**Article:**
Huang, D., Zhang, J., Liu, Z. et al. (1 more author) (2025) Prescriptive analytics for freeway traffic state estimation by multi-source data fusion. Transportation Research Part E: Logistics and Transportation Review, 198. 104105. ISSN 1366-5545

https://doi.org/10.1016/j.tre.2025.104105

Published in Transportation Research Part E.: https://doi.org/10.1016/j.tre.2025.104105

# Prescriptive Analytics for Freeway Traffic State Estimation by Multi-source Data Fusion

*Di Huang*[a], *Jinyu Zhang*[a], *Zhiyuan Liu*[a,]*, *Ronghui Liu*[b]

[a] *School of Transportation, Southeast University, China*

[b] *Institute for Transport Studies, University of Leeds, Leeds, UK*

***Corresponding author.** Email: zhiyuanl@seu.edu.cn

## Abstract

In the context of freeway traffic state estimation, this study introduces prescriptive analytics—also known as "predict-then-optimize"—for integrating data from Electronic Toll Collection (ETC) systems and traffic sensors. Traditional single-method data fusion techniques are constrained by inherent limitations. For instance, optimization-based methods are generally predicated on prior assumptions that may induce systematic biases, whereas machine learning approaches are frequently criticized for their lack of interpretability and their inability to elucidate underlying traffic mechanisms. To address these limitations, a novel "retrieval and matching" algorithm is proposed that integrates machine learning with optimization. First, the concept of the "state gene" is introduced to encapsulate traffic structural knowledge representing frequently occurring traffic patterns. In the retrieval phase, a heterogeneous graph conventional network is employed to predict potential state genes for a given scenario. In the matching phase, the predicted state genes are utilized to minimize the discrepancy with the current traffic state. This integration not only enhances the interpretability of the estimation process but also endows the optimization component with reverse inference capability through the incorporation of machine learning. Validation using real-world data from the G92 Freeway in Zhejiang, China, demonstrates high accuracy, yielding Mean Absolute Percentage Errors (MAPE) of 1.12−1.65% during peak periods and 1.28−1.67% during off-peak periods.

**Keywords**: traffic state estimation, data fusion, prescriptive analytics, predict-then-optimize, state genes

# 1 Introduction

The rapid increase in vehicles has made traffic congestion a critical concern on freeways worldwide. Addressing this challenge requires a comprehensive understanding of traffic characteristics, elevating traffic state estimation to a pivotal role in modern transportation systems. Traffic state estimation involves the assessment of current parameters—such as speed, density, and volume—to provide real-time insights into freeway performance (Fu et al., 2022; Huang et al., 2022). By understanding traffic flow and congestion patterns on freeways, transportation managers can make evidence-based decisions to enhance safety, reduce travel times, and optimize infrastructure utilization (Wang et al., 2024).

Accurate traffic state estimation on freeways relies on traffic sensing technologies, including Electronic Toll Collection (ETC) systems and various sensors (e.g., video cameras, radar, inductive loop detectors) (Klein, 2019). Each technology provides distinct insights. For example, ETC systems estimate average traffic states between adjacent gantries by matching toll transaction records, whereas sensors assess localized traffic states (e.g., speed and volume) at specific points along the freeway. However, each technology exhibits certain limitations. ETC data are characterized by relatively coarse granularity, while the coverage areas of sensors are limited.

To overcome the limitations of single-source data, existing studies integrate multiple data sources to leverage their respective strengths. Two major categories of methods are frequently adopted in the context of data fusion: machine learning and optimization. Machine learning (Bachmann et al., 2013; Bachmann et al., 2013; Adetiloye and Awasthi, 2019; Khan et al., 2021) excels in identifying patterns from large and complex datasets, thereby enabling the construction of end-to-end mappings from covariates to observed outcomes. However, because such models rely on numerous parameters, their interpretability is often limited, making it challenging to gain insights into the underlying structure of freeway traffic flow. In contrast, optimization methods (Huang et al., 2016; Canepa and Claudel, 2017; Lu et al., 2023;

Zhang et al., 2024a) provide precise solutions to problems with clearly defined objective functions and constraints, ensuring inherent interpretability and a transparent decision-making process (Huang et al., 2023). Nevertheless, the construction of an optimization model typically depends on certain prior assumptions, such as a specific traffic flow model or statistical model. When these assumptions do not hold under diverse real-world conditions, the reliability and generalizability of optimization approaches may be compromised.

To address these limitations, this study introduces a prescriptive analytics approach—also known as "predict-then-optimize" (Bertsimas and Kallus, 2020; Wang and Yan, 2022; Huang et al., 2024a)—for traffic state estimation through multi-source data fusion. This approach integrates machine learning and optimization to capitalize on the strengths of both. In comparison to machine learning alone, prescriptive analytics improves interpretability by incorporating predefined structural knowledge of freeway traffic. Compared to optimization alone, it does not require prior assumptions; instead, a prediction model is employed to estimate the core parameters of the traffic system.

## 1.1 Literature review

### 1.1.1 Application of machine learning and optimization

Machine learning has been widely employed in data fusion due to several advantages: (1) it can produce end-to-end estimation outputs with short inference times, facilitating rapid deployment (Zhao et al., 2021); (2) it generally does not rely on prior assumptions, enabling automatic knowledge extraction from data; (3) it effectively fits complex nonlinear relationships (Seo et al., 2017); and (4) it can be easily adapted to incorporate physical information. In the domain of freeway traffic state estimation, many machine learning models have been explored. These models include traditional regression models (Bachmann et al., 2013), random forests (Adetiloye and Awasthi, 2019), clustering (Wang et al., 2024), and artificial neural networks (Ivan, 1997; Bachmann et al., 2013). More recent studies have proposed advanced methods, such as convolutional neural networks (Khan et al., 2021), physics-informed neural networks

(Lu et al., 2023; Wang and Yang, 2024; Zhang et al., 2024b), graph neural networks (Jin et al., 2024; Lin et al., 2024), and transformer (Huang et al., 2024b). Nevertheless, machine learning—especially deep learning—often exhibits limited interpretability. The "black box" nature of these models makes it challenging to capture and represent the internal mechanisms of traffic states. Although progress has been made to enhance interpretability and explore underlying principles (Fan et al., 2024; Varshney et al., 2018), the black-box issue remains a central concern.

Another widely adopted approach for data fusion is optimization. Optimization methods offer several key advantages: (1) well-defined objective functions and constraints provide clear interpretability and facilitate the integration of structural knowledge about traffic (Zhang et al., 2024a); (2) deliver globally optimal solutions via dedicated solving algorithms; and (3) trace how different factors influence the final output (Chen et al., 2020). Despite these benefits, optimization methods also encounter challenges. First, the construction of an optimization model often depends on prior assumptions, typically involving specific traffic flow models. For instance, Canepa and Claudel (2017) integrate an optimization framework with the Lighthill–Whitham–Richards (LWR) equations to assimilate multiple sensor data streams under particular conditions. Other studies have combined optimization with statistical methods; for example, Zhang et al. (2024a) employ maximum likelihood estimation and maximin likelihood estimation as objective functions within an optimization model. Second, optimization can be computationally intensive. In contrast to most machine learning methods, where training is the main computational bottleneck, significant computation is required for each new instance when using optimization. Zhang et al. (2024a) report that their model needs 700 to 900 seconds to solve a 4.5 km freeway segment, posing challenges for real-time deployment.

### 1.1.2 Integration of machine learning and optimization

Three major themes have emerged in the integration of machine learning and optimization. The first theme is prescriptive analytics, originally introduced by

Bertsimas and Kallus (2020). This approach is naturally applied in settings where decision-making depends on uncertain parameters that must be predicted in advance; once the parameter distributions are estimated, the corresponding optimization problems can be formulated and solved.

The second theme focuses on leveraging machine learning to enhance optimization. Some studies employ machine learning to automatically select hyperparameters for optimization algorithms. For example, Kruber et al. (2017) predict whether Dantzig-Wolfe decomposition should be performed on mixed-integer linear programs (MILPs) to improve solution efficiency, while Bonami et al. (2018) train a classifier to determine whether a quadratic program should be linearized to reduce computation time. In addition, machine learning models have been embedded at various stages of optimization procedures—such as variable selection and cutting-plane generation. Alvarez et al. (2017) use decision trees to approximate optimal variable selection strategies, and Baltean-Lugojan et al. (2018) apply neural networks to identify the most effective cutting planes in the semidefinite programming relaxation.

The third theme centers on using optimization to enhance machine learning. In some studies, optimization has been employed to refine machine learning outputs to ensure that feasibility constraints are satisfied. For instance, Mahmood et al. (2018), Donti et al. (2023), and Wang et al. (2023) integrate optimization algorithms with machine learning layers to drive solutions back into the feasible region. In other studies, optimization is employed to further improve solutions proposed by machine learning models; for example, Han et al. (2023) implement a trust-region search to enhance model outputs.

### 1.2 Aims and contributions

In the literature, most studies employ a single method to fuse multi-source data. As discussed previously, machine learning approaches are constrained by their "black box" nature, limiting interpretability and making it difficult to intuitively reveal structural traffic knowledge on freeways. In contrast, optimization methods rely on

prior assumptions and often encounter computational efficiency issues. Moreover, to the best of the authors' knowledge, no existing study has applied a combined machine learning and optimization method for freeway traffic state estimation by multi-source data fusion.

To address these challenges, this study introduces a prescriptive analytics method, referred to as "predict-then-optimize", for freeway traffic state estimation by multi-source data fusion. In particular, a novel retrieval and matching algorithm is proposed. First, the concept of the state gene is developed to capture recurrent traffic patterns and encode essential structural knowledge. In the retrieval phase, a heterogeneous graph conventional network is used to predict the state genes most likely to apply under the current scenario. In the matching phase, the predicted state genes are aligned with real-time traffic data to minimize discrepancies between the estimated and observed conditions.

From the methodological perspective, this approach is classified as prescriptive analytics because it first predicts the state genes most likely to appear under a given scenario and subsequently fuses the data based on these state genes. The interaction between the retrieval and matching algorithm is advantageous: on the one hand, the retrieval model predicts the core parameters required by the matching model based on the scenario, endowing the matching model with "reverse inference" capabilities. On the other hand, the matching model enables the retrieval model to focus on conveying interpretable structural knowledge (i.e., the state genes) rather than producing end-to-end traffic estimates directly.

The contributions of this study are fourfold. First, a prescriptive analytics framework for freeway traffic state estimation is presented, which fuses data from Electronic Toll Collection (ETC) systems and sensors, thereby harnessing the advantages of both machine learning and optimization. Second, the concept of state genes is proposed within the context of traffic state estimation, offering a novel perspective on traffic structure knowledge and introducing an extraction method based

on the Louvain algorithm. Third, a prediction model called state gene retrieval is introduced, utilizing Learning to Rank (L2R) to identify the most relevant state genes for the current scenario. Fourth, an optimization model named state gene matching is developed to approximate the current traffic states with the retrieved state genes while fusing ETC and sensor observations.

The remainder of this study is organized as follows. Section 2 outlines the features of ETC systems and sensors, presents the decomposition method, and defines state genes. Section 3 describes the proposed retrieval and matching algorithm, including details of state gene extraction, the state gene retrieval model, and the state gene matching model. Section 4 validates the methodology using real-world data. Section 5 concludes the study, and Section 6 provides additional discussion.

## 2 Problem description

This section first presents the definition of traffic state, followed by an overview of the characteristics of ETC systems and sensors. Next, the decomposition method for the original problem is introduced, and finally, the concept of a state gene is defined.

### 2.1 Traffic state

Traffic state can be characterized by various parameters, such as speed, volume, and occupancy. Among these parameters, vehicle speed has traditionally served as a key indicator for evaluating traffic conditions (Mahmud et al., 2017; Zhang et al., 2024a). The Urban Congestion Report (UCR, 2023) defines congestion on road sections as conditions where speeds fall below 90 percent of the free-flow speed (e.g., speeds less than 54 mph when the free-flow speed is 60 mph). Numerous studies have demonstrated the effectiveness of using speed as a key indicator for assessing traffic states. One advantage of using speed is that most traffic monitoring technologies, such as ETC systems and sensors, can directly or indirectly measure vehicle speeds (Seo et al., 2017). Furthermore, speed offers a more precise reflection of traffic state compared to other indicators like volume, which can be ambiguous under certain conditions, and

it is more interpretable than occupancy. Given these benefits, this study also selects speed as the indicator for assessing traffic state on freeway.

### 2.2 Characteristics

This study aims to integrate ETC data and sensor data to enhance traffic state estimation. These two types of monitoring technologies have distinct characteristics. ETC systems do not directly measure vehicle speed; instead, they provide pass-through data, such as license plates information and timestamps (Zhang et al., 2024a). By matching data from two adjacent gantries with the same license plate, the travel time of vehicles between these gantries can be calculated. Subsequently, using the average travel time and the distance between gantries, an estimation of the average traffic state between gantries can be derived. In contrast, sensors refer to roadside detectors, such as video, radars, and ILD (Klein, 2024). These devices can directly measure the speed of nearby vehicles, thereby providing fine-grained traffic state information. However, their coverage area is relatively limited. In summary, ETC systems provide full-coverage but coarse-grained estimation, whereas sensors provide fine-grained but localized-coverage estimation (see Figure 1). This study focuses on fusing these two types of data to achieve a more comprehensive and detailed traffic state estimation.

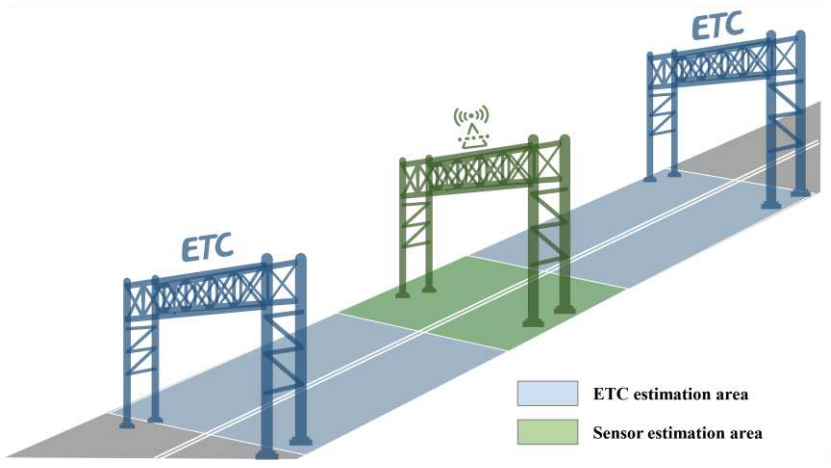



**Figure 1 Estimation areas of ETC and sensor.**

Two key assumptions are adopted:

**Assumption 1**: ETC systems can accurately estimate the average traffic state between adjacent gantries.

**Assumption 2:** Sensors can accurately estimate the traffic state of the specific segments where they are installed.

These assumptions allow the method to focus on the data fusion process without accounting for potential data biases or sensing errors inherent in the individual systems.

### 2.3 Decomposition

Decomposition is employed in this study for two primary reasons. First, partitioning the problem into smaller sub-problems enables the retrieval model to handle dynamic conditions more effectively. Second, solving the complete optimization problem in a single step would be computationally intractable. Zhang et al. (2024) propose a decomposition method for freeway data fusion that divides the original problem into multiple sub-problems based on ETC gantries (see Figure 2). The same approach is adopted herein, whereby the complete problem is split into several sub-problems that serve as the foundation for the subsequent retrieval and matching models.

The advantages of this decomposition are threefold: (1) it preserves the characteristics of ETC estimation by ensuring that each gantry-to-gantry segment is retained within the same sub-problem; (2) It produces sub-problems with similar structures. In each sub-problem, the two ETC gantries at the boundaries provide an estimate of the average traffic state, while varying numbers of sensors within the sub-problem estimate the traffic state of specific segments.; (3) decomposing the original problem into multiple sub-problems allows for independent parallel computation, accelerating the solution process.

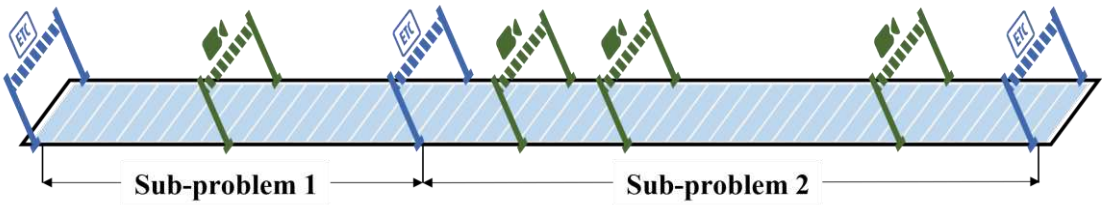



**Figure 2 Divide the original problem into multiple sub-problems.**

## 2.4 State gene

Figure 3 displays the spatial progression of traffic states along a freeway at a given moment, indicating that many segments exhibit remarkably similar patterns. The recurrence of these patterns suggests that, under certain conditions, specific traffic states occur with high frequency. This observation points to the existence of stable structural characteristics within the overall traffic system.

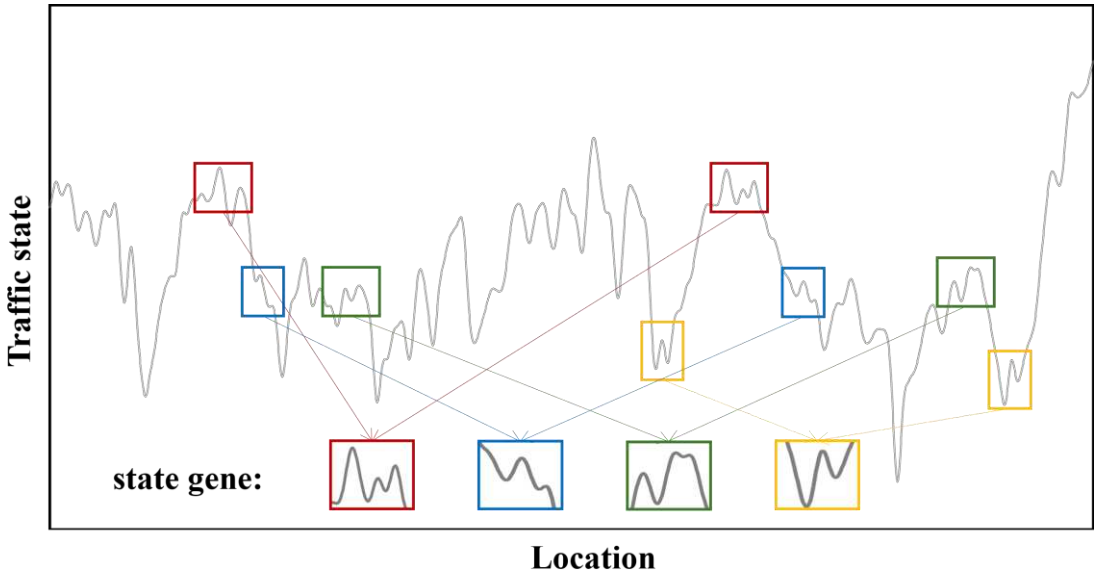



**Figure 3 Illustration of state genes.**

Motivated by these findings, this study introduces the concept of the state gene to capture and describe the structural knowledge encoded in these frequently recurring traffic states. Formally, the state gene is defined as

**Definition 1 (State gene)**

*A state gene is conceptualized as a frequently and repeatedly observed directed sequence of traffic states in historical data. Formally, let* $SG$ *be a length-*$L$ *sequence of consecutive freeway segments, where each segment* $p = 1, \ldots, L$ *is characterized by 25th percentile, mean, and 75th percentile of the traffic state. Thus, the state gene* $SG$ *can be expressed as:*

$$SG = \left( \underbrace{\left(V_1^{25}, V_1^{\mu}, V_1^{75}\right)}_{\text{the first position}}, \underbrace{\left(V_2^{25}, V_2^{\mu}, V_2^{75}\right)}_{\text{the second position}}, \ldots, \underbrace{\left(V_L^{25}, V_L^{\mu}, V_L^{75}\right)}_{\text{the } L\text{-th position}} \right). \quad (1)$$

*where $L$ is the length of the state gene (i.e., the number of contiguous segments), and $V^{25}$, $V^{\mu}$, and $V^{75}$ denote the 25th-percentile, mean, and 75th-percentile traffic states, respectively.*

## 3 Retrieval and matching

This section introduces the methodology of the retrieval and matching algorithm. The overall framework is presented first, followed by a description of the state gene extraction process. Subsequently, the state gene retrieval model and the state gene matching model are introduced.

### 3.1 Overall framework

The primary objective of retrieval and matching algorithm is to determine the traffic state of each segment in a given scenario. As illustrated in Figure 4, the scenario—comprising of the deployment of ETC gantries and sensors, along with their corresponding observations—serves as the input. The retrieval model is employed to rank and retrieve the most relevant state genes for the current scenario. These retrieved state genes are then used as parameters for the downstream matching model, which seeks to minimize the overall deviation between the estimated traffic states and the retrieved state genes.

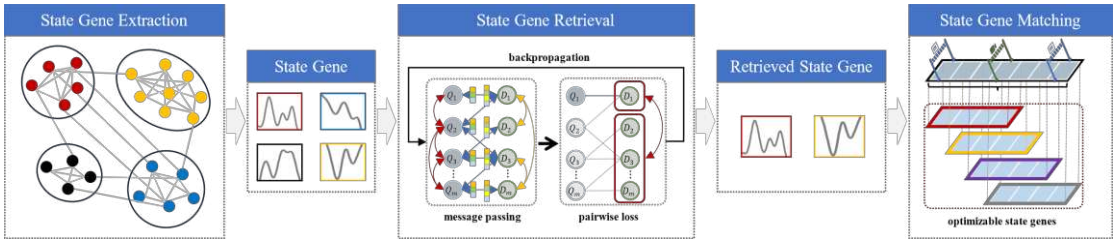



**Figure 4 The framework of the retrieval and matching algorithm.**

### 3.2 State gene extraction

The core concept of retrieval and matching algorithm is the state gene, which denotes frequently observed traffic state patterns. Hence, it is necessary to extract these state genes from historical data. Let $\mathcal{X}$ denote the spatiotemporal training dataset comprising traffic-state observations. Specifically, assume that there are $N$ consecutive road segments with observations collected over $T$ discrete time steps.

The spatiotemporal data can be represented as $\mathcal{X}=\{x_{i,t}:i=1,\ldots,N;t=1,\ldots,T\}$, where $x_{i,t}$ encodes the traffic state at location $i$ and time $t$. From this dataset, a collection of traffic state sequences $\{\mathbf{s}_{i,t}:i=1,\ldots,N-L+1;t=1,\ldots,T\}$ is extracted to capture various space patterns of interest, with each sequence having a length of $L$. Each sequence $\mathbf{s}_{i,t}$ is represented as $\mathbf{s}_{i,t}=[x_{i,t},x_{i+1,t},\ldots,x_{i+L-1,t}]$, $\forall i\in\{1,\ldots,N-L+1\}$, $t\in\{1,\ldots,T\}$. By sliding a temporal window of length $L$ over $\mathcal{X}$, a set of traffic-state sequences is obtained.

Next, the frequency with which these sequences recur in the historical dataset is determined. Let $c(\mathbf{s})$ denote the raw count of a given sequence $\mathbf{s}$, representing the total number of times it appears. Define the initial frequency $f$ $\mathbf{s}$ of $\mathbf{s}$ as

$$f(\mathbf{s}_{i,t})=\frac{c(\mathbf{s}_{i,t})}{\sum_i\sum_t c(\mathbf{s}_{i,t})},\forall i\in\{1,\ldots,N-L+1\},t\in\{1,\ldots,T\}. \tag{2a}$$

Although $f(\mathbf{s})$ characterizes the global importance of sequence $\mathbf{s}$, it may overemphasize sequences composed of highly prevalent individual traffic states. To mitigate this bias, let $p(x_{i+l-1,t})$ represent the proportion of the $l$-th traffic state within sequence $\mathbf{s}_{i,t}$ in the entire dataset $\mathcal{X}$. Define the mean proportion of the states in $\mathbf{s}$ as

$$\bar{p}(\mathbf{s}_{i,t})=\frac{1}{L}\sum_{l=1}^{L}p(x_{i+l-1,t}),\forall i\in\{1,\ldots,N-L+1\},t\in\{1,\ldots,T\}, \tag{2b}$$

To discount sequences that contain overly frequent states while preserving proper normalization, a strictly decreasing function $d(\mathbf{s})$ is introduced:

$$d(\mathbf{s}_{i,t})=\frac{1}{1+\alpha\bar{p}(\mathbf{s}_{i,t})},\forall i\in\{1,\ldots,N-L+1\},t\in\{1,\ldots,T\}, \tag{2c}$$

where $\alpha > 0$ is a hyperparameter governing the degree of discount. The adjusted frequency of sequence $\mathbf{s}$ is then computed by multiplying $f(\mathbf{s})$ by $d(\mathbf{s})$ and renormalizing over all sequences:

$$f'(\mathbf{s}_{i,t}) = \frac{f(\mathbf{s}_{i,t}) d(\mathbf{s}_{i,t})}{\sum_i \sum_t f(\mathbf{s}_{i,t}) d(\mathbf{s}_{i,t})}, \forall i \in \{1,\ldots,N-L+1\}, t \in \{1,\ldots,T\}, \quad (2d)$$

After computing $f'(\mathbf{s})$ for each sequence, the top $K_1$ are selected based on their adjusted frequencies.

A weighted network is constructed by treating each of these top-ranked traffic state sequences as a node. Consider two traffic state sequences $\mathbf{s}_{i,t}$ and $\mathbf{s}_{i',t'}$. Let $\mu_{i,t}$ and $\mu_{i',t'}$ be their respective mean values. The level difference is defined as

$$D_L(\mathbf{s}_{i,t}, \mathbf{s}_{i',t'}) = |\mu_{i,t} - \mu_{i',t'}| = \frac{1}{L}\left|\sum_{l=1}^{l=L} x_{i,t,l} - \sum_{l=1}^{l=L} x_{i',t',l}\right|, \forall i,i' \in \{1,\ldots,N-L+1\}, t,t' \in \{1,\ldots,T\},$$

(3a)

while the trend difference is given by

$$D_T(\mathbf{s}_{i,t}, \mathbf{s}_{i',t'}) = \frac{1}{L}\sum_{l=1}^{L} \left|(x_{i,t,l} - \mu_{i,t}) - (x_{i',t',l} - \mu_{i',t'})\right|, \forall i,i' \in \{1,\ldots,N-L+1\}, t,t' \in \{1,\ldots,T\}.$$

(3b)

These two terms can be combined into a composite difference

$$D(\mathbf{s}_{i,t}, \mathbf{s}_{i',t'}) = D_L(\mathbf{s}_{i,t}, \mathbf{s}_{i',t'}) + D_T(\mathbf{s}_{i,t}, \mathbf{s}_{i',t'}), \forall i,i' \in \{1,\ldots,N-L+1\}, t,t' \in \{1,\ldots,T\}.$$

(3c)

Any pair of sequence $(\mathbf{s}_{i,t}, \mathbf{s}_{i',t'})$ whose composite difference $D(\mathbf{s}_{i,t}, \mathbf{s}_{i',t'})$ does not exceed a threshold $\delta$ is connected by an edge. The weight of this edge is computed as the inverse of composite difference, assigning higher weights to more similar sequences. The Louvain algorithm (Blondel et al., 2008) is applied to the weighted

graph to detect communities of traffic state sequences. The algorithm seeks to maximize the weighted modularity

$$Q = \frac{1}{2W} \sum_{i,t} \sum_{i',t'} \left[ A_{i,t,i',t'} - \frac{k_{i,t} k_{i',t'}}{2W} \right] \delta\left(c_{i,t}, c_{i',t'}\right), \tag{4}$$

where $A_{i,j,i',j'}$ is the edge weight between sequence $\mathbf{s}_{i,t}$ and $\mathbf{s}_{i',t'}$, $W = \frac{1}{2} \sum_{i,t} \sum_{i',t'} A_{i,t,i',t'}$ the total edge weight in the network, $k_{i,t} = \sum_{i,t} A_{i,t,i',t'}$ the strength (i.e., weighted degree) of sequence $\mathbf{s}_{i,t}$, and $\delta\left(c_{i,t}, c_{i',t'}\right)$ equals 1 if $\mathbf{s}_{i,t}$ and $\mathbf{s}_{i',t'}$ are in the same community and 0 otherwise. The algorithm iteratively reassigns each node to neighboring communities to increase the overall modularity $Q$, aggregates the resulting communities into "super-nodes", and repeats this process until no further modularity gain is possible.

Once communities are identified, the mean adjusted frequency rank of the sequences in each community is computed. Only communities whose average rank is less than a threshold $K_2$ are retained. Each retained community is designated as a "state gene" (see Figure 5). For further characterization, key descriptive statistics such as the mean, 25th percentile, and 75th percentile are computed at each position of the state gene to reveal typical levels and the range of variability in traffic conditions.

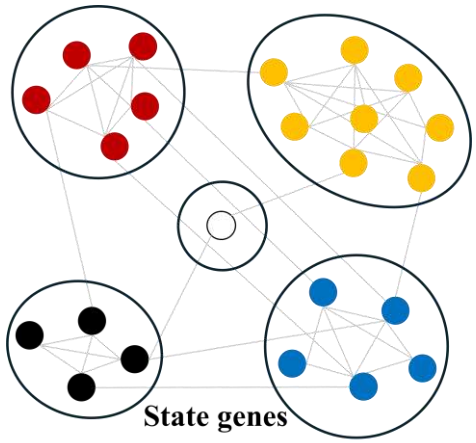



**Figure 5 State gene extraction using the Louvain algorithm.**

The entire procedure for extracting state genes from the spatiotemporal traffic dataset is summarized in Algorithm 1.

**Algorithm 1. State gene extraction.**

**Hyperparameter**: $\alpha$ (discount hyperparameter), $\delta_1$ (similarity threshold), $L$ (state gene length), $K_1$ (number of top traffic state sequence to retain), $K_2$ (state gene average rank threshold)

**Input**: $\mathcal{X}$ (spatiotemporal traffic state dataset of size $N \times T$)

1: Extract all traffic state sequences $\mathbf{s}$ by sliding a $L$-step window over the dataset $\mathcal{X}$.

2: Compute the adjusted frequency $f'(\mathbf{s})$ for all traffic state sequences using Eqs.(2a)−(2d) and retain top $K_1$ sequence.

3: Compute the difference between each pair of sequences using Eqs.(3a)−(3c).

4: Construct the weighted network by treating each sequence as a node. Connect nodes if their composite difference is less than $\delta_1$, with the edge weight defined as the inverse of the difference.

5: Detect sequence communities using the Louvain algorithm to maximize Eq.(4). Communities with an average rank less than $K_2$ are designated as state genes.

6: Compute the descriptive statistics (e.g., mean, 25th percentile, and 75th percentile) at each sequence position of each state gene.

**Output**: a set of identified state genes along with their descriptive statistics

### 3.3 State gene retrieval

The extracted state genes capture all frequently occurring traffic state patterns. However, for a specific scenario—defined by the deployment of ETC gantries, sensors and their observations—not all extracted state genes are suitable. Therefore, a prediction model is required to identify the subset of state genes that are potentially relevant to a given scenario.

#### 3.3.1 Commonalities with information retrieval

This problem can be analogous to the field of information retrieval (IR), which is based on two core concepts: document and query. A document represents a unit of stored information, such as an article, web page, or report, while a query is the search

request submitted by a user to locate relevant documents. For example, when a user enters keywords in a search engine, these keywords serve as the query, and the system ranks and retrieves the relevant documents accordingly. The background of IR is highly analogous to the problem discussed in this study: the scenario can be viewed as the query, and the extracted state genes as the documents. This analogy motivates the term "*state gene retrieval*", which involves using a prediction model to rank and retrieve the most relevant state genes based on the target scenario (see Figure 6).

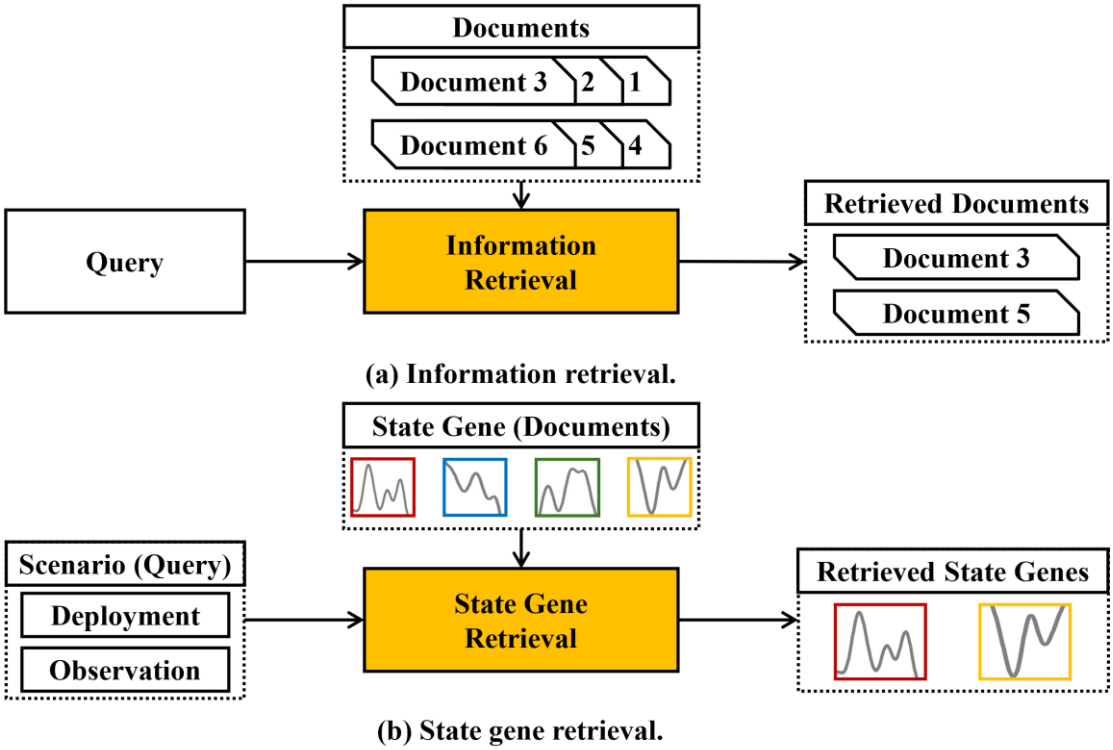



**Figure 6 Comparison between information retrieval and state gene retrieval.**

### 3.3.2 Learning to rank

A common method for ranking and retrieving tasks in IR is Learning to Rank (L2R), which employs machine learning techniques for construct rankings (Ergashev et al., 2023). L2R methods are typically categorized into three types—pointwise, pairwise, and listwise—depending on how the ranking is formulated (Mao et al., 2020). The pointwise approach evaluates each document's relevance to the query independently, the pairwise approach optimizes the relative order by comparing pairs of documents, and the listwise approach directly optimizes the ranking of the entire list.

For state gene retrieval, the objective is to predict which state genes are likely to occur in a given scenario. This task is formulated as a binary classification problem: predicting whether a state gene is relevant to the scenario. The pairwise approach is considered most suitable because the pointwise method does not account for relationships between different state genes, and the listwise method introduces additional computational complexity by explicitly optimizing ranks, which is not necessary for the current application.

Let $\mathcal{G}$ be the set of all state genes available for retrieval, $\mathcal{S}$ the set of scenarios in the training dataset. For each scenario $s \in \mathcal{S}$, an indicator $\gamma \in \{0,1\}$ is defined such that $\gamma = 1$ if state gene $g \in \mathcal{G}$ is observed in scenario $s$ (i.e., $g$ is relevant), and $\gamma = 0$ otherwise. The training dataset is then denoted as $\mathcal{D} \equiv \{(s, g, \gamma)\}_{q \in \mathcal{S}, g \in \mathcal{G}}$. The goal is to train the retrieval model to learn an optimal ranking function from the training data by minimizing a pairwise loss, thereby learning to differentiate relevant state genes from non-relevant ones for each scenario. Graph Convolutional Networks (GCNs) are employed to capture complex relationships between state genes and scenarios. Both state genes and scenarios are treated as nodes within a graph, and edges are used to indicate relationships between them. By leveraging the message passing capabilities of GCNs, the model can propagate information throughout the graph and capture both local and global dependencies.

### 3.3.3 Training process

The training process is designed to learn the correlation between the scenarios and state genes from real-world data. Figure 7 illustrates the overall architecture of this process. Initially, real-world data are used to generate the training set. Initial embeddings are then assigned to both scenarios and state genes, which serve as input representations for subsequent modeling. A heterogeneous graph is constructed to capture the relationships among various nodes. Finally, the GCN model is trained on this heterogeneous graph by minimizing the pairwise loss.

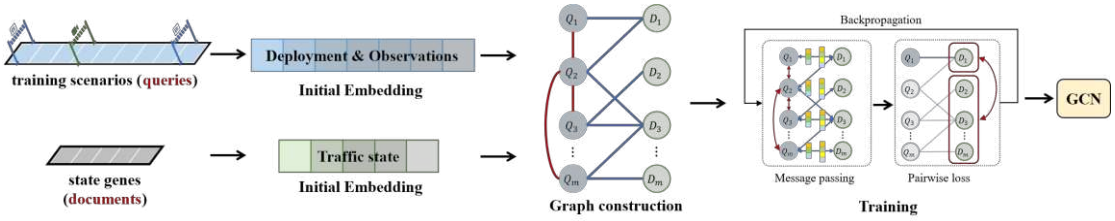



**Figure 7 Architecture of the training process.**

#### *3.3.3.1 Construction of training data*

To expand the dataset and enhance the model's robustness, the following steps are employed to generate the training data: (1) two consecutive ETC gantries are randomly and virtually placed on a freeway segment with fully known traffic states, ensuring the distance between these gantries remains within a reasonable range; (2) within the segment between these two ETC gantries, a number of sensors are randomly and virtually deployed. The sensor count is determined based on the distance between the gantries, in order to maintain a realistic deployment density; (3) it is determined which state genes appear in the resultant scenario, thereby linking specific traffic states to the given configuration. By repeating these steps, multiple scenarios and their corresponding state genes can be generated.

#### *3.3.3.2 Initial embedding*

Before constructing the GCN, it is necessary to encode both scenarios and state genes into appropriate representations. For each scenario, the observed data include the ETC systems observation (i.e., the average traffic state across all segments between two consecutive ETC gantries), and the sensors observations (i.e., the traffic states at the specific segments where sensors are located), together with the deployment positions of the gantries and sensors. Therefore, the initial embedding of a scenario can be formulated as

$$H^{sce} = \left( \underbrace{TS^{etc}}_{\text{ETC observations}}, \underbrace{TS_1^{sens}, \cdots, TS_D^{sens}}_{\text{sensor observations}}, \underbrace{P_1^{sens}, \cdots, P_D^{sens}}_{\text{sensor place}}, \underbrace{N}_{\text{number of road segment}} \right), \tag{5}$$

where $TS^{etc}$ denotes the observed traffic state by ETC systems, $TS_d^{sens}$ denotes the observed traffic state at sensor $d$, $P_d^{sens}$ represents the position of sensor $d$, and $N$ indicates the total number of road segments considered.

According to Definition 1, a state gene represents the traffic state distribution of a contiguous freeway segment, including the 25th percentile, the mean, and the 75th percentile of the traffic state at each position. These statistics can be directly employed as the initial embedding for the state gene:

$$H^{sg}=\left(\underbrace{V_1^{25},V_1^{\mu},V_1^{75}}_{\text{the first position feature}},\underbrace{V_2^{25},V_2^{\mu},V_2^{75}}_{\text{the second position feature}},\ldots,\underbrace{V_L^{25},V_L^{\mu},V_L^{75}}_{\text{the }L\text{-th position feature}}\right). \tag{6}$$

*3.3.3.3 Graph construction*

A heterogeneous graph $\mathcal{G}^H=(\mathcal{V},\mathcal{E},\mathcal{R})$ is constructed to represent the relationships among scenarios and state genes, where $\mathcal{V}$ is the set of nodes, $\mathcal{E}$ the set of edges, and $\mathcal{R}$ the set of edge relations. For the retrieval model, two categories of nodes are included: scenario nodes $\mathcal{V}^{sce}$ and state gene nodes $\mathcal{V}^{sg}$. The scenario nodes can be further classified based on sensor count. Based on the types of nodes at the ends of edges, $\mathcal{R}=\left\{\left\langle\mathcal{V}^{sce},\mathcal{V}^{sce}\right\rangle,\left\langle\mathcal{V}^{sce},\mathcal{V}^{sg}\right\rangle\right\}$ contains two relations, containing the connections between scenarios $\left\langle\mathcal{V}^{sce},\mathcal{V}^{sce}\right\rangle$, and the connections between scenarios and state genes $\left\langle\mathcal{V}^{sce},\mathcal{V}^{sg}\right\rangle$. The links in $\left\langle\mathcal{V}^{sce},\mathcal{V}^{sg}\right\rangle$ are determined by $\gamma$. If $\gamma=1$ (i.e., the state gene appears in the scenario), an edge is connected between the corresponding scenario and state gene nodes. The edges in $\left\langle\mathcal{V}^{sce},\mathcal{V}^{sce}\right\rangle$ are based on the weighted Manhattan Distance between their embeddings,

$$\begin{aligned}d\left(H_1^{sce},H_2^{sce}\right)=&\underbrace{\left|H_1^{sce}(1)-H_1^{sce}(1)\right|}_{\text{ETC observations}}+\underbrace{\beta_1\sum_{i=2}^{D+1}\left|H_1^{sce}(i)-H_1^{sce}(i)\right|}_{\text{sensor observations}}+\\&\underbrace{\beta_2\sum_{i=D+2}^{2D+1}\left|H_1^{sce}(i)-H_2^{sce}(i)\right|}_{\text{sensor locations}}+\underbrace{\beta_3\left|H_1^{sce}(2D+2)-H_2^{sce}(2D+2)\right|}_{\text{segment count}},\end{aligned} \tag{7}$$

where $\beta_1$, $\beta_2$ and $\beta_3$ are weighting coefficients controlling the relative importance of different components. When the distance is less than a specified threshold $\xi$, an edge is formed between two scenario nodes.

*3.3.3.4 Heterogenous Graph Neural Networks*

One way to handle heterogeneous graphs is the Relational Graph Convolutional Network (RGCN) (Schlichtkrull et al., 2018), which models information exchange across multi-relational structures. In an RGCN, each relation type in the graph is learned through a distinct set of parameters, enabling the model to capture the unique interactions associated with different edge types.

Consider a node $v$ whose representation in the $l$-th layer is denoted as $\tilde{H}_v^l$. The updated representation $\tilde{H}_v^{l+1}$ is computed as

$$\tilde{H}_v^{l+1} = \sigma\left( W_0^l \tilde{H}_v^l + \sum_{r\in\mathcal{R}} \sum_{u\in\mathcal{N}_r(v)} E_{v,u}^r W_r^l \tilde{H}_u^l \right), \tag{8}$$

where $\mathcal{R}$ is the set of all relation (edge) types, $\mathcal{N}_r(v)$ the set of neighbors of node $v$ connected by edges of relation type $r$, $W_0^l$ and $W_r^l$ the learnable parameter metrics associated with the $l$-th layer (with $W_r^l$ capturing the contribution of relation type $r$), $E_{v,u}^r$ is an edge-specific scalar weight for the edge linking nodes $v$ and $u$ under relation $r$, $\sigma(\cdot)$ a non-linear activation function.

*3.3.3.5 Pairwise loss function*

By applying the RGCN described above, each scenario node and state gene node is associated with a final embedding. These embeddings are aligned to the same dimensionality by configuring the matrices $W_0^l$ and $W_r^l$. Let $\mathcal{G}_q^+$ represent the set of state genes observed in scenario $q$ (i.e., $\gamma = 1$), and let $\mathcal{G}_q^-$ represent the set of state genes that do not observed (i.e., $\gamma = 0$).

A pairwise loss function is then defined to encourage the scenario embedding $\tilde{H}_q^{sce}$ to be closer to the embeddings of the observed state genes $\tilde{H}_{d^+}^{sg}$ than to those of the unobserved state genes $\tilde{H}_{d^-}^{sg}$. Formally, the loss $\mathcal{L}$ is given by

$$\mathcal{L} = -\sum_{q\in\mathcal{Q}} \sum_{d^+\in\mathcal{G}_q^+} \sum_{d^-\in\mathcal{G}_q^-} \left( \frac{\exp\left(\mathrm{sim}_{\cos}\left(\tilde{H}_q^{sce}, \tilde{H}_{d^+}^{sg}\right)\right)}{\exp\left(\mathrm{sim}_{\cos}\left(\tilde{H}_q^{sce}, \tilde{H}_{d^+}^{sg}\right)\right) + \exp\left(\mathrm{sim}_{\cos}\left(\tilde{H}_q^{sce}, \tilde{H}_{d^-}^{sg}\right)\right)} \right), \tag{9}$$

where $\mathrm{sim}_{\cos}(\cdot,\cdot)$ denotes the cosine similarity between the embedding of scenario and a state gene node. By minimizing this loss, the training process encourages the model to assign higher similarity scores to relevant state genes than to irrelevant ones.

### 3.3.4 Prediction process

The prediction process employs the trained GCN model to determine which state genes are likely to appear in a new scenario. The task is treated as a link prediction problem. As illustrated in Figure 8, the process begins by generating initial embeddings for both the new scenario and all extracted state gene, followed by graph sampling to select the relevant scenario node, and then applies the trained GCN to the sampled subgraph, thus obtaining the final embeddings for the scenario and state gene nodes. Subsequently, similarity is computed between the scenario and state gene embeddings, and the state gene with the highest similarity is regarded as the most relevant matching the current scenario.

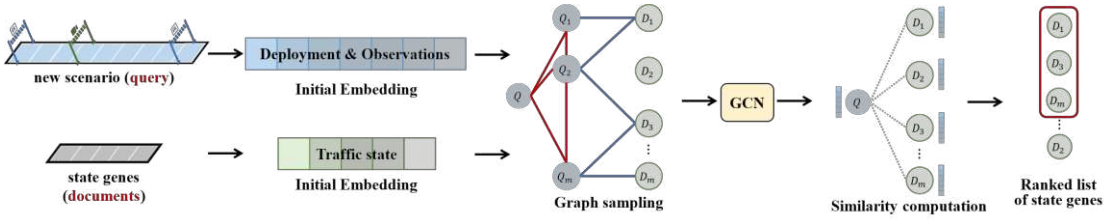



**Figure 8 Architecture of the prediction process.**

Both the initial embedding and the graph convolution are consistent with the methods described in the training phase, while the specific details of graph sampling and similarity computation are presented in subsequent sections.

#### *3.3.4.1 Graph sampling*

Graph sampling is introduced to accelerate prediction in the trained GCN, which contains many scenario nodes. Using the entire graph for prediction would `incur significant computational overhead. In general, graph sampling techniques aim to select a subset of nodes that preserves the essential structural information while reducing computation load. In the proposed framework, two types of nodes exist in the GCN: scenario nodes and state gene nodes. Because the number of state gene nodes is limited and the majority of nodes are scenario nodes, all state gene nodes are retained, while only a subset of scenario nodes is sampled. Specifically, the weighted Manhattan distance (see Eq.(7)) is computed between scenario nodes, and any scenario node exhibiting a distance smaller than a given threshold $\xi$ is preserved and linked accordingly.

#### *3.3.4.2 Similarity computation*

Once the final node embeddings are obtained via graph convolution, the next step is to compute the cosine similarity between the target scenario embedding and all extracted state gene embeddings. Any state gene node whose similarity score exceeds a predefined threshold $K$ is regarded as a retrieved state gene for the specified scenario.

### 3.4 State gene matching

The retrieval model predicts the most relevant state genes for a certain scenario. By leveraging these relevant state genes, ETC and sensor data are fused to obtain a comprehensive estimation of the traffic state. The core idea is to adjust the estimated traffic state to match the retrieved state genes as closely as possible while satisfying the constraints imposed by ETC systems and sensors. This optimization method is named *state gene matching*. For charity, Table 1 summarizes the list of notations used in this section.

**Table 1 List of notations.**

| Notation | Description |
|---|---|

***Sets***

| | |
|---|---|
| $\mathcal{D}$ | Set of road segments where sensors are located, indexed by $d$. |
| $\mathcal{G}^{opt}$ | Set of optimizable state genes, indexed by $j$, and $\mathcal{G}^{opt}=\{1,\ldots,I-L+1\}$. |
| $\mathcal{G}^{s}$ | Set of retrieved state genes for the current scenario, indexed by $k$. |
| $\mathcal{I}$ | Set of road segments included in the interval between two consecutive ETC gantries, indexed by $i$, and $\mathcal{I}=\{1,2,\cdots,I\}$. |
| $\mathcal{P}$ | Set of positions in state genes, indexed by $p$, and $\mathcal{P}=\{1,\ldots,L\}$. |

***Parameters***

| | |
|---|---|
| $V^{etc}$ | Traffic state measured by ETC systems. |
| $V_d^{sens}$ | Traffic state measured by the sensors located in $d^{th}$ road segments. |
| $V_{k,p}^{25}$ | 25th percentile of the traffic state at the $p$-th position of the $k$-th state gene. |
| $V_{k,p}^{75}$ | 75th percentile of the traffic state at the $p$-th position of the $k$-th state gene. |
| $V^{max}$ | Maximum traffic state. |
| $\delta_{down,up}^{95}$ | 95th percentile of the traffic state difference between the downstream and upstream road segments. |
| $\delta_{up,down}^{95}$ | 95th percentile of the traffic state difference between the upstream and downstream road segments. |

***Main Variables***

| | |
|---|---|
| $v_i$ | Continuous variable indicating the estimated traffic state of the $i$-th road segment. |
| $\delta_{j,k}$ | Continuous variable indicating the overall deviation between the $j$-th optimizable state gene and $k$-th retrieved state gene. |
| $\sigma_{j,k,p}$ | Continuous variable indicating the deviation between the $p$-th position of the $j$-th optimizable state gene and the $p$-th position of $k$-th retrieved state gene. |
| $\Delta_j$ | Continuous variable indicating the minimum deviation between the $j$-th optimizable state gene and all retrieved state genes. |

***Auxiliary Variables***

| | |
|---|---|
| $b_{j,k}$ | Auxiliary binary variable that equals 1 if the $k$-th retrieved state gene provides the minimum derivation with the $k$-th optimizable state genes, and 0 otherwise. |
| $w^{l}_{j,k,m}$ | Auxiliary continuous variable indicating the weight assigned to the $(j+m)$-th traffic state belonging to $l$-th piecewise interval. |
| $z^{l}_{j,k,m}$ | Auxiliary binary variable that equals 1 if the $(j+m)$-th traffic state belongs to the $l$-th interval, and 0 otherwise. |

### 3.4.1 Model formulation

To quantify the deviation between a target scenario and the set of retrieved state genes, the road network in the scenario is partitioned into contiguous fragments of length $L$, each referred to as an *optimizable state gene* (see Figure 9).

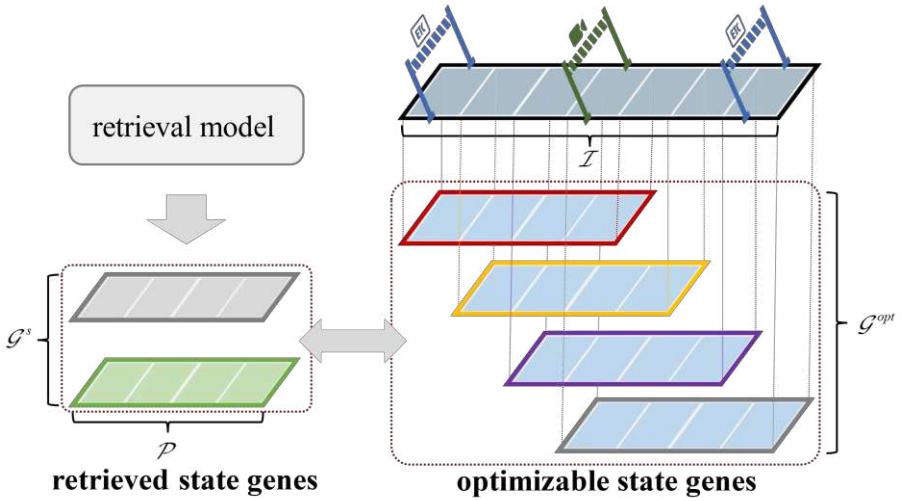



**Figure 9 Illustration of optimizable state gene.**

Formally, let $\mathcal{P}$ be the set of positions within a state gene, $\mathcal{I}=\{1,2,\cdots,I\}$ the set of road segments, and $\mathcal{G}^{opt}=\{1,\ldots,I-L+1\}$ the set of all optimizable state genes. Each fragment $j\in\mathcal{G}^{opt}$ corresponds to a set of consecutive segments $\{j, j+1,\ldots, j+L-1\}$. Let $\mathcal{G}^s$ represent the set of retrieved state genes obtained from the retrieval model, with each retrieved gene indexed by $k$. For each retrieved state gene $k$, let $V^{25}_{k,p}$ and $V^{75}_{k,p}$ be the 25$^{\text{th}}$- and 75$^{\text{th}}$- percentile traffic states at the $p$-th

position, respectively. Denote by $v_i$ the continuous traffic state variable of segment $i$, and let $\mathcal{D} \subseteq \mathcal{I}$ be the set of road segments where sensors are located, with $V_d^{sens}$ denoting the sensed traffic state. The matching model is formulated as

**State Gene Matching (Nonlinear form)**

$$\min_{\Delta, v, \delta, \sigma} \sum\nolimits_{j \in \mathcal{G}^{opt}} \Delta_j \tag{10a}$$

subject to

$$\frac{1}{I} \sum_{i \in \mathcal{I}} v_i = V^{etc}, \tag{10b}$$

$$v_d = V_d^{sens}, \forall d \in \mathcal{D}, \tag{10c}$$

$$v_i - v_{i-1} \le \delta_{95}^{-}, \forall i \in \mathcal{I} \setminus \{1\}, \tag{10d}$$

$$v_i - v_{i+1} \le \delta_{95}^{+}, \forall i \in \mathcal{I} \setminus \{I\}, \tag{10e}$$

$$\Delta_j = \min\nolimits_{k \in \mathcal{G}^s} \delta_{j,k}, \forall j \in \mathcal{G}^{opt}, \tag{10f}$$

$$\delta_{j,k} = \sum\nolimits_{p \in \mathcal{P}} \sigma_{j,k,p}, \forall j \in \mathcal{G}^{opt}, k \in \mathcal{G}^s, \tag{10g}$$

$$\sigma_{j,k,p} = \begin{cases} V_{k,p}^{25} - v_{j+p}, & \text{if } v_{j+p} \le V_{k,p}^{25}, \\ v_{j+p} - V_{k,p}^{75}, & \text{if } v_{j+p} \ge V_{k,p}^{75}, \\ 0, & \text{otherwise,} \end{cases} \forall j \in \mathcal{G}^{opt}, k \in \mathcal{G}^s, p \in \mathcal{P}, \tag{10h}$$

where the objective function (10a) aims to minimize the total deviation $\Delta_j$ between each optimizable state gene $j$ and its closest reference gene in $\mathcal{G}^s$. Constraint (10b) ensures that the average traffic state between adjacent ETC gantries corresponds with the observed ETC measurement $V^{etc}$. Similarly, constraint (10c) enforces that the estimated traffic state at sensor-instrumented segments match the corresponding sensor measurements $V_d^{sens}$. Constraints (10d) and (10e), as proposed by Zhang et al. (2024), limit abrupt traffic-state fluctuations across adjacent segments to promote spatial smoothness. Constraint (10f) forces each optimizable state gene $j$ to align as closely as possible with its nearest reference gene, while constraint (10g) aggregates the deviations $\sigma_{j,k,p}$ position-wise. Constraint (10h) ensures zero deviation when the

estimated traffic state $v_{j+p}$ lies within $\left[V_{k,p}^{25}, V_{k,p}^{75}\right]$, but penalizes states below the 25th percentile or above the 75th percentile of the reference distribution.

### 3.4.2 Linearization

This formulation belongs to the class of Nonlinear Programming (NLP) problems. The nonlinearities arise from two sources: (1) the minimization function in constraint (10f) and (2) the piecewise linear function in constraint (10h). To enable an exact solution via solvers, these constraints are linearized to transform the model into the MILP problem.

First, introduce an auxiliary binary variable $b_{j,k}$ to indicate whether the $k$-th retrieved state gene has the smallest difference from the $j$-th optimizable state gene. Constraint (10f) is equivalently transformed as

$$\Delta_j + M\left(1 - b_{j,k}\right) \geq \delta_{j,k}, \forall j \in \mathcal{G}^{opt}, k \in \mathcal{G}^{s}, \tag{11a}$$

$$\sum\nolimits_{k \in \mathcal{G}^{s}} b_{j,k} = 1, \forall j \in \mathcal{G}^{opt}, \tag{11b}$$

where $M$ is a sufficient large constant. However, if $M$ is set too high, this could cause numerical instabilities in the solver. Consequently, the goal is to choose $M$ as small as possible without affecting the feasibility of any valid solutions. This is achieved by specifying a distinct $M_k$ for each retrieved state gene, given by

$$M_k = \sum\nolimits_{p \in \mathcal{P}} \max\left\{V_{k,p}^{25}, V^{max} - V_{k,p}^{75}\right\}, \forall k \in \mathcal{G}^{s}. \tag{11c}$$

Subsequently, constraints (11a) are reformulated as

$$\Delta_j + \left(\sum\nolimits_{p \in \mathcal{P}} \max\left\{V_{k,p}^{25}, V^{max} - V_{k,p}^{75}\right\}\right)\left(1 - b_{j,k}\right) \geq \delta_{j,k}, \forall j \in \mathcal{G}^{opt}, k \in \mathcal{G}^{s}. \tag{11d}$$

Next, the piecewise linear function is addressed by introducing an auxiliary continuous variable $w_{j,k,m}^{l}$, which denotes the "weight" assigned to the $(j+m)$-th traffic state within the $l$-th piecewise interval. In parallel, an auxiliary binary variable $z_{j,k,p}^{l}$ indicates whether the $(j+m)$-th traffic state belongs to the $l$-th interval. With these variables, constraint (10g) is equivalently linearized as

$$\sum_{l=1}^{4} w_{j,k,p}^{l} = 1, \forall j \in \mathcal{G}^{opt}, k \in \mathcal{G}^{s}, p \in \mathcal{P}, \tag{12a}$$

$$\sum_{l=1}^{3} z_{j,k,p}^{l} = 1, \forall j \in \mathcal{G}^{opt}, k \in \mathcal{G}^{s}, p \in \mathcal{P}, \tag{12b}$$

$$w_{j,k,p}^{1} \le z_{j,k,p}^{1}, \forall j \in \mathcal{G}^{opt}, k \in \mathcal{G}^{s}, p \in \mathcal{P}, \tag{12c}$$

$$w_{j,k,p}^{2} \le z_{j,k,p}^{1} + z_{j,k,p}^{2}, \forall j \in \mathcal{G}^{opt}, k \in \mathcal{G}^{s}, p \in \mathcal{P}, \tag{12d}$$

$$w_{j,k,p}^{3} \le z_{j,k,p}^{2} + z_{j,k,p}^{3}, \forall j \in \mathcal{G}^{opt}, k \in \mathcal{G}^{s}, p \in \mathcal{P}, \tag{12e}$$

$$w_{j,k,p}^{4} \le z_{j,k,p}^{3}, \forall j \in \mathcal{G}^{opt}, k \in \mathcal{G}^{s}, p \in \mathcal{P}, \tag{12f}$$

$$\sigma_{j,k,p} = V_{k,p}^{25} w_{j,k,p}^{1} + \left(V^{max} - V_{k,p}^{75}\right) w_{j,k,p}^{4}, \forall j \in \mathcal{G}^{opt}, k \in \mathcal{G}^{s}, p \in \mathcal{P}, \tag{12g}$$

$$v_{j+p} = V_{k,p}^{25} w_{j,k,p}^{2} + V_{k,p}^{75} w_{j,k,p}^{3} + V^{max} w_{j,k,p}^{4}, \forall j \in \mathcal{G}^{opt}, k \in \mathcal{G}^{s}, p \in \mathcal{P}. \tag{12h}$$

Combining the above constraints, the linear form of matching model is constructed as follows.

**State Gene Matching (Linear form)**

$$\min_{\Delta, v, \delta, \sigma, b, w, z} \sum\nolimits_{j \in \mathcal{G}^{opt}} \Delta_j \tag{13a}$$

subject to

$$\frac{1}{I} \sum_{i \in \mathcal{I}} v_i = V^{etc}, \tag{13b}$$

$$v_d = V_d^{sens}, \forall d \in \mathcal{D}, \tag{13c}$$

$$v_i - v_{i-1} \le \delta_{95}^{-}, \forall i \in \mathcal{I} \setminus \{1\}, \tag{13d}$$

$$v_i - v_{i+1} \le \delta_{95}^{+}, \forall i \in \mathcal{I} \setminus \{I\}, \tag{13e}$$

$$\Delta_j + \left(\sum\nolimits_{m=0}^{L-1} \max\left\{V_{k,m}^{25}, V^{max} - V_{k,m}^{75}\right\}\right)\left(1 - b_{j,k}\right) \ge \delta_{j,k}, \forall j \in \mathcal{G}^{opt}, k \in \mathcal{G}^{s}, \tag{13f}$$

$$\sum\nolimits_{k \in \mathcal{G}^{s}} b_{j,k} = 1, \forall j \in \mathcal{G}^{opt}, \tag{13g}$$

$$\delta_{j,k} = \sum\nolimits_{m=1}^{L} \sigma_{j,k,m}, \forall j \in \mathcal{G}^{opt}, k \in \mathcal{G}^{s}, \tag{13h}$$

$$\sum_{l=1}^{4} w_{j,k,p}^{l} = 1, \forall j \in \mathcal{G}^{opt}, k \in \mathcal{G}^{s}, p \in \mathcal{P}, \tag{13i}$$

$$\sum_{l=1}^{3} z_{j,k,p}^{l} = 1, \forall j \in \mathcal{G}^{opt}, k \in \mathcal{G}^{s}, p \in \mathcal{P}, \tag{13j}$$

$$w_{j,k,p}^{1} \le z_{j,k,p}^{1}, \forall j \in \mathcal{G}^{opt}, k \in \mathcal{G}^{s}, p \in \mathcal{P}, \tag{13k}$$

$$w_{j,k,p}^{2} \le z_{j,k,p}^{1} + z_{j,k,p}^{2}, \forall j \in \mathcal{G}^{opt}, k \in \mathcal{G}^{s}, p \in \mathcal{P}, \tag{13l}$$

$$w_{j,k,p}^{3} \le z_{j,k,p}^{2} + z_{j,k,p}^{3}, \forall j \in \mathcal{G}^{opt}, k \in \mathcal{G}^{s}, p \in \mathcal{P}, \tag{13m}$$

$$w_{j,k,p}^{4} \le z_{j,k,p}^{3}, \forall j \in \mathcal{G}^{opt}, k \in \mathcal{G}^{s}, p \in \mathcal{P}, \tag{13n}$$

$$\sigma_{j,k,p} = V_{k,p}^{25} w_{j,k,p}^{1} + \left(V^{max} - V_{k,p}^{75}\right) w_{j,k,p}^{4}, \forall j \in \mathcal{G}^{opt}, k \in \mathcal{G}^{s}, p \in \mathcal{P}, \tag{13o}$$

$$v_{j+p} = V_{k,p}^{25} w_{j,k,p}^{2} + V_{k,p}^{75} w_{j,k,p}^{3} + V^{max} w_{j,k,p}^{4}, \forall j \in \mathcal{G}^{opt}, k \in \mathcal{G}^{s}, p \in \mathcal{P}. \tag{13p}$$

The linear form of matching model constitutes a MILP problem that can be solved efficiently with a MILP solver.

## 4 Case study

In this section, the performance of the retrieval and matching algorithm is evaluated by using real-world data. The algorithm is implemented in Python, and all experiments are conducted on an Intel Core i5-12400H CPU at 2.5 GHz with 16 GB RAM. The matching model is solved using the commercial solver GUROBI 10.0.1.

### 4.1 Experimental setup

This study utilizes real-world data from the G92 Freeway in Zhejiang, China. The traffic state data are derived from floating car data recorded between July 4 and July 13, 2023, with a temporal granularity of one minute. The coverage extends from kilometer markers 194.3 to kilometer markers 252.8, spanning 58.5 km and subdivided into 585 segments of 100 meters each. Specifically, data collected on July 5, 2023, covering both directions between kilometer markers 218 and kilometer markers 252.8, are used as the training set for the retrieval model. Data collected on July 6 and 7, 2023, for both directions between kilometer markers 194.3 and kilometer markers 217.1, are used as the testing set. The testing set comprises a total of 5,760 instances, with each instance containing 228 segments. This arrangement ensures that no temporal or spatial overlap exists between the training set and the testing scenarios. Within the testing range, seven ETC gantries are located at kilometer markers 194.3, 198.9, 206.8, 231.4, 213.9, 216.1, and 217.1, respectively. Additionally, 15 sensors are distributed at kilometer markers

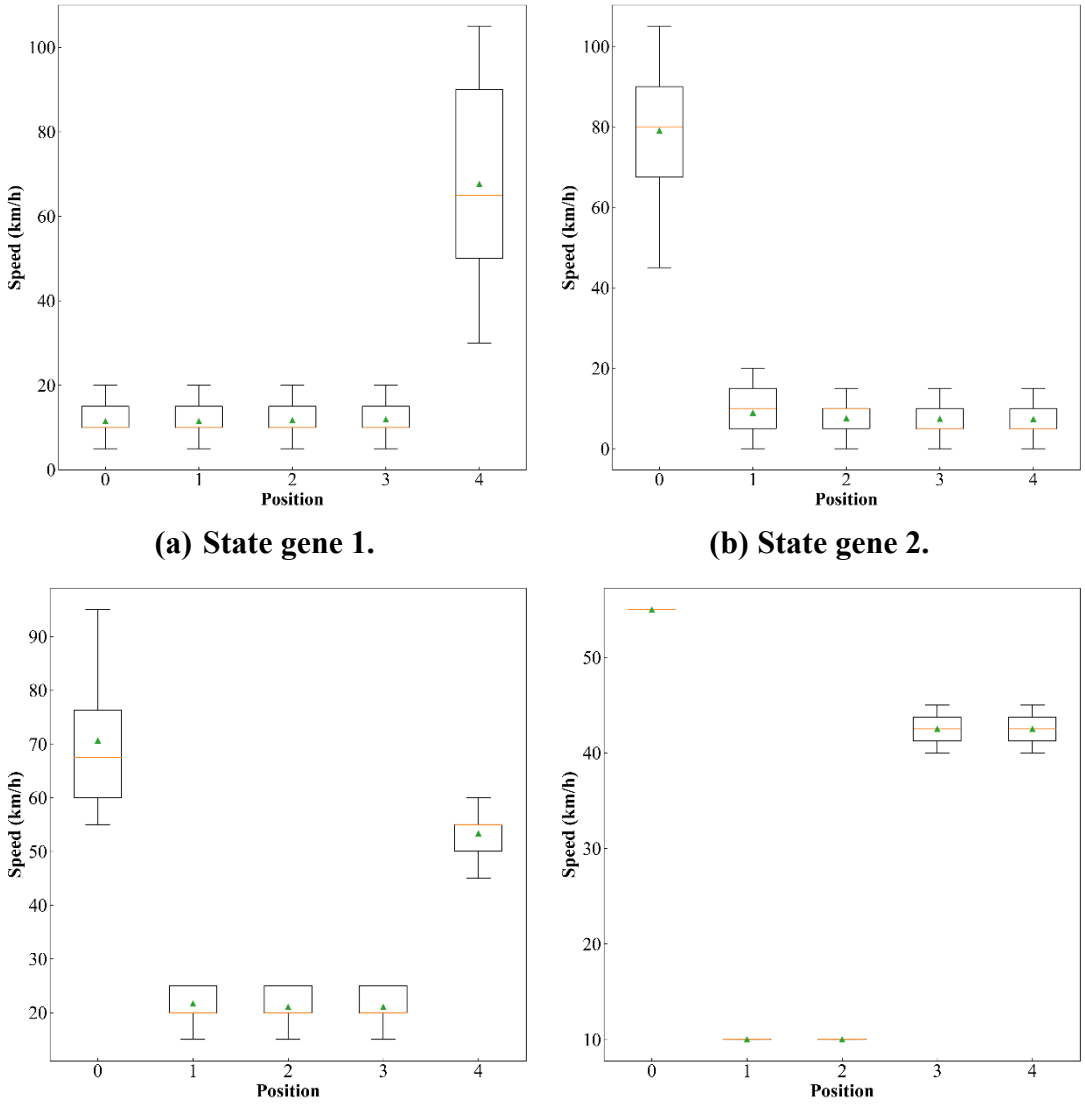
194.6, 196.1, 197.6, 199.1, 202.0, 202.8, 205.7, 207.2, 208.7, 209.4, 210.2, 211.7, 213.1, 214.6, and 216.3, with an average spacing of approximately 1.5 km.

### 4.2 Extracted state genes

In this study, each state gene is defined to encompass five consecutive freeway segments, covering a total distance of 500 meters. By following the procedure described in Algorithm 1, 45 distinct state genes are extracted from the training dataset.

Figure 10 illustrates four representative examples of state genes. In these figures, each state gene is depicted by the speed distributions across five segments (labeled positions 1 through 5). In Figure 10(a), the speeds at positions 1 through 4 exhibit relatively low, indicating congestion, while position 5 exhibits free-flow conditions, suggesting a rapid transition from congestion to free flow. By contrast, Figure 10(b) shows free flow at position 1, followed by four congested segments, reflecting a scenario where an initially uncongested segment evolves into sustained congestion. In Figure 10(c), a moderate to high speed is observed at position 1, followed by heavy congestion at positions 2 and 3, a modest recovery at position 4, and a return to higher speeds at position 5. Meanwhile, Figure 10(d) presents a multi-stage pattern: free-flow conditions at position 1, congestion at positions 2 and 3, and moderate speeds at positions 4 and 5. Collectively, these examples capture a range of recurring traffic speed patterns—from abrupt transitions between free-flow and congestion to more gradual shifts—offering structured insights into how congestion emerges, persists, and ultimately resolves along the freeway.

Overall, these examples show how state genes capture recurring spatial patterns in traffic states over short distances. Each state gene reflects a common configuration, whether it represents abrupt changes between free-flow and congestion or gradually shifting speeds. By encapsulating these frequently observed conditions, state genes provide structured insights into the processes of congestion formation, persistence, and resolution along the freeway.




**(a) State gene 1.** **(b) State gene 2.**

**(c) State gene 3.** **(d) State gene 4.**

**Figure 10 Illustration of state gene.**

## 4.3 Retrieval model results

In this section, the results of the retrieval model are presented. The retrieval model is designed to predict the relevant state genes from historical traffic data, which are subsequently utilized in the matching model. As a LTR model, its primary goal is to rank potential state genes according to their relevance rather than provide exact numerical frequency predictions. Consequently, evaluation is based on ranking metrics rather than traditional numerical measures like MAE or MAPE.

The performance of the retrieval model is assessed using several standard LTR metrics, including Mean Reciprocal Rank (MRR), Normalized Discounted Cumulative

Gain (NDCG), Mean Average Precision (MAP), and Precision at K (P@K). MRR evaluates the rank of the first relevant state gene, while NDCG measures the overall quality of the ranking by accounting for the positions of all relevant items. MAP aggregates the average precision across all relevant state genes, providing an overall measure of ranking performance across multiple instances. Finally, P@K calculates the proportion of relevant items within the top K predictions, offering insight into the model's performance when only the top-ranked results are considered.

Table 2 summarizes the model's performance on the test set. The MRR of 0.77 indicates that, on average, the most relevant state gene is positioned near the top of the ranked list. NDCG scores remain above 0.74 across multiple cutoffs, suggesting that high-relevance items are ranked consistently well. The MAP of 0.75 further demonstrates the model's strong overall ranking capability. Lastly, the precision values, ranging from 61% to 74% across various cutoffs, demonstrate that a substantial proportion of the model's top-ranked results are genuinely relevant state genes, supporting the model's effectiveness in identifying the most representative state genes for traffic state estimation.

**Table 2 Performance of the retrieval model.**

| **MRR** | **NDCG@1** | **NDCG@3** | **NDCG@5** |
|---|---|---|---|
| 0.77 | 0.74 | 0.81 | 0.80 |
| **MAP** | **Precision@1** | **Precision@3** | **Precision@5** |
| 0.75 | 74% | 67% | 61% |

One reason for the strong performance of the retrieval model is that the Louvain algorithm effectively reduces similarity among different state genes, enabling clear mapping relationships within the model. In addition, the number of scenarios greatly exceeds the number of state genes in the training set, providing numerous references for similarity matching. This abundance of scenarios facilitates the identification of relevant analogs for new conditions, further enhancing retrieval accuracy.

## 4.4 Estimation results

This section evaluates the performance of the retrieval and matching algorithm Section 4.4.1 presents the estimation results, while Section 4.3.2 compares the proposed method with several alternate data fusion approaches.

### 4.4.1 Estimation results

After extract all state genes and training the retrieval model, the performance of the retrieval and matching algorithm is tested. Two metrics, Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE), are used to evaluate the estimation accuracy, defined as

$$\mathrm{MAE} = \frac{1}{I}\sum_{i\in\mathcal{I}}\left|\hat{v}_i - v_i\right|, \tag{14a}$$

$$\mathrm{MAPE} = \frac{1}{I}\sum_{i\in\mathcal{I}}\frac{\left|\hat{v}_i - v_i\right|}{v_i}\times 100\%, \tag{14b}$$

where $\hat{v}_i$ represents the model result for the traffic state of the $i$ road segment. If $v_i$ is zero, a small positive value is substituted during the MAPE calculation.

Table 3 and Figure 11 compare the performance of the proposed retrieval and matching method against real-world observations for two directions on July 6 and July 7, 2023. Table 3 gives the model's accuracy in terms of MAE and MAPE. The results are broken down by hourly intervals over a 24-hour period, capturing the model's performance under both peak and off-peak conditions. Overall, the MAE and MAPE values remain relatively low across most time intervals, indicating that the method can accurately capture real-world traffic state variations. Moreover, a comparison between the two directions suggests that the approach generalizes well to different traffic flows, although minor discrepancies are observed during periods of heavy congestion.

Figure 11 provides a visual comparison by displaying heatmaps of the observed traffic state (left column) and estimated traffic state (right column) over time and spatial segments. In these heatmaps, green hues represent moderate speeds, while red zones highlight severe congestion. Despite some localized differences, the estimated

heatmaps closely mirror the temporal and spatial distribution of the observed speeds. Notably, major congestion episodes around midday and evening peak hours are clearly visible in both the observed and estimated heatmaps. In sum, the quantitative metrics in Table 3 and the visual evidence in Figure 11 demonstrate that the retrieval and matching algorithm reliably captures freeway traffic states under both congested and free-flow scenarios.

**Table 3 Comparison of model performance between observations and estimations. (MAE: km/h; MAPE: %)**

| **Time** | **2023.7.6** | | | | **2023.7.7** | | | |
|---|---|---|---|---|---|---|---|---|
| **Direction** | **1** | | **2** | | **1** | | **2** | |
| **Metric** | **MAE** | **MAPE** | **MAE** | **MAPE** | **MAE** | **MAPE** | **MAE** | **MAPE** |
| 00:00−00:59 | 1.17 | 1.26 | 1.45 | 1.61 | 1.17 | 1.29 | 1.51 | 1.66 |
| 01:00−01:59 | 1.21 | 1.32 | 1.45 | 1.63 | 1.34 | 1.46 | 1.45 | 1.67 |
| 02:00−02:59 | 1.15 | 1.27 | 1.43 | 1.66 | 1.32 | 1.47 | 1.32 | 1.53 |
| 03:00−03:59 | 1.08 | 1.21 | 1.26 | 1.50 | 1.20 | 1.35 | 1.26 | 1.49 |
| 04:00−04:59 | 1.16 | 1.37 | 1.61 | 1.79 | 0.98 | 1.14 | 1.36 | 1.51 |
| 05:00−05:59 | 1.33 | 1.52 | 1.70 | 1.86 | 1.30 | 1.45 | 1.65 | 1.83 |
| 06:00−06:59 | 1.05 | 1.12 | 1.53 | 1.68 | 1.10 | 1.20 | 1.57 | 1.67 |
| 07:00−07:59 | 0.96 | 1.03 | 1.85 | 2.05 | 1.10 | 1.17 | 1.28 | 1.36 |
| 08:00−08:59 | 0.82 | 0.88 | 1.85 | 2.12 | 0.78 | 0.82 | 1.05 | 1.12 |
| 09:00−09:59 | 0.75 | 0.81 | 1.80 | 2.11 | 0.96 | 1.15 | 0.94 | 1.02 |
| 10:00−10:59 | 0.97 | 1.14 | 1.60 | 1.88 | 0.79 | 0.88 | 1.02 | 1.10 |
| 11:00−11:59 | 1.23 | 1.38 | 1.86 | 2.14 | 1.09 | 1.27 | 1.21 | 1.29 |
| 12:00−12:59 | 1.07 | 1.22 | 1.78 | 1.96 | 1.06 | 1.20 | 0.90 | 0.98 |
| 13:00−13:59 | 1.14 | 1.35 | 1.03 | 1.10 | 0.91 | 1.06 | 1.07 | 1.19 |
| 14:00−14:59 | 1.01 | 1.16 | 1.10 | 1.20 | 1.32 | 1.56 | 1.06 | 1.18 |
| 15:00−15:59 | 0.87 | 1.05 | 1.06 | 1.15 | 1.31 | 1.86 | 0.95 | 1.05 |
| 16:00−16:59 | 1.05 | 1.22 | 1.10 | 1.19 | 1.39 | 1.76 | 1.14 | 1.27 |
| 17:00−17:59 | 1.04 | 1.21 | 1.21 | 1.33 | 1.11 | 1.26 | 1.18 | 1.31 |
| 18:00−18:59 | 1.06 | 1.17 | 1.24 | 1.36 | 0.85 | 0.94 | 2.36 | 5.10 |
| 19:00−19:59 | 1.09 | 1.24 | 1.16 | 1.26 | 1.54 | 1.85 | 1.07 | 1.18 |
| 20:00−20:59 | 1.18 | 1.33 | 1.16 | 1.26 | 1.10 | 1.29 | 1.01 | 1.11 |
| 21:00−21:59 | 1.23 | 1.36 | 1.20 | 1.28 | 1.25 | 1.42 | 1.12 | 1.19 |
| 22:00−22:59 | 1.35 | 1.47 | 1.27 | 1.36 | 1.16 | 1.21 | 1.33 | 1.41 |
| 23:00−23:59 | 1.57 | 1.74 | 1.33 | 1.43 | 0.83 | 0.86 | 1.82 | 1.99 |
| **Peak** | 1.01 | 1.16 | 1.46 | 1.65 | 1.06 | 1.28 | 1.02 | 1.12 |
| **Off-Peak** | 1.15 | 1.28 | 1.40 | 1.55 | 1.15 | 1.29 | 1.38 | 1.67 |

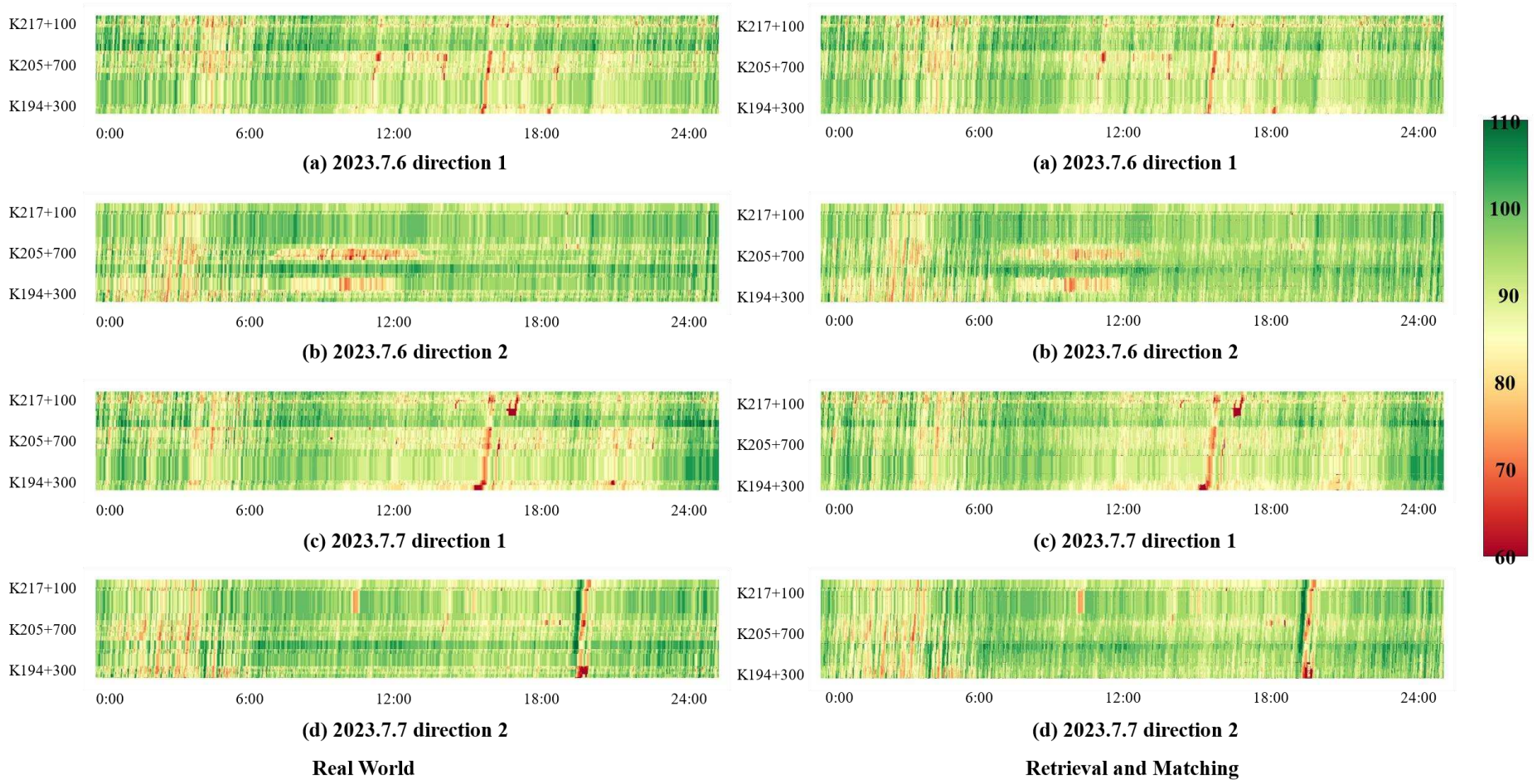



**Figure 11 Heatmaps of observation and estimation traffic states.**

### 4.4.2 Comparative experiments

This section presents comparative experiments to validate the proposed retrieval and matching algorithm against several alternative data fusion methods, including statistical, machine learning, and optimization methods.

#### *4.4.2.1 Baseline model*

Five models are selected for comparison:

(1) Dempster-Shafer (Faouzi et al., 2009): This model is based on Dempster-Shafer theory, with confusion matrices established using data on July 5, 2023 (two directions) to determine the reliability of each data type on a per-minute basis.

(2) Regression (Bachmann et al., 2013): This model employs linear regression, using features such as traffic state from ETC systems, sensors measurements, and coordinate information.

(3) Neural Networks (Khan et al., 2021): This approach utilizes artificial neural networks. Similar to the regression model, features from ETC and sensor observations, along with coordinate information, are used as inputs.

(4) Maximum Likelihood Estimation (Zhang et al., 2024a): This method pre-trains the spatial probability distribution and then maximizes the overall likelihood in a new scenario, formulating the problem as a MILP.

(5) Maximin Likelihood Estimation (Zhang et al., 2024a): Similar to the previous method, this approach pre-trains the spatial probability distribution but focuses on maximizing the minimum probability in a new scenario, with the problem also formulated as a MILP.

#### *4.4.2.2 Comparative results*

The computational time of the three optimization-based approaches is limit to 30 seconds. Table 4 and Table 5 compare the performance of the proposed retrieval and matching method with several existing approaches. Table 4 focuses on peak-hour periods, while Table 5 examines off-peak periods. Across all datasets, the retrieval and matching algorithm exhibits consistently lower MAE and MAPE values. During peak

periods, it significantly outperforms the single-source baselines (i.e., only ETC or only sensor), which tend to yield higher errors due to limited coverage or granularity. Additionally, compared with the Dempster–Shafer and linear regression methods, it better captures the nuanced shifts between free-flow and congested states, as indicated by smaller MAE/MAPE metrics. Even when benchmarked against more advanced techniques such as ANN and maximum/maximin likelihood estimation, the proposed method maintains a clear performance advantage, reinforcing the value of integrating machine learning with optimization. A similar trend emerges in off-peak periods: retrieval and matching algorithm achieves lower errors than the alternative methods. Overall, these results confirm that the proposed framework not only provides highly accurate estimates of freeway traffic states but also surpasses other fusion strategies across both peak and off-peak time windows.

**Table 4 Comparison results in peak period.**

| Dataset | Retrieval and matching | Only ETC | Only sensor | Dempster shafer | Linear regression | ANN | Maximum likelihood estimation | Maximin likelihood estimation |
|---|---|---|---|---|---|---|---|---|
| 2023.7.6 direction 1 | 1.01/1.16 | 4.05/4.50 | 3.62/4.17 | 3.67/4.2 | 3.93/4.48 | 2.10/2.42 | 1.66/1.90 | 1.18/1.31 |
| 2023.7.6 direction 2 | 1.46/1.65 | 4.75/5.19 | 4.84/5.53 | 4.75/5.26 | 4.32/4.83 | 2.32/2.64 | 1.62/1.80 | 1.99/2.19 |
| 2023.7.7 direction 1 | 1.06/1.28 | 3.74/4.30 | 3.68/4.33 | 3.63/4.25 | 3.76/4.41 | 2.04/2.43 | 1.7/1.96 | 1.73/1.94 |
| 2023.7.7 direction 2 | 1.02/1.12 | 3.47/3.70 | 3.25/3.59 | 3.42/3.68 | 3.21/3.5 | 1.72/1.89 | 1.55/1.7 | 1.98/2.14 |

**Table 5 Comparison results in off-peak period.**

| Dataset | Retrieval and matching | Only ETC | Only sensor | Dempster shafer | Linear regression | ANN | Maximum likelihood estimation | Maximin likelihood estimation |
|---|---|---|---|---|---|---|---|---|
| 2023.7.6 direction 1 | 1.15/1.28 | 3.98/4.29 | 4.05/4.52 | 3.97/4.35 | 3.89/4.29 | 2.16/2.4 | 1.96/2.19 | 1.78/1.98 |
| 2023.7.6 direction 2 | 1.4/1.55 | 4.21/4.5 | 4.15/4.62 | 4.14/4.52 | 3.96/4.33 | 2.16/2.4 | 1.86/2.06 | 2.34/2.56 |
| 2023.7.7 direction 1 | 1.15/1.29 | 3.92/4.29 | 4.06/4.57 | 3.92/4.34 | 3.83/4.28 | 2.12/2.39 | 2.0/2.22 | 1.93/2.2 |
| 2023.7.7 direction 2 | 1.38/1.67 | 4.07/4.5 | 4.01/4.66 | 4.06/4.63 | 3.87/4.39 | 2.12/2.42 | 1.79/2.02 | 2.17/2.38 |

# 5 Discussion

This study makes two key assumptions regarding ETC systems and sensor data, namely that they are free from bias, incompleteness, and outliers. Although these assumptions do not fully align with real-world conditions, the primary objective was to demonstrate how both data sources can be leveraged—assuming their true values are known—to exploit their unique strengths and fuse them effectively. Consequently, data preprocessing steps were not explicitly addressed. However, the proposed framework can readily incorporate existing data cleaning or preprocessing methods. For example, techniques for refining ETC data may follow approaches described by Zou et al. (2022) and Jedwanna et al. (2023), while preprocessing for sensor data can draw on methods outlined by Tian et al. (2018) and Kim, D and Kim, E (2023).

Although this study compares the proposed retrieval and matching algorithm with a range of methods—including statistical, machine learning, and optimization techniques—some state-of-the-art methods, such as graph neural networks and transformers, are not included. This decision is based on the current lack of research applying these advanced architectures to directly integrate ETC and sensor data or to handle the fusion of multiple traffic data types (e.g., speed, volume, occupancy). Future work may expand the set of baseline methods to include these emerging frameworks, thereby enabling a more comprehensive evaluation of the proposed approach.

State genes are the core of this work, capturing essential structural knowledge of freeway traffic. However, while state genes effectively represent spatial variations in traffic states, they do not incorporate temporal information. Consequently, the retrieval and matching algorithm relies solely on data corresponding to the same time step (i.e., deployment and observations), without fully exploiting inter-temporal dependencies. A promising direction for future research is to extend the concept of state genes to account for temporal dimensions, potentially by integrating sequence models (e.g., LSTM) to capture and utilize the evolving traffic state over time. This temporal extension would

allow the framework to leverage cross-time data more effectively, further enhancing the accuracy of freeway traffic state estimation.

The traffic state examined in this study is considered at the road-segment level, considering only the upstream and downstream relationships, as well as ETC and sensor deployment. However, additional covariates—such as ramps, merge and diverge areas, toll plazas, and lane configurations—are not explicitly considered. Future research could incorporate these factors so that the retrieval model does not simply select state genes at a broad scenario level but instead tailors them to specific locations. Such an expanded approach would likely increase precision in identifying critical bottlenecks or lane-specific dynamics, thereby enhancing the accuracy of the proposed framework in various network contexts.

Finally, the scale of the retrieved state gene set affects both the accuracy of the matching model and its computational efficiency. To address this, future work could consider column generation algorithms that incrementally add retrieved state genes based on their contribution to the overall solution quality. Such an approach would help maintain high prediction accuracy while controlling model complexity, providing a scalable solution for large-scale freeway traffic state estimation and ensuring its practical utility in real-world traffic networks.

## 6 Conclusions

This study introduces a prescriptive analytics framework for freeway traffic state estimation by fusing data from ETC systems and traffic sensors. To overcome the limitations of traditional single-method approaches, the proposed method combines machine learning with optimization, leveraging the inference capability of machine learning and the interpretability of optimization.

Central to this framework is the concept of state genes, which represent frequently recurring traffic patterns and capture essential structural knowledge. A novel retrieval and matching algorithm is developed, consisting of two core components: state gene retrieval, which predicts likely state genes using a heterogeneous graph model, and state

gene matching, which aligns these retrieved state genes with real-time traffic data to minimize difference.

Validation on real-world data from the G92 Freeway in Zhejiang, China, shows that the model achieves both high accuracy and fast inference. Specifically, the MAPE remains as low as 1.12–1.65% during peak periods and 1.28–1.67% during off-peak periods.

The introduced prescriptive analytics approach not only improves the accuracy of traffic state estimation but also provides a flexible framework for integrating various data sources. This methodology can be extended to other types of traffic monitoring technologies and a wide range of traffic scenarios. Future research may focus on refining the state gene selection process and exploring the integration of additional data sources to further enhance the precision of the proposed method.

**References**

Adetiloye, T., & Awasthi, A. (2019). Multimodal big data fusion for traffic congestion prediction. In Multimodal Analytics for Next-Generation Big Data Technologies and Applications (pp. 319–335).

Alvarez, A. M., Louveaux, Q., & Wehenkel, L. (2017). A machine learning-based approximation of strong branching. INFORMS Journal on Computing, 29(1), 185−195.

Andrychowicz, M., Denil, M., Gomez, S., Hoffman, M. W., Pfau, D., Schaul, T., ... & De Freitas, N. (2016). Learning to learn by gradient descent by gradient descent. Advances in Neural Information Processing Systems, 29.

Bachmann, C., Abdulhai, B., Roorda, M. J., & Moshiri, B. (2013). A comparative assessment of multi-sensor data fusion techniques for freeway traffic speed estimation using microsimulation modeling. Transportation Research Part C, 26, 33–48.

Baltean-Lugojan, R., Misener, R., Bonami, P., & Tramontani, A. (2018). Strong sparse cut selection via trained neural nets for quadratic semidefinite outer-approximations. Imperial College, London, Tech. Rep.

Bertsimas, D., & Kallus, N. (2020). From predictive to prescriptive analytics. Management Science, 66(3), 1025–1044.

Blondel, V. D., Guillaume, J. L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. Journal of Statistical Mechanics, 2008(10), P10008.

Bonami, P., Lodi, A., & Zarpellon, G. (2018). Learning a classification of mixed-integer quadratic programming problems. In Integration of Constraint Programming, Artificial Intelligence, and Operations Research: 15th International Conference (pp. 595−604).

Canepa, E. S., & Claudel, C. G. (2017). Networked traffic state estimation involving mixed fixed-mobile sensor data using Hamilton-Jacobi equations. Transportation Research Part B, 104, 686–709.

Chen, X., Liu, Z., Zhang, K., & Wang, Z., 2020. A parallel computing approach to solve traffic assignment using path-based gradient projection algorithm, Transportation Research Part C, 120, 102809.

Cvetek, D., Muštra, M., Jelušić, N., & Tišljarić, L. (2021). A survey of methods and technologies for congestion estimation based on multisource data fusion. Applied Sciences, 11(5), 2306.

Donti, P. L., Rolnick, D., & Kolter, J. Z. (2021). DC3: A learning method for optimization with hard constraints. arXiv preprint arXiv:2104.12225.

El Faouzi, N. E., Klein, L. A., & De Mouzon, O. (2009). Improving travel time estimates from inductive loop and toll collection data with Dempster-Shafer data fusion. Transportation Research Record, 2129(1), 73–80.

Ergashev, U., Dragut, E., & Meng, W. (2023). Learning to rank resources with GNN. In Proceedings of the ACM Web Conference 2023 (pp. 3247–3256).

Fan, M., Geng, B., Li, K., Wang, X., & Varshney, P. K. (2024). Interpretable Data Fusion for Distributed Learning: A Representative Approach via Gradient Matching. arXiv preprint arXiv:2405.03782.

Fu, H., Lam, W. H., Shao, H., Kattan, L., & Salari, M. (2022). Optimization of multi-type traffic sensor locations for estimation of multi-period origin-destination demands with covariance effects. Transportation Research Part E, 157, 102555.

Ghosh, N., Paul, R., Maity, S., Maity, K., & Saha, S. (2020). Fault Matters: Sensor data fusion for detection of faults using Dempster–Shafer theory of evidence in IoT-based applications. Expert Systems with Applications, 162, 113887.

Han, Q., Yang, L., Chen, Q., Zhou, X., Zhang, D., Wang, A., ... & Luo, X. (2023). A gnn-guided predict-and-search framework for mixed-integer linear programming. arXiv preprint arXiv:2302.05636.

Huang, D., Liu, Z., Liu, P., & Chen, J. (2016). Optimal transit fare and service frequency of a nonlinear origin-destination based fare structure. Transportation Research Part E, 96, 1–19.

Huang, D., & Wang, S. (2022). A two-stage stochastic programming model of coordinated electric bus charging scheduling for a hybrid charging scheme. Multimodal Transportation, 1(1), 100006.

Huang, D., Zhang, J., & Liu, Z. (2023). A robust coordinated charging scheduling approach for hybrid electric bus charging systems. Transportation Research Part D, 125, 103955.

Huang, D., Zhang, J., Liu, Z., He, Y., & Liu, P. (2024a). A novel ranking method based on Semi-SPO for battery swapping allocation optimization in a hybrid electric transit system. Transportation Research Part E, 188, 103611.

Huang, Y., Dong, Y., Tang, Y., & Li, L. (2024b). Leverage Multi-source Traffic Demand Data Fusion with Transformer Model for Urban Parking Prediction. arXiv preprint arXiv:2405.01055.

Ivan, J. N. (1997). Neural network representations for arterial street incident detection data fusion. Transportation Research Part C, 5(3–4), 245–254.

Jin, G., Yan, H., Li, F., Huang, J., & Li, Y. (2024). Spatio-temporal dual graph neural networks for travel time estimation. ACM Transactions on Spatial Algorithms and Systems, 10(3), 1−22.

Jedwanna, K., Athan, C., & Boonsiripant, S. (2023). Estimating Toll Road Travel Times Using Segment-Based Data Imputation. Sustainability, 15(17), 13042.

Khan, S., Nazir, S., García-Magario, I., & Hussain, A. (2021). Deep learning-based urban big data fusion in smart cities: Towards traffic monitoring and flow-preserving fusion. Computers & Electrical Engineering, 89, 106906.

Klein, L. A. (2019). Sensor and Data Fusion for intelligent transportation systems. Society of Photo-Optical Instrumentation Engineers.

Klein, L. A. (2024). Roadside Sensors for Traffic Management. IEEE Intelligent Transportation Systems Magazine. 16(4), 21−44.

Kruber, M., Lübbecke, M. E., & Parmentier, A. (2017). Learning when to use a decomposition. In International Conference on AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems (pp. 202−210).

Kong, Q., Li, Z., Chen, Y., & Liu, Y. (2009). An approach to urban traffic state estimation by fusing multisource information. IEEE Transactions on Intelligent Transportation Systems, 10(3), 499–511.

Kim, D., & Kim, E. (2023). Development of LSTM-MLR hybrid model for radar detector missing and outlier traffic volume correction. Transportation Planning and Technology, 46(2), 182-199.

Lin, W., Zhang, Z., Ren, G., Zhao, Y., Ma, J., & Cao, Q. (2025). MGCN: Mamba-integrated spatiotemporal graph convolutional network for long-term traffic forecasting. Knowledge-Based Systems, 309, 112875.

Lu, J., Li, C., Wu, X., & Zhou, X. (2023). Physics-informed neural networks for integrated traffic state and queue profile estimation: A differentiable programming

approach on layered computational graphs. Transportation Research Part C, 153, 104224.

Mahmud, S. S., Ferreira, L., Hoque, M. S., & Tavassoli, A. (2017). Application of proximal surrogate indicators for safety evaluation: A review of recent developments and research needs. IATSS Research, 41(4), 153–163.

Mahmood, R., Babier, A., McNiven, A., Diamant, A., & Chan, T. C. (2018). Automated treatment planning in radiation therapy using generative adversarial networks. In Machine learning for healthcare conference (pp. 484−499).

Mao, K., Xiao, X., Zhu, J., Lu, B., Tang, R., & He, X. (2020). Item tagging for information retrieval: A tripartite graph neural network based approach. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 2327–2336).

Nantes, A., Ngoduy, D., Bhaskar, A., Miska, M., & Chung, E. (2016). Real-time traffic state estimation in urban corridors from heterogeneous data. Transportation Research Part C, 66, 99–118.

Nie, Q., Xia, J., Qian, Z., An, C., & Cui, Q. (2015). Use of multisensor data in reliable short-term travel time forecasting for urban roads: Dempster–Shafer approach. Transportation Research Record, 2526(1), 61–69.

Schlichtkrull, M., Kipf, T. N., Bloem, P., Van Den Berg, R., Titov, I., & Welling, M. (2018). Modeling relational data with graph convolutional networks. In The semantic web: 15th international conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, proceedings 15 (pp. 593–607).

Seo, T., Bayen, A. M., Kusakabe, T., & Asakura, Y. (2017). Traffic state estimation on highway: A comprehensive survey. Annual Reviews in Control, 43, 128–151.

Shahrbabaki, M., Safavi, A., Papageorgiou, M., & Papamichail, I. (2018). A data fusion approach for real-time traffic state estimation in urban signalized links. Transportation Research Part C, 92, 525–548.

Tian, Y., Zhang, K., Li, J., Lin, X., & Yang, B. (2018). LSTM-based traffic flow prediction with missing data. Neurocomputing, 318, 297-305.

UCR. (2023). The Urban Congestion Report (UCR): Documentation and Definitions. https://ops.fhwa.dot.gov/perf_measurement/ucr/documentation.htm.

Varshney, K. R., Khanduri, P., Sharma, P., Zhang, S., & Varshney, P. K. (2018). Why interpretability in machine learning? An answer using distributed detection and data fusion theory. arXiv preprint arXiv:1806.09710.

Wang, Y., & Papageorgiou, M. (2005). Real-time freeway traffic state estimation based on extended Kalman filter: a general approach. Transportation Research Part B, 39(2), 141–167.

Wang, Y., Papageorgiou, M., & Messmer, A. (2007). Real-time freeway traffic state estimation based on extended Kalman filter: A case study. Transportation Science, 41(2), 167–181.

Wang, T., Huang, L., Tian, J., Zhang, J., Yuan, Z., & Zheng, J. (2024). Bus dwell time estimation and overtaking maneuvers analysis: A stochastic process approach. Transportation Research Part E, 186, 103577.

Wang, S., Li, F., Stenneth, L., & Yu, P. (2016). Enhancing traffic congestion estimation with social media by coupled hidden Markov model. In Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2016, Riva del Garda, Italy, September 19–23, 2016, Proceedings, Part II (pp. 247–264).

Wang, S. & Yan, R. (2022) “Predict, then optimize” with quantile regression: A global method from predictive to prescriptive analytics and applications to multimodal transportation. Multimodal Transportation, 1(4), 100035.

Wang, R., Zhang, Y., Guo, Z., Chen, T., Yang, X., & Yan, J. (2023). LinSATNet: The positive linear satisfiability neural networks. In International Conference on Machine Learning (pp. 36605−36625). PMLR.

Wang, Q., & Yang, K. (2024). Privacy-preserving data fusion for traffic state estimation: A vertical federated learning approach. arXiv preprint arXiv:2401.11836.

Wang, S., Dong, C., Shao, C., Luo, S., Zhang, J., & Meng, M. (2024). Traffic state estimation incorporating heterogeneous vehicle composition: A high-dimensional fuzzy model. Frontiers of Engineering Management, 1−19.

Yan, R. & Wang, S. (2022) Integrating prediction with optimization: Models and applications in transportation management. Multimodal Transportation, 1(3), 100018.

Zhang, J., Huang, D., Liu, Z., Zheng, Y., Han, Y., Liu, P., & Huang, W. (2024a). A data-driven optimization-based approach for freeway traffic state estimation based on heterogeneous sensor data fusion. Transportation Research Part E, 189, 103656.

Zhang, J., Mao, S., Yang, L., Ma, W., Li, S., & Gao, Z. (2024b). Physics-informed deep learning for traffic state estimation based on the traffic flow model and computational graph method. Information Fusion, 101, 101971.

Zhao, J., Jing, X., Yan, Z., & Pedrycz, W. (2021). Network traffic classification for data fusion: A survey. Information Fusion, 72, 22–47.

Zou, F., Ren, Q., Tian, J., Guo, F., Huang, S., Liao, L., & Wu, J. (2022). Expressway speed prediction based on electronic toll collection data. Electronics, 11(10), 1613.