



Delft University of Technology

Document Version

Final published version

Licence

CC BY

Citation (APA)

Gao, T., Isufi, E., Daamen, W., Smits, E. S., & Hoogendoorn, S. (2026). Q-Net: Queue length estimation via Kalman-based neural networks. *Transportation Research Part C: Emerging Technologies*, 190, Article 105809. <https://doi.org/10.1016/j.trc.2026.105809>

Important note

To cite this publication, please use the final published version (if applicable). Please check the document version above.

Copyright

In case the licence states "Dutch Copyright Act (Article 25fa)", this publication was made available Green Open Access via the TU Delft Institutional Repository pursuant to Dutch Copyright Act (Article 25fa, the Taverne amendment). This provision does not affect copyright ownership. Unless copyright is transferred by contract or statute, it remains with the copyright holder.

Sharing and reuse

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights. We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.



Q-Net: Queue length estimation via Kalman-based neural networks

Ting Gao ^{a,*}, Elvin Isufi ^b, Winnie Daamen ^a, Erik-Sander Smits ^c,
Serge Hoogendoorn ^a

^a Department of Transport and Planning, Faculty of Civil Engineering and Geosciences, Delft University of Technology, 2628 CN, Delft, the Netherlands

^b Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Delft, 2628 CD, the Netherlands

^c Arane Adviseurs in verkeer en vervoer, Gouda, 2805 TD, the Netherlands

ARTICLE INFO

Keywords:

Queue estimation
Kalman filter
Deep learning

ABSTRACT

Estimating queue lengths at signalized intersections is a long-standing challenge in traffic management. Partial observability of vehicle flows complicates this task despite the availability of two privacy-preserving data sources: (i) aggregated vehicle counts from loop detectors near stop lines, and (ii) aggregated floating car data (aFCD) that provide segment-wise average speed measurements. However, how to integrate these sources with differing spatial and temporal resolutions for queue length estimation is rather unclear. Addressing this question, we present Q-Net: a queue estimation framework built upon a state-space formulation. This design addresses key challenges in queue modeling, such as violations of traffic conservation assumptions. Q-Net follows the Kalman predict-update structure and maintains physical interpretability in both the state evolution and measurement models. Q-Net uses an AI-augmented Kalman filter to learn time-varying gain dynamics from data. The framework supports real-time implementation and improves spatial transferability by grouping aFCD measurements into fixed-size local groups, making the number of learnable parameters independent of section length. Evaluations on urban main roads in Rotterdam, the Netherlands, show that Q-Net outperforms baseline methods, tracks queue formation and dissipation accurately, and mitigates aFCD-induced delays. By combining data efficiency, interpretability, real-time applicability, and spatial transferability, Q-Net makes accurate queue length estimation possible without costly sensing infrastructure like cameras or radar.

1. Introduction

Traffic queues raise operational, safety, and environmental concerns that require effective monitoring and control. Urban congestion reduces network throughput and limits agglomeration economies (Sweet, 2011). Long queues also increase accident likelihood (Pasquale et al., 2018), elevate noise levels (Chevallier et al., 2009), contribute to driver stress and mental health issues (Nadrian et al., 2019), and generate localized air pollution (Matzoros and Van Vliet, 1992). Queue length estimation helps address these issues by enabling real-time adaptive signal control and, more broadly, supporting the development of queue-based control algorithms and

* Corresponding author.

E-mail addresses: t.gao-1@tudelft.nl (T. Gao), e.isufi-1@tudelft.nl (E. Isufi), w.daamen@tudelft.nl (W. Daamen), e.smits@arane.nl (E.-S. Smits), s.p.hoogendoorn@tudelft.nl (S. Hoogendoorn).

<https://doi.org/10.1016/j.trc.2026.105809>

Received 27 November 2025; Received in revised form 5 June 2026; Accepted 9 June 2026

Available online 17 June 2026

0968-090X/© 2026 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

network-wide optimization strategies (Li et al., 2021; Hoogendoorn et al., 2016). However, accurate estimation remains difficult because of incomplete sensor coverage, limited data availability, unpredictable driver behavior, and complex network layouts.

Queue length estimation at signalized intersections relies on either Lagrangian or Eulerian input data. Lagrangian data are collected from mobile sensors moving with traffic flow, mainly GPS-equipped probe vehicles. Recent studies have used probe trajectories for queue estimation through methods ranging from shockwave theory to probabilistic likelihood-based approaches (Cheng et al., 2011; Hu et al., 2022; Zhao et al., 2019; Cheng et al., 2012; Zhao et al., 2021). These methods often require prior knowledge of probe vehicle penetration rates and may raise privacy concerns because individual positions are shared with external monitoring systems (Rose, 2006). In contrast, Eulerian data capture traffic conditions over fixed spatial segments, coming from either stationary sensors or aggregated mobile sources. Computer vision systems can detect queue ends (Umair et al., 2021; Albiol et al., 2011), but their performance may degrade under poor lighting or bad weather. Radar systems can provide direct queue measurements, but are expensive to install and maintain (Bernas et al., 2018).

This study focuses on two readily available Eulerian data sources: loop detector data and aggregated floating car data (aFCD). Loop detectors remain widely used because they provide accurate vehicle counts, occupancy, and speed measurements, and extensive detector infrastructure already exists in many cities (MÜCK, 2002; Liu et al., 2009; Geroliminis and Skabardonis, 2011; Abewickrema et al., 2023). However, they provide point-based observations and cannot directly capture the queue length, especially under incomplete detector coverage. aFCD complements loop detectors by providing segment-level speed information. It is derived from anonymized mobile observations aggregated over road segments and time intervals, without requiring high penetration rates (van der Loop et al., 2017; Altuntaşı, 2018; Mobiliteitsscanner). The key challenge is to integrate these heterogeneous data, with different spatial and temporal resolutions, into a unified and principled queue estimation framework.

1.1. Literature review

Queue length estimation at signalized intersections can be categorized into four approaches: input-output models (Webster, 1958; Sharma et al., 2007), shockwave theory (Lighthill and Whitham, 1955; Liu et al., 2009), speed-based methods (Dinh et al., 2014), and filtering algorithms (Yin et al., 2018).

Input-output models estimate queue length based on the conservation equation of cumulative traffic inflow and outflow at signalized sections. Early studies used vehicle counts from road detectors to characterize the queuing process (Webster, 1958), and later work incorporated real-time flow and occupancy measurements to adjust queue estimates (Vigos et al., 2008). While input-output models are straightforward, they do not capture the spatiotemporal dynamics of queues and are sensitive to measurement errors. Recent studies have improved them by integrating probe vehicle data, enabling real-time queue-tail estimation (Wang et al., 2021; Tang et al., 2023); however, they require sufficient probe penetration.

Shockwave theory describes queue formation and dissipation, with shockwave speed indicating traffic state transitions. Liu et al. (2009) used loop detector data for cycle-wise queue estimation, later extended by Abewickrema et al. (2023) to both undersaturated and oversaturated conditions. These methods identify occupancy breakpoints across signal phases, but such patterns can be obscured by noise and traffic heterogeneity. To address this, studies have fused probe vehicle data with detector measurements to estimate shockwave speeds and derive cycle-wise maximum and residual queues (Cai et al., 2014; Wang et al., 2017). An et al. (2018) proposed a hybrid method that applies input-output models when breakpoints are unreliable and shockwave analysis otherwise. Although shockwave-based models offer a theoretical basis for queue evolution, they assume uniform traffic conditions, which rarely hold in urban settings.

Speed-based methods estimate queue tails by analyzing speed patterns across road segments over time, typically using floating car data (FCD). Some methods detect queue tails from spatial speed drops in probe FCD (Dinh et al., 2014). Because aFCD provides segment-level average speeds, Altuntaşı (2018) generated synthetic probe trajectories using four-node quadratic interpolation and constructed iso-speed contours to estimate queue length. van Erp et al. (2018) combined aFCD with loop detector data to estimate vehicle accumulation. While such speed-based methods are intuitive and data-efficient, their accuracy depends heavily on data quality and congestion speed thresholds. More recently, Luo et al. (2022) avoided fixed speed thresholds by combining aFCD, loop detector, and traffic light event data within a decision-tree and RPROP-NN framework. However, such neural network-based methods require section-specific training and act as black boxes, limiting transferability and interpretability.

Kalman filtering algorithms use state-space models with a hidden state evolution equation and a measurement equation. The hidden state represents queue-related variables that evolve over time but are not directly observed, and different data sources can be integrated into either the evolution or measurement model. The recursive prediction-update structure of Kalman filtering enables real-time tracking of queue dynamics. Several studies have applied this framework to queue estimation. Abewickrema et al. (2023) used an extended Kalman filter in which cycle-wise maximum and residual queues are modeled as states that remain static within each signal cycle, with a shockwave-based measurement model capturing phase breakpoints. Hu et al. (2022) used two independent extended Kalman filters to track queue formation and discharge waves, with measurements derived from shockwave location and speed. Wang et al. (2021) applied a standard Kalman filter in which the hidden state is the number of queued vehicles; its evolution model follows input-output theory, while its measurement model accounts for probe vehicle penetration. Similarly, Heshami and Kattan (2021) used an input-output evolution model combined with a case-based reasoning measurement model. Despite their flexibility, traditional Kalman filters rely on linear or linearized system models and require full prior knowledge of system dynamics, including noise covariances. However, these assumptions are often unrealistic in urban traffic. For example, noise can be complex, unknown, and time-varying, which degrades model performance (Li et al., 2015). Queue dynamics are inherently nonlinear and heterogeneous (Ahmed et al., 2021; Chauhan et al., 2021), and become even more difficult to model under partial observability.

Table 1
Overview of data sources and their characteristics.

Input Data		Benchmark Data	
Data Source	Loop Detector	aFCD	Radar
Traffic Variable	Vehicle Count	Average Speed	Queue Length
Spatial Coverage	Lane-level at stop lines (upstream and downstream)	Section segments	Section-level
Temporal Resolution	10 s	60 s	10 s

1.2. Research gap and contribution

Current queue length estimation approaches face three main limitations. First, many studies are developed and validated in well-controlled settings, overlooking partial observability of traffic flows. In urban networks, vehicles may enter or leave the study area through adjacent roads or unsignalized intersections without being monitored, due to sensor placement constraints or failures (Federal Highway Administration, 2006; Shang et al., 2024). Second, existing methods often depend strongly on specific data conditions. Loop-detector-based methods built on input-output or shockwave theory require accurate boundary counts or clear occupancy patterns, which are frequently unmet in practice. Meanwhile, aFCD, despite its low cost, wide coverage, and privacy advantages, remains underutilized for queue estimation. As a result, many methods generalize poorly across imperfect real-world sensor settings. Third, traditional and extended Kalman filters rely on linear or linearized models and predefined noise covariances, assumptions that are often unrealistic under dynamic intersection traffic, limiting adaptability and scalability.

To address these limitations, we propose Q-Net, a queue length estimation approach tailored to urban environments. Q-Net is built upon a state-space modeling approach integrated with an AI-augmented Kalman filtering framework. It handles partially observed traffic flows, uses widely available and privacy-preserving loop detector and aFCD inputs, and preserves interpretability through physically meaningful state evolution and measurement models. To overcome the limitations of traditional Kalman filters, Q-Net learns the Kalman gain dynamically using neural networks, allowing it to capture time-varying and nonlinear noise characteristics. This learning process builds on the KalmanNet principle (Revach et al., 2022), which has gained attention in signal processing through several recent variants (Buchnik et al., 2023; Revach et al., 2023; Choi et al., 2023).

The main contributions of this paper are summarized as follows:

- 1. Interpretable state-space formulation under partial observability:** We develop a state-space model for queue length estimation using only loop detector and aFCD data. Unlike approaches that assume fully observed boundary flows, we explicitly address partial observability: our evolution model separates high-frequency queue dynamics from slow estimation drift caused by unmonitored flows. We also design a nonlinear measurement function that maps queue length to expected trajectory-based macroscopic speeds, enabling physics-informed fusion of data sources with different spatial and temporal resolutions.
- 2. Spatial transferability via AI-augmented filtering:** Q-Net balances modeling capacity and interpretability by using neural networks to compute the Kalman gain while maintaining physically meaningful states. To support spatial transferability, we introduce a local measurement grouping strategy considering that measurement covariances mainly depend on neighboring aFCD speeds. This makes the learnable parameter dimension independent of road section length, allowing Q-Net to be trained on one section and tested on another.
- 3. Real-world validation:** Q-Net is validated on main urban road sections in Rotterdam, the Netherlands. It outperforms several baseline models, transfers to adjacent road sections without fine-tuning, and maintains reliable performance under different traffic patterns and infrastructure configurations. Its online version further demonstrates potential for real-time adaptive traffic control. To our knowledge, this is the first application of AI-augmented Kalman filtering to queue length estimation.

The remainder of this paper is organized as follows. Section 2 defines the problem setup and challenges. Section 3 reviews Kalman filter fundamentals. Sections 4 and 5 present the Q-Net state-space formulation and filtering procedure, respectively. Section 6 reports the Rotterdam case study. Finally, Section 7 discusses limitations and future directions, and Section 8 concludes the paper.

2. Problem setup and challenges

In this study, a *lane* refers to a portion of the roadway used by a single line of vehicles. A *section* is a group of lanes that share the same travel direction. As illustrated in Fig. 1, a lane may accommodate multiple movement directions, and a section typically contains lanes that diverge into different movements at downstream intersections.

We estimate queue length at signalized intersections using indirect data sources, since direct queue measurement is often costly or impractical. The focus is section-wise queue length, defined as the maximum queuing distance across all lanes within a section. Consistent with Ramezani and Geroliminis (2015), a queue is not restricted to vehicles fully stopped at zero speed. Instead, we define it macroscopically as the spatial length of the section operating in a congested traffic state. In our formulation, the unqueued portion operates at free-flow speed (v_f), while the queued portion operates at a representative low speed (v_j). Since v_j represents the average speed of the congested state, this definition includes both stopped vehicles and vehicles moving slowly in stop-and-go waves.

The problem involves three data sources, summarized in Table 1. Loop detectors installed near stop lines record vehicle passages, aggregated into 10-s vehicle counts. aFCD divides each road section into segments and provides average speed for each segment at

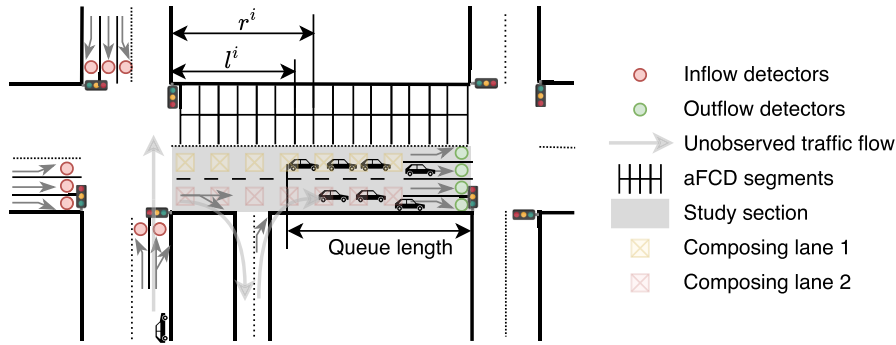


Fig. 1. Layout of a signalized intersection and the section-wise queue length estimation setup. Segment i extends from position l^i to r^i , measured as distances from the stop line. Loop detectors provide inflow/outflow counts, while aFCD provides segment-level speed measurements. The figure also illustrates partial observability caused by multi-directional upstream lanes and unmonitored entering or leaving flows.

Table 2
Overview of notation, descriptions, and units.

Notation	Description	Unit
t	Discrete time index	–
x_t	State variable at time t : section-level queue length	m
u_t	Control input at time t : estimated queue change	m
y_t	Measurement vector at time t (aFCD speeds)	m/s
y_t^i	Speed measurement for i th section segment at time t	m/s
N	Number of aFCD segments within the section	–
w_t, Q_t	Process noise and its variance matrix	m, m ²
v_t, R_t	Measurement noise and its covariance matrix	m/s, (m/s) ²
h	Measurement model function	–
L_t^q	Recovered queue length using only cumulative vehicle counts	m
Σ_t	Variance of state variable x_t	m ²
S_t	Covariance matrix of measurements y_t	(m/s) ²
λ_c	Unobserved traffic flow rate	vehicles/s
m	Number of lanes in the section	–
L	Length of the road section	m
A_t	Cumulative inflow (vehicle arrivals) count from upstream	vehicles
D_t	Cumulative outflow (vehicle departures) count from the section	vehicles
k_j, k_f	Jam density and free-flow density	vehicles/m
v_j, v_f	Jam speed and free-flow speed	m/s
$Q_{\max}$	Maximum reachable queue length for the section	m
$\hat{x}_{t t}$	Posteriori estimate of x_t given measurements up to time t	m
$\hat{y}_{t t-1}$	Predicted measurement at time t based on state at time $t-1$	m/s
l^i, r^i	Start and end positions of segment i measured from the stop line	m
$\Delta \hat{x}_t$	Forward evolution difference	m
$\Delta \hat{x}_t$	Forward update difference	m
$\Delta \hat{y}_t$	Measurement difference	m/s
Δy_t	Innovation difference	m/s
$\mathcal{K}_t(\theta)$	Kalman gain parameterized by θ	–

1-min intervals. Radar sensors provide section-level queue length estimates at 10-s intervals. The radar data, hereafter referred to as the benchmark data, are used for model training and validation. Although treated as benchmark observations, they may still contain sensor-specific measurement noise.

Under this setup, queue length estimation faces three challenges. First, traffic flows are only partially observed. Fig. 1 illustrates two sources of unobserved vehicle movements: (i) vehicles from multi-directional upstream lanes that do not enter the downstream study section, and (ii) vehicles entering or leaving through unmonitored neighboring roads or facilities. Unmonitored entering flows are typically minor, whereas unmonitored exiting flows can be substantial, making direct aggregation of detector counts unreliable. Second, the data sources differ in spatial granularity and temporal resolution, as summarized in Table 1. Third, all data sources are subject to measurement noise: loop detectors may miscount due to calibration errors or occlusion, aFCD speeds may be affected by reporting delays and aggregation artifacts, and radar may mismeasure queues when sensor limits are exceeded.

These challenges motivate a state-space estimation framework that can combine indirect measurements with explicit uncertainty modeling. Before presenting Q-Net, the next section briefly reviews the Kalman filtering principles on which it builds.

3. Background in Kalman filtering

State-space models describe how a system's latent state and measurements (observations) evolve across time steps. Let $\mathbf{x}_t$ denote the latent state of the system at time t , $\mathbf{u}_t$ the control input, and $\mathbf{y}_t$ the observed measurement. The system is governed by a state evolution and a measurement (observation) model.

(1) State evolution describes how the state $\mathbf{x}_t$ evolves over time:

$$\mathbf{x}_t = \mathbf{F}_t \mathbf{x}_{t-1} + \mathbf{G}_t \mathbf{u}_t + \mathbf{w}_t, \quad (1)$$

where $\mathbf{F}_t$ is the state transition matrix, $\mathbf{G}_t$ is the control input matrix, and $\mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_t)$ is zero-mean Gaussian process noise.

(2) Measurement model relates the observed measurement to the current state:

$$\mathbf{y}_t = \mathbf{H}_t \mathbf{x}_t + \mathbf{v}_t, \quad (2)$$

where $\mathbf{H}_t$ is the observation matrix and $\mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_t)$ is zero-mean Gaussian measurement noise. The noise $\mathbf{w}_t$ and $\mathbf{v}_t$ are mutually independent and also independent of both the states and the measurements. The initial state $\mathbf{x}_0$ is Gaussian with known mean $\hat{\mathbf{x}}_0$ and covariance matrix $\hat{\Sigma}_0$.

The Kalman filter recursively estimates the *a posteriori* distribution of the latent state $\mathbf{x}_t$ given all control inputs and measurements up to time t . Let Σ_t and S_t denote the state and measurement covariance matrices, respectively. At each time step, the filter alternates between prediction and update.

Prediction. The filter first computes the *a priori* state estimate $\hat{\mathbf{x}}_{t|t-1}$ and covariance $\Sigma_{t|t-1}$ from the previous *a posteriori* estimate $\hat{\mathbf{x}}_{t-1|t-1}$ and $\Sigma_{t-1|t-1}$:

$$\hat{\mathbf{x}}_{t|t-1} = \mathbf{F}_t \hat{\mathbf{x}}_{t-1|t-1} + \mathbf{G}_t \mathbf{u}_t, \quad (3)$$

$$\Sigma_{t|t-1} = \mathbf{F}_t \Sigma_{t-1|t-1} \mathbf{F}_t^\top + \mathbf{Q}_t, \quad (4)$$

where $(\cdot)^\top$ denotes the matrix transpose operation. The predicted measurement is:

$$\hat{\mathbf{y}}_{t|t-1} = \mathbf{H}_t \hat{\mathbf{x}}_{t|t-1}, \quad (5)$$

$$\mathbf{S}_{t|t-1} = \mathbf{H}_t \Sigma_{t|t-1} \mathbf{H}_t^\top + \mathbf{R}_t. \quad (6)$$

Update. Using the new measurement $\mathbf{y}_t$, the filter updates the state estimate:

$$\Delta \mathbf{y}_t = \mathbf{y}_t - \hat{\mathbf{y}}_{t|t-1}, \quad (7)$$

$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathcal{K}_t \Delta \mathbf{y}_t, \quad (8)$$

$$\mathcal{K}_t = \Sigma_{t|t-1} \mathbf{H}_t^\top \mathbf{S}_{t|t-1}^{-1}, \quad (9)$$

$$\Sigma_{t|t} = \Sigma_{t|t-1} - \mathcal{K}_t \mathbf{S}_{t|t-1} \mathcal{K}_t^\top. \quad (10)$$

The innovation $\Delta \mathbf{y}_t$ is the prediction error between the observed measurement $\mathbf{y}_t$ and its one-step-ahead prediction $\hat{\mathbf{y}}_{t|t-1}$. The Kalman gain $\mathcal{K}_t$ weights this innovation and, under a linear state-space model with fixed Gaussian noise, minimizes the mean squared error of the state estimate. This recursive prediction-update structure is the basis for online state estimation. For nonlinear systems, the Extended Kalman Filter applies the same principle by locally linearizing the state and measurement models using Jacobian matrices (Maybeck, 1982).

4. Q-Net state-space formulation

Let $x_t \in \mathbb{R}$ be the section-level queue length at time step t , $u_t \in \mathbb{R}$ the queue change as the control input, and $\mathbf{y}_t = [y_t^1, y_t^2, \dots, y_t^N]^\top \in \mathbb{R}^N$ the segment-wise aFCD speed measurements over N section segments. Q-Net adopts the following state-space model:

$$x_t = x_{t-1} + u_t + w_t, \quad (11)$$

$$\mathbf{y}_t = \mathbf{h}(x_t) + \mathbf{v}_t. \quad (12)$$

The evolution Eq. (11) models the current queue length as the previous queue length plus the queue change derived from *loop detector counts* and process noise. The measurement Eq. (12) relates the queue length to *aFCD speeds* through a nonlinear mapping $\mathbf{h}(\cdot)$. Here, $w_t \sim \mathcal{N}(0, Q_t)$ denotes Gaussian process noise, and $\mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_t)$ denotes Gaussian measurement noise. The derivation of u_t is given in Section 4.1, and the measurement function $\mathbf{h}(\cdot)$ is detailed in Section 4.2.

We do not treat noise covariances as fixed or known; instead, Q_t and $\mathbf{R}_t$ are modeled as time-varying quantities, whose effects on the Kalman gain are learned from data. The state variance and measurement covariance are denoted respectively by Σ_t and S_t . Table 2 summarizes the notation used throughout the paper.

4.1. Control queue change input

This subsection derives the control input u_t in Eq. (11). The approach first reconstructs a queue-length proxy from *cumulative vehicle counts* and then uses its temporal differences as queue changes.

Let A_t and D_t denote cumulative vehicle inflow and outflow up to time t , and let L_t^q denote the reconstructed queue length. The control input is obtained as:

$$L_t^q = C(A_t - D_t), \quad (13)$$

$$\bar{L}_t^q = \mathcal{F}(L_t^q), \quad (14)$$

$$u_t = \bar{L}_t^q - \bar{L}_{t-1}^q, \quad (15)$$

where C reconstructs L_t^q from vehicle count data while accounting for unobserved flows, $\mathcal{F}$ applies a Fourier band-pass filter to produce the smoothed queue length $\bar{L}_t^q$, and u_t is computed as the temporal difference of $\bar{L}_t^q$. Applying correction and filtering before differencing yields a noise-robust queue-change input for Eq. (11).

Step 1: Queue length recovery $C(A_t - D_t)$. Let λ_c be the unobserved net traffic flow rate, defined as the net number of vehicles leaving the section per unit time without passing through the designated outflow detector, partially offset by unobserved minor inflow. We model a section of length L with m lanes and consider approximately balanced queues across lanes. The queued portion operates at jam density k_j and speed v_j , while the remaining portion operates at free-flow density k_f and speed v_f . Applying macroscopic vehicle conservation gives:

$$mL_t^q k_j + m(L - L_t^q)k_f = A_t - D_t - \lambda_c t. \quad (16)$$

The left-hand side represents the total amount of vehicles in the queued and free-flow portions of the section, while the right-hand side represents net accumulation adjusted for unobserved flows. Solving for L_t^q yields:

$$L_t^q = \frac{A_t - D_t - m L k_f - \lambda_c t}{m(k_j - k_f)}. \quad (17)$$

Step 1.1: Estimate λ_c . Since $A_t - D_t$ is obtained from loop detector counts, the remaining unknown governing the evolution of L_t^q is λ_c . We estimate it using boundary conditions at the beginning t_0 (e.g., 6:00 AM) and end t_T (e.g., 8:00 PM) of each daily analysis period. This relies on the physical assumption that queues are negligible ($L_t^q \approx 0$) at these pre-peak (t_0) and post-peak (t_T) times.

Applying these conditions yields: $\lambda_c \approx \frac{(A_{t_T} - D_{t_T}) - (A_{t_0} - D_{t_0})}{t_T - t_0}$.

Step 1.2: Estimate L_t^q . Eq. (17) shows that L_t^q is an affine transformation of $(A_t - D_t - \lambda_c t)$. Rather than estimating k_j and k_f , which can be time-varying and site-specific, we directly shift and scale this signal to the empirically observed queue range $[0, Q_{\max}]$, where $Q_{\max}$ is the section-specific maximum reachable queue length. Fig. 2a shows the resulting reconstructed queue length L_t^q .

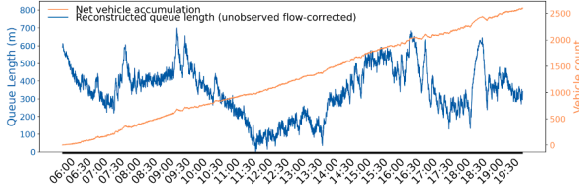
Step 2: Fourier band-pass filtration $\mathcal{F}(L_t^q)$. As shown in Fig. 2a, the reconstructed queue length L_t^q may contain both low-frequency drift and high-frequency noise. The drift can arise from time-varying unobserved flow rate λ_c , such as changes in turning ratios, while high-frequency noise may originate from detector errors. We apply a Fourier-based band-pass filter $\mathcal{F}(\cdot)$ to suppress both components and retain medium-frequency variations that better represent queue buildup and dissipation. Fig. 2b shows the filtered profile $\bar{L}_t^q = \mathcal{F}(L_t^q)$, from which u_t is computed. Because $\bar{L}_t^q$ is used only to derive the control input, its *absolute magnitude is less important than its temporal variation*. Thus, even if the reconstructed queue levels have similar average values in the morning and afternoon, this has limited impact as long as the relative queue changes are preserved.

Remark on oversaturation and spillback. The queue evolution model relies on measured boundary flows rather than expected phase capacities. If downstream congestion limits green-phase discharge, the downstream detector records the reduced outflow; if spillback blocks upstream entry, the upstream detector records zero inflow. When the section is fully congested and lacks a free-flow portion, the aFCD measurement model reports aggregated speeds near the congested speed v_j . Since Q-Net models the section as an aggregated storage unit, partial blocking across movements, such as a left-turn queue blocking adjacent through traffic, is not modeled explicitly at the lane level but is reflected in the aggregate measured boundary flows. The adaptive process noise captures the increased uncertainty under oversaturated conditions, while aFCD measurements help correct possible estimation drift.

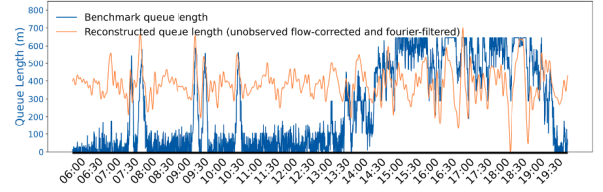
4.2. Measurement function

The measurement function maps the section-level queue length x_t to the expected aFCD speed on each segment. We use the two-state traffic representation introduced in Section 4.1: the queued portion operates at congested speed v_j , while the unqueued portion operates at free-flow speed v_f . Let segment i extend from position l^i to r^i , measured as distances from the stop line with $r^i > l^i$. At time t , the expected speed for segment i is:

$$[h(x_t)]_i = \begin{cases} v_f & x_t \leq l^i & (\text{segment is entirely unqueued}), \\ \frac{v_f(r^i - l^i) - v_j(x_t - l^i)}{r^i - l^i} & l^i < x_t \leq r^i & (\text{segment is partially queued}), \\ v_j & r^i < x_t & (\text{segment is entirely queued}). \end{cases} \quad (18)$$



(a) Reconstructed queue length L_t^q is obtained by removing the unobserved traffic flow and applying an affine transformation to align with the observed queue range.



(b) Fourier filter suppresses low-frequency drift and high-frequency noise, producing a smoothed queue profile $\bar{L}_t^q$ suitable for control input u_t .

Fig. 2. Queue estimation from net vehicle accumulation under unobserved traffic flow conditions for section N1-IN in Rotterdam on 2023-11-10.

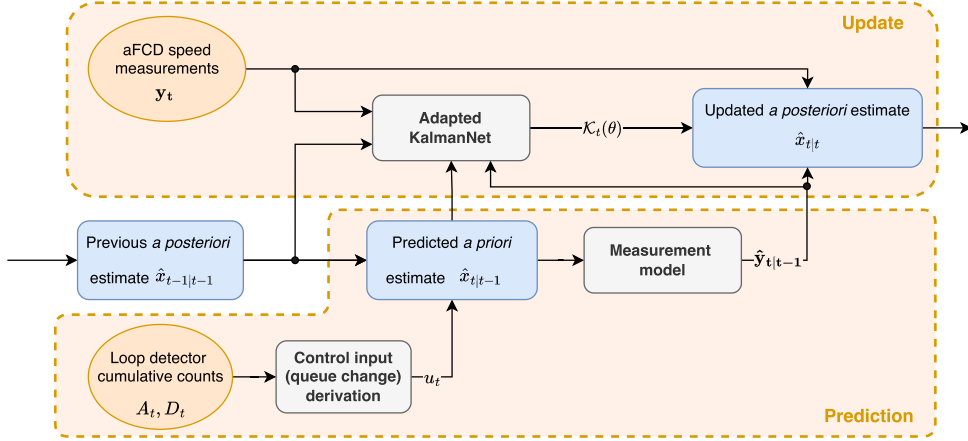


Fig. 3. Overview of the Q-Net filtering framework at time t . The predict step uses cumulative vehicle counts A_t, D_t to derive the control input (queue change) u_t and predict the queue length $\hat{x}_{t|t-1}$ and expected aFCD speeds $\hat{y}_{t|t-1}$. The update step compares $\hat{y}_{t|t-1}$ with observed aFCD speeds y_t and uses the adapted KalmanNet to compute the Kalman gain $\mathcal{K}_t$ and update the queue estimate $\hat{x}_{t|t}$.

For partially queued segments, where the queue boundary lies within the segment, the expected speed is computed as segment length divided by total travel time, with travel time accumulated over the queued and unqueued portions. The noisy measurement is given by $y_t^i = [h(x_t)]_i + v_t^i$, consistent with Eq. (12).

The characteristic speeds v_j and v_f are estimated as the two peak modes of the historical aFCD speed distribution. This empirical calibration aligns the measurement function with section-specific aFCD observations, while short-term cycle-to-cycle fluctuations are treated as stochastic measurement noise.

The speed $[h(x_t)]_i$ is defined from a macroscopic, travel-time-based perspective, consistent with aFCD aggregation. aFCD providers aggregate observations from passing probe vehicles over time intervals; thus, the reported segment speeds represent aggregated traversal speeds rather than instantaneous arithmetic averages of all vehicles present on the segment (NDW, 2026). Accordingly, Eq. (18) computes expected speed from the travel time needed to traverse the free-flow and queued portions. Because v_j is typically much lower than v_f , vehicles spend more time in the congested portion, so the resulting speed is naturally weighted toward v_j as the queue grows.

5. Queue-length estimation framework

This section presents the Q-Net filtering procedure. As shown in Fig. 3 and Algorithm 1, Q-Net follows a Kalman-style predict-update loop.

Predict. The predict step estimates the *a priori* queue length $\hat{x}_{t|t-1}$ from the previous *a posteriori* estimate $\hat{x}_{t-1|t-1}$ and the control input u_t . It then predicts the expected aFCD speeds through the measurement function $\mathbf{h}(\cdot)$:

$$\hat{x}_{t|t-1} = \mathbb{P}_{[0, Q_{\max}]}(\hat{x}_{t-1|t-1} + u_t), \quad (19)$$

$$\hat{y}_{t|t-1} = \mathbf{h}(\hat{x}_{t|t-1}), \quad (20)$$

where $\mathbb{P}_{[0, Q_{\max}]}(\cdot)$ projects the state estimate onto the physically feasible range $[0, Q_{\max}]$, preventing negative queues or queues exceeding the section capacity. This projection introduces a nonlinear constraint, rendering it sub-optimal under classical Kalman filter theory. Nevertheless, Q-Net learns the Kalman gain directly from data rather than relying on linear-Gaussian assumptions. The neural network is expected to adapt to the nonlinearity introduced by the projection during training.

Algorithm 1 Queue-length estimation algorithm.**Input:** Initial state $\hat{x}_{0|0}$, control inputs $\{u_t\}$, measurements $\{y_t\}$, maximum queue length $Q_{\max}$, parameters θ **Output:** Queue length estimates $\{\hat{x}_{t|t}\}$ **For** $t = 1, 2, \dots, T$:**Predict:**

$$\hat{x}_{t|t-1} = \mathbb{P}_{[0, Q_{\max}]}(\hat{x}_{t-1|t-1} + u_t) \leftarrow [\text{Eq.}](19)$$

$$\hat{y}_{t|t-1} = \mathbf{h}(\hat{x}_{t|t-1}) \leftarrow [\text{Eqs.}](18), (20)$$

Update:

$$\text{context}_t = (\Delta\tilde{x}_t, \Delta\hat{x}_t, \Delta\tilde{y}_t, \Delta y_t) \leftarrow [\text{Eqs.}](22)–(25)$$

$$\mathcal{K}_t = \text{NeuralNetwork}(\theta, \text{context}_t)$$

$$\hat{x}_{t|t} = \mathbb{P}_{[0, Q_{\max}]}(\hat{x}_{t|t-1} + \mathcal{K}_t(y_t - \hat{y}_{t|t-1})) \leftarrow [\text{Eq.}](21)$$

Update. The update step corrects the predicted state using the innovation $y_t - \hat{y}_{t|t-1}$ weighted by a learned Kalman gain $\mathcal{K}_t(\theta)$. Q-Net learns $\mathcal{K}_t(\theta)$ from data following the KalmanNet principle. This is important because our state-space model is nonlinear (Section 4) and the process and measurement uncertainties vary over time, as illustrated by the different noise patterns in the morning and afternoon in Fig. 2. The projection in Eq. (21) again enforces physically meaningful estimates and is consistent with constrained filtering practice (Simon, 2010; Prakash et al., 2014):

$$\hat{x}_{t|t} = \mathbb{P}_{[0, Q_{\max}]}(\hat{x}_{t|t-1} + \mathcal{K}_t(\theta)(y_t - \hat{y}_{t|t-1})). \quad (21)$$

Loop detector counts are available every 10 s, while aFCD speeds arrive at a coarser 60s resolution. Since aFCD represents an aggregated one-minute speed rather than an instantaneous snapshot, we use a sample-and-hold strategy: when a new aFCD measurement arrives, it replaces the previous value; otherwise, the most recent value is retained. Although the held measurement remains constant within the one-minute interval, the innovation remains time-varying because $\hat{y}_{t|t-1}$ changes every 10 s as $\hat{x}_{t|t-1}$ in Eq. (20) is driven by loop detector inputs. Because aFCD speeds are temporally aggregated whereas $\mathbf{h}(x_t)$ maps an instantaneous queue state to an expected speed, the resulting temporal mismatch is treated as part of the measurement uncertainty. *Since aFCD is used as a lower-frequency correction signal, while high-resolution loop detector counts drive state propagation in the prediction step, residual aFCD timing errors may still introduce some delay but are less dominant in determining short-term queue dynamics in the offline experiments.* The separate issue of causal aFCD availability, which can further increase the effective temporal mismatch, is examined through the online variants in Section 6.3.

5.1. Learned Kalman gain

Eq. (21) is the key departure from traditional Kalman filtering: Q-Net learns the Kalman gain from data instead of computing it analytically. Following Revach et al. (2022), four temporal features are used to represent the statistical relationship between the state-space model and the gain:

$$\text{The forward evolution difference: } \Delta\tilde{x}_t = \hat{x}_{t|t} - \hat{x}_{t-1|t-1}; \quad (22)$$

$$\text{The forward update difference: } \Delta\hat{x}_t = \hat{x}_{t|t} - \hat{x}_{t|t-1}; \quad (23)$$

$$\text{The measurement difference: } \Delta\tilde{y}_t = y_t - \hat{y}_{t-1}; \quad (24)$$

$$\text{The innovation difference: } \Delta y_t = y_t - \hat{y}_{t|t-1}. \quad (25)$$

The *forward evolution difference* $\Delta\tilde{x}_t$ captures the change between consecutive *a posteriori* state estimates, while the *forward update difference* $\Delta\hat{x}_t$ captures the correction from the *a priori* to the *a posteriori* estimate. Since $\hat{x}_{t|t}$ is unavailable before the Kalman gain is computed, the previous values $\Delta\hat{x}_{t-1}$ and $\Delta\tilde{x}_{t-1}$ are used during inference. In this setup, the pair $(\Delta\tilde{x}_{t-1}, \text{measurement difference } \Delta\tilde{y}_t)$ captures state and measurement evolution, while $\Delta\hat{x}_{t-1}$ and the *innovation difference* Δy_t reflect the current correction and innovation uncertainty.

These inputs are fed into the KalmanNet architecture, which uses Gated Recurrent Units (GRUs) to learn time-varying uncertainty representations. The GRUs control how past information impacts the current prediction, mirroring the recursive nature of the traditional Kalman gain calculation (Eqs. (3)–(10)). The resulting parametric Kalman gain, detailed in Appendix A, is computed as:

$$\mathcal{K}_t = \text{KalmanNet}(\Delta\tilde{x}_t, \Delta\hat{x}_t, \Delta\tilde{y}_t, \Delta y_t). \quad (26)$$

Training pipeline. Q-Net is trained end-to-end to learn the Kalman gain. Given a sequence of T time steps, the loss is the Root Mean Square Error (RMSE) between the estimated queue length $\hat{x}_{t|t}$ and the radar benchmark x_t :

$$\mathcal{L} = \sqrt{\frac{1}{T} \sum_{t=0}^{T-1} (x_t - \hat{x}_{t|t})^2} \quad (27)$$

During training, the projections in Eqs. (19) and (21) are disabled to ensure uninterrupted gradient flow. They are applied during validation and testing to enforce physical queue length bounds.

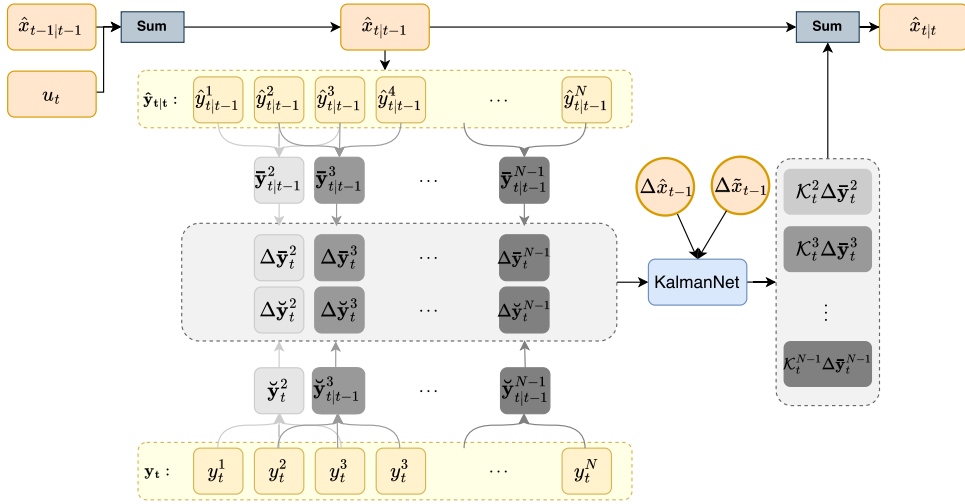


Fig. 4. Local measurement grouping strategy for transferable Kalman gain estimation. aFCD measurements are grouped into fixed-size vectors consisting of the previous, current, and next segments. Each group is processed by the same KalmanNet function to obtain a local gain, and the final state update is obtained by summing all local updates.

5.2. Adaptation for spatial transferability

A segment is the spatial unit over which aFCD aggregates speed measurements. Since benchmark queue data are difficult and costly to obtain, a model trained on one instrumented section should ideally be transferable to other sections. However, standard KalmanNet requires a fixed-dimensional measurement vector determined by the road section's aFCD segmentation, which varies across sections. This mismatch motivates our spatial transferability adaptation. To address it, Q-Net uses fixed-size local measurement groups, enabling the same trained model to be applied to sections with different lengths or segmentations.

The grouping is based on the assumption that measurement covariances depend only on the segment itself and its immediate neighbors (previous and next segments), which is justified by the short 10-s filtering interval (benchmark data frequency). As shown in Fig. 4, for each intermediate segment $1 < i < N$, we form:

- The *grouped measurement difference* $\Delta \tilde{y}_t^i = [y_t^{i-1} - y_{t-1}^{i-1}, y_t^i - y_{t-1}^i, y_t^{i+1} - y_{t-1}^{i+1}]^T$;
- The *grouped innovation difference* $\Delta \tilde{y}_t^i = [y_t^{i-1} - \hat{y}_{t|t-1}^{i-1}, y_t^i - \hat{y}_{t|t-1}^i, y_t^{i+1} - \hat{y}_{t|t-1}^{i+1}]^T$.

Boundary segments ($i = 1$ and $i = N$) are excluded because queue dynamics are mainly reflected by intermediate segments. The first segment near the stop line can be erroneous due to localized dissipation patterns, while the last segment adds limited information since intermediate segments repeatedly confirm the presence of a long queue.

Each grouped feature vector is fed to the same KalmanNet function. Instead of learning one global gain vector in Eq. (8), Q-Net learns multiple, local gain vectors $\mathcal{K}_t^i$ for each group i :

$$\mathcal{K}_t^i = \text{KalmanNet}(\Delta \tilde{x}_t, \Delta \hat{x}_t, \Delta \tilde{y}_t^i, \Delta \tilde{y}_t^j), \quad \text{where } \mathcal{K}_t^i \in \mathbb{R}^{1 \times 3}, \quad (28)$$

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + \sum_{i=2}^{N-1} \mathcal{K}_t^i \Delta \tilde{y}_t^i, \quad \text{where } \Delta \tilde{y}_t^i \in \mathbb{R}^3. \quad (29)$$

Since a single measurement innovation Δy_t^j appears in three groups ($j-1$, j , and $j+1$), its total effective gain is implicitly the sum of components from three adjacent group gains ($\mathcal{K}_t^{j-1}$, $\mathcal{K}_t^j$, and $\mathcal{K}_t^{j+1}$). While this is not a globally optimal update in the classical sense, this design localizes the gain calculation. This localization makes the architecture independent of the total section segment number (N) and thereby achieves spatial transferability.

6. Numerical results

In this section, we evaluate Q-Net in terms of temporal transferability (Section 6.1), spatial transferability (Section 6.2), online capability (Section 6.3), and component contributions (Section 6.4). Training is supervised and relies on benchmark queue data. Sections 6.1 and 6.2 report offline ex-post evaluations using full-day data. In these evaluations, each aFCD timestamp is shifted backward by one minute to align the reported speed with the preceding aggregation interval. Section 6.3 assesses online performance.

Case study. We consider a real-world case study in Rotterdam, the Netherlands, a dense urban road network with recurrent congestion. The dataset covers 16 days from 6:00 AM to 8:00 PM, including 12 weekdays and 4 weekend days. We exclude sections with

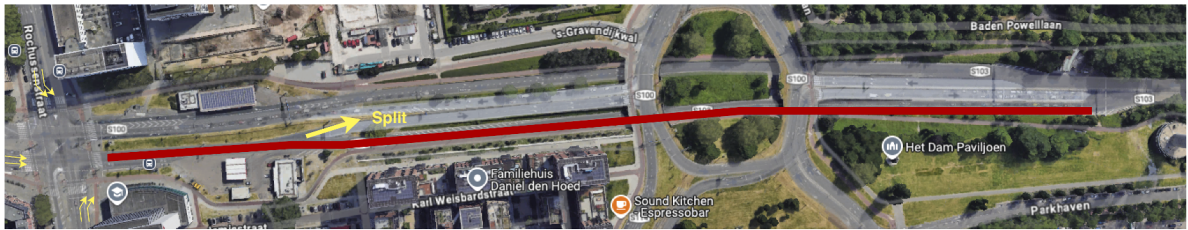


Fig. 5. Satellite view of N1-IN. The study section is highlighted in dark red. Downstream, part of the traffic flow exits toward a grade-separated interchange connected to an open bridge, leading to substantial unobserved outflow.

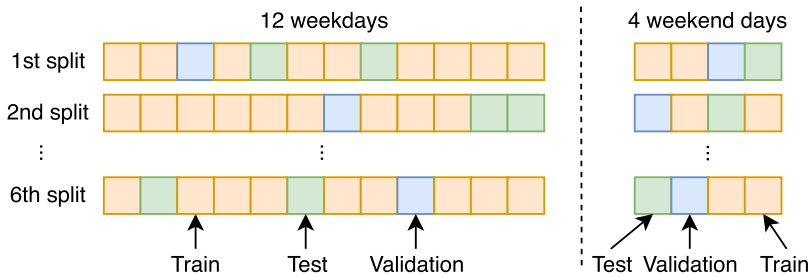


Fig. 6. Six-fold data split: for each split, weekday and weekend samples are randomly assigned to the training, validation, and test sets.

insufficient data availability due to contractual restrictions or substantial missing values, as the evaluation focuses on queue estimation rather than data imputation. Among the remaining candidates, we select N1-IN as the main study section (Fig. 5). This major urban section connects directly to the Maastunnel, an important entry point to Rotterdam and a critical focus of traffic management. It is challenging because considerable unobserved traffic exits toward a ring interchange. Missing aFCD values are imputed using the most recent available value within the same segment.

Baselines. We compare Q-Net with relevant baseline methods. Many existing approaches are not directly comparable because they rely on assumptions incompatible with our setting. For example, loop-detector-based methods estimate cycle-wise maximum queues (Liu et al., 2009; Abewickrema et al., 2023; Sengupta et al., 2021a; An et al., 2018) or assume fully observed and conserved traffic flow (Heshami and Kattan, 2021; Lee et al., 2019; Yang et al., 2019; Karnati et al., 2021). These assumptions do not hold in our study section, which has limited loop coverage and substantial unobserved traffic flow. In addition, cycle-wise methods typically estimate per-lane or per-signal-group maximum queues, whereas our benchmark data provide section-level queue lengths every 10 s.

Given these constraints, we select six baselines. The first two use only aFCD: OSD (Altıntaş, 2018), which detects queue ends from observed speed drops using fixed upstream and downstream thresholds of 16 km/h; and ISC (Dinh et al., 2014), which constructs synthetic trajectories from aFCD and estimates queues using isospeed contours with thresholds from 14 to 22 km/h. The third, Tree-RPROP (Luo et al., 2022), is a multisource baseline combining aFCD, loop detector, and traffic light event data through a decision tree and a resilient propagation neural network. We adopt the same setup as the original paper. Finally, we include three supervised data-driven baselines using the same input features as Q-Net: Long Short-Term Memory networks (LSTM) (Hochreiter and Schmidhuber, 1997), Gaussian Processes (GP) (Williams and Rasmussen, 2006), and eXtreme Gradient Boosting (XGBoost) (Chen and Guestrin, 2016). Their hyperparameters are selected by grid search.

Q-Net implementation and hyperparameters. The Fourier band-pass filter is tuned on the validation set. The lower and upper cutoff frequencies are set to $1/360$ and $1/30$ cycles per step. With a 10-s system time step, this preserves signal periods between 300s (30 steps) and 3600s (360 steps). The KalmanNet implementation follows Revach et al. (2022).¹ Q-Net is trained with the Adam optimizer using a learning rate of 0.001. It contains 952 trainable parameters, and each training epoch takes approximately 40 s on an Apple M2 Pro-chip.

Training strategy. At 10-s resolution, each daily series contains 5040 time steps. To reduce accumulated prediction errors during the early stage of training, we divide each sequence into 10-min windows. The state is initialized to zero for the first window of each day. For subsequent windows, the initial state is set to the *a posteriori* estimate from the previous window, mimicking practical deployment where the true state is unobserved.

Evaluation protocol. Due to the limited data, we use six independent train-validation-test splits for robust evaluation. For each split, the model is trained and tested independently: 11 days are randomly selected for training, 2 for validation, and 3 for testing, while preserving a mix of weekdays and weekends (Fig. 6). This ensures that every day appears in the test set at least once. The day-wise splits are reported in Table B.1, Appendix B.

¹ https://github.com/KalmanNet/KalmanNet_TSP

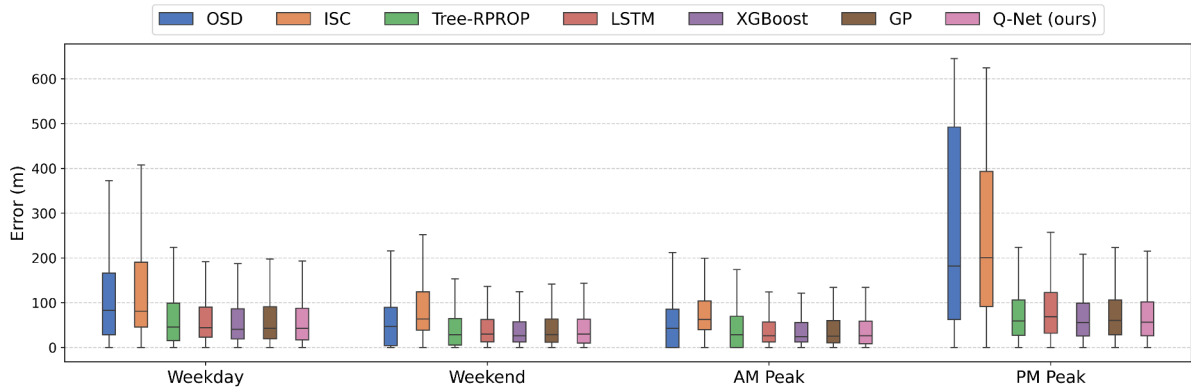


Fig. 7. Box plots of absolute estimation errors for baseline models and Q-Net for section N1-IN.

Table 3

Performance comparison of the proposed and baseline models when trained and tested on N1-IN.

	All day			Morning peak			Afternoon peak		
	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE
OSD	200.17	121.56	95.48	122.94	71.24	118.53	349.54	269.32	58.02
ISC	191.18	131.22	94.03	165.84	108.07	135.33	297.51	241.49	50.3
Tree-RPROP	114.25	67.96	65.11	111.17	58.5	91.74	101.37	75.49	19.43
GP	92.56	61.26	51.79	84.88	50.11	67.99	93.25	73.32	18.5
LSTM	89.32	61.04	49.35	80.85	49.11	66.32	103.32	82.54	18.16
XGBoost	87.81	58.17	48.52	82.17	48.56	63.58	85.82	67.71	16.96
Q-Net (ours)	86.77	58.14	56.22	78.62	47.62	74.64	88.36	69.25	17.41

Models with random initialization are trained with five random seeds. Performance is evaluated using Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). Metrics are computed by concatenating predictions and benchmark queue lengths across all test splits. To avoid inflated relative errors near zero, MAPE is computed only when the benchmark queue length exceeds 10 m, covering 71.73% of the test set. Morning and afternoon peaks are defined as 07:00–09:00 and 16:00–18:00, following Dutch Ministry of Infrastructure guidelines ([Mobiliteitsscanner](#)).

6.1. Performance

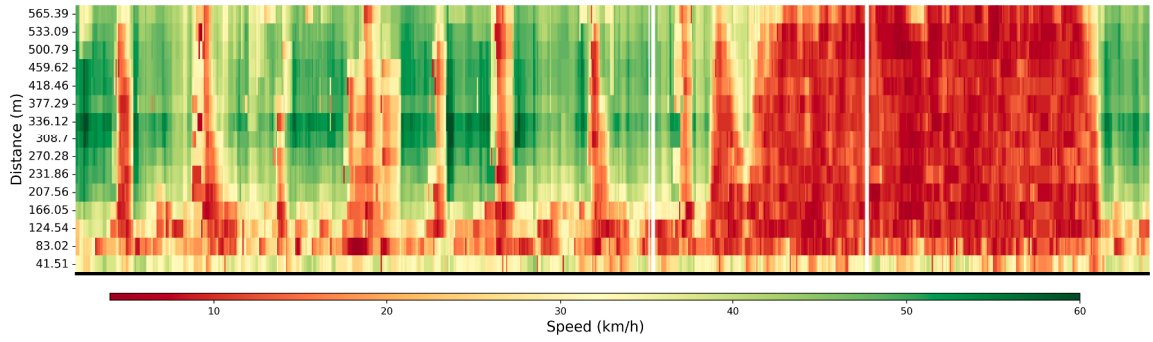
This subsection evaluates temporal transferability, where Q-Net is trained and tested on the same section using different time periods. This setting applies when benchmark sensors, such as radar, can be temporarily installed to collect training data and then removed to reduce long-term sensing costs.

Quantitative analysis. Table 3 reports model performance, and Fig. 7 shows absolute error distributions. Q-Net achieves the best all-day RMSE and MAE and remains competitive across peak periods. OSD and ISC perform worst because they rely directly on aFCD and are therefore sensitive to speed measurement errors. Q-Net also improves over Tree-RPROP, despite Tree-RPROP using additional data sources. Although standard data-driven regression models, especially XGBoost, achieve comparable time-averaged metrics, these aggregate values do not fully reveal transient estimation errors. The qualitative analysis below further examines these differences.

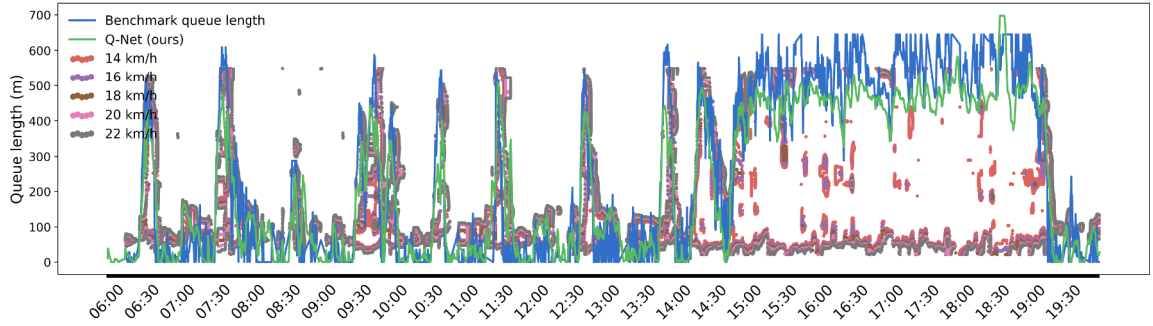
A notable observation from Table 3 is the higher morning-peak MAPE across all methods. The benchmark data indicate that queues do not consistently extend across the full morning peak window. Since MAPE is relative to the benchmark value, shorter queues produce larger percentage errors. Fig. 7 also shows smaller absolute errors during weekends and morning peaks, when queues are generally less pronounced and therefore easier to estimate.

Qualitative analysis. Fig. 8 shows a visual case study for 2023-11-14. For clarity, we plot the best-performing model from each baseline category: ISC, Tree-RPROP, and XGBoost. ISC suffers from discontinuities and may generate multiple queue estimates at a single time step. Tree-RPROP produces many unrealistic maximum queues because it relies on a fixed loop detector occupancy threshold under noisy conditions. Although XGBoost achieves competitive time-averaged metrics, it exhibits clear estimation lag during rapid queue formation and dissipation, as shown in the zoomed comparison in Fig. 8d. Without an explicit queue-evolution structure, purely data-driven models tend to respond reactively to observed input patterns. In contrast, Q-Net uses loop-detector-derived queue changes to drive the prediction step, enabling it to track rapid buildup and dissipation more promptly, while aFCD measurements correct slower estimation drift in the update step. This prediction-correction structure reduces temporal lag while maintaining smooth estimates.

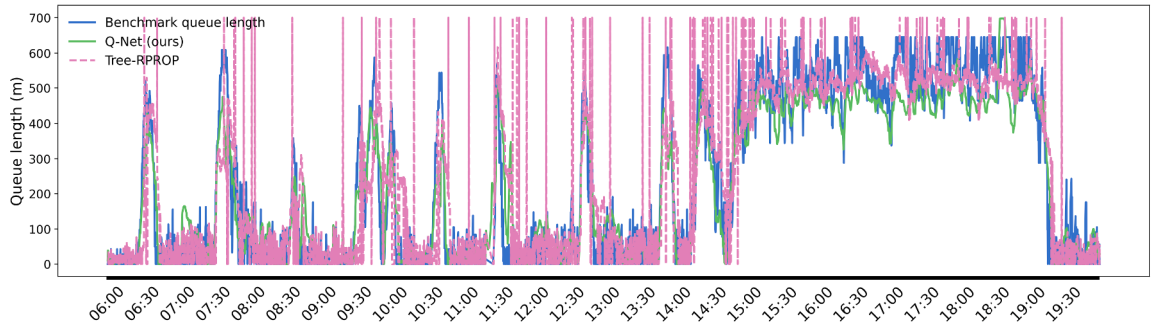
The baseline errors in Fig. 8 are associated with aFCD noise and temporal lag. Fig. 8a shows several aFCD artifacts: (a) erroneous high speeds within the queue during congestion, such as 15:00–19:00, which can place the queue end incorrectly in the middle of



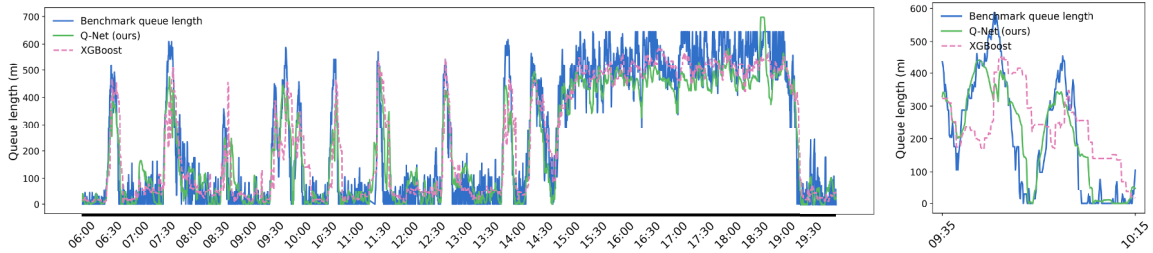
(a) The aFCD.



(b) Q-Net vs ISC.



(c) Q-Net vs Tree-RPROP.



(d) Q-Net vs XGBoost. Left: full-day estimation. Right: zoomed-in comparison during rapid queue variation.

Fig. 8. The aFCD visualization and queue length estimation comparison among selective baselines (ISC, Tree-RPROP, and XGBoost), and our proposed method for section N1-IN on 2023-11-14. Results are generated using the model trained with random seed 42.

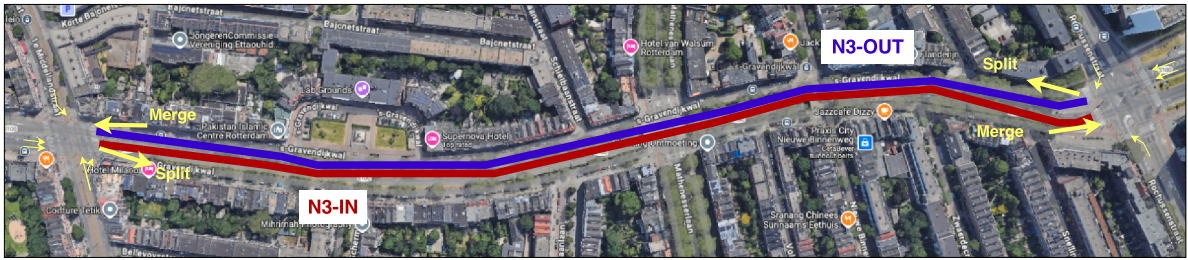


Fig. 9. Satellite view of N3-IN and N3-OUT. In both cases, the downstream area divides into an underground study section and an above-ground branch, which merge at the downstream intersection.

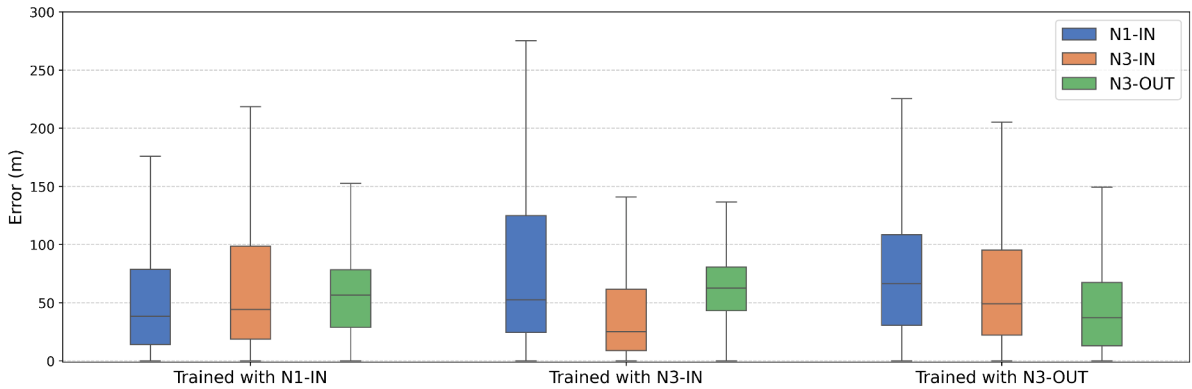


Fig. 10. Box plots of absolute estimation errors across all test datasets using different training sections.

the section; (b) high speeds near the stop line, likely due to sparse probe coverage or aggregation artifacts, causing ISC to estimate near-zero queues despite congestion; and (c) isolated low-speed patches that cause inconsistent queue-end detection. In addition, despite temporal alignment, aFCD can still exhibit delay due to probe sparsity, segment aggregation, and vendor-side smoothing (NDW, 2026). Q-Net is less affected by these errors because the learned Kalman gain adapts to changing model uncertainty, while loop-detector-derived queue changes provide a more responsive prediction of queue dynamics. During 09:35-10:15, for example, aFCD shows delayed queue formation, but Q-Net still captures the queue onset because loop detector cumulative counts respond more promptly to queue changes.

6.2. Transferability

This subsection evaluates spatial transferability, where Q-Net is trained on one section and tested on others without fine-tuning. This setting is relevant when radar benchmark data are available for one instrumented section but not for every target section. The grouping strategy in Section 5.2 enables this transferability.

We introduce two additional Rotterdam sections adjacent to N1-IN: N3-IN and N3-OUT (Fig. 9). Both are underground sections with an above-ground branch that diverges after the upstream intersection and merges back downstream. In N3-IN, combined through/right-turn lanes at the upstream intersection create additional unobserved flow in the studied direction. We train Q-Net separately on N1-IN, N3-IN, and N3-OUT, and evaluate each trained model on all three sections.

Fig. 10 shows the performance for all training-testing combinations. As expected, each section performs best when the model is trained on itself, indicating that Q-Net captures location-specific traffic dynamics. Interestingly, the model trained on N1-IN transfers better to N3-IN and N3-OUT than vice versa. Although N1-IN has a simpler infrastructure layout, its unobserved upstream split exposes the model to uncertainty patterns that transfer effectively to the two N3 sections. By contrast, N3-IN and N3-OUT contain additional complexity: above-ground branches diverge and rejoin downstream, introducing unobserved traffic at both boundaries, and downstream detector counts may include vehicles that were not part of the initial inflow. These ambiguities make it harder for the model to separate multiple uncertainty sources when training data are limited, reducing transfer performance.

Fig. 11 visualizes transfer results on N3-IN and N3-OUT for 2023-11-14 using Q-Net trained on N1-IN. The model captures multiple queue peaks and follows the benchmark during most queue formation and dissipation phases, demonstrating transferability to unseen sections and time periods. Two observations are worth noting. First, the benchmark queue never drops to 0 m, suggesting possible radar malfunction because continuous queues during early morning hours are unrealistic. Second, during stable afternoon queues, Q-Net sometimes overestimates the benchmark. From a traffic management perspective, this conservative bias is acceptable once the minimum queue needed to trigger control action is reached.

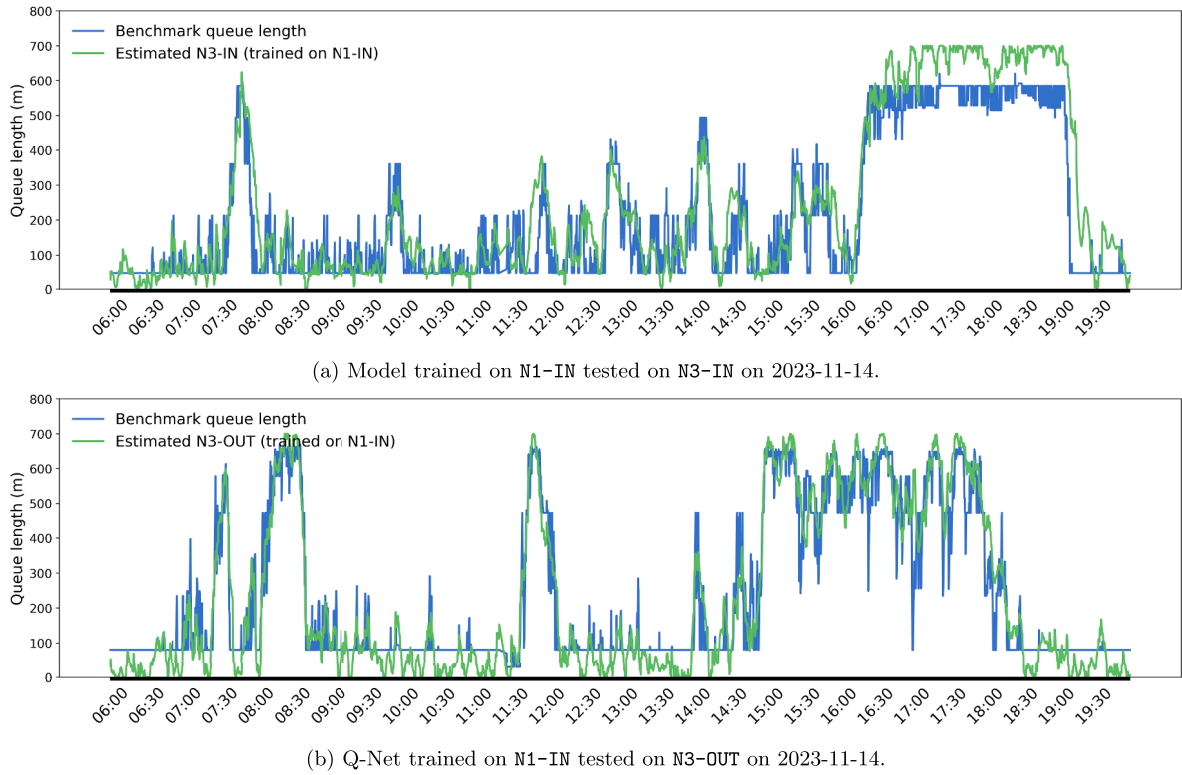


Fig. 11. Queue length estimation for section N3-IN and N3-OUT using a model trained on N1-IN without fine-tuning (random seed 42).

6.3. Online estimation for real-time traffic control

The default Q-Net implementation estimates a fixed unobserved traffic flow rate using full-day cumulative net flow data (Section 4.1) and backward-aligns each aFCD timestamp to the preceding aggregation interval. This ex-post formulation is suitable for performance evaluation but not for causal real-time operation. To assess online applicability and isolate the effect of aFCD temporal delay, we implement two causal variants. The semi-online version estimates the unobserved traffic flow rate using only cumulative flow data available up to the current time, while retaining backward-aligned aFCD timestamps. The fully online version estimates the unobserved traffic flow rate in real time and uses each aFCD observation only after it becomes available. All other hyperparameters remain unchanged.

Table 4 and Fig. 12 compare the default, semi-online, and fully online versions. The semi-online version performs similarly to the default Q-Net in terms of average metrics, indicating that causal estimation of the unobserved flow rate has limited impact. The fully online version shows lower all-day accuracy mainly because delayed aFCD availability amplifies the response delay during rapid queue transitions. However, during the afternoon peak, its performance remains close to the default Q-Net. This indicates that the impact of aFCD time lag is limited in periods where the queue state changes more gradually, whereas it becomes more pronounced during abrupt transitions. During rapid queue transitions, the default Q-Net reacts most promptly, the semi-online version shows a slightly longer delay, and the fully online version exhibits the largest delay. For example, during the queue formation and dissipation period around 08:30-09:30 on 2023-11-22, both online variants lag behind the default estimate but still follow the main trend. Overall, the online variants capture the dominant queue dynamics, suggesting that Q-Net remains promising for real-time monitoring and control applications where causal implementation is required.

6.4. Ablation study

As Q-Net's main components are the state-space formulation, learned Kalman gain, and grouped measurement strategy, we evaluate three ablation variants. The first, Q-Net w/o u_t , removes the control input from Eq. (11), yielding $x_t = x_{t-1} + w_t$. Since u_t is derived from cumulative vehicle counts, this variant relies on aFCD only and tests the contribution of loop-detector-driven queue dynamics. The second variant, Q-EKF, replaces KalmanNet with a standard Extended Kalman Filter (EKF), isolating the benefit of learning the Kalman gain under nonlinear and time-varying uncertainty. The third variant, Q-Net w/o grouping, removes the local measurement grouping strategy in Eq. (29), resulting in a location-specific model without spatial transferability. Results are shown in Table 5 and Fig. 13.

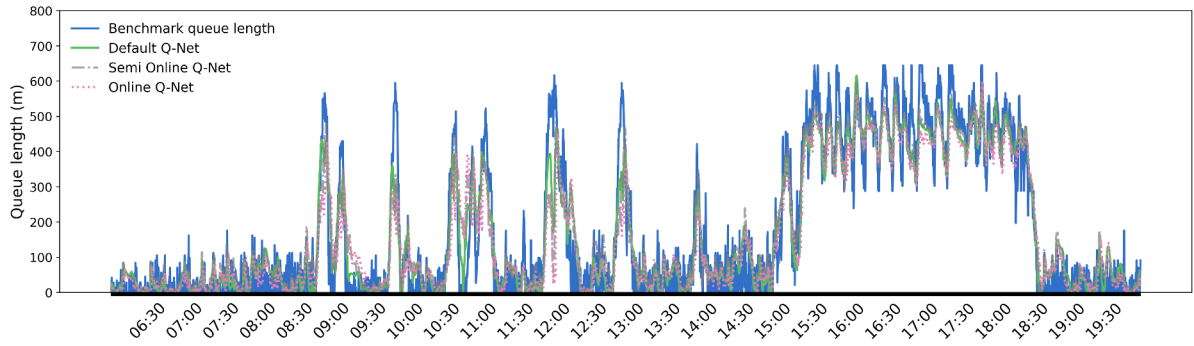


Fig. 12. Comparison of queue length estimation for section N1-IN on 2023-11-22 (Wednesday) using the default, semi-online, and fully online versions of Q-Net. The default version estimates a fixed rate from full-day data via linear regression and temporally aligns aFCD. The semi-online version updates the rate using data available up to the current time. Building upon the semi-online model, the fully online version uses aFCD only after it becomes available.

Table 4

Performance comparison of Q-Net and its online version when trained and tested on N1-IN.

	All day			Morning peak			Afternoon peak		
	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE
Default Q-Net	86.77	58.14	56.22	78.62	47.62	74.64	88.36	69.25	17.41
Semi-online Q-Net	85.33	56.74	56.34	76.24	46.93	74.88	81.07	62.6	15.56
Online Q-Net	100.58	67.92	70.28	89.73	56.56	92.27	89.08	69.89	18.19

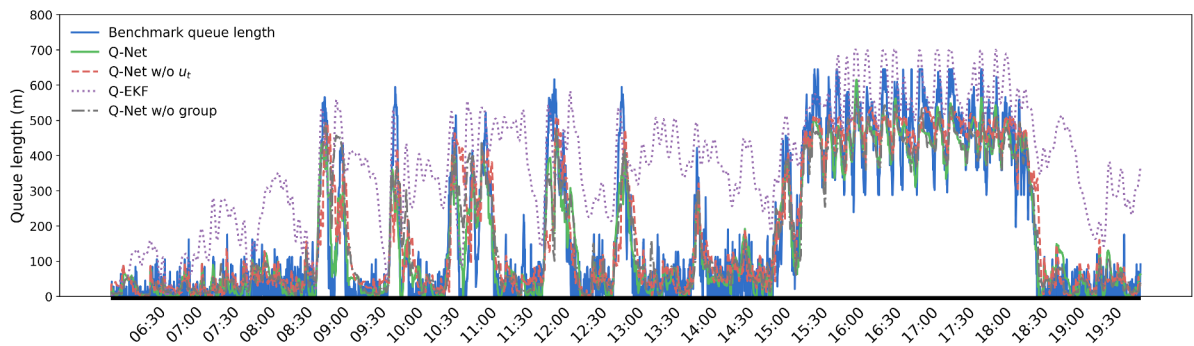


Fig. 13. Queue length estimation results for different model variants on 2023-11-22 for section N1-IN.

Table 5

Performance comparison of model variants in the ablation study on section N1-IN.

	All day			Morning peak			Afternoon peak		
	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE
Default	86.77	58.14	56.22	78.62	47.62	74.64	88.36	69.25	17.41
Q-Net w/o u_t	90.2	59.75	56.81	83.31	49.24	73.24	87.54	68.67	17.27
Q-EKF	217.58	175.78	259.58	213.45	165.46	353.47	135.55	111.59	34.39
Q-Net w/o group	90.12	60.26	58.0	88.1	51.4	83.91	89.78	71.3	16.63

State-space formulation and control input (u_t). Comparing Q-Net w/o u_t with the aFCD-only baselines in Table 3 shows the value of the state-space formulation: even without loop-detector-derived queue changes, Q-Net w/o u_t substantially improves over direct speed-based methods. Comparing Q-Net w/o u_t with the default Q-Net then isolates the role of u_t . Without u_t , the model relies only on aFCD and is more affected by temporal lag, especially during queue formation around 11:30-13:00 in Fig. 13. By incorporating flow-count-based queue changes u_t , the default Q-Net tracks dynamic buildup and dissipation more promptly.

Q-Net w/o u_t performs slightly better during the afternoon peak in Table 5, when queues are long and relatively stable. In this condition, the true queue change is often close to zero, so the assumption $u_t \approx 0$ becomes reasonable, while the default model introduces minor performance degradation through noisy cumulative-count-based u_t . This does not diminish the overall value of the control input, which is more critical for queue formation and dissipation.

Data-driven Kalman filtering. Q-EKF performs worst across all metrics and fails to track queue formation and dissipation accurately (Fig. 13). This highlights the importance of learning the Kalman gain from data. A standard EKF relies on predefined noise covariances, which are insufficient for the nonlinear and time-varying uncertainties observed in urban queue estimation.

Grouped measurement strategy. Q-Net w/o group performs slightly worse than the default Q-Net (Table 5), even though it is trained as a location-specific model. This suggests that the grouping strategy does not impose an accuracy penalty for enabling spatial transferability. Instead, neighboring aFCD segments provide useful local context for gain estimation. The grouped design therefore supports spatial transferability (Section 6.2) without compromising single-location performance.

7. Discussion

Q-Net addresses a practical queue estimation setting in which traffic flows are only partially observed and direct queue measurements are unavailable during deployment. It uses two widely available and privacy-preserving data sources: loop detector counts and aggregated floating car data (aFCD). While loop detector placement may differ across networks, our method remains applicable by accounting for the travel time between vehicle detection and arrival at the stop line, though performance may vary depending on detector placement and traffic conditions.

Architectural considerations. Q-Net uses loop detector data to derive queue change as the control input. This component is obtained through affine transformation and Fourier-based band-pass filtering, as described in Section 4.1. More specialized neural architectures could be designed for estimating queue changes, and we experimented with LSTM and attention-based alternatives. However, under limited training data, these models struggled to learn queue-change dynamics and noise uncertainties at the same time, and the current formulation performed better. A separate framework focused only on queue-change prediction and subsequent queue reconstruction is possible but beyond the scope of this study.

Q-Net introduces two practical deviations from classical Kalman filtering. First, it applies a projection operation to keep queue estimates within physically feasible bounds. Although this sacrifices some classical Kalman optimality, such ad hoc constraints are widely used in engineering practice to enforce physical feasibility (Simon, 2010; Prakash et al., 2014) and performed well in our experiments. Second, Q-Net learns the Kalman gain with an unconstrained neural network, so the KalmanNet implementation does not enforce positive semi-definiteness of learned covariance surrogates. In traditional recursive estimation, violating this property may destabilize the filter and lead to divergence (Simon, 2006). Recent methods enforce positive definiteness using Cholesky decomposition (Ko and Shafieezadeh, 2024), but they require empirically tuned activation functions and more complex loss functions. We use an unconstrained architecture to retain representational flexibility and did not observe divergence or numerical instability during our experiments. Establishing formal stability guarantees for neural gain-learning filters remains an important direction for future research.

Evaluation limitations. The empirical evaluation has two main limitations. First, although the dataset is real-world, it covers only 16 days within a single month. Such data limitations are common in queue estimation studies (Liu et al., 2009; Abewickrema et al., 2023; Luo et al., 2022). Simulation tools such as SUMO or AIMSUN (Abewickrema et al., 2025) could provide larger datasets, but simulating unobserved traffic flows in a high-fidelity manner is itself a complex research problem and was not pursued here. Second, the spatial transferability evaluation is limited to N1-IN, N3-IN, and N3-OUT, which are located within the same corridor in Rotterdam due to data availability constraints. These sections differ in infrastructure layout, traffic direction, and unobserved traffic-flow patterns, allowing us to evaluate local spatial transferability across connected but non-identical sections. The shared corridor context may still limit the generality of the findings. Broader validation across more independent test settings, such as different corridors, network types, and cities, is therefore needed to assess whether the observed transferability generalizes beyond the local corridor context.

In this transfer setting, the learned gain adaptation mechanism is reused, while the unobserved net flow rate is derived from local loop detector data rather than fixed global parameters. However, because Q-Net treats each section as an aggregated storage unit, highly complex lane configurations may increase uncertainty in the effective net inflow estimation.

Advanced spatiotemporal architectures such as graph neural networks and Transformers could also be considered for transfer learning. However, defining spatial dependencies for such baselines is not straightforward in the current setup, where only one labeled training section is available and the target sections have different layouts and aFCD segmentations. More generally, rigorous benchmarking of such models is constrained by the limited availability of ground-truth queue length data across multiple locations, particularly for constructing graph relationships across multiple sections and training data-intensive models. Benchmarking Q-Net against these architectures under larger-scale transfer settings is therefore left for future work.

aFCD penetration and data requirements. The estimated aFCD penetration rate in our study region is approximately 18.5% (Eisinga and Lorkowski, 2025). Q-Net is designed to reduce reliance on high probe coverage because loop detector inputs drive the queue evolution model, and aFCD periodically corrects prediction drift through macroscopic average speeds. Previous studies suggest that penetration rates as low as 2–3% can capture stable macroscopic speed trends (Herrera et al., 2010). This does not directly guarantee accurate queue estimation at such low penetration rates, but it suggests that useful macroscopic information may remain available. Quantifying Q-Net's performance degradation under reduced aFCD penetration remains an important future task.

Future extensions and emerging data. Q-Net represents queue length as a section-level macroscopic state. Lane-level heterogeneity, movement-specific blocking, and lane-specific spillback are not explicitly modeled. This choice is consistent with the available data,

which consist of aggregated detector counts, segment-level aFCD speeds, and section-level radar benchmark queues. Future work could extend the framework to lane-resolved state estimation or multi-regime traffic representations for complex oversaturated intersections.

Such extensions may be supported by emerging high-resolution data sources, including lane-specific trajectories from Connected and Autonomous Vehicles (CAVs) and localized queue detections from camera-based computer vision systems. These complementary, high-granularity streams would provide the microscopic detail necessary to move beyond current macroscopic approximations.

8. Conclusion

This paper addresses the challenge of estimating queue length in partially observed environments by proposing Q-Net, a Kalman-based framework that integrates model-driven and data-driven principles. Evaluated on multiple main road sections in Rotterdam, Q-Net achieves competitive or improved performance compared with baseline methods and effectively mitigates the lag typically observed in aFCD-based estimates. The model's strong performance across temporal, spatial, and real-time online settings validates our hybrid approach. These results demonstrate Q-Net's practicality as a robust solution for real-world deployment, offering accurate queue estimation without the need to maintain or install costly radar infrastructure.

CRedit authorship contribution statement

Ting Gao: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Formal analysis, Data curation, Conceptualization; **Elvin Isufi:** Writing – review & editing, Supervision, Software, Methodology, Conceptualization; **Winnie Daamen:** Writing – review & editing, Supervision, Software, Methodology, Conceptualization; **Erik-Sander Smits:** Resources, Methodology, Data curation; **Serge Hoogendoorn:** Writing – review & editing, Supervision, Methodology, Funding acquisition, Conceptualization.

Declaration of generative AI use

During the preparation of this work, the authors used ChatGPT (OpenAI) and Gemini (Google) to assist with writing revision and language polishing. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Acknowledgments

The authors gratefully acknowledge the support of the Rotterdam municipality for providing the data. This research has been produced by the “EMERALDS” project which has received funding under the Horizon Europe research and innovation programme (GA No. 101093051).

Data availability

The authors do not have permission to share data.

Appendix A. KalmanNet algorithm

Recall that Σ_t and S_t denote the covariance matrices of the state x_t and the measurements y_t , while Q_t and R_t represent the process and measurement noise covariances. In KalmanNet, we use the notation $\hat{(\cdot)}$ to denote learned, time-varying uncertainty representations inspired by these quantities, and $(\cdot)^h$ to denote internal latent states of recurrent modules. As shown in Fig. A.1, KalmanNet can be globally divided into five submodules:

The **process uncertainty module** produces a time-varying representation $\hat{Q}_t$ based on the *forward update difference* $\Delta\hat{x}_{t-1}$. Both $\hat{Q}_t$ and the *forward evolution difference* $\Delta\hat{x}_{t-1}$ are then fed into the **state uncertainty module** to produce the state-uncertainty embedding $\hat{\Sigma}_t$. Its associated latent state $\hat{\Sigma}_t^h$ is later refined by the **hidden-state update module**. Next, $\hat{\Sigma}_t$, the *innovation difference* Δy_t , and the *measurement difference* $\Delta\hat{y}_t$ are provided to the **measurement uncertainty module** to compute measurement-uncertainty embedding $\hat{S}_t$. Finally, $\hat{\Sigma}_t$ and $\hat{S}_t$ are passed to the **Kalman gain module**, which outputs the Kalman gain $\mathcal{K}_t$. All uncertainty-related matrices are represented in vectorized form throughout the network, which simplifies operations such as concatenation and allows the network to learn each entry without explicit matrix structure constraints.

- a. **Process uncertainty Module:** The design is motivated by Eqs. (1) and (3), which suggest that evolution noise influences the predicted state estimate. The estimation follows:

$$\hat{Q}_t = \text{GRU}(\text{FC}(\Delta\hat{x}_{t-1})). \quad (\text{A.1})$$

The fully connected (FC) layers map the input into a higher-dimensional feature space. The gated recurrent unit (GRU) maintains a hidden state that is updated at each time step through its gating mechanism, allowing $\hat{Q}_t$ to reflect the complete history of *forward update differences* rather than a static noise model.

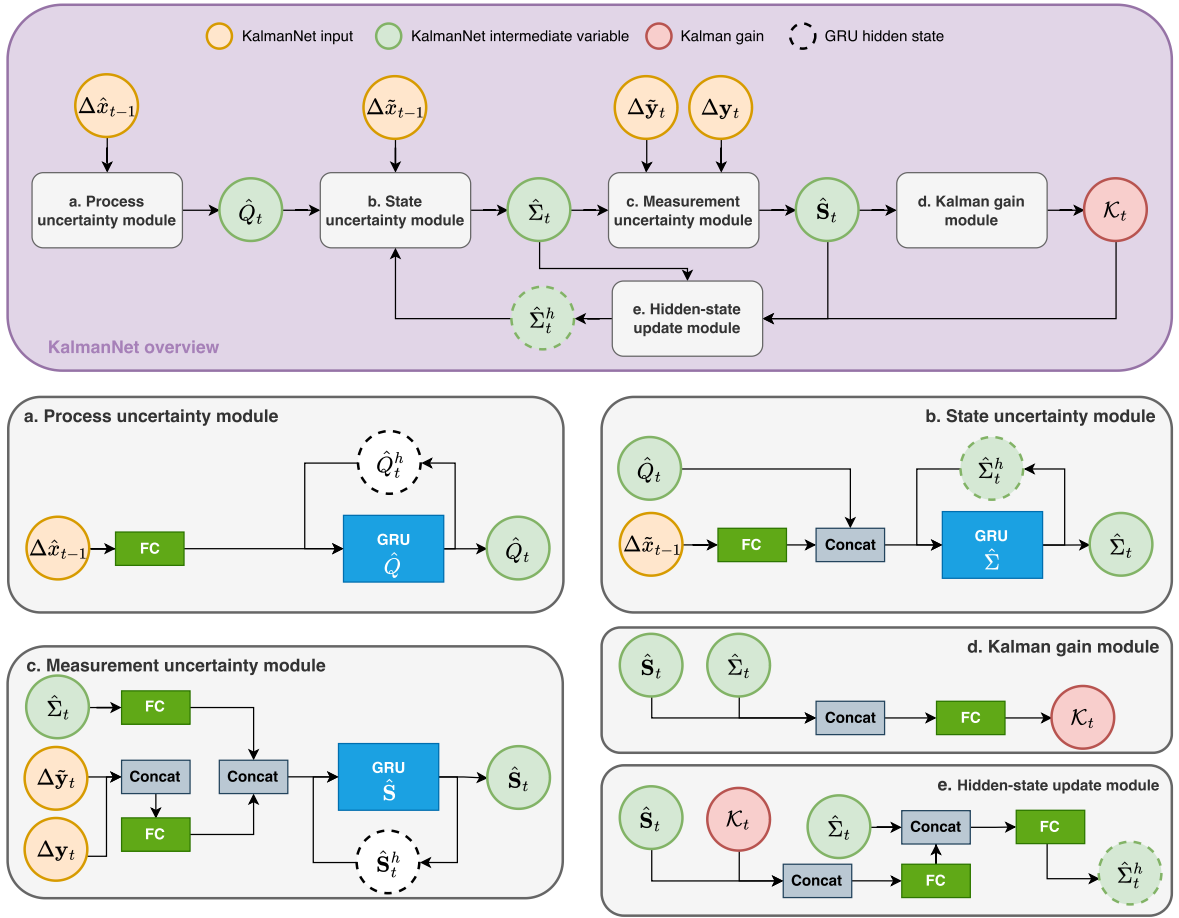


Fig. A.1. KalmanNet architecture for dynamic Kalman gain estimation. The top panel illustrates the overall KalmanNet pipeline, which takes in four temporal input features (*forward evolution difference* $\Delta\hat{x}_{t-1}$, *forward update difference* $\Delta\tilde{x}_{t-1}$, *measurement difference* $\Delta\tilde{y}_t$, *innovation difference* Δy_t) to estimate the Kalman gain. Intermediate representations ($\hat{Q}_t$, $\hat{\Sigma}_t$, $\hat{S}_t$ and $\hat{\Sigma}_t^h$) are computed through sequential submodules. The lower panels detail the submodules a, b, c, d, and e.

- b. State uncertainty Module:** This module estimates state-uncertainty embedding $\hat{\Sigma}_t$ using the learned process uncertainty representation $\hat{Q}_t$ (per Eq. (4)) and the *forward evolution difference* $\Delta\tilde{x}_{t-1}$, which captures queue length evolution and system dynamics not reflected in process uncertainty alone:

$$\hat{\Sigma}_t, \hat{\Sigma}_t^h = \text{GRU}(\text{Concat}[\hat{Q}_t, \text{FC}(\Delta\tilde{x}_{t-1})]), \quad (\text{A.2})$$

where $\hat{\Sigma}_t^h$ denotes the GRU hidden state (later refined in module e) and Concat denotes concatenation. The GRU selectively retains relevant historical uncertainty patterns while discarding outdated information.

- c. Measurement uncertainty module:** Following KalmanNet conventions, we denote the learned surrogate of the measurement covariance term by $\hat{S}_t$. From Eq. (6), $\hat{S}_t$ depends on the *innovation difference* Δy_t (speed prediction errors) and $\hat{\Sigma}_t$ (queue length uncertainty). We additionally include the *measurement difference* $\Delta\tilde{y}_t$ to capture temporal variations in raw aFCD speed measurements, and the GRU gates control how past measurement patterns and current uncertainties influence the estimate:

$$\hat{S}_t = \text{GRU}(\text{Concat}[\text{FC}(\hat{\Sigma}_t), \text{FC}(\text{Concat}[\Delta\tilde{y}_t, \Delta y_t])]). \quad (\text{A.3})$$

- d. Kalman gain module:** This module computes the Kalman gain $\mathcal{K}_t$ based on the queue length and speed measurement uncertainty embeddings, consistent with the inputs used by Kalman gain computation (state and measurement uncertainties):

$$\mathcal{K}_t = \text{FC}(\text{Concat}[\hat{\Sigma}_t, \hat{S}_t]). \quad (\text{A.4})$$

The recursive nature of Kalman gain calculation (Eqs. (3)–(10)), where each Kalman gain updates the state estimate and subsequently influences the next gain computation, is reflected through the GRU-based temporal dependencies in modules b and c.

- e. Hidden-state update: This module updates the internal hidden state $\hat{\Sigma}_t^h$ to retain temporal context across time steps, which is reflected by Eq. (10):

$$\hat{\Sigma}_t^h = \text{FC}(\text{Concat}[\hat{\Sigma}_t, \text{FC}(\text{Concat}[\mathcal{K}_t, \hat{\Sigma}_t])]). \quad (\text{A.5})$$

In summary, the Kalman gain is computed adaptively, leveraging both memory of past information and current predicted state and measurement data:

$$\mathcal{K}_t = \text{KalmanNet}(\Delta \tilde{x}_t, \Delta \hat{x}_t, \Delta \tilde{y}_t, \Delta y_t). \quad (\text{A.6})$$

For additional implementation details, please refer to Revach et al. (2022).

FC layers are typically followed by ReLU activations to produce non-negative feature representations. However, KalmanNet does not provide theoretical guarantees of positive semi-definiteness for the estimated covariance surrogates or strict non-negativity for scalar variance surrogates. In particular, outputs from the GRU layers are unconstrained, and the Kalman gain is computed via a two-layer network without explicit non-negativity enforcement on the final output.

Appendix B. Data split

Table B.1

All data were collected in November 2023. The table presents the specific days assigned to the training, validation, and test sets in each split.

Split	Train days	Validation days	Test days
1	16, 17, 10, 27, 23, 13, 21, 15, 24, 25, 19	20, 11	14, 22, 18
2	14, 22, 16, 17, 15, 24, 20, 21, 10, 18, 11	23, 25	27, 13, 19
3	14, 24, 13, 23, 15, 17, 20, 10, 27, 19, 11	22, 18	21, 16, 25
4	27, 20, 21, 23, 14, 17, 10, 22, 16, 19, 18	13, 25	15, 24, 11
5	20, 10, 22, 14, 16, 27, 15, 21, 13, 19, 11	24, 18	23, 17, 25
6	27, 22, 23, 24, 21, 15, 16, 17, 14, 11, 25	13, 19	18, 10, 20

References

- Abewickrema, W., Yildirimoglu, M., Kim, J., 2023. Multivariate time-varying Kalman filter approach for cycle-based maximum queue length estimation. *Transp. Res. Part C Emerg. Technol.* 154, 104238.
- Abewickrema, W., Yildirimoglu, M., Kim, J., 2025. An ensemble deep learning framework for real-time queue length estimation at signalized intersections. *Data Sci. Transp.* 7 (2), 15.
- Ahmed, A., Outay, F., Zaidi, S. O.R., Adnan, M., Ngoduy, D., 2021. Examining queue-jumping phenomenon in heterogeneous traffic stream at signalized intersection using UAV-based data. *Pers. Ubiquitous Comput.* 25, 93–108.
- Albiol, A., Albiol, A., Mossi, J.M., 2011. Video-based traffic queue length estimation. In: 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops). IEEE, pp. 1928–1932.
- Altuntaşı, O., 2018. Detecting Queue Length on Urban Arterials Using Floating Car Data (FCD). Ph.D. thesis. Middle East Technical University (Turkey).
- An, C., Wu, Y.-J., Xia, J., Huang, W., 2018. Real-time queue length estimation using event-based advance detector data. *J. Intell. Transp. Syst.* 22 (4), 277–290.
- Bernas, M., Placzek, B., Korski, W., Loska, P., Smyła, J., Szymała, P., 2018. A survey and comparison of low-cost sensing technologies for road traffic monitoring. *Sensors* 18 (10), 3243.
- Buchnik, I., Revach, G., Steger, D., Van Sloun, R. J.G., Routtenberg, T., Shlezinger, N., 2023. Latent-KalmanNet: learned Kalman filtering for tracking from high-dimensional signals. *IEEE Trans. Signal Process.* 72, 352–367.
- Cai, Q., Wang, Z., Zheng, L., Wu, B., Wang, Y., 2014. Shock wave approach for estimating queue length at signalized intersections by fusing data from point and mobile sensors. *Transp. Res. Rec.* 2422 (1), 79–87.
- Chauhan, R., Dhamaniya, A., Arkatkar, S., 2021. Driving behavior at signalized intersections operating under disordered traffic conditions. *Transp. Res. Rec.* 2675 (12), 1356–1378.
- Chen, T., Guestrin, C., 2016. XGBoost: a scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794.
- 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.
- Cheng, Y., Qin, X., Jin, J., Ran, B., Anderson, J., 2011. Cycle-by-cycle queue length estimation for signalized intersections using sampled trajectory data. *Transp. Res. Rec.* 2257 (1), 87–94.
- Chevallier, E., Can, A., Nadji, M., Leclercq, L., 2009. Improving noise assessment at intersections by modeling traffic dynamics. *Transp. Res. D Transp. Environ.* 14 (2), 100–110.
- Choi, G., Park, J., Shlezinger, N., Eldar, Y.C., Lee, N., 2023. Split-KalmanNet: a robust model-based deep learning approach for state estimation. *IEEE Trans. Veh. Technol.* 72 (9), 12326–12331.
- Dinh, T.-U.J., Billot, R., Pillet, E., El Faouzi, N.-E., 2014. Real-time queue-end detection on freeways with floating car data: practice-ready algorithm. *Transp. Res. Rec.* 2470 (1), 46–56.
- Eisinga, K., Lorkowski, S., 2025. Network-wide traffic volume estimation based on probe vehicle data. *Transp. Res. Rec.* 2679 (4), 264–277.
- Federal Highway Administration, 2006. *Traffic Detector Handbook: Third Edition—Volume II*. Technical Report FHWA-HRT-06-139. U.S. Department of Transportation, Federal Highway Administration. Washington, DC. Accessed: 2025-09-18. <https://www.fhwa.dot.gov/publications/research/operations/its/06139/chapt5b.cfm>.
- Geroliminis, N., Skabardonis, A., 2011. Identification and analysis of queue spillovers in city street networks. *IEEE Trans. Intell. Transp. Syst.* 12 (4), 1107–1115.
- Herrera, J.C., Work, D.B., Herring, R., Ban, X.J., Jacobson, Q., Bayen, A.M., 2010. Evaluation of traffic data obtained via GPS-enabled mobile phones: the mobile century field experiment. *Transp. Res. C Emerg. Technol.* 18 (4), 568–583.
- Heshami, S., Kattan, L., 2021. A queue length estimation and prediction model for long freeway off-ramps. *J. Intell. Transp. Syst.* 25 (1), 122–134.
- Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. *Neural Comput.* 9 (8), 1735–1780.
- Hoogendoorn, S.P., Van Kooten, J., Adams, R., 2016. Lessons learned from field operational test of integrated network management in Amsterdam. *Transp. Res. Rec.* 2554 (1), 111–119.

- Hu, S., Zhou, Q., Li, J., Wang, Y., Roncoli, C., Zhang, L., Lehe, L., 2022. High time-resolution queue profile estimation at signalized intersections based on extended Kalman filtering. *IEEE Trans. Intell. Transp. Syst.* 23 (11), 21274–21290.
- Karnati, Y., Sengupta, R., Rangarajan, A., Ranka, S., 2021. Subcycle waveform modeling of traffic intersections using recurrent attention networks. *IEEE Trans. Intell. Transp. Syst.* 23 (3), 2538–2548.
- Ko, M., Shafieezadeh, A., 2024. Cholesky-KalmanNet: model-based deep learning with positive definite error covariance structure. *IEEE Signal Process. Lett.* 32, 326–330.
- Lee, S., Xie, K., Ngoduy, D., Keyvan-Ekbatani, M., 2019. An advanced deep learning approach to real-time estimation of lane-based queue lengths at a signalized junction. *Transp. Res. Part C Emerg. Technol.* 109, 117–136.
- Li, L., Okoth, V., Jabari, S.E., 2021. Backpressure control with estimated queue lengths for urban network traffic. *IET Intell. Transp. Syst.* 15 (2), 320–330.
- Li, Q., Li, R., Ji, K., Dai, W., 2015. Kalman filter and its application. In: 2015 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS). IEEE, pp. 74–77.
- Lighthill, M.J., Whitham, G.B., 1955. On Kinematic Waves I. Flood Movement in Long Rivers. II A Theory of Traffic Flow on Long Crowded Roads. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences* 229 (1178), 281–345.
- Liu, H.X., Wu, X., Ma, W., Hu, H., 2009. Real-time queue length estimation for congested signalized intersections. *Transp. Res. Part C Emerg. Technol.* 17 (4), 412–427.
- Luo, X., Ma, X., Munden, M., Wu, Y.-J., Jiang, Y., 2022. A multisource data approach for estimating vehicle queue length at metered on-ramps. *J. Transp. Eng. A Syst.* 148 (2), 04021117.
- Matzoros, A., Van Vliet, D., 1992. A model of air pollution from road traffic, based on the characteristics of interrupted flow and junction control: Part I-model description. *Transp. Res. A Policy Pract.* 26 (4), 315–330.
- Maybeck, P.S., 1982. *Stochastic Models, Estimation, and Control*. Vol. 3. Academic Press.
- Mobiliteitsscan, Floating car data (FCD). Accessed: 2025-05-21. <https://mobiliteitsscan-info.nl/analyse/modulegroepen/modulegroep-externe-bronnen/floating-car-data/>.
- MÜCK, J., 2002. Using detectors near the stop-line to estimate traffic flows. *Traffic Eng. Control* 43 (11), 429–434.
- Nadrian, H., Taghdisi, M.H., Pouyesh, K., Khazaei-Pool, M., Babazadeh, T., 2019. “I am sick and tired of this congestion”: perceptions of sanandaj inhabitants on the family mental health impacts of urban traffic jam. *J. Transp. Health* 14, 100587.
- NDW, 2026. National data warehouse for traffic information, floating car data (FCD). Accessed: 2026-03-22. <https://docs.ndw.nu/en/producten/fcd/>.
- Pasquale, C., Saccone, S., Siri, S., Papageorgiou, M., 2018. Optimal control for reducing congestion and improving safety in freeway systems. *IEEE Trans. Intell. Transp. Syst.* 19 (11), 3613–3625.
- Prakash, J., Huang, B., Shah, S.L., 2014. Recursive constrained state estimation using modified extended Kalman filter. *Comput. Chem. Eng.* 65, 9–17.
- Ramezani, M., Geroliminis, N., 2015. Queue profile estimation in congested urban networks with probe data. *Computer-Aided Civ. Infrastruct. Eng.* 30 (6), 414–432.
- Revach, G., Ni, X., Shlezinger, N., Van Sloun, R. J.G., Eldar, Y.C., 2023. RTSNet: learning to smooth in partially known state-space models. *IEEE Trans. Signal Process.* 71, 4441–4456.
- Revach, G., Shlezinger, N., Ni, X., Escoriza, A.L., Van Sloun, R. J.G., Eldar, Y.C., 2022. KalmanNet: neural network aided Kalman filtering for partially known dynamics. *IEEE Trans. Signal Process.* 70, 1532–1547.
- Rose, G., 2006. Mobile phones as traffic probes: practices, prospects and issues. *Transp. Rev.* 26 (3), 275–291.
- Sengupta, R., Karnati, Y., Rangarajan, A., Ranka, S., 2021a. TQAM: temporal attention for cycle-wise queue length estimation using high-resolution loop detector data. In: 2021 IEEE International Intelligent Transportation Systems Conference (ITSC). IEEE, pp. 3313–3320.
- Shang, Q., Tang, Y., Yin, L., 2024. A hybrid model for missing traffic flow data imputation based on clustering and attention mechanism optimizing LSTM and adaboost. *Sci. Rep.* 14 (1), 26473.
- 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.
- Simon, D., 2006. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. John Wiley & Sons.
- Simon, D., 2010. Kalman filtering with state constraints: a survey of linear and nonlinear algorithms. *IET Control Theory Appl.* 4 (8), 1303–1318.
- Sweet, M., 2011. Does traffic congestion slow the economy? *J. Plan. Lit.* 26 (4), 391–404.
- Tang, Y., Ozbay, K., Jin, L., 2023. Robust queue length estimation for ramp metering in a connected vehicle environment. In: 2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC). IEEE, pp. 4402–4407.
- Umair, M., Farooq, M.U., Raza, R.H., Chen, Q., Abdulhai, B., 2021. Efficient video-based vehicle queue length estimation using computer vision and deep learning for an urban traffic scenario. *Processes* 9 (10), 1786.
- van der Loop, H., Kouwenhoven, M., van Bekkum, P., 2017. Validation of floating car data for transport policy analysis and transport models. Technical Report. KIM Netherlands Institute for Transport Policy Analysis. KIM Netherlands Institute for Transport Policy Analysis, Significance, MuConsult. <https://www.kimnet.nl/publicaties/papers/2017/10/04/validation-of-floating-car-data-for-transport-policy-analysis-and-transport-models>.
- van Erp, P.B., Knoop, V.L., Hoogendoorn, S.P., 2018. Estimating the vehicle accumulation: data-fusion of loop-detector flow and floating car speed data. In: *Proceedings of the 97th TRB Annual Meeting*, Washington, DC, USA, pp. 7–11.
- Vigos, G., Papageorgiou, M., Wang, Y., 2008. Real-time estimation of vehicle-count within signalized links. *Transp. Res. C Emerg. Technol.* 16 (1), 18–35.
- Wang, Y., Yao, Z., Cheng, Y., Jiang, Y., Ran, B., 2021. Kalman filtering method for real-time queue length estimation in a connected vehicle environment. *Transp. Res. Rec.* 2675 (10), 578–589.
- Wang, Z., Cai, Q., Wu, B., Zheng, L., Wang, Y., 2017. Shockwave-based queue estimation approach for undersaturated and oversaturated signalized intersections using multi-source detection data. *J. Intell. Transp. Syst.* 21 (3), 167–178.
- Webster, F.V., 1958. *Traffic Signal Settings*. Technical Report.
- Williams, C. K.I., Rasmussen, C.E., 2006. *Gaussian Processes For Machine Learning*. Vol. 2. MIT Press Cambridge, MA.
- Yang, G., Tian, Z., Wang, D., Xu, H., 2019. Queue length estimation for a metered on-ramp using mesoscopic simulation. *Transp. Lett.* 11 (10), 570–579.
- Yin, J., Sun, J., Tang, K., 2018. A Kalman filter-based queue length estimation method with low-penetration mobile sensor data at signalized intersections. *Transp. Res. Rec.* 2672 (45), 253–264.
- Zhao, Y., Wong, W., Zheng, J., Liu, H.X., 2021. Maximum likelihood estimation of probe vehicle penetration rates and queue length distributions from probe vehicle data. *IEEE Trans. Intell. Transp. Syst.* 23 (7), 7628–7636.
- Zhao, Y., Zheng, J., Wong, W., Wang, X., Meng, Y., Liu, H.X., 2019. Various methods for queue length and traffic volume estimation using probe vehicle trajectories. *Transp. Res. C Emerg. Technol.* 107, 70–91.