in a different lane accelerates past another vehicle before changing lanes, as exemplified by vehicle 5 in Fig. 1(a). From the perspective of LPR data analysis, these two types of behaviors manifest similarly and can therefore be unified in this paper. Specifically, we define overtaking based on detection times t_k^{d} and upstream departure times t_k^{u}: for vehicle i, if vehicle j is detected earlier ($t_i^{\mathrm{d}} > t_j^{\mathrm{d}}$) but departs upstream later ($t_i^{\mathrm{u}} < t_j^{\mathrm{u}}$), we consider that vehicle j has overtaken vehicle i. In this case, we define the overtaken vehicle i as having violated the FIFO principle.

To systematically analyze these detection patterns and their implications, we can represent the data using cumulative arrival and departure curves, as shown in Fig. 1(b). Assuming no miss detection, the LPR camera of a specific lane (e.g., the through lane)

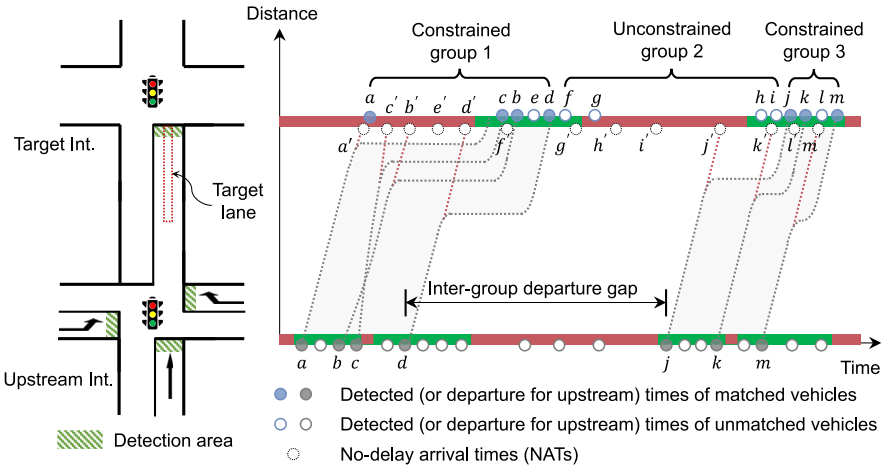

Fig. 2. Illustration of the license plate matching result and no-delay arrival times (NATs).

can detect complete departures, as shown by the blue cumulative departure curve. In contrast, only partial arrivals can be observed, depending on the number of vehicles matched from the upstream intersection, as indicated by the gray cumulative arrival curve. The solid circles represent matched vehicles, while the hollow circles represent unmatched vehicles like vehicle 7. The red solid circles denote FIFO violators resulting from overtaking, detailed in the partial vehicle trajectories and lane-based queue lengths in Fig. 1(b). This limitation makes it impossible to directly apply the input–output model to describe the traffic state of the link. Previous studies (Zhan et al., 2015; An et al., 2021) have addressed this issue by removing all FIFO violators. This would require adjusting their departure times to reasonable values based on certain rules (e.g., uniform arrival), and inferring the departure times of all unmatched vehicles. However, this adjustment process can introduce errors, specifically from incorrect inferred upstream departure times and neglecting interactions between vehicle trajectories.

Instead of reconstructing the cumulative arrival and departure curves, which may incur issues such as FIFO violations, we focus on the vehicle's *no-delay arrival time* (NAT), i.e., the arrival time without experiencing any delay. In Fig. 2, the NATs are denoted in the dashed hollow circle. Given the specific NATs of all vehicles, the queue length can be easily derived by accumulating vehicles that queued. Thereby, the problem of queue length estimation is transformed into the estimation of NATs of vehicles. A straightforward way for NAT estimation is to assume a fixed running time for all vehicles, which ignores the vehicle speed variance and overtaking behaviors:

$$t_k^{\mathrm{a}} = t_k^{\mathrm{u}} + T_0, \quad T_0 \text{ is a constant} \tag{1}$$

where t_k^{a} is the NAT of vehicle k, t_k^{u} is the upstream departure time of vehicle k, and T_0 is a fixed running time. Obviously, this approach introduces significant errors.

Wu et al. (2024) considered the distribution of running time to better characterize the uncertain vehicle behaviors on the link and traversed possible overtaking cases. It only used the information provided by upstream LPR data, which can be summarized as

$$t_k^{\mathrm{a}} = t_k^{\mathrm{u}} + T_k, \quad T_k \sim D(\cdot|\theta_{\mathrm{up}}) \tag{2}$$

where T_k is the running time of vehicle k, $D(\cdot|\theta_{\mathrm{up}})$ represents the distribution of T, and θ_{up} is the upstream information used to generate possible overtaking cases.

However, Eq. (2) ignores the actual overtaking information provided by the target intersection's LPR data. In Fig. 2, vehicles that departed during the first cycle from the target lane followed the order of a, c, b, e, d, among which vehicle c arrived before b due to overtaking. As lane-changing was not permitted in the queuing area, the sequence of NATs mirrors that of departure times. Subsequently, we can utilize the prior running time distribution and the license plate matching result (using information from both the upstream and the target intersection) to establish the NAT conditions, thereby indirectly considering the impact of overtaking. This process can be expressed as

$$t_k^{\mathrm{a}} = t_k^{\mathrm{u}} + T_k, \quad T_k \sim D(\cdot|\theta_{\mathrm{up}}, \theta_{\mathrm{target}}) \tag{3}$$

where θ_{target} is the information from the target intersection.

Since we only utilize LPR data from a single lane of the target intersection in this scenario, we refer to this queue length estimation approach as *single-lane estimation*. This approach is highly suitable for online applications due to its modest data requirements. It only

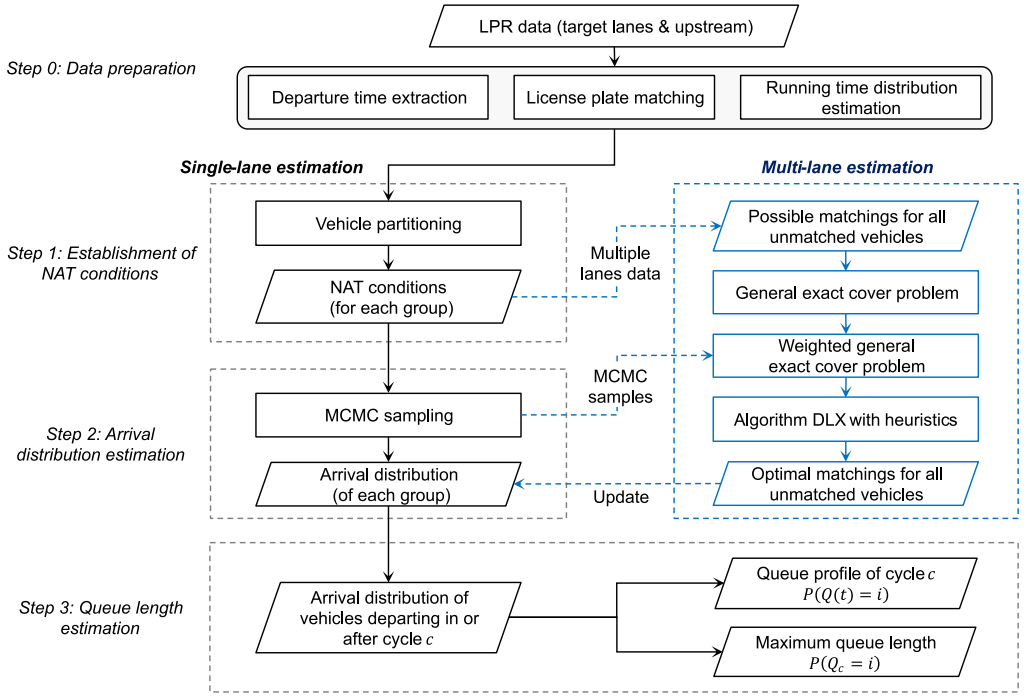

Fig. 3. The general framework of the methodology.

requires LPR data from the target lane, without needing LPR coverage for all relevant lanes at the upstream intersection. Vehicles entering from unmonitored lanes, such as the right-turn lane, are filtered out during the license plate matching process and classified as unmatched vehicles. This treatment aligns with cases of recognition errors, as the methodology focuses on the binary state of vehicle matching rather than the specific causes of mismatches.

The single-lane estimation approach, however, does not utilize information from upstream unmatched vehicles that may eventually depart from any lane of the target intersection. To address this limitation, we propose *multi-lane estimation*, which builds upon the single-lane approach by determining optimal matches between unmatched vehicles from the upstream intersection and target lanes. This enhancement requires comprehensive input–output information for the link, including LPR data from all lanes at both the target approach and relevant upstream intersection lanes. While more demanding in terms of data requirements and computational complexity, multi-lane estimation is primarily suited for offline analysis and offers significant improvements in queue length estimation accuracy, making it particularly valuable for detailed post-hoc traffic analysis and research purposes.

4. Methodology

4.1. Framework

The general framework of the methodology is shown in Fig. 3 and explained below.

In the single-lane estimation, we first process the LPR data to obtain data used in the next steps, including departure times, license plate matching results, and the running time distribution. Second, we can list the conditions all vehicles passing through the target lane must satisfy in NAT. Due to the large number of NAT conditions during the study period, we propose a DP-based vehicle partitioning algorithm to divide all vehicles passing through the target lane into constrained and unconstrained groups. The NAT conditions are then divided accordingly. Third, uniform samples of the region represented by the conditions can be obtained for each set of NAT conditions using an MCMC sampling algorithm. The conditional probability can be used to calculate the arrival distribution of all vehicles by integral calculation. Finally, the queue profile and maximum queue length for each cycle can be obtained by analyzing the arrival distribution of vehicles departing in or after each cycle in sequence.

In the multi-lane estimation, we extend the approach by using NAT conditions of multiple lanes and then update the arrival distribution. To achieve this, we first obtain possible matchings for each unmatched vehicle with upstream unmatched vehicles, and the problem of finding all possible matchings can be formulated as a general exact cover problem. Utilizing the uniform samples from

the MCMC sampling, the problem can be reformulated to a weighted version and solved by the Algorithm DLX (an implementation of Knuth's Algorithm X (Knuth, 2000) via dancing links) with heuristics. Finally, most previously unmatched vehicles will have been matched with upstream vehicles, enabling a more accurate computation of the arrival distribution and an improved estimation of queue profile and maximum queue length.

4.2. Data preparation

4.2.1. Departure time extraction

The LPR system provides only the detected times of vehicles. For vehicles detected during the red phase, their detected times do not represent their actual departure times. The detected time can be adjusted to determine the departure time using the existing interpolation method (Mo et al., 2017). Additionally, it is essential to account for the travel time within the upstream intersection, defined as the time elapsed between a vehicle leaving the stop line and entering the link. This travel time varies according to movement type, denoted as Δt_{LT}, Δt_{TH}, and Δt_{RT} for left-turn, through, and right-turn movements, respectively, as depicted in Fig. 1(a). For the purposes of this study, it is assumed that the travel time within the intersection remains constant throughout the study period.

4.2.2. License plate matching

The license plate matching process consists of three steps. First, an outer join is executed on the LPR data from both the target lane and the upstream intersection, using the license plate number as a common key. Next, the difference between the downstream detected time and the upstream departure time, namely the travel time, is computed. Finally, records with travel times that are either less than 0 or exceed a certain threshold (e.g., 5 min) are filtered out.

4.2.3. Running time distribution estimation

To model the running time distribution, we employ the approach proposed by Li et al. (2023), which involves fitting the travel time data using a mixed model with a log-normal distribution. After removing the noise component, the component with the smallest mean value is identified as the representative distribution of running time. Given that running times are inherently bounded, we truncate the log-normal distribution using the maximum and minimum values of travel time within this component:

$$g(t)=\begin{cases}\dfrac{\frac{1}{t\sigma_T}\phi\left(\frac{\ln t-\mu_T}{\sigma_T}\right)}{\Phi\left(\frac{\ln T_{\max}-\mu_T}{\sigma_T}\right)-\Phi\left(\frac{\ln T_{\min}-\mu_T}{\sigma_T}\right)}, & T_{\min}\le t\le T_{\max}\\ 0, & \text{otherwise}\end{cases} \tag{4}$$

Here, $g(\cdot)$ represents the probability density function (PDF) of the running time T. The parameters μ_T and σ_T denote the location and scale parameters of the log-normal distribution, respectively. The bounds $T_{\min}$ and $T_{\max}$ define the truncation interval, which corresponds to the minimum and maximum possible running times. Finally, $\phi(\cdot)$ and $\Phi(\cdot)$ represent the standard normal distribution's PDF and CDF (cumulative distribution function), respectively.

4.3. Single-lane estimation

4.3.1. Establishment of NAT conditions

Given the information derived from the LPR data, we can establish the following conditions for all vehicles departing from the target lane:

1. Departure sequence conditions

These conditions indicate that the sequence of the NATs must correspond to the sequence of departure times. If vehicle $k+1$ departs after vehicle k, then the NAT of vehicle $k+1$ should also be greater than that of vehicle k. Meanwhile, the headway condition should be satisfied:

$$t_{k+1}^{\mathrm{a}}-t_k^{\mathrm{a}}\ge\min(h,t_{k+1}^{\mathrm{d}}-t_k^{\mathrm{d}}),\quad \forall k,k+1\in\mathcal{K}\cup\mathcal{K}_{\mathrm{u}} \tag{5}$$

where $\mathcal{K}$ is the set of vehicles in constrained groups; $\mathcal{K}_{\mathrm{u}}$ is the set of vehicles in unconstrained groups; t_k^{a} is the NAT of vehicle k; t_k^{d} is the detected time of vehicle k; h is the predefined saturation headway; $t_{k+1}^{\mathrm{d}}-t_k^{\mathrm{d}}$ is the actual arrival headway.

2. Running time conditions

The running time of each vehicle follows a certain distribution and has a minimum and maximum value. Therefore, for all matched vehicles, we have

$$t_k^{\mathrm{u}}+T_k^{\min}\le t_k^{\mathrm{a}}\le t_k^{\mathrm{u}}+T_k^{\max},\quad \forall k\in\mathcal{M} \tag{6}$$

where $\mathcal{M}$ is the set of matched vehicles; t_k^{u} is the departure time from the upstream of vehicle k; $T_k^{\min}$ and $T_k^{\max}$ are the minimum and maximum running time for vehicles k, respectively.

3. Detected time conditions

For all vehicles, the NATs should not exceed the actual detected times:

$$t_k^{\mathrm{a}}\le t_k^{\mathrm{d}},\quad \forall k\in\mathcal{K}\cup\mathcal{K}_{\mathrm{u}} \tag{7}$$

After that, the arrival distribution for each vehicle is estimated through an MCMC sampling method in Section 4.3.2. The computational efficiency of this sampling process is largely determined by the dimensionality of the sampling space, which directly corresponds to the number of vehicles being analyzed. To improve computational efficiency, we partition vehicles into several groups before sampling. Our partitioning principle ensures that vehicles in adjacent constrained groups do not influence each other while maximizing the number of groups, thereby reducing the number of vehicles in each group and increasing sampling efficiency.

It is important to distinguish our vehicle partitioning approach from platoon identification in traffic engineering. While platoon identification focuses on detecting groups of vehicles traveling together along a roadway segment with similar speeds and small spacing — typically using CV data for arterial signal coordination (He et al., 2012; Tiaprasert et al., 2018) — our vehicle partitioning serves a different purpose. Our method specifically examines vehicles heading toward the same target lane, considering whether their departure times from the upstream intersection are sufficiently close to affect each other's arrival patterns. In contrast, platoon identification tracks vehicle grouping behavior during their journey, regardless of their eventual lane choices.

We propose a DP-based algorithm, as presented in Algorithm 1, to achieve this partitioning. Given that matched vehicles must be in constrained groups, we first select all matched vehicles, sorted by their arrival order, and use their upstream departure times as the first input, denoted lst with size n. Next, we set the minimum inter-group departure gap $\delta_{\min}$ to separate adjacent constrained groups, forming the second input. In the absence of overtaking behavior, the partitioning problem could be efficiently solved using a greedy approach by creating new partitions whenever the $\delta_{\min}$ condition is satisfied, achieving an $\mathcal{O}(n)$ time complexity. However, due to the occurrence of overtaking, lst may not necessarily increase monotonically, necessitating a more comprehensive search strategy with $\mathcal{O}(n^2)$ time complexity to guarantee optimality.

We first implement a preprocessing step that calculates prefix maximum and suffix minimum arrays in $\mathcal{O}(n)$ time. This optimization transforms the condition for non-influence into an $\mathcal{O}(1)$ operation: $suffix\_min[j+1] - prefix\_max[j] > \delta_{\min}$, where $suffix\_min$ and $prefix\_max$ are precomputed arrays storing the minimum and maximum upstream departure times for vehicles after index j and up to index j, respectively. The dynamic programming array dp stores the maximum achievable number of partitions up to each vehicle in the sequence, while the auxiliary array $break\_points$ maintains the indices in lst where optimal partitions begin. The state transition equation $dp[i] = \max(dp[j]+1, dp[i])$ is applied when the non-influence condition is satisfied, indicating the formation of a new partition. Furthermore, we enhance the algorithm's efficiency by maintaining the last valid j position that resulted in a successful partition, as this position represents a promising starting point for subsequent searches. This optimization leverages the observation that if a particular j yields an optimal partition for index i, it often serves as a good candidate for index $i+1$ as well.

Algorithm 1: Matched vehicle partitioning

Input: A list lst of upstream departure times for matched vehicles with size n, and a minimum inter-group departure gap $\delta_{\min}$

Output: A list of partitions of matched vehicles

```
Precompute suffix_min and prefix_max arrays;
Initialize a list dp of size n with all elements set to 1;
Initialize a list break_points of size n + 1 with all elements set to 0;
break_points[dp[0]] ← 1;
last_valid_j ← 0;
for i ← 1 to n − 1 do
    for j ← last_valid_j to i − 1 do
        if suffix_min[j + 1] − prefix_max[j] > δ_min then
            dp[i] = max(dp[j] + 1, dp[i]);
            if dp[j] + 1 > dp[i] then
                last_valid_j ← j;
    break_points[dp[i]] ← i + 1;
Initialize an empty list partitions;
for i ← 0 to dp[n − 1] − 1 do
    append lst[break_points[i] : break_points[i + 1]] to partitions;
return partitions
```

After obtaining the optimal partitioning through the DP-based algorithm, we proceed to classify all vehicles into constrained and unconstrained groups. This classification process follows a systematic approach: each partition of matched vehicles identified by Algorithm 1, together with any unmatched vehicles departing from the target lane between them, constitutes a constrained group. Conversely, unmatched vehicles that fall between these constrained groups form unconstrained groups. Fig. 2 illustrates this classification, where vehicles a, c, b, e, d along with vehicles j, k, l, m comprise two distinct constrained groups, while vehicles f, g, h, i form an unconstrained group between them. Subsequently, we can formulate the preceding conditions from Eqs. (5) to (7)

in matrix form. For each constrained group, these conditions are expressed as

$$\underbrace{\begin{bmatrix} 1 & -1 & 0 & \cdots & 0 \\ 0 & 1 & -1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & -1 \\ 1 & 0 & 0 & \cdots & 0 \\ -1 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & -1 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}}_{\text{Matrix } \boldsymbol{A}} \cdot \underbrace{\begin{bmatrix} t_1^{\mathrm{a}} \\ t_2^{\mathrm{a}} \\ t_3^{\mathrm{a}} \\ \vdots \\ t_n^{\mathrm{a}} \end{bmatrix}}_{\text{Vector } \boldsymbol{t}} \leq \underbrace{\begin{bmatrix} -\min(h, t_2^{\mathrm{d}} - t_1^{\mathrm{d}}) \\ -\min(h, t_3^{\mathrm{d}} - t_2^{\mathrm{d}}) \\ \vdots \\ -\min(h, t_n^{\mathrm{d}} - t_{n-1}^{\mathrm{d}}) \\ t_1^{\mathrm{u}} + T_1^{\max} \\ -t_1^{\mathrm{u}} - T_1^{\min} \\ \vdots \\ t_n^{\mathrm{u}} + T_n^{\max} \\ -t_n^{\mathrm{u}} - T_n^{\min} \\ t_1^{\mathrm{d}} \\ t_2^{\mathrm{d}} \\ t_3^{\mathrm{d}} \\ \vdots \\ t_n^{\mathrm{d}} \end{bmatrix}}_{\text{Vector } \boldsymbol{b}} \tag{8}$$

1. *Departure sequence conditions*
2. *Running time conditions*
3. *Detected time conditions*

where $\boldsymbol{A}$ represents an $m \times n$ matrix, with m denoting the number of conditions and $n = |K|$ representing the number of vehicles in the constrained group $K \in \mathcal{K}$. The vector $\boldsymbol{t}$ represents an n-dimensional NAT vector, while $\boldsymbol{b}$ is an m-dimensional constant vector. To enhance computational efficiency, we employ an established redundant constraints identification method (Telgen, 1983) to reduce the number of conditions while preserving the solution space.

The treatment of unconstrained groups requires special consideration, as the absence of running time conditions makes it impossible to determine their NAT ranges in isolation. Therefore, we extend our analysis to include the departure sequence conditions imposed by vehicles immediately preceding and following each unconstrained group, as exemplified by vehicles d and j in Fig. 2. These extended conditions can be formulated in a matrix form analogous to Eq. (8), where $\boldsymbol{A}$ becomes an $m \times (n+2)$ matrix to accommodate the additional boundary vehicles, $\boldsymbol{t}$ expands to an $(n+2)$-dimensional NAT vector, and $\boldsymbol{b}$ remains an m-dimensional constant vector.

4.3.2. Arrival distribution estimation

After establishing NAT conditions for each vehicle group, we first compute the arrival distributions of vehicles in each constrained group. Assuming we know the prior probability distributions of running times for matched vehicles, we can then derive the prior probability distribution of their NATs, given that their upstream departure times have been captured by the LPR camera. Due to the lack of upstream information for unmatched vehicles, we assume that their prior NATs follow a uniform distribution. If we consider the NATs of all vehicles as independent variables, their joint PDF can be calculated through multiplication. Under the conditions obtained in the previous step, the probability of each vehicle's NAT falling in $[t, t+1]$ can be calculated through the conditional probability:

$$\begin{aligned} P(t \leq t_k^{\mathrm{a}} \leq t+1 | \boldsymbol{A}\boldsymbol{t} \leq \boldsymbol{b}) &= \frac{P((\boldsymbol{A}^\top, \boldsymbol{A}_{k,t}^\top)^\top \boldsymbol{t} \leq (\boldsymbol{b}^\top, \boldsymbol{b}_{k,t}^\top)^\top)}{P(\boldsymbol{A}\boldsymbol{t} \leq \boldsymbol{b})} \\ &= \frac{\int_{\Omega_{k,t}} f(\boldsymbol{t}) d\boldsymbol{t}}{\int_{\Omega} f(\boldsymbol{t}) d\boldsymbol{t}}, \quad k \in K \end{aligned} \tag{9}$$

where $\boldsymbol{A}_{k,t}\boldsymbol{t} \leq \boldsymbol{b}_{k,t}$ indicates that vehicle k's NAT falls in $[t, t+1]$, which is the matrix representation of $t \leq t_k^{\mathrm{a}} \leq t+1$. $\Omega = \{\boldsymbol{t} \in \mathbb{R}^n | \boldsymbol{A}\boldsymbol{t} \leq \boldsymbol{b}\}$ and $\Omega_{k,t} = \{\boldsymbol{t} \in \mathbb{R}^n | (\boldsymbol{A}^\top, \boldsymbol{A}_{k,t}^\top)^\top \boldsymbol{t} \leq (\boldsymbol{b}^\top, \boldsymbol{b}_{k,t}^\top)^\top\}$ are polytopes composed of linear inequality constraints. $f(\boldsymbol{t}) = \prod_{k \in K} f_k(t_k^{\mathrm{a}})$ is the joint PDF of $\boldsymbol{t}$, and for matched and unmatched vehicles, we have

$$f_k(t_k^{\mathrm{a}}) = \begin{cases} g_k(t_k^{\mathrm{a}} - t_k^{\mathrm{u}}), & k \in K \cap \mathcal{M} \\ \text{constant}, & k \in K - \mathcal{M} \end{cases} \tag{10}$$

where $g_k(\cdot)$ represents the PDF of vehicle k's running time. For unmatched vehicles, their prior PDFs for NATs are the same for each second. However, this specific value does not impact subsequent computations and can thus be ignored when calculating joint PDF $f(\boldsymbol{t})$.

Since the dimension of the vector $\boldsymbol{t}$, $n = |K|$, is usually high, it is challenging to compute using deterministic integration algorithms. Hence, we employ the naive Monte Carlo integration to calculate the above probabilities. For integrals in the form $\int_{\Omega} f(\boldsymbol{t}) d\boldsymbol{t}$, we first calculate the volume of polytope Ω:

$$V = \int_{\Omega} \mathrm{d}\boldsymbol{t} \tag{11}$$

Next, we uniformly sample $\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_N \in \Omega$ within the polytope Ω. We chose an MCMC sampling algorithm called Vaidya walk (Chen et al., 2018) to achieve that. Vaidya walk is a random walk derived from interior point methods and based on the volumetric-logarithmic barrier introduced by Vaidya (1989). For a polytope in $\mathbb{R}^n$ defined by m constraints, the Vaidya walk mixes in $\mathcal{O}(m^{0.5}n^{1.5})$ steps, with an effective cost of $\mathcal{O}(m^{1.5}n^{3.5})$ per sample. This allows for dynamic adjustment of the number of iterations N_{iter} based on mixing time:

$$N_{\text{iter}} = \max\{N_{\text{iter}}^{\min}, \alpha m^{0.5} n^{1.5}\} \tag{12}$$

where $N_{\text{iter}}^{\min}$ is the minimum number of iterations to ensure estimation stability and α is a scaling factor. Since $m \geq n$ for closed polytopes in $\mathbb{R}^n$, the Vaidya walk is more efficient than other random walks, such as the Dikin walk (Kannan and Narayanan, 2012), which has an effective cost of $\mathcal{O}(m^2 n^3)$ per sample. The samples obtained are then used to compute the integral value, which approximates the average joint PDF values of the samples multiplied by the volume of the polytope:

$$\int_{\Omega} f(\boldsymbol{t})\,\mathrm{d}\boldsymbol{t} \approx V \frac{1}{N} \sum_{i=1}^{N} f(\boldsymbol{t}_i) \tag{13}$$

Since vehicles must arrive and can only arrive once, any point uniformly sampled in Ω will always fall into one of the $\Omega_{k,t}$. Let us denote the points that fall into $\Omega_{k,t}$ as $\boldsymbol{t}_{k,t,1}, \boldsymbol{t}_{k,t,2}, \ldots, \boldsymbol{t}_{k,t,N_{k,t}} \in \Omega_{k,t}$, with a count of $N_{k,t}$. Then, approximately, we have

$$\frac{V_{k,t}}{V} \approx \frac{N_{k,t}}{N} \tag{14}$$

where $V_{k,t}$ is the volume of $\Omega_{k,t}$. Therefore, the conditional probability in Eq. (9) can be expressed as the ratio of the sums of the joint PDF values of the sampling points, where the difficult-to-calculate volume can be canceled out in the ratio:

$$\frac{\int_{\Omega_{k,t}} f(\boldsymbol{t})\,\mathrm{d}\boldsymbol{t}}{\int_{\Omega} f(\boldsymbol{t})\,\mathrm{d}\boldsymbol{t}} \approx \frac{V_{k,t}\frac{1}{N_{k,t}}\sum_{i=1}^{N_{k,t}} f(\boldsymbol{t}_{k,t,i})}{V\frac{1}{N}\sum_{i=1}^{N} f(\boldsymbol{t}_i)} \approx \frac{\sum_{i=1}^{N_{k,t}} f(\boldsymbol{t}_{k,t,i})}{\sum_{i=1}^{N} f(\boldsymbol{t}_i)} \tag{15}$$

The procedure to calculate the arrival distribution is summarized in Algorithm 2.

Algorithm 2: Arrival distribution estimation

Input: Vehicle group K along with the corresponding NAT conditions, denoted as $\boldsymbol{At} \leq \boldsymbol{b}$
Output: Arrival distribution of each vehicle

```
Uniformly sample t_1, t_2, ..., t_N ∈ Ω = {t ∈ R^n | At ≤ b} and denote T = (t_1, t_2, ..., t_N);
Calculate Σ_{i=1}^{N} f(t_i);
foreach vehicle k ∈ K do
    Compute the minimum value t_k^{a,min} and the maximum value t_k^{a,max} of t_k^a based on At ≤ b, then round these values down
      and up respectively;
    for t ← t_k^{a,min} to t_k^{a,max} do
        Select N_{k,t} points from T which satisfy A_{k,t} t ≤ b_{k,t};
        Compute Σ_{i=1}^{N_{k,t}} f(t_{k,t,i}) to get the approximation of Eq. (9) using Eq. (15);
```

Following the calculation of the arrival distribution for adjacent constrained groups, each unconstrained group can be bounded. As outlined by Algorithm 1, we ensure minimal interference between vehicles in distinct constrained groups, and then their NATs are deemed conditionally independent. In Fig. 4, the vehicles f, g, h, i constitute an unconstrained group, and the arrival distributions of both the preceding and following vehicles, i.e., vehicle d and j, have already been obtained. The procedure to compute the arrival distributions of vehicles in an unconstrained group is similar to Algorithm 2, with the primary difference being the calculation of the joint PDF $f(\boldsymbol{t})$, as shown in Eq. (16). At this point, the arrival distributions of the previous and following vehicles are known, so the PDFs of their NATs can be treated as second-based piecewise functions, as demonstrated in the histogram in Fig. 4:

$$f(\boldsymbol{t}) = f_{\text{prev.}}(t^{\text{a}}_{\text{prev.}}) \cdot f_{\text{foll.}}(t^{\text{a}}_{\text{foll.}}) \tag{16}$$

where $f_{\text{prev.}}(t^{\text{a}}_{\text{prev.}})$ represents the arrival distribution of the previous vehicle, and $f_{\text{foll.}}(t^{\text{a}}_{\text{foll.}})$ signifies that of the following vehicle. Given that the remainder of the vehicles are all unmatched, their PDFs are disregarded.

4.3.3. Queue length estimation

After obtaining the arrival distribution of all vehicles departing from the target lane, we can further calculate the queue profile and the maximum queue length per cycle.

To obtain the queue profile, we need to calculate the distribution of queue length $Q(t)$ at each time step t based on the arrival distribution estimated in the previous section. First, we must determine the potential queuing time range for each vehicle within a cycle. Fig. 5 shows a typical cycle's vehicle trajectory, where cycle $c-1$ is over-saturated, resulting in a residual queue. For cycle c, the queuing time range for the ith vehicle (whose trajectory is shown in dark gray in the figure) can be calculated using the

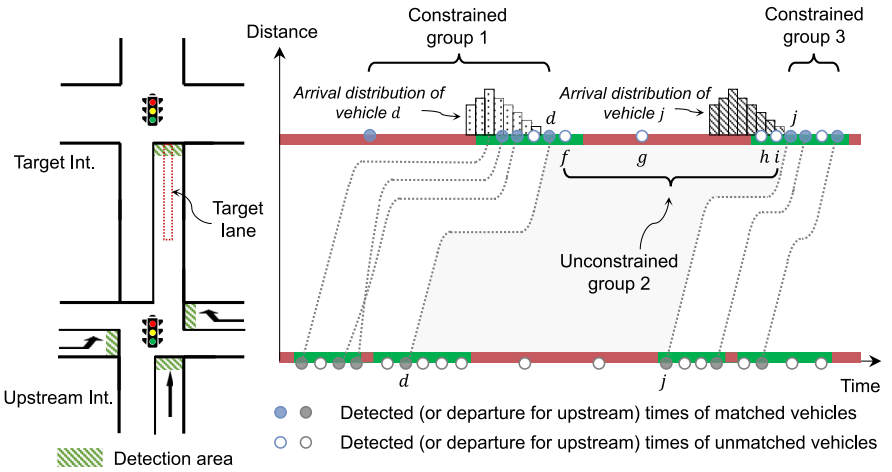

Fig. 4. Illustration of the arrival distribution estimation for an unconstrained group.

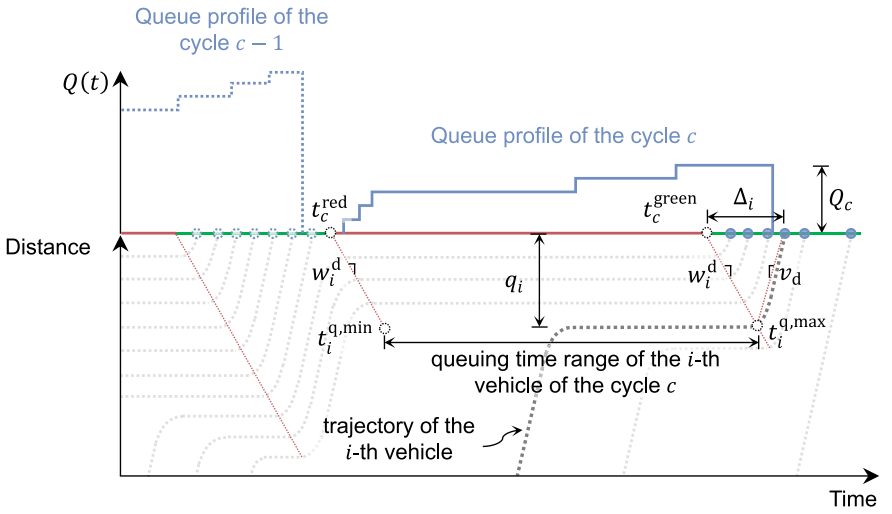

Fig. 5. Illustration of the queuing time range of each vehicle per cycle.

dissipation wave speed w_i^{d} before the ith vehicle departs:

$$t_i^{\text{q,min}} = t_c^{\text{red}} + \frac{q_i}{w_i^{\text{d}}} \tag{17}$$

$$t_i^{\text{q,max}} = t_c^{\text{green}} + \frac{q_i}{w_i^{\text{d}}} \tag{18}$$

where $t_i^{\text{q,min}}$ and $t_i^{\text{q,max}}$ represent the potential queuing time range in cycle c for the ith vehicle departing in or after cycle c. t_c^{red} and t_c^{green} are the start times of the red and green lights in cycle c, respectively. $q_i = \sum_{k=1}^{i} x_k$ is the queue length in meters if there are i vehicles in the queue, where x_k is the queuing space in meters of the kth vehicle. To calculate w_i^{d}, we assume that vehicles travel at a fixed discharging speed v_{d} before passing the stop line. According to the shockwave theory, we obtain:

$$w_i^{\text{d}} = \frac{v_{\text{d}} \cdot q_i}{v_{\text{d}} \cdot \Delta_i - q_i} \tag{19}$$

$$\Delta_i = \min\{h \cdot i, t_i^{\text{dd}} - t_c^{\text{green}}\} \tag{20}$$

where Δ_i is the minimum time difference between the start time of the green light and when the ith vehicle departs, h is the saturation headway and t_i^{dd} is the actual departure time of the ith vehicle. Notably, for vehicles appearing in multiple cycles, each cycle has a corresponding potential queuing time range.

For any time t, we can calculate the lower bound of the queue length at time t based on the arrival distribution of all vehicles whose queuing time range covers that moment. As shown in Fig. 6(a), the underlying concept is that if the ith vehicle departing in

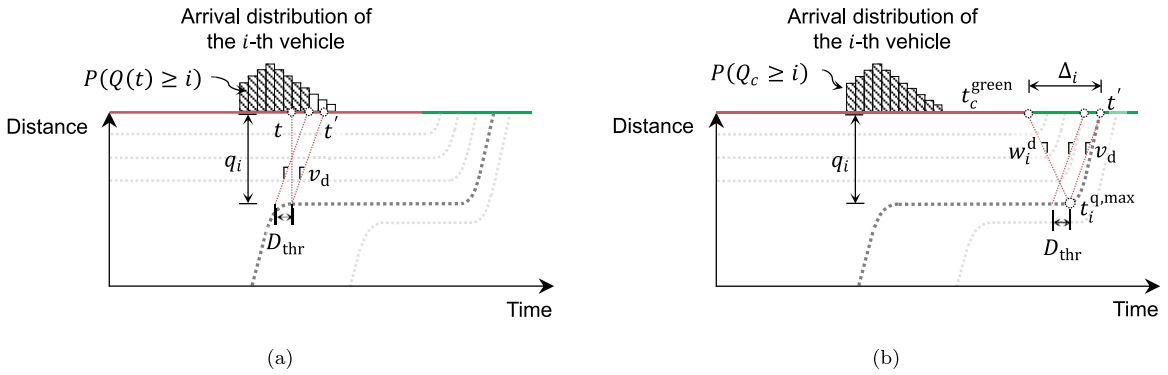

Fig. 6. Illustration of the queue length estimation process, where (a) depicts the probability distribution of queue length at time step t and (b) shows the probability distribution of the maximum queue length for cycle c.

or after cycle c arrives at its queuing position before time step t, it can be inferred that the queue length is at least i. However, the previously estimated arrival distribution is based on arrivals at the stop line. Given the queue length in meters q_i when there are i vehicles in the queue, we can calculate the corresponding arrival time at the stop line:

$$t' = t + \frac{q_i}{v_d} \tag{21}$$

where t is the current time step, and t' is the time when the ith vehicle is expected to arrive at the stop line.

To eliminate the impact of acceleration and deceleration delays, we set a delay threshold D_{thr} to avoid treating minor fluctuations in arrival times as stopped delays. A vehicle is only considered to be queuing if its delay exceeds D_{thr}, i.e. if its NAT is before $t' - D_{thr}$. This enables us to calculate the probability that the queue length at time step t equals or exceeds i:

$$P(Q(t) \geq i) = P(t_i^a \leq t' - D_{thr}) \tag{22}$$

Then the probability mass function (PMF) of $Q(t)$ can be obtained by subtraction:

$$P(Q(t) = i - 1) = P(Q(t) \geq i - 1) - P(Q(t) \geq i) \tag{23}$$

However, for several time steps at the beginning of the green light, the vehicles before the ith vehicle may no longer be queuing, and thus $P(Q(t) \geq i - 1)$ cannot be obtained, making Eq. (23) inapplicable. A typical example is as follows, where the ith element represents the value of $P(Q(t) \geq i)$:

$$[1, \text{NA}, \text{NA}, \text{NA}, P(Q(t) \geq 4), P(Q(t) \geq 5), 0]$$

In this instance, probability estimates for queue lengths of 1, 2, and 3 are unavailable due to the first three vehicles being outside their queuing time range. Consequently, we interpolate the NA values by adopting the subsequent valid estimation, following the relationship:

$$P(Q(t) \geq 1) = P(Q(t) \geq 2) = P(Q(t) \geq 3) = P(Q(t) \geq 4)$$

Subsequently, Eq. (23) can be calculated straightforwardly.

To estimate the maximum queue length for cycle c, we can bypass the estimation of the entire cycle's queue profile. As illustrated in Fig. 6(b), for the ith vehicle, the position with the highest queuing probability occurs at the maximum possible queuing time step $t_i^{q,max}$:

$$P(Q_c \geq i) = P(Q(t_i^{q,max}) \geq i) \tag{24}$$

Therefore, we only need to sequentially calculate the probability for all vehicles departing in or after cycle c until $P(Q_c \geq i) = 0$. Then, similar to Eq. (23), we can compute the PMF of Q_c:

$$P(Q_c = i - 1) = P(Q_c \geq i - 1) - P(Q_c \geq i) \tag{25}$$

The queue profile and maximum queue length estimation processes are summarized in Algorithm 3. Notably, if we only need to calculate the maximum queue length, the parameter v_d is not required, as t' can be directly determined by Δ_i.

Algorithm 3: Estimation of the queue profile and the maximum queue length

Input: The set K_c of all vehicles departing in or after cycle c and their corresponding information, the delay threshold D_{thr}, the saturation headway h, and the discharging speed v_{d}

Output: The queue profile $Q(t) = i$ and the maximum queue length Q_c

Calculate the queue length in meters $q_i = \sum_{k=1}^{i} x_k$ if there are i vehicles in the queue;

// Queue profile estimation

foreach *time step* t *in cycle* c **do**

  $P(Q(t) \geq 0) = 1$;

  for $i = 1$ **to** $|K_c|$ *such that* $t_i^{\text{q,min}} \leq t \leq t_i^{\text{q,max}}$ **do**

    $t' \leftarrow t + \frac{q_i}{v_{\text{d}}}$;

    $P(Q(t) \geq i) = P(t_i^{\text{a}} \leq t' - D_{\text{thr}})$;

    if $P(Q(t) \geq i) = 0$ **then**

      break;

  Fill the NA values in the sequence of $P(Q(t) \geq i)$ by using the next valid estimation;

  Calculate $P(Q(t) = i - 1) = P(Q(t) \geq i - 1) - P(Q(t) \geq i)$ for each i until $P(Q(t) \geq i) = 0$;

// Maximum queue length estimation

for $i = 1$ **to** $|K_c|$ **do**

  $t' \leftarrow t_c^{\text{green}} + \Delta_i$;

  $P(Q_c \geq i) = P(t_i^{\text{a}} \leq t' - D_{\text{thr}})$;

  if $P(Q_c \geq i) = 0$ **then**

    break;

Calculate $P(Q_c = i - 1) = P(Q_c \geq i - 1) - P(Q_c \geq i)$ for all iterated i;

4.3.4. Online application

Our single-lane estimation methodology can be effectively adapted for online implementation through modifications to each component.

For the vehicle partitioning component (Algorithm 1), we implement a sliding window approach. As new vehicles enter the system, they are added to the current window. The partitioning algorithm activates when the time gap between the newest vehicle and previous vehicles exceeds δ_{min}. This design maintains optimal partitioning within each window while supporting real-time processing. The *last_valid_j* optimization significantly reduces computational costs during sequential vehicle arrivals, making it well-suited for online operation.

The arrival distribution estimation (Algorithm 2) processes the partitioned vehicle groups sequentially. For constrained groups, the estimation begins as soon as a group is identified and meets the δ_{min} time gap criterion. For unconstrained groups, we compute their distributions once we obtain the distributions of their adjacent constrained groups. To enhance computational efficiency, we parallelize the MCMC sampling process by running multiple independent sampling chains simultaneously.

For queue profile estimation (Algorithm 3), we maintain vehicle sets K_c for each cycle c. The algorithm calculates the distribution of $Q(t)$ using currently available arrival distributions, while continuously updating estimates for previous time steps within the same cycle as new data becomes available. The maximum queue length estimation follows a similar progressive approach, with the final distribution determined when we observe the first vehicle that has zero probability of queuing.

4.4. Multi-lane estimation

For each lane in the approach, we have several groups (both constrained and unconstrained) that contain unmatched vehicles, as shown in Fig. 7. For each unmatched vehicle, we can compute the minimum value $t_k^{\text{a,min}}$, and the maximum value $t_k^{\text{a,max}}$ of its NAT according to the NAT conditions established in the previous section. Then based on the minimum value T_k^{min} and the maximum value T_k^{max} of its running time, the departure time range of upstream vehicles that could match with it can be expressed as $[t_k^{\text{a,min}} - T_k^{\text{max}}, t_k^{\text{a,max}} - T_k^{\text{min}}]$. According to this range, we can obtain the upstream candidates for this vehicle. Subsequently, the upstream candidates for each group can be represented as the union of the upstream candidates of each vehicle within it. For all groups within the study period, if the union of upstream candidates of some groups has no intersection with the union of upstream candidates of other groups, then these groups can be classified into one cluster, such as groups 1, 2, 3, and groups 4, 5, 6 in Fig. 7.

For individual vehicles, upstream candidates indicate possible matchings to upstream vehicles. For each group, the possible matchings are the combination of upstream candidates of each vehicle. For example, for group 2 in Fig. 7, the upstream candidates for each vehicle are $[1]$, $[1, 2, 3]$, and $[2, 3, 4, 5]$ respectively, then valid combinations include $[1, 2, 3]$, $[1, 2, 4]$, $[1, 2, 5]$, $[1, 3, 2]$, etc. However, combinations such as $[1, 1, 2]$ and $[1, 3, 3]$ are not permitted, as vehicles are duplicated. The figure also shows possible matchings for other groups. Our task is to find the optimal matching for each group within each cluster, that is, to determine the optimal global matching for each cluster.

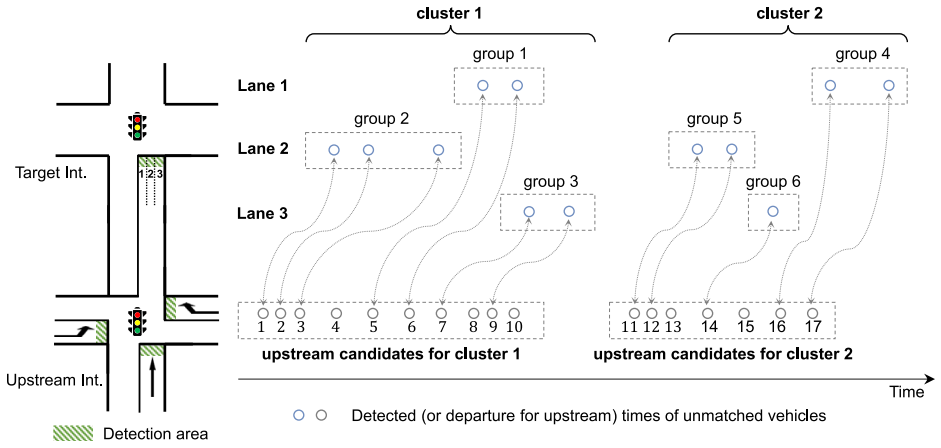

Fig. 7. Illustration of possible matchings with upstream vehicles for each group.

Remark. When faced with an exponentially large number of combinations, sampling techniques can be employed to select a representative subset, thereby avoiding the computational burden of considering all possible combinations.

4.4.1. Problem transformation

For each cluster, the problem of identifying all possible global matchings can be formulated as a general exact cover problem. Furthermore, considering the total weight associated with each global matching, this problem can be transformed into a weighted version. Before proceeding, let us briefly introduce the concept of the exact cover problem:

Definition 1 (*Exact Cover Problem*)**.** Given a collection S of subsets of a set X, the exact cover problem is to find a subcollection S^* of S that satisfies the following condition:

- Each element in X is contained in exactly one subset in S^*.

The exact cover problem can be generalized slightly to involve not only exactly-once constraints but also at-most-once constraints:

Definition 2 (*General Exact Cover Problem*)**.** Given a collection S of subsets of a set X, the general exact cover problem is to find a subcollection S^* of S that satisfies the following conditions:

- Each element in a subset $X^* \subseteq X$ is contained in at most one subset in S^*.
- Each element in $X \setminus X^*$ is contained in exactly one subset in S^*.

Fig. 8 illustrates the general exact cover problem for finding a possible global matching for cluster 1. Set X includes all the numbers of upstream candidates as well as the numbers of groups (represented by negative numbers). Among these, the numbers of upstream candidates make up the subset X^*, meaning that upstream unmatched vehicles are not necessarily matched with any group, but may instead leave the link midway. For example, in Fig. 7, the number of upstream candidates for cluster 1 is 10, but the total number of vehicles in all groups is only 7. Subsets in collection S define the possible matchings for different groups. Since sets are inherently unordered, a subset s_i may correspond to multiple matchings for group j, denoted as $\Theta_j(s_i)$. For instance, the two unmatched vehicles in group 1 can be matched with upstream vehicles 5 and 6, or 6 and 5, both of which can be represented by the subset $s_1 = \{5, 6, -1\}$.

For clearer explanation, the subsets can be expressed in matrix form as shown on the right side of Fig. 8. Each row represents a subset of set X, and the column corresponding to the position of 1 is the element of that subset. Columns corresponding to subset X^* are called secondary columns, while all others are referred to as primary columns. Therefore, our task is to select several rows from the matrix such that primary columns contain exactly one 1, and secondary columns contain at most one 1. It is easy to see that $S^* = \{s_1, s_5, s_{10}\}$, $S^* = \{s_2, s_7, s_{11}\}$, and so on, are possible solutions.

Furthermore, to evaluate different solutions, we reformulated the problem to find an optimal global matching as a maximum likelihood estimation (see Appendix). The global match's log-likelihood function $l(\theta)$ can be represented as the sum of the log-likelihood functions of different groups $l_j(\theta_j)$:

$$l(\theta) = \sum_{j \in \mathcal{G} \cup \mathcal{G}_u} l_j(\theta_j) \tag{26}$$

Here, θ_j denotes a matching for group j, and $\theta = \{\theta_j\}$ represents a global matching for a cluster. $\mathcal{G}$ and $\mathcal{G}_u$ represent the set of constrained and unconstrained groups in the cluster, respectively. Specifically, for each subset s_i corresponding with group j, we

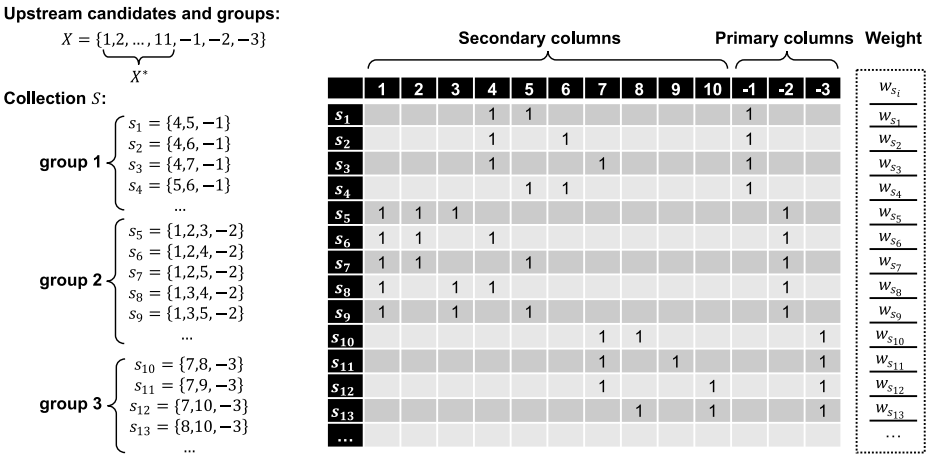

Fig. 8. Illustration of the weighted general exact cover problem to find the optimal global matching.

can calculate $\max_{\theta_j \in \Theta_j(s_i)} l_j(\theta_j)$ as the weight w_{s_i} for this subset, as illustrated in the right-most column of Fig. 8. Consequently, we can redefine the problem as a weighted version.

Definition 3 (*Weighted General Exact Cover Problem*)**.** Given a collection S of subsets of a set X, where each subset $s \in S$ is associated with a weight $w_s \in \mathbb{R}$, the weighted general exact cover problem is to find a general exact cover S^* that maximizes the total weight of the selected subsets.

Remark. In the weighted general exact cover problem, we only care about whether selecting a subset affects the selection of other subsets, not the specific order of elements within this subset. Once the optimal solution is found, the matching θ_j with the highest log-likelihood function within $\Theta_j(s_i)$ is chosen as the final result for group j.

4.4.2. Solution algorithm

Knuth's Algorithm X (Knuth, 2000) is a backtracking algorithm that finds all solutions to a (general) exact cover problem. A technique named dancing links (DLX) is designed to implement Algorithm X efficiently, at which point the algorithm is also known as Algorithm DLX. The Algorithm DLX employs circular doubly linked lists for each row and column to store the 1 s in the matrix instead of a two-dimensional array. Each column list also includes a special node known as the column header, which forms a special row consisting of all the columns that still exist in the matrix. This design accelerates operations such as removing and recovering columns or rows. However, since it is essentially a backtracking algorithm, Algorithm DLX has exponential complexity, $\mathcal{O}(c^n)$, where c is a constant very close to 1, and n is the number of 1s in the matrix. In summary, the direct application of the Algorithm DLX faces the following challenges: (1) when the problem size increases, solution efficiency cannot be guaranteed, as a cluster in a real traffic scenario may contain dozens of vehicles; (2) it can only obtain all feasible solutions and cannot evaluate the quality of the solutions, hence it is unable to select the optimal solution.

To address these challenges, we have incorporated heuristics into the Algorithm DLX, as shown in Algorithm 4. The algorithm operates on a binary matrix A, where the objective is to find a general exact cover that maximizes the total weight, given a set of weights w_r for each row r. The core functionality is built upon three key components: matrix operations, heuristic evaluation, and backtracking search.

Matrix operations are handled through the `Cover_Row` function and its inverse `Uncover_Row`. The `Cover_Row` function is designed to cover all elements associated with row r in matrix A by first removing columns corresponding to the 1 s in row r from the dancing links structure, and subsequently eliminating rows corresponding to the 1 s in those columns. In the context of our study, covering a row implies that if a possible matching for a group is selected, that group and its corresponding upstream vehicles are marked as used. Consequently, conflicting possible matchings are temporarily removed to prevent the reuse of upstream vehicles or the rematching of the group. Conversely, the function `Uncover_Row` reverses the operations performed by `Cover_Row`, restoring the matrix to its state prior to the covering. This reverse operation is crucial in the backtracking process.

Our primary improvement to Algorithm DLX is the incorporation of heuristics to guide the search for an optimal solution. The `Heuristic` function estimates the potential weight contribution if row r is included in the solution. It covers row r, computes an initial heuristic value as the weight of row r, and adds the maximum weight of the remaining rows for each primary column in the covered matrix. The matrix is then uncovered to preserve the original state.

The `Backtracking` function is the core of our algorithm, which recursively explores potential solutions. If no primary columns are left in matrix A, the current solution and its weight are recorded as the best found so far. Otherwise, it selects a primary column

with the fewest 1 s, computes heuristics for the rows in this column, sorts them in descending order of their heuristic values, and iterates through each row. If the current solution's weight plus the heuristic does not exceed the best-known total weight, it prunes the search space. For each row, it covers the row, adds it to the current solution, and recurses. After exploring, it uncovers the row and backtracks.

Remark. When matrix A is too large, consider the following measures to find a suboptimal solution:

- Retain only the top n rows (top n most likely matchings) for each group by sorting the weights.
- Set a timeout for the algorithm.

Algorithm 4: Algorithm DLX with heuristics

Input: The matrix A consisting of 0s and 1s, weights w_r for each row r
Output: Solution of the weighted general exact cover problem

Function `Cover_Row(`A, r`)`
  foreach *column* c *such that* $A[r,c]=1$ **do**
    Remove column c from A;
    foreach *row* i *such that* $A[i,c]=1$ **do**
      Remove row i from A;

Function `Uncover_Row(`A, r`)`
  foreach *column* c *such that* $A[r,c]=1$ **do**
    foreach *row* i *such that* $A[i,c]=1$ **do**
      Recover row i from A;
    Recover column c from A;

Function `Heuristic(`A, r`)`
  `Cover_Row(`A, r`)`;
  $h \leftarrow w_r$;
  foreach *primary column* c' *in matrix* A **do**
    $h \leftarrow h + \max_{r:A[r,c']=1} w_r$;
  `Uncover_Row(`A, r`)`;
  return h

Function `Backtracking(`A, *cur_sol*, *cur_weight*`)`
  if *matrix* A *has no primary columns* **then**
    $sol \leftarrow cur\_sol$;
    $total\_weight \leftarrow cur\_weight$;
    return;
  Choose a primary column c' with the lowest number of 1s;
  $lst\_row \leftarrow \{r : A[r,c']=1\}$;
  foreach *row* r *in* lst_row **do**
    $h_r \leftarrow$ `Heuristic(`A, r`)`;
  Sort lst_row in descending order of h_r;
  foreach *row* r *in* lst_row **do**
    if $total\_weight \neq -\infty$ *and* $cur\_weight + h_r \leq total\_weight$ **then**
      return;
    `Cover_Row(`A, r`)`;
    Append r to cur_sol;
    `Backtracking(`A, cur_sol, $cur\_weight + w_r$`)`;
    Pop r from cur_sol;
    `Uncover_Row(`A, r`)`;

Initialize the optimal solution sol to an empty list;
Initialize the corresponding total weight $total\_weight$ to $-\infty$;
`Backtracking(`A, `[]`, 0`)`;
return sol, $total\_weight$;

5. Evaluation

The single-lane estimation approach was first evaluated using an empirical case study. For comparison, the method proposed by Zhan et al. (2015) using multi-section LPR data for lane-based queue length estimation, was also tested. This method, referred to as the Gaussian-process-car-following (GP-CF) method, was selected as the baseline due to its alignment with our research objectives in three key aspects: its focus on lane-based queue length estimation using multi-section LPR data, its capability to handle unmatched vehicles through Gaussian process-based interpolation, and its similar parameter calibration approach that minimizes a loss function using true labels. Subsequently, the multi-lane estimation approach was evaluated using a simulation case study, accompanied by a sensitivity analysis of matching rates, volume-to-capacity (V/C) ratios, and FIFO violation rates.

5.1. Calibration of the key parameters

In the proposed approach, four key parameters require calibration: the minimum inter-group departure gap ($\delta_{\min}$), the saturation headway (h), the discharging speed (v_{d}), and the delay threshold (D_{thr}). The minimum inter-group departure gap can be calibrated as the range of the running time. To calibrate the saturation headway, we employ the 15th percentile of observed headways on the target lane. Since the discharging speed and the delay threshold cannot be directly obtained from the LPR data, we utilize a grid search approach to systematically explore the parameter space and minimize the following loss function:

$$L(v_{\mathrm{d}}, D_{\mathrm{thr}}) = w_1 \sum_{c=1}^{N_{\mathrm{calib.}}} \left(\hat{y}_c(v_{\mathrm{d}}, D_{\mathrm{thr}}) - y_c\right)^2 + w_2 \sum_{c=1}^{N_{\mathrm{calib.}}} \left(\hat{t}_c^{\mathrm{q,max}}(v_{\mathrm{d}}, D_{\mathrm{thr}}) - t_c^{\mathrm{q,max}}\right)^2 \tag{27}$$

where $N_{\mathrm{calib.}}$ signifies the number of calibration cycles; $\hat{y}_c(v_{\mathrm{d}}, D_{\mathrm{thr}})$ and $\hat{t}_{q,c}^{max}(v_{\mathrm{d}}, D_{\mathrm{thr}})$ denote the estimated maximum queue length and the time of occurrence of the maximum queue length for cycle c given parameters v_{d} and D_{thr}, respectively; y_c and $t_c^{\mathrm{q,max}}$ represent the corresponding ground truth; while w_1 and w_2 are weights assigned to the two terms. Importantly, v_{d} and D_{thr} only participate in the final step described in Section 4.3, thereby facilitating a rapid search process. For the exclusive purpose of estimating the maximum queue length, calibration is required solely for the delay threshold by minimizing the loss function defined as

$$L'(D_{\mathrm{thr}}) = \sum_{c=1}^{N_{\mathrm{calib.}}} \left(\hat{y}_c(v_{\mathrm{d}}, D_{\mathrm{thr}}) - y_c\right)^2 \tag{28}$$

The baseline method, GP-CF, performed a similar calibration procedure by searching the car-following parameters to minimize the following loss function:

$$L''(d_{\mathrm{safe}}, v_{\mathrm{desired}}) = \sum_{c=1}^{N_{\mathrm{calib.}}} \left(\hat{y}_c(d_{\mathrm{safe}}, v_{\mathrm{desired}}) - y_c\right)^2 \tag{29}$$

where d_{safe} is the safe driving distance and v_{desired} is the desired speed.

5.2. Empirical case study

The intersection of Jinling Road and Taihu Road in Changzhou, China, was selected for the empirical case study. Two through lanes of the southbound approach were studied, as illustrated in Fig. 9(a). The upstream intersection is 580 m away. All lanes at both intersections are captured by LPR cameras. Due to a branch road (Daduhe Road) within the link, vehicles in the studied lanes are not necessarily captured by the upstream intersection. Therefore, experiments for the multi-lane estimation were not conducted here. Given that branch roads within links are common in real-world scenarios, selecting this site helps to evaluate the general applicability of the proposed approach.

The LPR data and the corresponding recorded videos from both the upstream and the target intersections were available on September 7th, 2020. We selected the morning peak period from 7:00 am to 8:50 am as the study period. The cycle-by-cycle maximum queue lengths of the two studied lanes were obtained as the ground truth data using the opposite side camera. The recorded videos also captured the traffic light, so signal timing data could be extracted manually. Both the upstream and target intersections adopted fixed timing plans with a common cycle length of 160 s and a mainstream flow offset of 60 s, as illustrated in Fig. 9(b). Each phase has a yellow interval of 3 s, and there is no all-red clearance time. Subsequently, we proceed to the license plate matching process. The average matching rate is 87.5%, which decreases to 73.0% after filtering out FIFO violations for the GP-CF method. Notably, vehicles from upstream right-turn movements account for only 8.3% of the total, and excluding these vehicles reduces the average matching rate to 79.5%. The matching rates for the two studied lanes are 90.5% and 89.2%, respectively, and after excluding upstream right turns, these rates decrease to 84.3% and 80.0%. Through video verification, we confirmed a negligible miss detection rate (<1%). Additionally, travel times within the upstream intersection were manually measured by reviewing video footage, and the medians of the sampled observations were used as estimates. We utilize the ground truth queue length data from the first 8 cycles to calibrate the delay threshold D_{thr} of the proposed approach and the car-following model of the GP-CF method. Considering the limited number of vehicles traveling at free-flow speed during peak hours, we utilize the travel time data from 6:00 am to 12:00 am to estimate the distribution of running time, thereby obtaining the minimum inter-group departure gap $\delta_{\min}$. The calibrated values are summarized in Table 1.

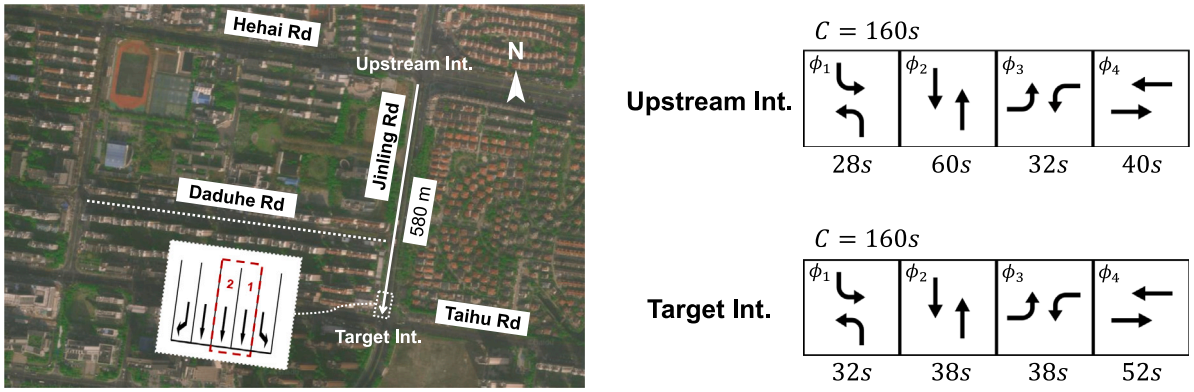

Fig. 9. Overview of the empirical case study site in Changzhou, showing (a) the geographic layout with the target intersection and upstream intersection along Jinling Road spaced 580 m apart, and (b) signal timing plans with a cycle length of 160 s for both intersections.

Table 1
Parameters for the proposed approach and the baseline method.

Parameter	Value	Description
h	2.0	Saturation headway (s)
D_{thr}	5.1	Delay threshold (s)
μ_T	3.89	Location parameter of the log-normal distribution
σ_T	0.15	Scale parameter of the log-normal distribution
$T_{\min}$	31.1	Minimum running time (s)
$T_{\max}$	58.0	Maximum running time (s)
$\delta_{\min}$	26.9	Minimum inter-group departure gap (s)
Δt_{LT}	8	Left-turn travel time within the upstream intersection (s)
Δt_{TH}	6	Through travel time within the upstream intersection (s)
Δt_{RT}	5	Right-turn travel time within the upstream intersection (s)
d_{safe}	6.2	Safe driving distance for the baseline method (m)
v_{desired}	17.0	Desired speed for the baseline method (m/s)

Since the calculation result of the maximum queue length is in probabilistic form, to verify it, we first calculate the mean μ_{Q_c} of Q_c, and the lower L_{Q_c} and upper U_{Q_c} bounds of the 95% confidence interval (CI):

$$\mu_{Q_c} = \sum_i i \cdot P(Q_c = i) \tag{30}$$

$$L_{Q_c} = \max_{P(Q_c \geq i) \geq 0.975} i \tag{31}$$

$$U_{Q_c} = \min_{P(Q_c \geq i) \leq 0.025} i \tag{32}$$

Then the performance of the maximum queue length estimation was evaluated using four metrics: the mean absolute error (MAE), the root mean square error (RMSE), the mean absolute percentage error (MAPE), and the coverage rate. These metrics were calculated as follows:

$$\text{MAE}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{N_{\text{cycle}}} \sum_{c=1}^{N_{\text{cycle}}} |y_c - \hat{y}_c| \tag{33}$$

$$\text{RMSE}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \sqrt{\frac{1}{N_{\text{cycle}}} \sum_{c=1}^{N_{\text{cycle}}} (y_c - \hat{y}_c)^2} \tag{34}$$

$$\text{MAPE}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{N_{\text{cycle}}} \sum_{c=1}^{N_{\text{cycle}}} \left| \frac{y_c - \hat{y}_c}{y_c} \right| \times 100\% \tag{35}$$

$$\gamma_{\text{queue}} = \frac{N_{\text{covered}}}{N_{\text{cycle}}} \times 100\% \tag{36}$$

where $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$ are the ground truth and estimated values, respectively; N_{cycle} is the number of cycles; N_{covered} is the number of cycles covered by the 95% confidence interval; and γ_{queue} is the coverage rate of queue length.

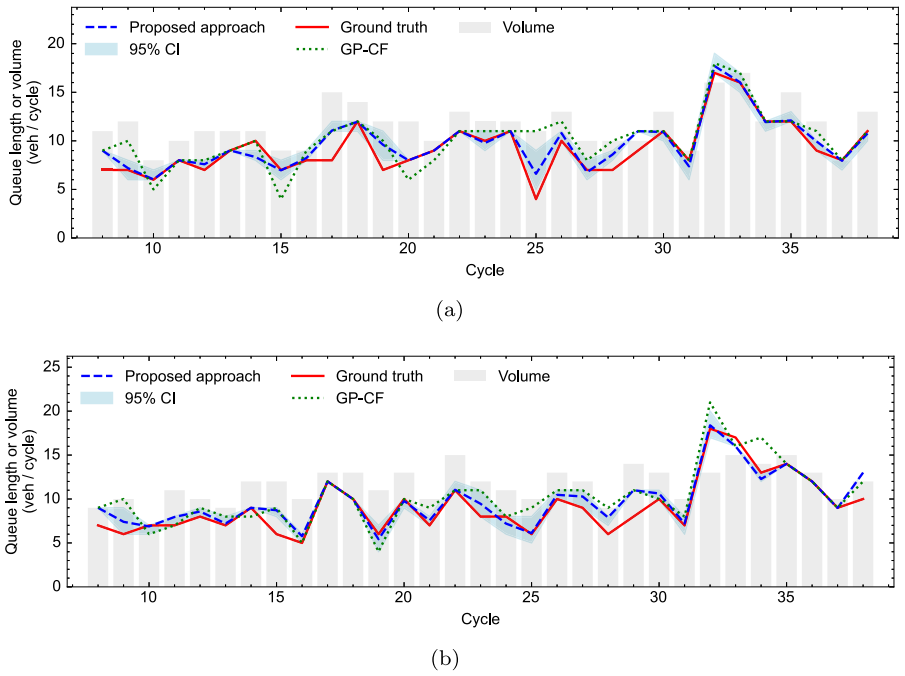

Fig. 10. Maximum queue length estimation results comparing the evaluated methods with ground truth data: (a) through lane 1 and (b) through lane 2.

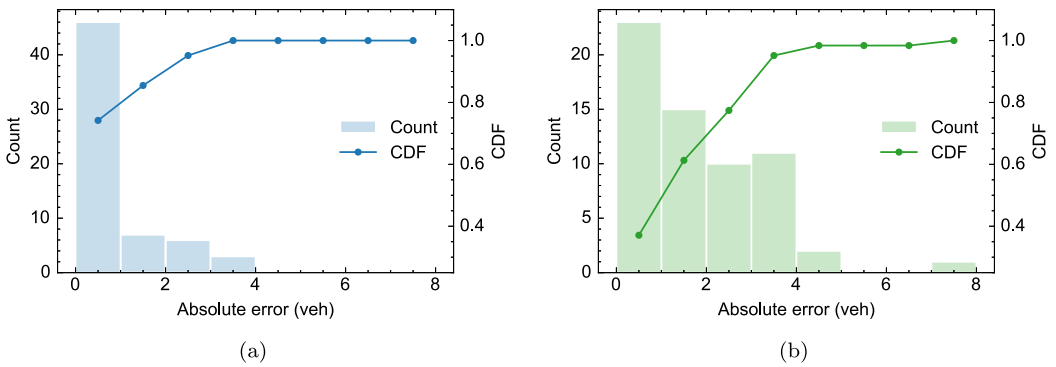

Fig. 11. Distribution of absolute errors in maximum queue length: (a) proposed approach and (b) GP-CF method.

5.2.1. *Maximum queue length estimation*

Fig. 10 presents the detailed queue length estimation results of the proposed approach and the GP-CF method, where the gray bars represent the traffic volume. The estimation error of the proposed approach is significantly smaller than the GP-CF method for half the cycles in both lanes, and the difference between the two is also relatively small in other cycles. From the error histograms in Fig. 11, we can find that the majority of the errors for the proposed approach are concentrated below 1 veh, accounting for 74% of all cycles of two lanes, with the maximum error being less than 4 veh. In contrast, errors below 1 veh only account for 37% of the cycles for the GP-CF method, while most errors in other cycles are uniformly distributed between 1 veh and 4 veh, with a maximum error of 7 veh. These findings demonstrate that the proposed approach significantly outperforms the GP-CF method in terms of accuracy and reliability in empirical cases. Another advantage of the proposed approach is that we provide confidence intervals, and it can be found that the vast majority of cycles fall within the confidence intervals. Existing studies have demonstrated that such confidence interval information can improve the performance of traffic signal control by robust theory (Tan et al., 2024b).

Table 2 quantifies the comprehensive performance advantages of the proposed method across both through lanes. The improvements are substantial and consistent across all evaluation metrics, with reductions in error measures exceeding 37% compared to the GP-CF method. These consistent improvements across different error metrics indicate that the method enhances both average performance and extreme case handling. Furthermore, the confidence interval coverage rates of 77.42% and 80.65% for the two lanes confirm the reliability of the uncertainty quantification.

Table 2
Maximum queue length estimation results.

Metric	Proposed approach	GP-CF	Improvement
Through lane 1			
MAE (veh/cycle)	0.67	1.29	48.06% ↓
RMSE (veh/cycle)	1.13	1.98	42.93% ↓
MAPE (%)	9.29	19.13	51.44% ↓
Coverage rate (%)	77.42	–	–
Through lane 2			
MAE (veh/cycle)	0.81	1.39	41.73% ↓
RMSE (veh/cycle)	1.20	1.91	37.17% ↓
MAPE (%)	10.65	17.88	40.44% ↓
Coverage rate (%)	80.65	–	–

Table 3
Sensitivity analysis of maximum queue length estimation across various matching rates.

Lane	Matching rate (%)	Proposed approach			GP-CF		
		MAE (veh/cycle)	RMSE (veh/cycle)	MAPE (%)	MAE (veh/cycle)	RMSE (veh/cycle)	MAPE (%)
TH1	30.0	0.97	1.40	12.77	2.27	2.91	30.25
	40.0	0.92	1.33	12.15	2.02	2.67	27.65
	50.0	0.85	1.22	11.45	1.76	2.43	24.43
	60.0	0.79	1.18	10.63	1.61	2.31	22.73
	70.0	0.73	1.14	9.90	1.50	2.18	21.39
	80.0	0.70	1.13	9.59	1.38	2.04	20.03
	84.3[a]	0.68	**1.12**	9.35	1.35	2.03	19.52
	90.5[b]	**0.67**	1.13	**9.29**	1.29	1.98	19.13
TH2	30.0	1.33	1.74	17.13	2.27	2.89	28.19
	40.0	1.19	1.55	15.01	2.11	2.75	26.66
	50.0	1.08	1.44	13.93	1.88	2.59	24.07
	60.0	1.00	1.37	12.81	1.69	2.38	21.73
	70.0	0.93	1.30	11.99	1.50	2.15	19.34
	80.0	0.87	1.25	11.21	1.42	2.01	18.35
	80.0[a]	**0.81**	**1.18**	**10.45**	1.42	2.09	19.09
	89.2[b]	**0.81**	1.20	10.65	1.39	1.91	17.88

[a] All upstream right-turning vehicles were excluded from the dataset.
[b] Complete dataset without any modifications.

5.2.2. Impact of matching rates

In the worst case, the matching rate of the LPR data at two consecutive intersections can degrade to 50%–70% (Zhan et al., 2015). To test the performance of the proposed approach under extreme conditions, we intentionally degraded the matching rate by randomly removing vehicle IDs in the database using five different random seeds, resulting in six test scenarios with matching rates ranging from 30% to 80%. To further assess the effectiveness of the proposed approach in the absence of LPR cameras in the right-turn lane, we removed all upstream right-turning vehicles from the dataset. This yielded matching rates of 84.3% and 80.0% for the two studied lanes, respectively.

The sensitivity analysis presented in Table 3 reveals the distinct performance characteristics of both methods under varying matching rates. While the GP-CF method shows significant degradation at lower matching rates, with MAE increasing from 1.38 to 2.27 veh/cycle as matching rates drop from 80% to 30% in through lane 1 (TH1), the proposed approach demonstrates remarkable stability. For through lane 1, our method achieves an MAE of 0.97 veh/cycle at 30% matching rate, outperforming even the GP-CF method's performance at 80% matching rate. Similarly, for through lane 2 (TH2), our approach exhibits consistent performance across different matching rates, with improvements of 38%–44% in MAE compared to the GP-CF method.

A notable phenomenon emerges in the right-turn-excluded scenario. For through lane 2, which experiences a more significant decline in matching rate (9.2%) due to a higher likelihood of upstream right-turn selection, the right-turn-excluded scenario achieved better performance than the full dataset at 89.2% matching rate. This improvement can be attributed to the variable behavior patterns of right-turning vehicles. Specifically, upstream right-turning vehicles, often uncontrolled, may depart during red light phases, resulting in running times that differ from other vehicles. Including these vehicles in constrained vehicle groups could introduce noise into the queue length estimation process. Conversely, when these vehicles remain undetected upstream (becoming unmatched vehicles), they are likely categorized into the unconstrained group (see Fig. 4). Therefore, their exclusion from the dataset actually eliminates a significant source of noise in the queue length estimation process. These findings suggest that our approach's effectiveness depends more on the consistency of vehicle behavior patterns than on the total number of matched vehicles.

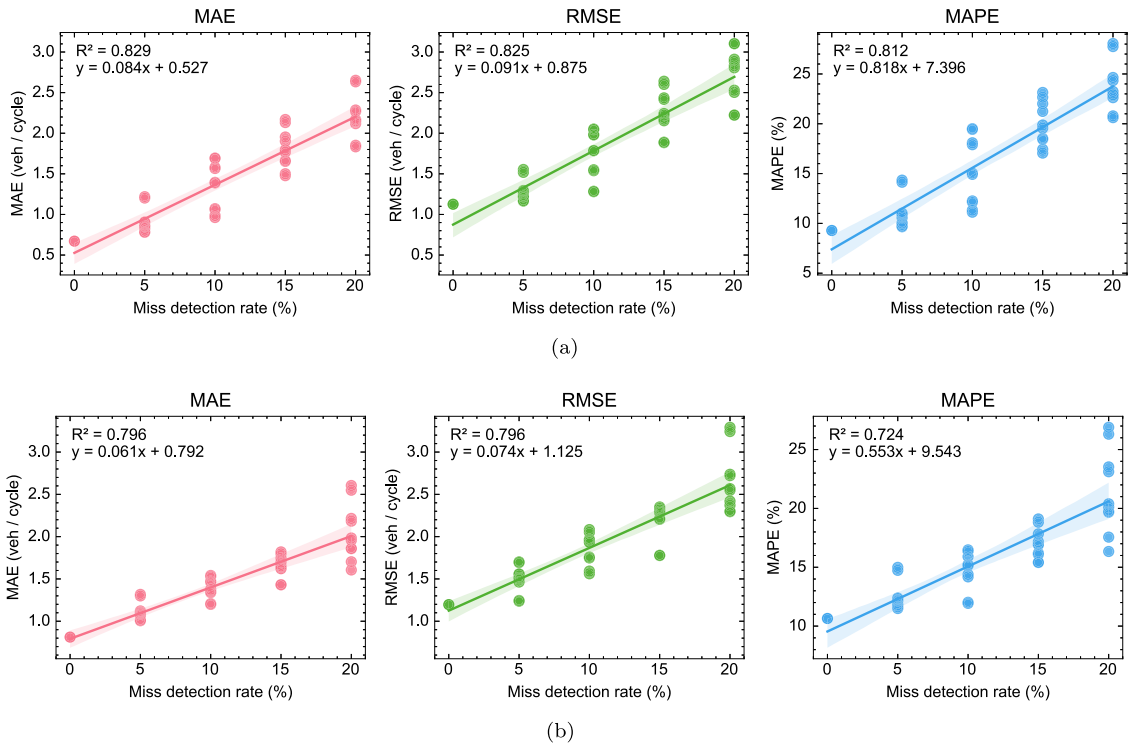

Fig. 12. Sensitivity analysis of maximum queue length estimation with miss detection rates ranging from 0% to 20%, illustrating the linear relationship between MAE and miss detection rate: (a) through lane 1 and (b) through lane 2.

5.2.3. Impact of miss detection rates

Miss detection is a common challenge in LPR systems due to factors such as illumination fluctuations, perspective distortion, and environmental interference (Wu et al., 2024). Although the proposed approach does not explicitly address this issue given its low occurrence rate in typical scenarios, it is essential to evaluate the method's robustness under such conditions. We simulated miss detection scenarios by randomly removing LPR records from both the upstream and target intersections with equal probabilities, generating test cases with miss detection rates from 5% to 20% at 5% increments using five different random seeds.

Fig. 12 presents the sensitivity analysis results, revealing strong linear relationships between detection performance and estimation accuracy across all metrics and both lanes. This linear degradation can be attributed to two key factors. First, miss detection at the target intersection directly reduces the number of available LPR records for queue length estimation, leading to proportionally reduced accuracy in queue estimates. Second, miss detection at the upstream intersection decreases the matching rates, which exhibits a linear impact on estimation accuracy within the tested range. The slightly different slopes between the two lanes ($R^2 > 0.78$ for all cases) reflect their distinct traffic patterns and queue formation characteristics, with through lane 2 showing marginally better resilience to detection failures despite its higher baseline error. The uniformity of these linear relationships indicates that the proposed approach maintains graceful degradation under increasing miss detection rates. Notably, even at the maximum tested miss detection rate of 20%, the estimation errors remain within practical bounds, with MAE values not exceeding 2.5 veh/cycle for either lane. In comparison with the matching rate analysis presented in Section 5.2.2, these findings suggest that the proposed approach exhibits higher sensitivity to miss detections than to matching rate degradation. The difference in sensitivity can be explained by the fact that miss detection results in complete loss of vehicle information, while matching failures still preserve the presence information of individual vehicles at both intersections. This insight carries important implications for system design and maintenance strategies, suggesting that maintaining reliable vehicle detection should be prioritized over achieving high license plate recognition rates in practical deployments.

5.3. Simulation case study

To evaluate the multi-lane estimation approach, we constructed a small network comprising two intersections in Tongxiang, China, and implemented the simulation on the SUMO platform. As shown in Fig. 13(a), one through lane of the eastbound approach at the intersection of Qingfeng Road and Yuqiao Road was selected for study. The stop line of the approach is about 910 m from the upstream, and variations in travel times and overtaking behaviors are significantly different from the empirical case. The signal phases and timing are illustrated in Fig. 13(b), showing that the two intersections are not coordinated. The intergreen periods consist

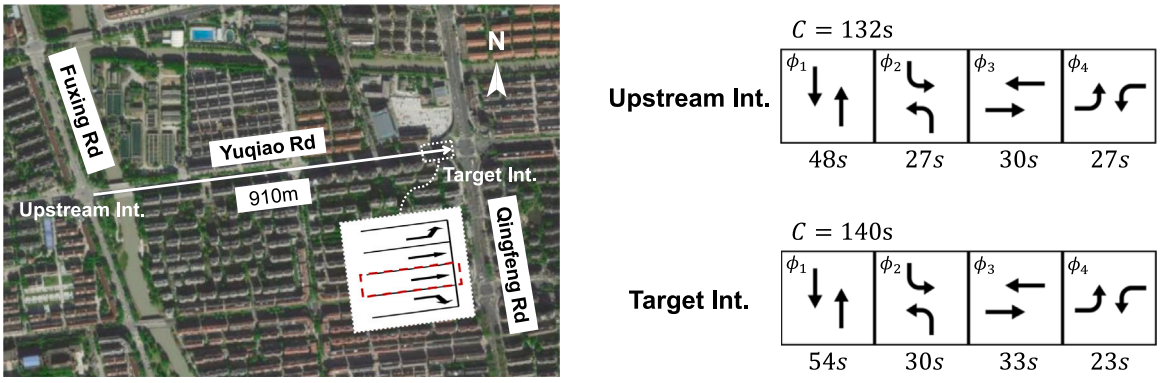

Fig. 13. Overview of the simulation case study site in Changzhou, showing (a) the geographic layout with the target intersection and upstream intersection along Yuqiao Road spaced 910 m apart, and (b) signal timing plans with different cycle lengths (132 s for upstream and 140 s for target intersection).

of a 3-s yellow interval followed by a 1-s all-red clearance. The simulation model was calibrated using LPR data from 8:00 am to 9:00 am, including arrival rate, speed, turning ratio, and saturation headway. Then, based on the basic scenario, we considered different V/C ratios, matching rates, and FIFO violation rates, which yielded various simulation scenarios. The simulation ran for 7800 s with a warm-up period of 600 s. The trajectory data of each vehicle, including vehicle IDs, timestamps, and locations, were recorded in the database to verify queue length estimation. Finally, we used the first 12 cycles after the warm-up period to calibrate the proposed approach for each scenario, including the discharging speed v_d and the delay threshold D_{thr}. Both weights w_1 and w_2 were assigned a value of 1. The car-following models of the GP-CF method were calibrated accordingly.

Since we can obtain the true queue profile in the simulation, the estimated queue profile is evaluated in this section. Like the evaluation of the maximum queue length, we first calculate the mean $\mu_{Q(t)}$ of $Q(t)$, and the lower $L_{Q(t)}$ and upper $U_{Q(t)}$ bounds of the 95% confidence interval:

$$\mu_{Q(t)} = \sum_i i \cdot P(Q(t) = i) \tag{37}$$

$$L_{Q(t)} = \max_{P(Q(t)\geq i)\geq 0.975} i \tag{38}$$

$$U_{Q(t)} = \min_{P(Q(t)\geq i)\leq 0.025} i \tag{39}$$

Then our queue profile estimation performance was evaluated using three metrics: the MAE during red, the MAE during green, and the coverage rate. These metrics were calculated as follows:

$$\mathrm{MAE}_{\mathrm{red}}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{|\mathcal{T}_{\mathrm{red}}|} \sum_{t\in\mathcal{T}_{\mathrm{red}}} |y_t - \hat{y}_t| \tag{40}$$

$$\mathrm{MAE}_{\mathrm{green}}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{|\mathcal{T}_{\mathrm{green}}|} \sum_{t\in\mathcal{T}_{\mathrm{green}}} |y_t - \hat{y}_t| \tag{41}$$

$$\gamma_{\mathrm{profile}} = \frac{|\mathcal{T}_{\mathrm{covered}}|}{|\mathcal{T}|} \times 100\% \tag{42}$$

where $\mathcal{T} = \mathcal{T}_{\mathrm{red}} \cup \mathcal{T}_{\mathrm{green}}$ is the set of time steps in the study period; $\mathcal{T}_{\mathrm{red}}$ is the set of time steps during red phases; $\mathcal{T}_{\mathrm{green}}$ is the set of time steps during yellow and green phases; $\mathcal{T}_{\mathrm{covered}}$ is the set of time steps covered by the 95% confidence interval; and $\gamma_{\mathrm{profile}}$ is the coverage rate of queue profile. The reason for separately calculating the error during green phases is that the queue length will instantly drop to zero at some point during the green phase, causing small deviations to result in a large MAE, and part of the green phase has no queue.

5.3.1. Impact of V/C ratios

In this section, we scaled the arrival rate of the upstream intersection to create five scenarios with varying V/C ratios, ranging from 0.5 to 0.9. We then randomly removed vehicle IDs from the database using five different random seeds, obtaining results with a matching rate of 60% as representative.

Fig. 14(a) presents the performance metrics of maximum queue length estimation under different V/C ratios. Across all scenarios, the results demonstrate the consistent superiority of multi-lane estimation over single-lane estimation. As traffic conditions intensify with increasing V/C ratios, both approaches show a gradual decline in accuracy, with MAE and RMSE reaching their peaks at a V/C ratio of 0.9. However, these errors remain within acceptable bounds, staying below 0.8 veh/cycle and 1.2 veh/cycle, respectively. The MAPE exhibits an inverse relationship with V/C ratios, decreasing as ratios increase due to larger ground truth queue lengths in high V/C scenarios, while consistently remaining below 11%. The performance advantage of multi-lane estimation is most

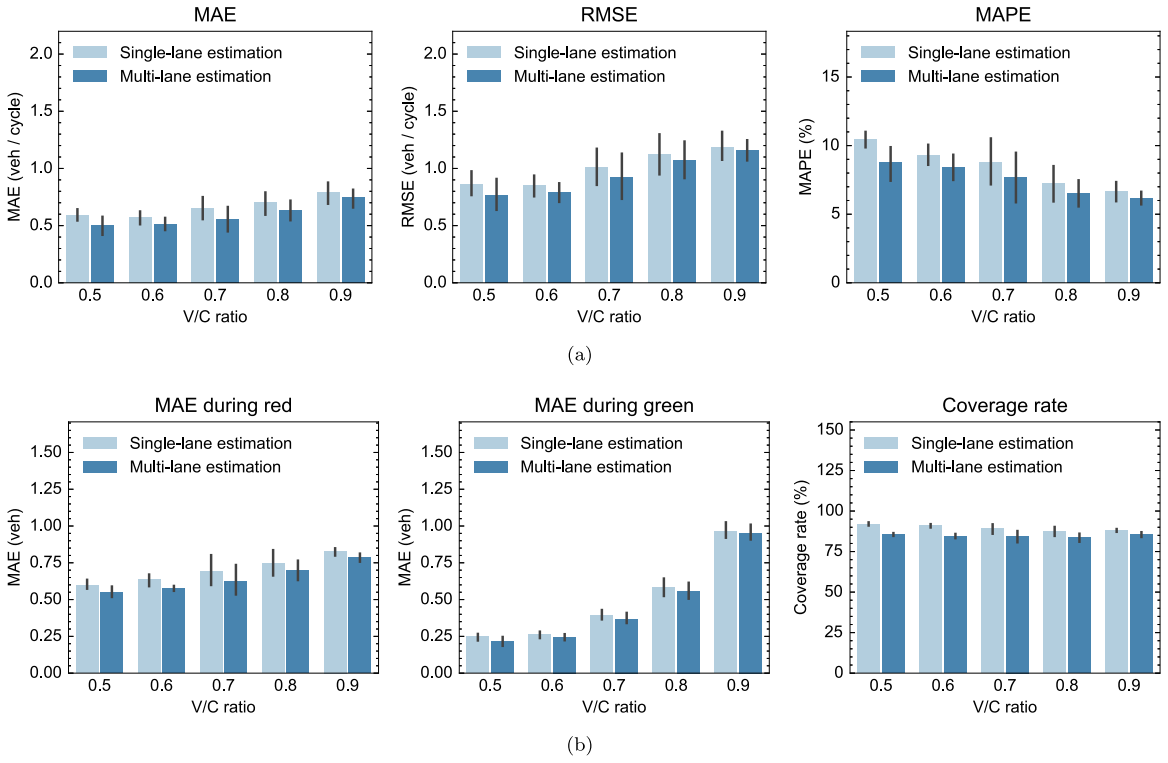

Fig. 14. Sensitivity analysis of queue length estimation across V/C ratios (0.5–0.9) for single-lane and multi-lane approaches: (a) maximum queue length estimation and (b) queue profile estimation.

pronounced at lower V/C ratios, though this margin gradually narrows under more congested conditions.

Fig. 14(b) illustrates the queue profile estimation performance during red and green phases, along with the coverage rate across different V/C ratios. Similar to the maximum queue length estimation, the multi-lane approach maintains its superior performance across all scenarios. Analysis of the MAE reveals distinct patterns between signal phases: during the red phase, the error increases gradually with V/C ratios, while the green phase shows a more pronounced rise, particularly under higher V/C ratios. This divergence can be attributed to the presence of longer queues during the green phase in high V/C scenarios, where small deviations in queue clearance time estimation can lead to significant profile errors. Despite these variations in MAE, the coverage rate remains relatively stable across V/C ratios, showing only a slight decrease in single-lane estimation while maintaining stability in multi-lane estimation, with both methods consistently achieving rates above 80%.

5.3.2. Impact of matching rates

Building upon findings from our empirical case study, which demonstrated the robust performance of the proposed single-lane estimation approach compared to the baseline method, we conducted a comparative analysis to further demonstrate the improvement of the multi-lane estimation approach at a fixed V/C ratio of 0.7 across varying matching rates.

For maximum queue length estimation, Fig. 15(a) demonstrates that 60% represents an inflection point across all three error metrics (MAE, RMSE, and MAPE). Beyond this threshold, the error reduction becomes more gradual, indicating that the core traffic patterns are sufficiently captured. The multi-lane approach consistently achieves lower error rates compared to the single-lane approach, with the most significant improvements observed in the 60%–80% matching rate range. At 30% matching rate, both approaches exhibit similar performance limitations due to the high proportion of unmatched vehicles, which complicates optimal matching.

Queue profile estimation results, shown in Fig. 15(b), reveal distinct patterns during different signal phases. The multi-lane estimation exhibits particularly strong performance during red phases, with notable improvements for matching rates between 50% and 80%. Green phase estimation improvements are more modest but still positive. The coverage rate shows steady improvement with higher matching rates, eventually stabilizing around 91%. When the matching rate is less than 80%, the coverage rate of the multi-lane estimation is significantly lower than that of the single-lane estimation. This trend is due to our efforts to reduce uncertainty by matching all unmatched vehicles in the multi-lane estimation, which means that even minor estimation errors can place the ground truth outside the 95% confidence interval.

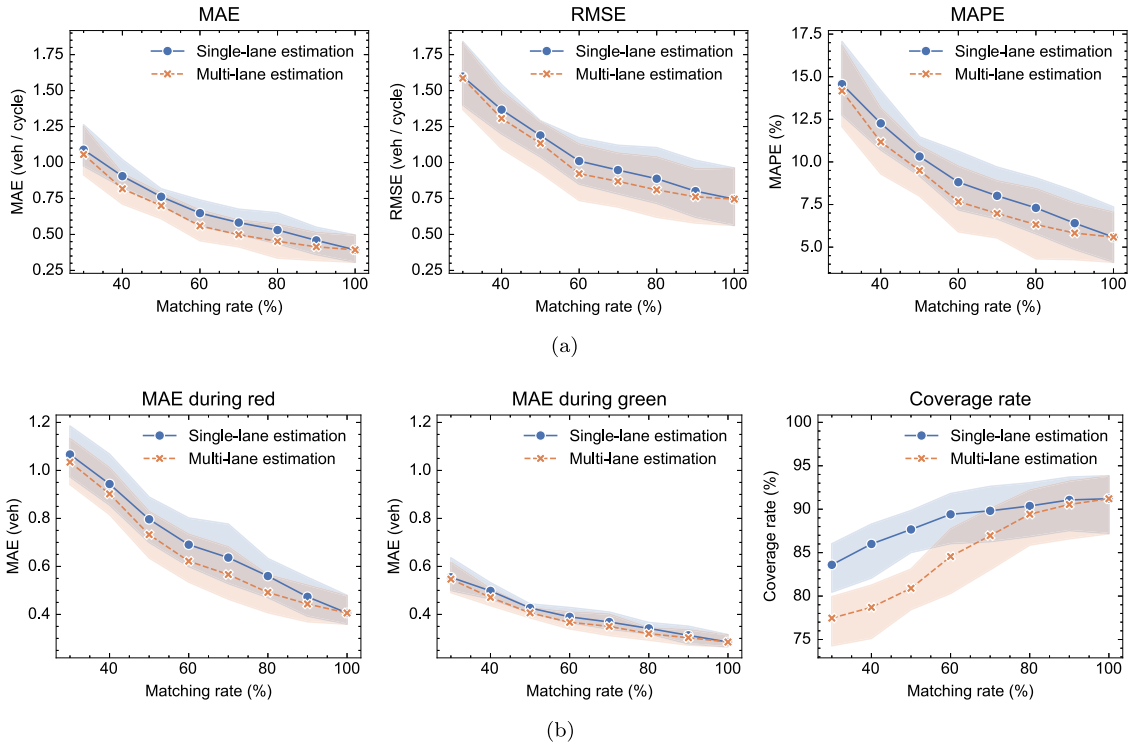

Fig. 15. Sensitivity analysis of queue length estimation across matching rates (30%–100%) for single-lane and multi-lane approaches: (a) maximum queue length estimation and (b) queue profile estimation.

5.3.3. *Impact of FIFO violation rates*

A key difference between the proposed approach and existing methods lies in the relaxation of the FIFO assumption, making it necessary to study the impact of the FIFO violation rate on this approach. The FIFO violation rate is defined as the proportion of vehicles that violate the FIFO rule among all vehicles. Specifically, we can sort all vehicles passing through the target lane by their arrival order. For any matched vehicle, if there exists any vehicle that arrives before it but has a later upstream departure time, it is considered a FIFO violation. The FIFO violation rate can be expressed as

$$\text{FIFO violation rate} = \frac{\left|\left\{ i \in \mathcal{M} \middle| \exists j \in \mathcal{M} : t_i^{\mathrm{d}} < t_j^{\mathrm{d}} \text{ and } t_i^{\mathrm{u}} > t_j^{\mathrm{u}} \right\}\right|}{|\mathcal{V}|} \tag{43}$$

where $\mathcal{V}$ is the set of all vehicles and $\mathcal{M}$ is the set of matched vehicles. For each matched vehicle i, we examine whether any other matched vehicle j has overtaken vehicle i, which indicates a violation of the FIFO principle by vehicle i.

To analyze the sensitivity of our approach to these violations, we conducted experiments using a scenario with a V/C ratio of 0.7 and a matching rate of 60%. We controlled vehicle overtaking frequency by modifying the speed factor distribution in SUMO, where the speed factor serves as a vehicle-specific multiplier of the road speed limit to determine each vehicle's desired free-flow driving speed. The default speed factor for passenger cars follows a truncated normal distribution $\mathcal{N}(1, 0.1, 0.2, 2)$, implying that about 95% of the vehicles drive between 80% and 120% of the legal speed limit. By increasing the speed factor deviation from 0.10 to 0.20, we observed the relationship between deviation and FIFO violation rates, as shown in Fig. 16. The analysis revealed a clear linear correlation ($R^2 = 0.954$), with FIFO violation rates increasing from approximately 19% at a deviation of 0.1 to 29% at a deviation of 0.2. This demonstrates that varying the speed factor deviation in SUMO allows us to investigate the impact of FIFO violation rates on the performance of our proposed approach.

Fig. 17(a) presents the sensitivity analysis results for maximum queue length estimation. Both approaches show a slight upward trend in error metrics as speed factor deviation increased, though the overall performance remained favorable. Specifically, the MAE stays within 0.5–0.8 veh/cycle, the RMSE between 0.8–1.2 veh/cycle, and the MAPE ranges from 7% to 11%. The multi-lane estimation approach maintains its performance advantage, particularly at speed factor deviations of 0.1 and 0.14. Further examination of queue profile estimation, depicted in Fig. 17(b), yields comparable findings, with the MAE remaining below 0.8 veh/cycle during red phases and 0.4 veh/cycle during green phases. The coverage rate exhibited a gradual decline with increasing speed factor deviation but consistently remained above 80% for both estimation approaches.

These results demonstrate the approach's effectiveness in handling frequent overtaking behaviors. Even at a speed factor deviation of 0.2, where vehicle speeds varied substantially (60% to 140% of the speed limit), the estimation accuracy remained within

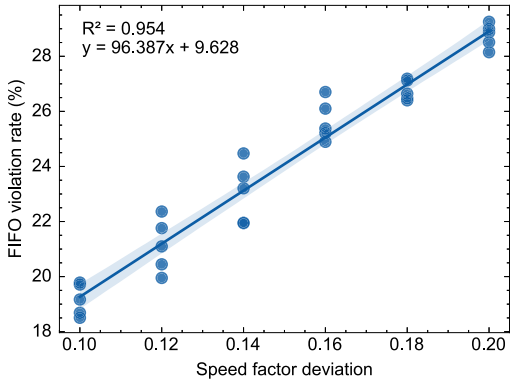

Fig. 16. The correlation between the speed factor deviation and the FIFO violation rate.

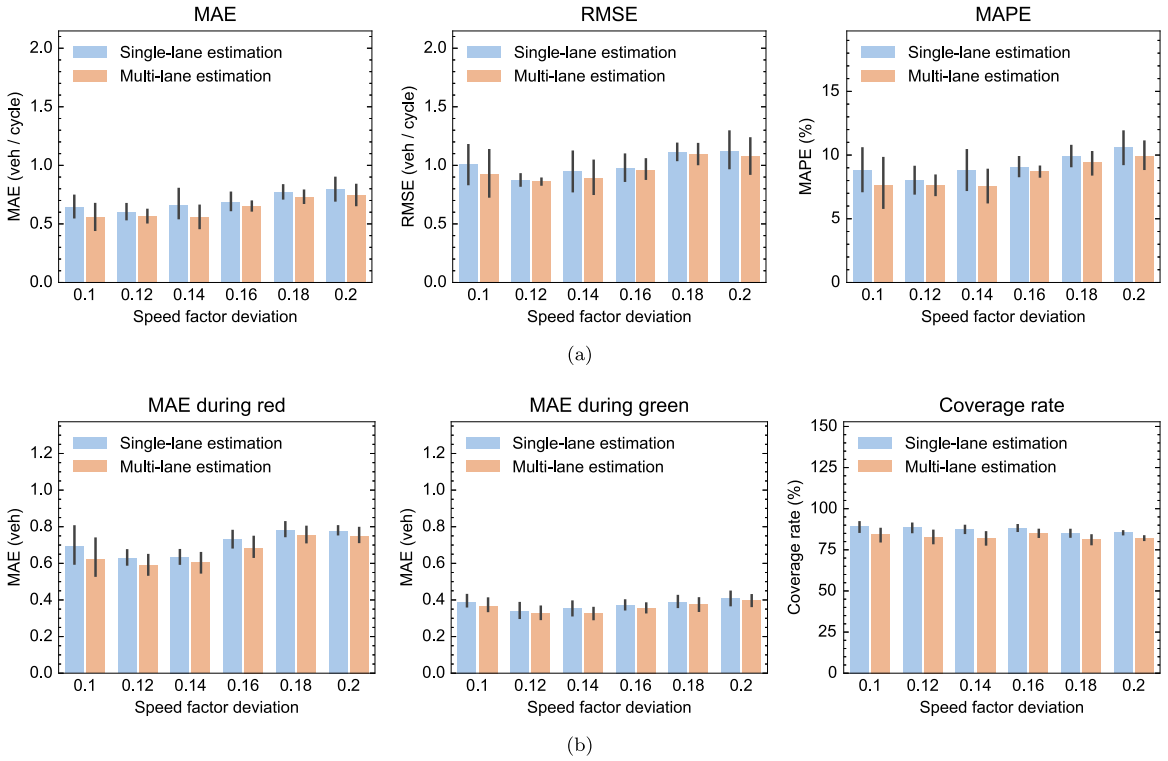

Fig. 17. Sensitivity analysis of queue length estimation across speed factor deviations (0.1–0.2) for single-lane and multi-lane approaches: (a) maximum queue length estimation and (b) queue profile estimation.

acceptable bounds. This robustness can be attributed to our approach's greater emphasis on the departure sequence of the target lanes, rather than the upstream departure sequence, thereby considering the overtaking and the potential impact on vehicle running times.

5.3.4. Detailed queue profile estimation

Figs. 18 and 19 present the partial cycle results of the estimated queue profiles from the proposed approach (including both the single-lane and multi-lane estimations) and the GP-CF method. Fig. 18 shows the results for a V/C ratio of 0.7, while Fig. 19 shows the results for a V/C ratio of 0.9, both with a matching rate of 60%.

At a V/C ratio of 0.7, the proposed approach outperforms the GP-CF method, with the multi-lane estimation showing a slight advantage over the single-lane estimation. Since the queue mostly forms during the red phase, the improvement in the MAE during the red phase is more significant. For instance, in cycles 17, 18, 20, and 21, the multi-lane estimation results show noticeably smaller deviations from the ground truth, with a narrower 95% confidence interval, which also reduces the error in estimating

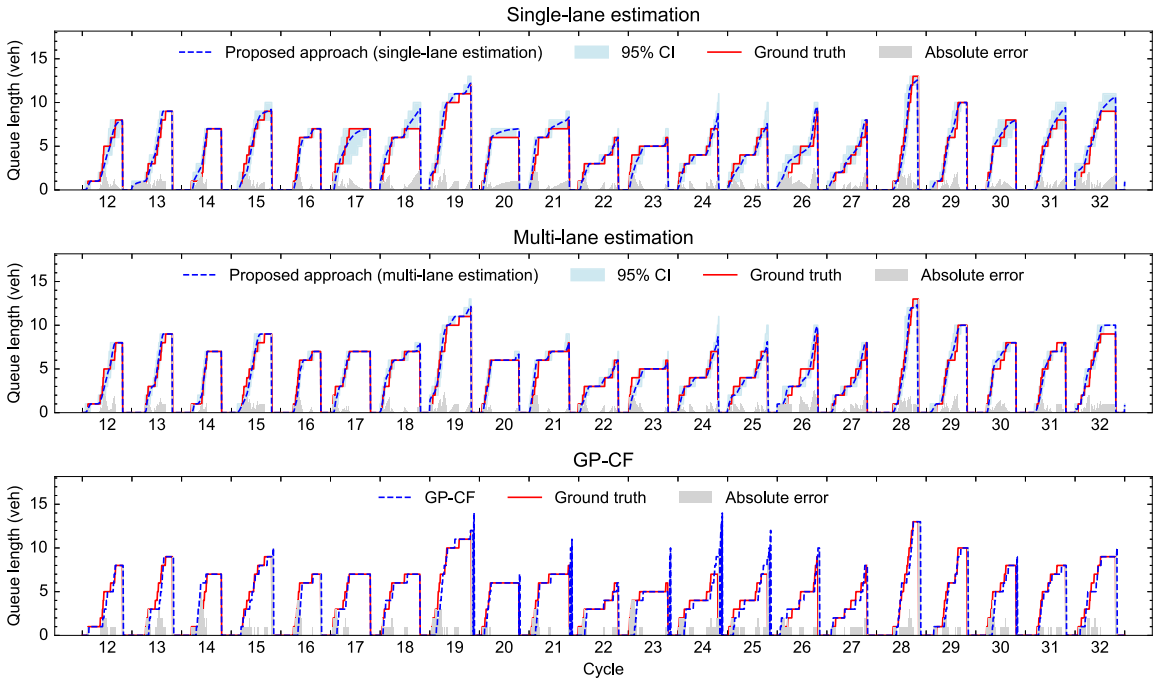

Fig. 18. Partial cycle result of estimated queue profile with a V/C ratio of 0.7.

the maximum queue length. The GP-CF method estimates the arrival curve using signal timing information from the upstream intersection, leading to a good estimation of the approximate upstream departure times and acceptable queue profile estimation during the queue formation period. However, since the GP-CF method relies on a car-following model that assumes only vehicles heading to the target lane are present, it is challenging to accurately determine the actual back of the queue.

At a V/C ratio of 0.9, the proposed approach consistently outperforms the GP-CF method across all metrics, though the difference between single-lane and multi-lane estimations is less pronounced. Notably, some cycles in this scenario experience over-saturation, resulting in residual queues, such as cycles 22 to 28. For each over-saturated cycle, the GP-CF method tends to overestimate the maximum queue length, leading to a rightward shift in the queue profile at the start of the next cycle. In contrast, both the single-lane and multi-lane estimations accurately estimate the queue profile for these cycles, indicating the approach's suitability for over-saturated scenarios.

To gain a deeper understanding of the estimation results, we averaged the absolute errors over multiple cycles within the study period, as illustrated in Fig. 20. As shown, the absolute error during one cycle in both the single-lane and multi-lane estimations follows a similar trend. During the red phase, the absolute error continuously increases from its initial value, reaching a peak and then slightly decreasing. The absolute error increases more rapidly at a V/C ratio of 0.9, primarily due to the residual queue from over-saturated cycles. With the onset of the green light, the absolute error rapidly increases to the maximum value of the entire cycle and then quickly decreases to zero. The GP-CF method shows a slightly different pattern, with the absolute error rising faster during the red phase and peaking much higher during the green phase compared to the proposed approach. This is primarily due to the deviation at the moment when the queue fully clears. Notably, for the scenario with a V/C ratio of 0.9, the GP-CF method's curve shifts significantly to the right, which is consistent with the previously observed rightward shift in queue profile estimation results for over-saturated cycles.

5.3.5. *Traffic signal control application*

To validate the proposed queue length estimation approach in traffic management, this section investigates the effectiveness of offset optimization using arrival distribution information, which can be converted to the queue profile as described in Section 4.3.3. Specifically, we selected the arterial scenario depicted in Fig. 13(a) with the highest traffic volume (V/C ratio of 0.9) while maintaining a consistent 60% match rate. Initially, we tested the multi-lane estimation on the eastbound approach of the target intersection, obtaining lane-specific arrival distribution data for all vehicles. The arrival rate for lane l at time step t, denoted as $\alpha_l^{\text{arrive}}(t)$, is given by

$$\alpha_l^{\text{arrive}}(t) = \sum_{k \in \mathcal{V}_l} P(t \le t_k^{\text{a}} \le t+1) \tag{44}$$

where $\mathcal{V}_l$ represents the set of vehicles departing from lane l.

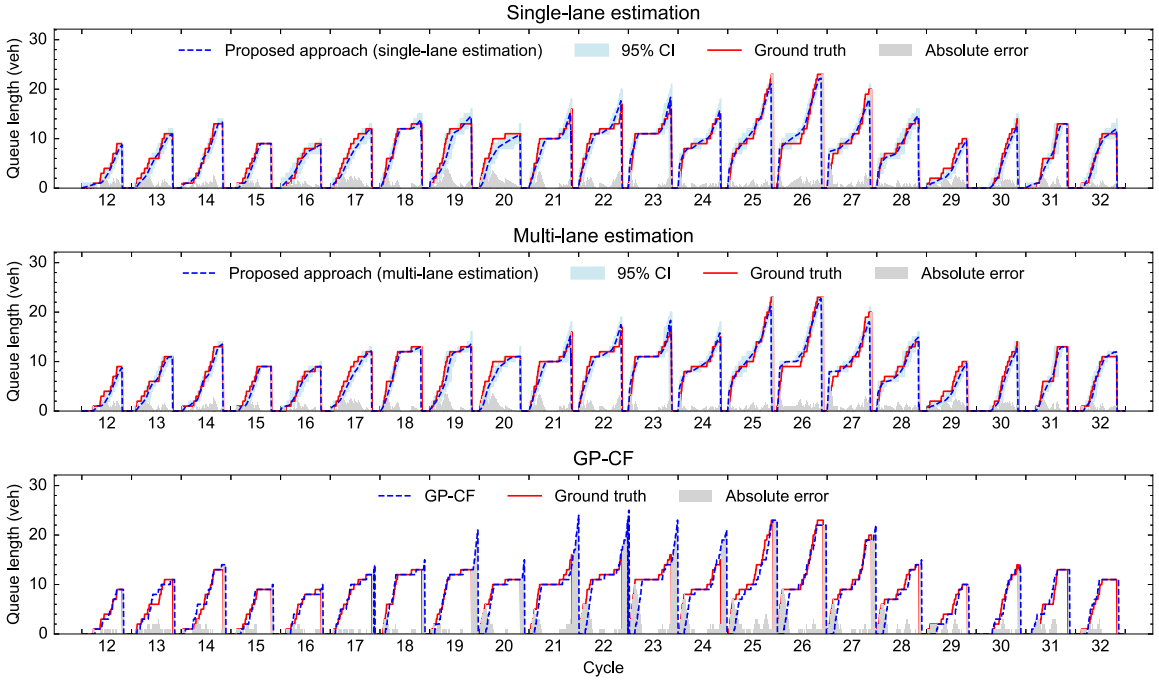

Fig. 19. Partial cycle result of estimated queue profile with a V/C ratio of 0.9.

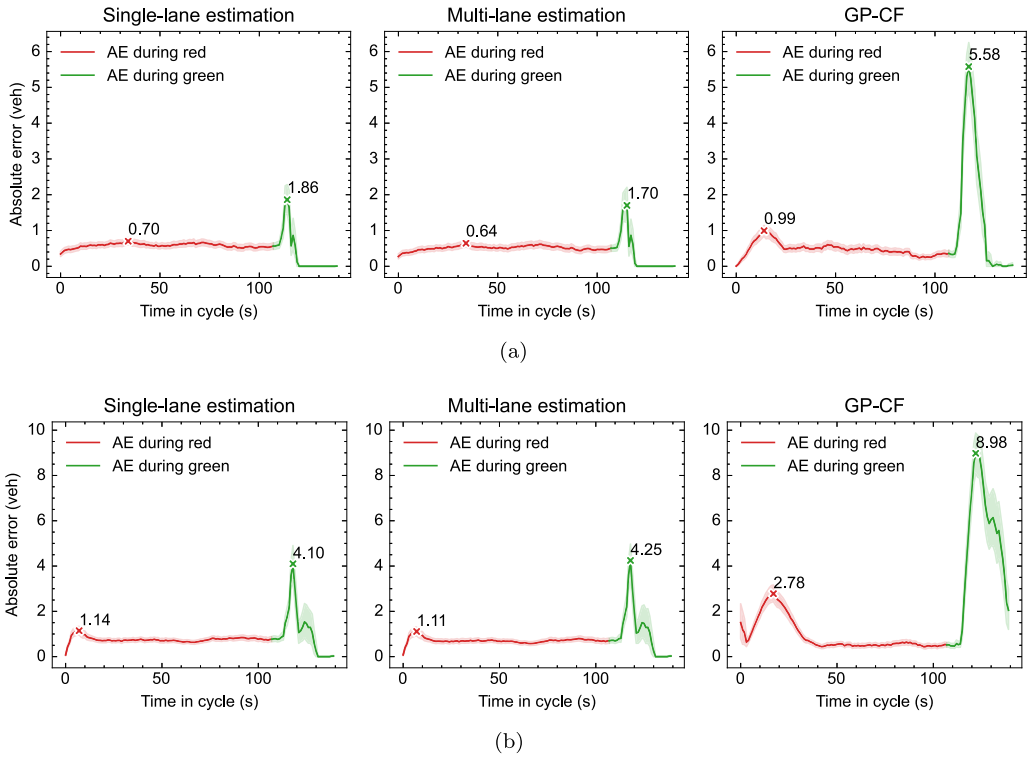

Fig. 20. Error analysis of queue profile estimation, comparing evaluated methods under (a) moderate congestion (V/C = 0.7) and (b) near-saturation conditions (V/C = 0.9).

Subsequently, by adjusting the mainstream direction's offset, we can optimize the control benefits for the target intersection. To ensure a common cycle, the target intersection's cycle length was set to match the upstream intersection (132 s), while maintaining

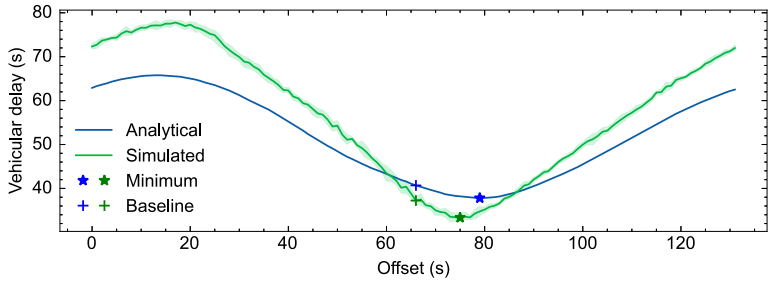

Fig. 21. Comparison of vehicular delay across various offsets.

the duration of phase 3. Without queue length information, the baseline method for setting the offset uses the expected speed along the road segment, calculated as 66 s in this scenario. However, by leveraging lane-based arrival distribution, we can compute the vehicular delay for the eastbound approach under different offsets:

$$d(\phi) = \frac{\sum_{t\in\mathcal{T},l\in\mathcal{L}} q_l(t)}{\sum_{t\in\mathcal{T},l\in\mathcal{L}} \alpha_l^{\text{arrive}}(t)} \tag{45}$$

$$q_l(t+1) = q_l(t) + \alpha_l^{\text{arrive}}(t) - \alpha_l^{\text{depart}}(t) \tag{46}$$

$$\alpha_l^{\text{depart}}(t) = \min\{q_l(t) + \alpha_l^{\text{arrive}}(t), \mu_l u_l(t;\phi)\} \tag{47}$$

where $d(\phi)$ represents the vehicular delay given the offset ϕ. The numerator is calculated using the incremental queue accumulation (IQA) (Strong et al., 2006) method to determine total delay, while the denominator represents the number of vehicles. Here, $\mathcal{T}$ is the set of time steps in the study period, and $\mathcal{L}$ is the set of controlled lanes. $q_l(t)$ and $\alpha_l^{\text{depart}}(t)$ denote the queue length and departure rate at time t, respectively. The departure rate is determined by the number of vehicles that can leave and the traffic signal, with μ_l representing the saturation flow rate for lane l, and $u_l(t;\phi)$ indicating whether lane l is given the right of way at time step t.

Using Eqs. (45) to (47), we determined the optimal offset that minimizes the delay, as illustrated by the analytical curve in Fig. 21. For comparison, we conducted simulations using five different random seeds to traverse the offset values, resulting in the simulated curve shown in the same figure. The analytical curve demonstrates strong agreement with the simulated results, particularly in capturing the general convex shape and the location of the optimal region, though it exhibits slightly more conservative delay estimates throughout the offset range. The analytical curve reaches its minimum delay of 37.81 s at an offset of 79 s, while the corresponding simulated delay is 34.76 s, representing a difference of approximately 8.8%. The simulated curve achieves its minimum delay of 33.35 s at an offset of 75 s, with a 4-s difference in optimal offset values. Using the baseline offset of 66 s, which was calculated based on expected speed, the analytical and simulated delays are 40.68 s and 37.23 s respectively. The offset derived from our analytical approach reduces the delay by 6.63% compared to the baseline while remaining only 4.22% higher than the empirically determined optimal value. These results demonstrate that our queue length estimation approach can effectively support downstream traffic signal control applications, providing a reliable foundation for offset optimization without requiring extensive simulation efforts.

6. Conclusion

This paper presents a novel probabilistic approach for queue length estimation using multi-section LPR data that effectively addresses the limitations of existing methods in multi-lane scenarios. By introducing a conditional probability model for no-delay arrival time (NAT) estimation that incorporates overtaking behaviors, our approach eliminates the need for traditional assumptions such as the FIFO rule and specific arrival processes. The proposed methodology consists of three key components: a DP-based vehicle group partitioning algorithm that reduces computational complexity, an MCMC sampling method for calculating NAT distributions, and a weighted general exact cover problem formulation that extends the estimation to multi-lane scenarios.

Empirical and simulation case studies demonstrated the effectiveness of both single-lane and multi-lane queue length estimation. In the empirical study, the proposed approach significantly outperformed the baseline method, achieving improvements of over 37% across all evaluation metrics, with most estimation errors below one vehicle and maximum errors under four vehicles. Our approach demonstrated strong robustness in two key aspects: first, it maintained reliable performance even with severely reduced matching rates, achieving better accuracy at a 30% matching rate than the baseline method at an 80% matching rate; second, it exhibited predictable linear performance degradation when subjected to miss detection rates up to 20%, while still maintaining practically useful accuracy.

The simulation study further validated these findings and revealed the approach's effectiveness across varying traffic conditions, with V/C ratios ranging from 0.5 to 0.9, and its resilience to FIFO violations. Notably, the multi-lane estimation consistently outperformed single-lane estimation, particularly during red phases and at moderate V/C ratios. Both approaches exhibited a clear performance inflection point at a 60% matching rate, beyond which error reduction became more gradual, suggesting that core traffic

patterns were sufficiently captured at this threshold. Detailed queue profile results confirm the approach's ability to handle oversaturated traffic conditions. The practical applicability of the approach was demonstrated through a traffic signal control application, where the analytically derived optimal offset achieved a 6.63% delay reduction compared to the baseline method, with results closely matching simulation outcomes.

Several promising directions exist for future research: (1) The proposed approach currently assumes dedicated exit lanes, and extending the methodology to accommodate shared lanes represents an important research direction. This is particularly crucial for through-right turn lanes, where the complex departure patterns and uncertain vehicle movements require new probabilistic frameworks. (2) Another key aspect to explore is the impact of geometric design features, specifically the presence of short lanes, on queue formation and dissipation. This would require developing models to quantify and predict queue spillback probability. (3) As LPR data cannot directly provide vehicle trajectories within links, exploring data fusion methods, such as integrating CV data (Tan et al., 2020), can improve the accuracy of queue profile estimation. Furthermore, investigating algorithms for vehicle trajectory reconstruction based on LPR data or the fusion of LPR and CV data, utilizing car-following and lane-changing models, represents another promising direction.

CRediT authorship contribution statement

Lyuzhou Luo: Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Hao Wu:** Writing – review & editing, Validation, Investigation, Data curation. **Jiahao Liu:** Writing – review & editing, Data curation, Conceptualization. **Keshuang Tang:** Writing – review & editing, Supervision, Funding acquisition, Conceptualization. **Chaopeng Tan:** Writing – review & editing, Visualization, Validation, Methodology, Formal analysis, Data curation.

Acknowledgments

This research was sponsored by the National Key Research & Development Program, China (Grant No. 2023YFB4301900) and the National Natural Science Foundation of China (Grant No. 52372319).

Appendix. Log-likelihood function of global match θ

Let θ_j denote a matching for group j, and let $\theta = \{\theta_j | j \in \mathcal{G} \cup \mathcal{G}_u\}$ denote a global matching for a cluster, where $\mathcal{G}$ and $\mathcal{G}_u$ represent the set of constrained and unconstrained groups in the cluster, respectively. Therefore, using the NAT conditions derived from Section 4.3.1, we can formulate a maximum likelihood estimation:

$$\hat{\theta} = \arg\max_{\theta \in \Theta} P\left(\bigcap_{j\in\mathcal{G}} C_j, \bigcap_{j\in\mathcal{G}_u} C_j^u; \theta\right) \tag{48}$$

where C_j represents the NAT conditions for the constrained group j, C_j^u represents the NAT conditions for the unconstrained group j, and Θ denotes all possible global matchings. We assume that the NAT conditions for different constrained groups are mutually independent. Given the NAT conditions for all constrained groups, those for different unconstrained groups are considered mutually conditionally independent. As such, we have

$$\begin{aligned} P\left(\bigcap_{j\in\mathcal{G}} C_j, \bigcap_{j\in\mathcal{G}_u} C_j^u; \theta\right) &= P\left(\bigcap_{j\in\mathcal{G}_u} C_j^u \middle| \bigcap_{j\in\mathcal{G}} C_j; \theta\right) P\left(\bigcap_{j\in\mathcal{G}} C_j; \theta\right) \\ &= \prod_{j\in\mathcal{G}_u} P\left(C_j^u \middle| \bigcap_{j\in\mathcal{G}} C_j; \theta\right) \prod_{j\in\mathcal{G}} P\left(C_j; \theta\right) \end{aligned} \tag{49}$$

Each product term can be calculated using the results from previous MCMC sampling results and Eq. (13):

$$P(C_j; \theta) \approx V_j \frac{1}{N_j} \sum_{i=1}^{N_j} f(\boldsymbol{t}_{j,i}; \theta), \quad j \in \mathcal{G} \tag{50}$$

$$P\left(C_j^u \middle| \bigcap_{j\in\mathcal{G}} C_j; \theta\right) \approx V_j \frac{1}{N_j} \sum_{i=1}^{N_j} f(\boldsymbol{t}_{j,i}; \theta), \quad j \in \mathcal{G}_u \tag{51}$$

where $\boldsymbol{t}_{j,i}$ is the ith sampling point for group j, with each element t_k^a representing vehicle k's NAT. V_j is the volume of the polytope represented by the NAT conditions of group j. N_j is the number of samples of group j. Note that for unconstrained groups, changes in the NAT conditions of adjacent constrained groups can cause changes in their own NAT conditions, theoretically requiring resampling. However, for simplicity, this effect is ignored in this context.

Given the global match θ, the upstream departure times t_k^u for all vehicles $k \in K$ in group j are known, and we have

$$\begin{aligned} f(\boldsymbol{t}_{j,i}; \theta) &= \prod_{k\in K} f_k(t_k^a) \\ &= \prod_{k\in K} g_k(t_k^a - t_k^u) \end{aligned} \tag{52}$$

Therefore, the log-likelihood function corresponding to Eq. (48) can be expressed as

$$l(\theta) = \sum_{j \in \mathcal{G} \cup \mathcal{G}_u} \log \sum_{i=1}^{N_j} f(\boldsymbol{t}_{j,i}; \theta) \tag{53}$$

Since matching θ is only partially related to group j, we have

$$\begin{aligned} l(\theta) &= \sum_{j \in \mathcal{G} \cup \mathcal{G}_u} \log \sum_{i=1}^{N_j} f(\boldsymbol{t}_{j,i}; \theta_j) \\ &= \sum_{j \in \mathcal{G} \cup \mathcal{G}_u} l_j(\theta_j) \end{aligned} \tag{54}$$

From this, it is clear that the global match's log-likelihood function $l(\theta)$ can be represented as the sum of the log-likelihood functions of different groups $l_j(\theta_j)$.

References

An, C., Guo, X., Hong, R., Lu, Z., Xia, J., 2021. Lane-based traffic arrival pattern estimation using license plate recognition data. IEEE Intell. Transp. Syst. Mag. 14 (4), 133–144. http://dx.doi.org/10.1109/MITS.2021.3051489.

Ban, X., Hao, P., Sun, Z., 2011. Real time queue length estimation for signalized intersections using travel times from mobile sensors. Transp. Res. C 19 (6), 1133–1156. http://dx.doi.org/10.1016/j.trc.2011.01.002.

Berry, D.S., Belmont, D.M., et al., 1951. Distribution of vehicle speeds and travel times. In: Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability. Vol. 31, University of California Press, pp. 589–602. http://dx.doi.org/10.1525/9780520411586-044.

Cao, Q., Yuan, J., Ren, G., Qi, Y., Li, D., Deng, Y., Ma, W., 2024. Tracking the source of congestion based on a probabilistic sensor flow assignment model. Transp. Res. C 165, 104736. http://dx.doi.org/10.1016/j.trc.2024.104736.

Chen, Y., Dwivedi, R., Wainwright, M.J., Yu, B., 2018. Fast MCMC sampling algorithms on polytopes. J. Mach. Learn. Res. 19 (55), 1–86.

Cheng, Y., Qin, X., Jin, J., Ran, B., 2012. An exploratory shockwave approach to estimating queue length using probe trajectories. J. Intell. Transp. Syst. 16 (1), 12–23. http://dx.doi.org/10.1080/15472450.2012.639637.

Comert, G., Cetin, M., 2011. Analytical evaluation of the error in queue length estimation at traffic signals from probe vehicle data. IEEE Trans. Intell. Transp. Syst. 12 (2), 563–573. http://dx.doi.org/10.1109/TITS.2011.2113375.

He, Q., Head, K.L., Ding, J., 2012. PAMSCOD: Platoon-based arterial multi-modal signal control with online data. Transp. Res. C 20 (1, SI), 164–184. http://dx.doi.org/10.1016/j.trc.2011.05.007.

Kannan, R., Narayanan, H., 2012. Random walks on polytopes and an affine interior point method for linear programming. Math. Oper. Res. 37 (1), 1–20. http://dx.doi.org/10.1287/moor.1110.0519.

Knuth, D.E., 2000. Dancing links. arXiv Preprint cs/0011047.

Li, M., Tang, J., Chen, Q., Liu, Y., 2023. Traffic arrival pattern estimation at urban intersection using license plate recognition data. Phys. A 625, 128995. http://dx.doi.org/10.1016/j.physa.2023.128995.

Li, F., Tang, K., Yao, J., Li, K., 2017. Real-time queue length estimation for signalized intersections using vehicle trajectory data. Transp. Res. Rec. 2623 (1), 49–59. http://dx.doi.org/10.3141/2623-06.

Liu, H.X., Wu, X., Ma, W., Hu, H., 2009. Real-time queue length estimation for congested signalized intersections. Transp. Res. C 17 (4), 412–427. http://dx.doi.org/10.1016/j.trc.2009.02.003.

Luo, X., Ma, D., Jin, S., Gong, Y., Wang, D., 2019. Queue length estimation for signalized intersections using license plate recognition data. IEEE Intell. Transp. Syst. Mag. 11 (3), 209–220. http://dx.doi.org/10.1109/MITS.2019.2919541.

Ma, D., Luo, X., Jin, S., Guo, W., Wang, D., 2018. Estimating maximum queue length for traffic lane groups using travel times from video-imaging data. IEEE Intell. Transp. Syst. Mag. 10 (3), 123–134. http://dx.doi.org/10.1109/MITS.2018.2842047.

Ma, D., Luo, X., Li, W., Jin, S., Guo, W., Wang, D., 2017. Traffic demand estimation for lane groups at signal-controlled intersections using travel times from video-imaging detectors. IET Intell. Transp. Syst. 11 (4), 222–229. http://dx.doi.org/10.1049/iet-its.2016.0233.

Ma, D., Xiao, J., Song, X., Ma, X., Jin, S., 2021. A back-pressure-based model with fixed phase sequences for traffic signal optimization under oversaturated networks. IEEE Trans. Intell. Transp. Syst. 22 (9), 5577–5588. http://dx.doi.org/10.1109/TITS.2020.2987917.

Mo, B., Li, R., Dai, J., 2020. Estimating dynamic origin–destination demand: A hybrid framework using license plate recognition data. Comput.-Aided Civ. Infrastruct. Eng. 35 (7), 734–752. http://dx.doi.org/10.1111/mice.12526.

Mo, B., Li, R., Zhan, X., 2017. Speed profile estimation using license plate recognition data. Transp. Res. C 82, 358–378. http://dx.doi.org/10.1016/j.trc.2017.07.006.

Noaeen, M., Mohajerpoor, R., Far, B.H., Ramezani, M., 2021. Real-time decentralized traffic signal control for congested urban networks considering queue spillbacks. Transp. Res. C 133, http://dx.doi.org/10.1016/j.trc.2021.103407.

Ramezani, M., Geroliminis, N., 2015. Queue profile estimation in congested urban networks with probe data. Comput.-Aided Civ. Infrastruct. Eng. 30 (6), 414–432. http://dx.doi.org/10.1111/mice.12095.

Rao, W., Wu, Y.-J., Xia, J., Ou, J., Kluger, R., 2018. Origin-destination pattern estimation based on trajectory reconstruction using automatic license plate recognition data. Transp. Res. C 95, 29–46. http://dx.doi.org/10.1016/j.trc.2018.07.002.

Sharma, A., Bullock, D.M., Bonneson, J.A., 2007. Input–output and hybrid techniques for real-time prediction of delay and maximum queue length at signalized intersections. Transp. Res. Rec. 2035 (1), 69–80. http://dx.doi.org/10.3141/2035-08.

Skabardonis, A., Geroliminis, N., 2008. Real-time monitoring and control on signalized arterials. J. Intell. Transp. Syst. 12 (2), 64–74. http://dx.doi.org/10.1080/15472450802023337.

Strong, D.W., Nagui, R.M., Courage, K., 2006. New calculation method for existing and extended HCM delay estimation procedure. In: Proceedings of the 87th Annual Meeting Transportation Research Board. Transportation Research Board, Washington, DC, pp. 06–0106.

Tan, C., Cao, Y., Ban, X., Tang, K., 2024a. Connected vehicle data-driven fixed-time traffic signal control considering cyclic time-dependent vehicle arrivals based on cumulative flow diagram. IEEE Trans. Intell. Transp. Syst. http://dx.doi.org/10.1109/TITS.2024.3360090.

Tan, C., Ding, Y., Yang, K., Zhu, H., Tang, K., 2024b. Connected vehicle data-driven robust optimization for traffic signal timing: Modeling traffic flow variability and errors. arXiv preprint arXiv:2406.14108.

Tan, C., Liu, L., Wu, H., Cao, Y., Tang, K., 2020. Fuzing license plate recognition data and vehicle trajectory data for lane-based queue length estimation at signalized intersections. J. Intell. Transp. Syst. 24 (5), 449–466. http://dx.doi.org/10.1080/15472450.2020.1732217.

Tan, C., Wu, H., Tang, K., Tan, C., 2022a. An extendable gaussian mixture model for lane-based queue length estimation based on license plate recognition data. J. Adv. Transp. 2022, e5119209. http://dx.doi.org/10.1155/2022/5119209.

Tan, C., Yang, K., 2024. Privacy-preserving adaptive traffic signal control in a connected vehicle environment. Transp. Res. C 158, 104453. http://dx.doi.org/10.1016/j.trc.2023.104453.

Tan, C., Yao, J., Ban, X., Tang, K., 2022b. Cumulative flow diagram estimation and prediction based on sampled vehicle trajectories at signalized intersections. IEEE Trans. Intell. Transp. Syst. 23 (8), 11325–11337. http://dx.doi.org/10.1109/TITS.2021.3102750.

Tan, C., Yao, J., Tang, K., Sun, J., 2021. Cycle-based queue length estimation for signalized intersections using sparse vehicle trajectory data. IEEE Trans. Intell. Transp. Syst. 22 (1), 91–106. http://dx.doi.org/10.1109/TITS.2019.2954937.

Tang, K., Cao, Y., Chen, C., Yao, J., Tan, C., Sun, J., 2021. Dynamic origin-destination flow estimation using automatic vehicle identification data: A 3D convolutional neural network approach. Comput.-Aided Civ. Infrastruct. Eng. 36 (1), 30–46. http://dx.doi.org/10.1111/mice.12559.

Tang, K., Wu, H., Yao, J., Tan, C., Ji, Y., 2022. Lane-based queue length estimation at signalized intersections using single-section license plate recognition data. Transp. B: Transp. Dyn. 10 (1), 293–311. http://dx.doi.org/10.1080/21680566.2021.1991504.

Telgen, J., 1983. Identifying redundant constraints and implicit equalities in systems of linear constraints. Manage. Sci. 29 (10), 1209–1222. http://dx.doi.org/10.1287/mnsc.29.10.1209.

Tiaprasert, K., Zhang, Y., Ye, X., 2018. Platoon recognition using connected vehicle technology. J. Intell. Transp. Syst. 23 (1), 12–27. http://dx.doi.org/10.1080/15472450.2018.1476146.

Vaidya, P.M., 1989. A new algorithm for minimizing convex functions over convex sets. In: 30th Annual Symposium on Foundations of Computer Science. pp. 338–343. http://dx.doi.org/10.1109/SFCS.1989.63500.

Vigos, G., Papageorgiou, M., Wang, Y., 2008. Real-time estimation of vehicle-count within signalized links. Transp. Res. C 16 (1), 18–35. http://dx.doi.org/10.1016/j.trc.2007.06.002.

Wu, X., Liu, H.X., Gettman, D., 2010. Identification of oversaturated intersections using high-resolution traffic signal data. Transp. Res. C 18 (4), 626–638. http://dx.doi.org/10.1016/j.trc.2010.01.003.

Wu, H., Luo, L., Oguchi, T., Tang, K., Zhu, H., 2024. Stochastic queue profile estimation using license plate recognition data. Phys. A 643, 129790. http://dx.doi.org/10.1016/j.physa.2024.129790.

Wu, H., Yao, J., Liu, L., Cao, Y., Tang, K., 2019. Left-turn spillback identification based on license plate recognition data. In: Proceedings of the 98th Annual Meeting Transportation Research Board. Transportation Research Board, Washington, DC.

Yang, J., Sun, J., 2015. Vehicle path reconstruction using automatic vehicle identification data: An integrated particle filter and path flow estimator. Transp. Res. C 58, 107–126. http://dx.doi.org/10.1016/j.trc.2015.07.003.

Yao, Z., Jiang, Y., Zhao, B., Luo, X., Peng, B., 2020. A dynamic optimization method for adaptive signal control in a connected vehicle environment. J. Intell. Transp. Syst. 24 (2), 184–200. http://dx.doi.org/10.1080/15472450.2019.1643723.

Yin, J., Chen, P., Tang, K., Sun, J., 2021. Queue intensity adaptive signal control for isolated intersection based on vehicle trajectory data. J. Adv. Transp. 2021 (1), 8838922. http://dx.doi.org/10.1155/2021/8838922.

Zhan, X., Li, R., Ukkusuri, S.V., 2015. Lane-based real-time queue length estimation using license plate recognition data. Transp. Res. C 57, 85–102. http://dx.doi.org/10.1016/j.trc.2015.06.001.

Zhan, X., Li, R., Ukkusuri, S.V., 2020. Link-based traffic state estimation and prediction for arterial networks using license-plate recognition data. Transp. Res. C 117, 102660. http://dx.doi.org/10.1016/j.trc.2020.102660.

Zhang, H., Liu, H.X., Chen, P., Yu, G., Wang, Y., 2020. Cycle-based end of queue estimation at signalized intersections using low-penetration-rate vehicle trajectories. IEEE Trans. Intell. Transp. Syst. 21 (8), 3257–3272. http://dx.doi.org/10.1109/TITS.2019.2925111.