mathematics

MDPI

Article

Reconstruction of Highway Vehicle Paths Using a Two-Stage Model

Weifeng Yin [1], Junyong Zhai [1,*] and Yongbo Yu [2]

[1] School of Automation, Southeast University, Nanjing 210096, China; 230209097@seu.edu.cn
[2] Jiangsu Communications Holding Digital Transportation Research Institute Co., Ltd., Nanjing 210019, China; yuyongbo@email.jchc.cn
* Correspondence: jyzhai@seu.edu.cn

Abstract: The accurate reconstruction of vehicle paths is essential for effective highway toll management. To address the challenge of multiple possible paths due to missing trajectory data, this study proposes a novel two-stage model for vehicle path reconstruction. In the first stage, a Gaussian Mixture Model (GMM) is integrated into a path choice model to estimate the mean and standard deviation of travel times for each road segment, utilizing an improved Expectation Maximization (EM) algorithm. In the second stage, based on the estimated time parameters, path choice prior probabilities and observed data are combined using maximum likelihood estimation to infer the most probable paths among candidate routes. The results indicate that the improved EM algorithm achieved convergence in 17 iterations compared to 41 iterations for the traditional EM algorithm. The two-stage model outperforms the Shortest Path and Bidirectional Long Short-Term Memory models in path reconstruction, particularly with a high number of missing trajectory points. Additionally, when the number of candidate paths $K = 4$, the path reconstruction performance is optimal. These results demonstrate the effectiveness of the proposed method in handling sparse and incomplete trajectory data, offering robust and accurate vehicle path estimations that enhance traffic management and toll calculation precision.

Keywords: vehicle path reconstruction; Gaussian mixture models; path choice model; expectation maximization algorithm

MSC: 90B20

check for updates

Academic Editor: Bo Wang

Received: 12 January 2025
Revised: 11 February 2025
Accepted: 12 February 2025
Published: 13 February 2025

Citation: Yin, W.; Zhai, J.; Yu, Y. Reconstruction of Highway Vehicle Paths Using a Two-Stage Model. *Mathematics* **2025**, *13*, 618. https://doi.org/10.3390/math13040618

1. Introduction

The accurate acquisition of vehicle trajectory information is essential for effective toll management on highways. Currently, the following two primary types of devices are employed to collect vehicle-related data: high-definition (HD) cameras and Electronic Toll Collection (ETC) systems combined with Compound Pass Card (CPC) sensors. HD cameras capture images of vehicle license plates and frontal views, while ETC systems and CPC sensors detect the presence of ETC devices installed in vehicles and the CPC cards issued at toll stations. These devices are typically mounted on gantries along highways, and by integrating location data from these gantries, vehicle trajectories can be obtained. However, the existing tolling framework often relies on the Shortest Path (SP) algorithm to estimate vehicle travel paths between adjacent gantries, introducing several limitations. Under high-traffic conditions, camera images may capture multiple vehicles in close proximity, leading to the partial or complete loss of license plate information or difficulties in obtaining clear frontal images. Traditional license plate recognition (LPR) systems are also susceptible to errors, particularly with counterfeit or unregistered plates. Furthermore, ETC and

CPC sensors can experience detection failures, resulting in unregistered vehicles passing through certain gantries. Consequently, these challenges contribute to incomplete or inaccurate trajectory data, especially when significant gaps exist between consecutive trajectory points. In such cases, the SP assumption may fail to accurately reflect the vehicle's actual travel route, thereby compromising toll calculation precision. The accurate reconstruction of vehicle travel paths remains a critical challenge in transportation research. Existing methods typically rely on Automatic Vehicle Identification (AVI) data, such as LPR and gantry sensing data obtained from tolling systems [1,2]. For simpler road networks, the SP algorithm [3,4] has been widely adopted due to its computational efficiency and ease of implementation. However, its performance degrades significantly in complex road networks with incomplete trajectory data, as real-world driving behavior often deviates from the shortest path assumption [5]. Factors such as driver preferences, real-time traffic conditions, and external constraints are not accounted for in SP-based models, limiting their applicability.

To address these shortcomings, researchers have developed methods that incorporate path choice behavior [6,7]. These approaches optimize path selection based on the spatial topology of road networks, offering enhanced flexibility over static assumptions. Nevertheless, they often neglect subjective factors like individual driver preferences and dynamic traffic conditions, which constrains their effectiveness in large-scale and real-world applications. Probabilistic methods have been introduced to overcome these challenges by considering both micro-level and macro-level factors in path reconstruction [8]. Micro-level approaches model individual vehicle trajectories, while macro-level models estimate path flow distributions through equilibrium-based frameworks. These methods are effective at handling uncertainty but are computationally intensive, particularly in high-dimensional state spaces or real-time scenarios. Additionally, data-driven approaches that combine AVI data with origin–destination (OD) estimates [9] have shown potential for accurate path reconstruction. However, their reliance on high-quality data often undermines their robustness in practical settings where data may be incomplete or noisy.

The rapid advancements in artificial intelligence and the availability of large-scale data have facilitated the application of deep learning to path reconstruction. Deep learning frameworks leverage neural networks to capture complex patterns in trajectory data [10,11], offering improved performance in complex road networks. Sequence-based models [12] have been employed to learn temporal dependencies in vehicle paths, addressing the limitations of traditional methods. However, some models still fail to account for spatial correlations between road segments, reducing their accuracy in highly interconnected networks. Recent approaches have integrated bidirectional sequence learning [13] to simultaneously capture spatial and temporal dependencies in vehicle trajectories. These models are fully data-driven and adaptive, eliminating the reliance on static path selection assumptions. Despite their advantages, challenges remain in scaling these methods to large road networks and maintaining accuracy in the presence of incomplete or noisy data. Emerging techniques, such as graph-based neural networks and attention mechanisms [14,15], show promise for addressing these limitations and improving both scalability and precision.

Despite significant advancements in vehicle path reconstruction, existing methodologies exhibit notable limitations in addressing the complexity and variability of real-world traffic scenarios. Factors such as driver preferences, dynamic traffic conditions, and inherent uncertainties in vehicle trajectories are often not simultaneously considered, and computational efficiency tends to be low in large-scale road networks. Additionally, many data-driven approaches heavily rely on high-quality data, which undermines their robustness in practical applications. To overcome these challenges, a novel two-stage path reconstruction model based on Gaussian Mixture Models (GMM) is proposed in this study.

The GMM [16] assumes that data originate from a weighted combination of multiple Gaussian distributions, each referred to as a component. The linear combination of these components represents the overall distribution of the dataset. The objective of GMM is to determine the model parameters—namely, the mean, covariance, and weight of each component—in order to maximize the likelihood function of the entire dataset. Due to the additive property of Gaussian distributions, the sum of two or more independent Gaussian random variables is also a Gaussian random variable. Consequently, the probability density function of GMM is a weighted sum of individual Gaussian distributions. The widespread adoption of GMM stems from the Expectation Maximization (EM) algorithm proposed in [17], a parameter learning method based on maximum likelihood estimation. The EM algorithm iteratively estimates missing data or latent variables in probabilistic models by alternating between the E-step (Expectation) and M-step (Maximization). It is widely employed for parameter estimation in models such as GMMs and those dealing with incomplete or censored data [18]. Due to its ability to handle incomplete or noisy data, the EM algorithm has been widely applied in various domains, including image segmentation [19], dynamic traffic prediction [20], and network traffic identification [21]. The advantage of GMM in handling complex data distributions has led to its extensive application in the transportation sector. It has demonstrated excellent performance in various areas, including traffic flow prediction [22], traffic accident analysis [23], vehicle tracking [24], and traffic flow anomaly detection [25]. Unlike traditional GMMs, the GMM proposed in this paper accounts for the different paths corresponding to various OD pairs when estimating the mean and standard deviation of road vehicle travel times. Traditional GMMs typically assume that data are generated from a weighted combination of several Gaussian distributions, with the weights of each Gaussian component determined independently through maximum likelihood estimation or the EM algorithm, without incorporating specific path selection behaviors. In contrast, our approach determines the weights of the GMM based on a path selection model, meaning that the weight of each Gaussian component reflects the probability of the corresponding path being chosen. This mechanism enables the model to more accurately represent the path selection preferences of different OD pairs, enhancing its adaptability and interpretability in complex traffic networks with diverse paths.

The main contributions of this paper are as follows:

1. A novel two-stage model is introduced for reconstructing vehicle travel paths from sparse trajectory data. In the first stage, a GMM is constructed to estimate the mean and standard deviation of vehicle travel times on each road segment. This estimation accounts for the probabilistic distribution of travel times, thereby capturing the inherent uncertainties and variations in vehicle behavior. In the second stage, the estimated time parameters are combined with path choice probabilities using maximum likelihood estimation to infer the most probable vehicle trajectories. This approach enhances the accuracy and flexibility of path reconstruction by integrating both spatial and temporal factors.
2. An initial value selection algorithm is designed to optimize the EM algorithm, and the Limited-memory Broyden–Fletcher–Goldfarb–Shanno with Bound (L-BFGS-B) optimization method is incorporated into the M-step to solve the Q-function's extremum, significantly enhancing the convergence speed of the EM algorithm.
3. Vehicle re-identification technology is integrated into traditional license plate recognition methods to construct a reliable vehicle travel trajectory dataset. This integration effectively addresses issues such as unregistered or counterfeit license plates, thereby further improving the accuracy of the detection data.

The structure of this paper is as follows. Section 2 outlines the proposed two-stage model for vehicle path reconstruction and the optimized EM algorithm. Section 3 presents the experimental results and evaluates the performance of the model. Section 4 discusses the advantages of the proposed method relative to existing approaches. Section 5 provides a summary of the entire work.

2. Methods

2.1. Assumptions

To reduce the complexity of the model, the following assumptions are made:

Assumption 1. *Within short time intervals, the travel times of vehicles on each road segment approximately follow a normal distribution. Although the travel times for individual vehicles may not strictly follow a normal distribution, according to the Central Limit Theorem, the distribution of average travel times over a short period is more likely to resemble a normal distribution. One advantage of using a normal distribution is its additivity, which can significantly reduce the model's complexity.*

Assumption 2. *The travel times of different road segments are statistically independent of each other. This assumption simplifies the construction of the GMM, enabling the travel time of each road segment to be modeled independently.*

Assumption 3. *Traffic conditions, including factors such as traffic flow and road status, remain relatively stable within a short time interval without significant fluctuations. This ensures the consistency of the model parameters throughout the time period.*

2.2. Model Construction

A two-stage path reconstruction model is proposed to accurately infer vehicle paths from sparse and incomplete trajectory data. In the first stage, GMM is employed to estimate the mean and standard deviation of travel times for each road segment. Subsequently, these estimated parameters are combined with path choice probabilities to infer the most probable vehicle paths through maximum likelihood estimation in the second stage.

2.2.1. GMM-Based Segment Travel Times Estimation

Based on Assumption 1, let y_i^t denote the travel times observed for the i-th data point in time period t. This data corresponds to multiple paths for the same OD pair. The selection probability for path k is modeled using a neural network-enhanced approach and is incorporated as weights into the GMM. The probability of observing y_i^t, given path k and the parameters $\overrightarrow{\mu_t}$ and $\overrightarrow{\sigma_t}$, is defined as follows:

$$P(y_i^t \mid k, \overrightarrow{\mu_t}, \overrightarrow{\sigma_t}) = \mathcal{N}\left(y_i^t \mid g_k^i(\overrightarrow{\mu_t}), h_k^i(\overrightarrow{\sigma_t})\right) \tag{1}$$

where $\overrightarrow{\mu_t}$ represents the vector of average travel times for all road segments in time period t. $\overrightarrow{\sigma_t}$ represents the vector of the standard deviations of travel times for all road segments in time period t. $g_k^i(\overrightarrow{\mu_t}) = \sum_{l \in k} \mu_t^l$ is the expected travel time for path k, and μ_t^l is the expected travel time for segment l contained in path k. $h_k^i(\overrightarrow{\sigma_t}) = \sum_{l \in k} (\sigma_t^l)^2$ is the variance of travel time for path k, and σ_t^l represents the standard deviation of the travel time for segment l contained in path k. $\mathcal{N}(y \mid \mu, \sigma^2)$ denotes the Gaussian distribution with the mean μ and variance σ^2.

In large road networks, a single trip may involve multiple path choices. The objective of this stage is to estimate the travel time characteristics of each road segment, utilizing the prior probability of path choices λ as weights in the estimation process.

Path choice models should incorporate various factors that influence travelers' decisions, including path length, travel times, congestion levels, and vehicle type [26–29]. By integrating these factors, the models can accurately estimate path selection probabilities, which are subsequently utilized as weights within the GMM. This comprehensive consideration of influencing variables ensures that the GMM effectively captures the underlying variability and dependencies in travel behavior, thereby enhancing the accuracy of travel time estimations.

The utility of path choice U_k^i is defined as follows:

$$U_k^i = f_{\text{NN}}(\boldsymbol{x}_k^{i,\text{att}}; \Theta) \tag{2}$$

where f_{NN} represents a feedforward neural network (FNN) [30], which is utilized to model complex nonlinear relationships within the data. Θ denotes the parameters (weights and biases) of the neural network. $\boldsymbol{x}_k^{i,\text{att}} = \sum_{m=1}^{H} \boldsymbol{\alpha}_k^{i,m} \cdot \boldsymbol{v}_k^{i,m}$ is the multi-head attention-weighted feature vector of path k for observation i. H is the number of attention heads in the self-attention mechanism. $\boldsymbol{\alpha}_k^{i,m}$ is the attention weight vector for path k in attention head m, computed as follows:

$$\boldsymbol{\alpha}_k^{i,m} = \text{softmax}\left(\frac{\boldsymbol{q}_k^{i,m} \cdot \boldsymbol{k}_k^{i,m}}{\sqrt{d_k}}\right) \tag{3}$$

where $\boldsymbol{q}_k^{i,m} = \boldsymbol{W}_q^m \boldsymbol{x}_k^i$ is the query vector for attention head m, $\boldsymbol{W}_q^m \in \mathbb{R}^{d_k \times d_k}$ is a learnable weight matrix, and $\mathbb{R}$ denotes the set of real numbers. $\boldsymbol{k}_k^{i,m} = \boldsymbol{W}_k^m \boldsymbol{x}_k^i$ is the key vector for attention head m, $\boldsymbol{W}_k^m \in \mathbb{R}^{d_k \times d_k}$ is a learnable weight matrix. $\boldsymbol{v}_k^{i,m} = \boldsymbol{W}_v^m \boldsymbol{x}_k^i$ is the value vector for attention head m, $\boldsymbol{W}_v^m \in \mathbb{R}^{d_k \times d_k}$ is a learnable weight matrix. $\boldsymbol{x}_k^i$ is the feature vector for path k in observation i. This vector includes various attributes such as path length, travel times, congestion level, vehicle type, and the straight-line distance between the origin and destination. d_k is the dimensionality of the feature vector $\boldsymbol{x}_k^i$. softmax$(\cdot)$ is a normalization function typically used in multi-class classification tasks. It converts raw scores (logits) into probabilities that sum to 1, making it suitable for scenarios where each input should be assigned a probability distribution over multiple classes or options.

The path choice prior probability λ_k^i is modeled as followed:

$$\lambda_k^i = \frac{\exp(U_k^i)}{\sum_{s \in R_i} \exp(U_s^i)} \tag{4}$$

where R_i is the set of possible paths for observation i. It encompasses all possible paths that a vehicle could take between the origin and destination for a given observation.

The neural network model is trained by minimizing the cross-entropy loss as follows:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k \in R_i} z_k^i \cdot \log(\lambda_k^i) \tag{5}$$

where z_k^i is the observed path choice label (a binary indicator for whether path k was chosen in observation i). N represents the number of observed data points.

Under the objective parameters, the probability of observing the travel time for the i-th data point in time period t is modeled as the expectation of the travel time across multiple paths for the corresponding OD pair, leading to the GMM formulation expressed as follows:

$$P(y_i^t \mid \overrightarrow{\mu_t}, \overrightarrow{\sigma_t}) = \sum_{k \in R_i} \lambda_k^i P(y_i^t \mid k, \overrightarrow{\mu_t}, \overrightarrow{\sigma_t}) \tag{6}$$

2.2.2. Bayesian-Based Path Reconstruction

Given an observed travel time y_i^t, the posterior probability for each path $k \in R_i$ is computed using Bayes' theorem. The posterior probability is expressed as follows:

$$P(k \mid y_i^t) \propto P(y_i^t \mid k)P(k) \tag{7}$$

where $P(y_i^t \mid k)$ is the likelihood of observing y_i^t given path k. Assuming that the travel time for path k follows a normal distribution, this likelihood is defined as follows:

$$P(y_i^t \mid k) = \frac{1}{\sqrt{2\pi\sigma_t^k}} \exp\left(-\frac{(y_i^t - \mu_t^k)^2}{2(\sigma_t^k)^2}\right) \tag{8}$$

$P(k)$ is the prior probability of path k, representing the preference for selecting path k. In our model, $P(k)$ is equivalent to λ_k^i, which is derived from the self-attention-enhanced neural network path choice model.

Thus, the posterior probability of path k is given by the following:

$$P(k \mid y_i^t) \propto \frac{1}{\sqrt{2\pi\sigma_i^k}} \exp\left(-\frac{(y_i^t - \mu_i^k)^2}{2(\sigma_i^k)^2}\right)\lambda_k^i \tag{9}$$

To identify the most likely path, the posterior probability for each path $k \in R_i$ is computed, and the path k_g with the maximum posterior probability is selected as follows:

$$k_g = \arg\max_{k \in R_i} P(k \mid y_i^t) = \arg\max_{k \in R_i}\left[P(y_i^t \mid k)\lambda_k^i\right] \tag{10}$$

2.3. Algorithm Design

To enhance the convergence speed and stability of the EM algorithm used in the GMM-based travel time estimation, an initial value selection algorithm with multiple strategies and the L-BFGS-B optimization method are proposed.

2.3.1. Multi-Strategy Initial Value Selection Algorithm

In the EM algorithm, the choice of initial values is crucial. Poor initial values may cause the algorithm to converge to local optima or require more iterations to reach the global optimum. To address this, a multi-strategy initial value selection algorithm based on the Mahalanobis distance is developed to increase the diversity and robustness of the initial values. This ensures that the initial parameters comprehensively cover the distribution characteristics of the data, thereby improving the stability and convergence efficiency of the EM algorithm.

The algorithm for initial value selection is described as follows:

Step 1: Calculate the theoretical time range.

For each segment l, the theoretical minimum travel time $\tau_l^{\min}$ and maximum travel time $\tau_l^{\max}$ are calculated by $\tau_l^{\min} = \frac{l_s}{v_{\max}}$ and $\tau_l^{\max} = \frac{l_s}{v_{\min}}$, where l_s is the length of segment l, $v_{\min}$ is the minimum speed limit for the segment (taken as 20 km/h in this study), and $v_{\max}$ is the maximum speed limit for the segment (taken as 135 km/h in this study). The initial mean range is set as $\mu_l \in [\tau_l^{\min}, \tau_l^{\max}]$ and the initial standard deviation range is set as $\sigma_{\max} = \frac{\tau_l^{\max} - \tau_l^{\min}}{3}$.

Step 2: Generate candidate parameters based on multiple strategies.

Strategy 1: Equidistant sampling rule-based candidate parameter generation.

Within the specified mean range $[\tau_l^{\min}, \tau_l^{\max}]$, W equidistant values are uniformly sampled to estimate the mean using the following:

$$\mu_l^{\text{rules}} = \left\{\tau_l^{\min} + \frac{a}{W-1} \cdot (\tau_l^{\max} - \tau_l^{\min}),\ a = 0, 1, \ldots, W-1\right\} \tag{11}$$

Within the standard deviation range $[0, \sigma_{\max}]$, W equidistant values are uniformly sampled to estimate the standard deviation using the following:

$$\sigma_l^{\text{rules}} = \left\{ 0 + \frac{a}{W-1} \cdot \sigma_{\max}, \, a = 0, 1, \ldots, W-1 \right\} \tag{12}$$

The candidate parameter set S_{rules} is generated by combining the means and standard deviations as follows:

$$S_{\text{rules}} = \left\{ (\mu, \sigma) \mid \mu \in \mu_l^{\text{rules}}, \sigma \in \sigma_l^{\text{rules}} \right\} \tag{13}$$

Strategy 2: Random candidate parameter generation.

The Y values for the mean are sampled randomly within the range $[\tau_l^{\min}, \tau_l^{\max}]$ and the Y values for the standard deviation are sampled randomly within the range $[0, \sigma_{\max}]$. The candidate parameter set S_{random} is generated by combining the means and standard deviations as follows:

$$S_{\text{random}} = \{ (\mu, \sigma) \mid \mu \text{ randomly generated}, \sigma \text{ randomly generated} \} \tag{14}$$

Step 3: Filter candidate parameters using the Mahalanobis distance.

For each candidate parameter (μ, σ), the Mahalanobis distance M_l is computed as follows:

$$M_l = \sqrt{\frac{(\mu - \mu_{\text{global}})^2}{\sigma_{\text{global}}^2}} \tag{15}$$

where the theoretical mean center is $\mu_{\text{global}} = \frac{\tau_l^{\min} + \tau_l^{\max}}{2}$, and the theoretical maximum standard deviation is $\sigma_{\text{global}} = \sigma_{\max}$.

The candidate parameters are then sorted in descending order of the Mahalanobis distance, and the top three candidates are selected as follows:

$$S_{\text{selected}} = \text{Top-3}(S_{\text{rules}} \cup S_{\text{random}}, M_l) \tag{16}$$

Step 4: Compute the final initial parameters.

For each selected candidate parameter (μ_i, σ_i) in S_{selected}, the weight ϕ_i is computed as follows:

$$\phi_i = \frac{M_i}{\sum_{j \in S_{\text{selected}}} M_j} \tag{17}$$

The initial mean $\mu^{(0)}$ is computed as follows:

$$\mu^{(0)} = \sum_{i \in S_{\text{selected}}} \phi_i \cdot \mu_i \tag{18}$$

and the initial standard deviation $\sigma^{(0)}$ is computed as follows:

$$\sigma^{(0)} = \sqrt{\sum_{i \in S_{\text{selected}}} \phi_i \cdot \sigma_i^2} \tag{19}$$

2.3.2. EM Algorithm with L-BFGS-B Optimization

To further enhance the convergence speed and stability of the EM algorithm, the L-BFGS-B optimization method is integrated into the Maximization (M)-step. The overall algorithm flow is described as follows:

Step 1: Initial value selection.

The multi-strategy initial value selection algorithm is employed to determine the initial values $\overrightarrow{\mu}^{(0)}$ and $\overrightarrow{\sigma}^{(0)}$.

Step 2: E-step—posterior probability calculation.

For a given observed trip with travel time y_i^t, origin–destination, and vehicle type information, the prior probability of path k is λ_k^i. The likelihood of the observed travel time y_i^t, given path k, is $P(y_i^t|k, \overrightarrow{\mu_t}, \overrightarrow{\sigma_t})$. Therefore, the posterior probability γ_k^i of path k being selected is computed as follows:

$$\gamma_k^i = \frac{\lambda_k^i P(y_i^t|k, \overrightarrow{\mu_t}, \overrightarrow{\sigma_t})}{\sum_{s\in R_i} \lambda_s^i P(y_i^t|s, \overrightarrow{\mu_t}, \overrightarrow{\sigma_t})} \tag{20}$$

The initial value of this parameter can be computed directly from the initialized $\overrightarrow{\mu}^{(0)}$ and $\overrightarrow{\sigma}^{(0)}$.

Step 3: M-step—maximizing the Q-function.

The expected log-likelihood function $Q(\overrightarrow{\mu_t}, \overrightarrow{\sigma_t})$ is the conditional expectation of the model parameters based on the posterior probabilities calculated in the E-step. Specifically, $Q(\overrightarrow{\mu_t}, \overrightarrow{\sigma_t})$ is given by the following:

$$Q(\overrightarrow{\mu_t}, \overrightarrow{\sigma_t}) = \frac{1}{N}\sum_{i=1}^{N}\sum_{k\in R_i} \gamma_k^i \Big[\ln \lambda_k^i + \ln P(y_i^t|k, \overrightarrow{\mu_t}, \overrightarrow{\sigma_t})\Big] \tag{21}$$

The task in this step is to maximize the function Q. Due to travel speed limits, the range of the travel time is constrained. The maximum speed corresponds to the minimum travel time $\tau_l^{\min}$, and the minimum speed corresponds to the maximum travel time $\tau_l^{\max}$. The standard deviation $\overrightarrow{\sigma_t}$ is a non-negative value. Therefore, the optimization problem can be expressed as follows:

$$\begin{aligned} &\max Q(\overrightarrow{\mu}_t, \overrightarrow{\sigma}_t) \\ &s.t. \begin{cases} \tau_l^{\min} \le \mu_l^t \le \tau_l^{\max} \\ \sigma_l^t \ge 0 \end{cases} \end{aligned} \tag{22}$$

Step 4: Optimization using L-BFGS-B.

The L-BFGS-B algorithm is employed to solve the constrained optimization problem defined in Step 3. The algorithm proceeds as follows:

Step 4.1: Initialization.

Let $f(x) = Q(\overrightarrow{\mu}_t, \overrightarrow{\sigma}_t)$, initialize the starting point $x_0 = (\overrightarrow{\mu}^{(0)}, \overrightarrow{\sigma}^{(0)})$, set the tolerance $\varepsilon > 0$, store the number of recent iterations as $u = 6$, set the current iteration number $j = 0$, and define the maximum iteration number $j_{\max}$. The Hessian matrix is initialized as the identity matrix $H_0 = I$, and the initial gradient change is set as $r = \nabla f(x_0)$.

Step 4.2: Convergence check.

If the condition $\|\nabla f(x_{j+1})\| \le \varepsilon$ is satisfied, the algorithm terminates and returns the optimal solution x_{j+1}. Otherwise, proceed to Step 4.3.

Step 4.3: Compute search direction.

Calculate the feasible direction for this iteration as $p_j = -r_j$, and perform a one-dimensional line search to find the step size $\rho_j > 0$ that minimizes the function $f(x_j + \rho_j p_j)$ as follows:

$$\rho_j = \arg\min_{\rho \ge 0} f(x_j + \rho p_j) \tag{23}$$

Step 4.4: Update the parameters.

Update the parameter values using $x_{j+1} = x_j + \rho_j p_j$.

Step 4.5: Maintain limited memory.

When $j > u$, retain only the most recent u vector pairs and discard the information related to (z_{j-u}, c_{j-u}).

Step 4.6: Update the Hessian approximation.

Compute the difference $z_j = x_{j+1} - x_j$ and the gradient difference $c_j = \nabla f(x_{j+1}) - \nabla f(x_j)$. These are used to approximate the inverse of the Hessian matrix, and the updated gradient is calculated as $r_j = H_j \nabla f(x_j)$.

Step 4.7: Increment iteration number.

Increment the iteration number $j = j + 1$. If $j > j_{\max}$, terminate the algorithm. Otherwise, return to Step 4.2.

Step 5: Parameter update and convergence.

Update the values of $\overrightarrow{\mu_t}$ and $\overrightarrow{\sigma_t}$, and repeat Steps 2 through 4 until the Q function converges.

3. Results

The computational experiments in this study are performed on a high-performance workstation featuring 64 GB of RAM and an NVIDIA A6000 graphics card. The software environment is configured with Python 3.8, CUDA 11.8, and PyTorch 2.5 to ensure the compatibility and optimal performance of the neural network models. Firstly, a vehicle trajectory dataset is constructed. Secondly, a vehicle path choice model is trained using complete trajectories. Thirdly, the parameters of a GMM are estimated based on the data with missing trajectories and validated using complete trajectory data. Finally, the effectiveness of the proposed path reconstruction method is demonstrated through comparisons with other approaches.

3.1. Dataset

3.1.1. Vehicle Trajectory Dataset

Five highways in China—G4011, G4221, S19, G15, and S28—are included in this study, as shown in Figure 1. The study area includes 244 bidirectional road segments with a total length of approximately 2000 km. The road segments are represented by blue lines, with each segment demarcated by gantries located at its origin. These gantries mark the endpoints of each road segment, thus defining a complete segment of the highway between two consecutive gantries.

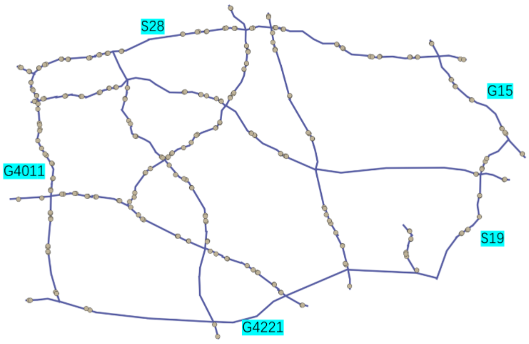

Figure 1. Spatial scope of the case study.

Data collected from HD cameras and ETC/CPC sensors were integrated to enhance the density and accuracy of the vehicle trajectory data. The specific algorithmic process for data fusion is outlined as follows.

Vehicle identity determination: The vehicle's identity is established by combining LPR and Vehicle Re-Identification (Re-ID) [31]. Initially, license plates are detected and recognized using LPR, and all associated vehicle records corresponding to the identified plate number are retrieved from the database. Simultaneously, the visual features of the vehicle, such as color, shape, and texture, are extracted through Re-ID, and matching records are obtained based on similarity measures. Furthermore, the resultant data are organized into a structured table comprising the following five fields: $vehicle\_id$, $timestamp$, $gantry\_id$, $gantry\_longitude$, and $gantry\_latitude$.

ETC/CPC data organization: Data obtained from ETC/CPC sensors are organized into records of the same form.

Chronological sorting: All records from the Vehicle Identity Determination and ETC/CPC Data Organization for each vehicle are sorted by timestamp in ascending order. This chronological ordering prepares the data for the subsequent merging processes.

Record merging: For each vehicle, adjacent records with the same $gantry\_id$ are merged, and the timestamp is updated by averaging the timestamps of the merged records.

Vehicle type classification: Vehicle type features are extracted from vehicle image data using a lightweight Convolutional Neural Network (CNN) method [32]. Vehicles are categorized into the following four classes:

- Light-Duty Passenger Vehicles (LPV)
- Coaches or Large Buses (LB)
- Light-Duty Trucks (LT)
- Heavy-Duty Trucks (HT)

Through the processing of the above steps, the resulting Vehicle Trajectory Dataset (VTD) is constructed. The dataset comprises the following six fields: $vehicle\_id$, $timestamp$, $gantry\_id$, $gantry\_longitude$, $gantry\_latitude$, and $vehicle\_type$. A comprehensive dataset comprising approximately 150,000 vehicle trajectories is collected for this study over a three-day period from 1 February to 3 February 2024. Data acquisition occurred daily between 07:00 and 17:00 h within the spatial boundaries previously delineated. A total of 44,702 trajectories are complete and the distribution of the number of trajectory points n is shown in Figure 2. The largest proportion of trajectories (46.73%) contains exactly two trajectory points, as indicated by the segment labeled $n = 2$. The second-largest group (30.87%) includes vehicles with between two and six trajectory points, followed by vehicles with between five and nine trajectory points (14.1%). Smaller proportions of complete trajectories fall into the other categories, including those with between 8 and 12 points (5.28%) and those with more than 14 points (0.9%). This dataset provides a robust foundation for analyzing traffic patterns and validating the proposed path reconstruction model.

For the remaining vehicles, trajectory point pairs are extracted sequentially. Specifically, if a vehicle has m observed trajectory points, $m - 1$ trajectory point pairs are generated. Based on the shortest path principle, the number of missing trajectory points between each pair n_lack is determined, with the results presented in Figure 3. Most of the vehicles (57.28%) have between zero and four missing trajectory points. The second-largest proportion (31.12%) has between three and seven missing points. Smaller percentages of vehicles have more than 7 missing points, with 9.32% falling within the 6 to 10 missing point range and 1.74% falling within the 9 to 13 missing point range. Only a small fraction (0.54%) have more than 12 missing trajectory points.

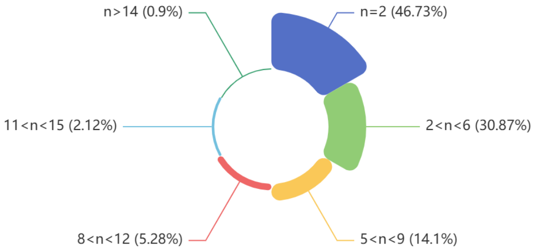

Figure 2. Distribution of the number of complete trajectories.

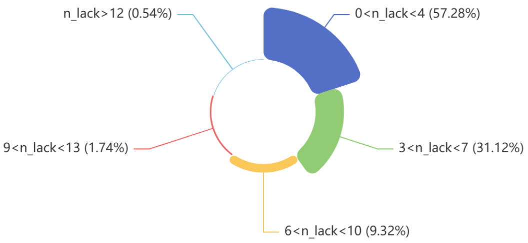

Figure 3. Distribution of missing trajectory points between trajectory point pairs.

3.1.2. Candidate Path Dataset

For each observed trip record, a candidate path set R_i is constructed through a systematic process comprising historical path selection, the application of the K-Shortest Paths (KSP) algorithm [33], and path set construction.

Historical path selection: Multiple distinct paths between the origin and destination points are identified from historically observed data, specifically derived from the complete trajectories mentioned above. The number of these historical paths is denoted as n_k, and they are ordered by increasing path length.

KSP algorithm: The KSP algorithm is applied to the road network data to obtain the shortest K paths between the OD pairs, ensuring that $K \geq n_k$.

Path set construction: The n_k observed historical paths are retained. The remaining $K - n_k$ paths, which are not present in the observed set, are selected from the KSP-derived shortest paths and incorporated into the candidate path set in order of increasing path length.

Consequently, the final candidate path set R_i comprises K paths, combining both the observed historical paths and the additional paths obtained through the KSP algorithm. Except for the sensitivity analysis conducted on K, a value of $K = 4$ was employed throughout this study.

3.1.3. Dataset for the Path Choice Model

Complete vehicle trajectories were utilized to train the path choice model. The path choice model outputs path encodings from the set of candidate paths using one-hot encoding. The input features include the length of the path, travel time, level of congestion, vehicle type, and the straight-line distance between the origin and destination. Travel time and vehicle type can be obtained from the complete vehicle trajectories directly. The straight-line distance between the origin and destination can be computed using the Haversine formula [34] with the unit in kilometers (km). The length of the path can be calculated as the sum of segment lengths between adjacent trajectory points. For a complete road network, Dijkstra's algorithm can be directly used for the computation.

To calculate the level of congestion, road operation speed data from Amap are introduced. These data, based on floating vehicles, provide information on the average vehicle speed for each road segment during each time period in the road network.

The level of congestion is quantified using a congestion index. For a given road segment, the congestion index during a specific time period is defined as the ratio of 85% of the segment's speed limit to the average traffic speed during that time period. Specifically, the congestion index C_l for segment l is expressed as follows:

$$C_l = \frac{0.85 \cdot V_{\max,l}}{V_{\text{avg},l}} \tag{24}$$

where $V_{\max,l}$ represents the speed limit of segment l, and $V_{\text{avg},l}$ denotes the average traffic speed on segment l during the given time period.

To evaluate the overall congestion level of a path, the weighted sum of the congestion indices of all segments along the path is calculated. The weights are proportional to the lengths of segments relative to the total path length. The congestion level C_k for a path k is thus defined as follows:

$$C_k = \frac{\sum_{l \in k} C_l \cdot L_l}{L_k} \tag{25}$$

where L_l is the length of segment l, and $L_k = \sum_{l \in k} L_l$ is the total length of path k.

3.2. Evaluation Indicators

This study defines three metrics to assess the performance of the two-stage model. In stage one, a weighted 95% confidence interval coverage ratio $ratio_{95}$ is employed to evaluate the estimated vehicle travel time parameters for each road segment. This ratio quantifies the proportion of observed travel time that falls within the 95% confidence interval of the Gaussian distribution. A coverage ratio exceeding 95% indicates that the travel time parameters estimated by the GMM are accurate.

$ratio_{95}$ is defined as follows:

$$ratio_{95} = \frac{\sum\limits_{l \in L} ratio_{95}^{l} \times n_l}{\sum\limits_{l \in L} n_l} \tag{26}$$

where L is the set of all road segments in the road network, n_l is the number of vehicles passing through road segment l, $ratio_{95}^{l} = \frac{n_{95}^{l}}{n_l}$, n_{95}^{l} is the number of vehicles passing through road segment l whose travel time fall within the 95% range.

In stage two, two metrics are utilized to evaluate the reconstructed paths. The Mean Absolute Percentage Error ($MAPE$) was introduced to evaluate the deviation between the sum of the segment travel time along the most probable path, calculated based on the OD pair and the observed values. The calculation formula was as follows:

$$MAPE = \frac{1}{N} \sum_{i=1}^{N} \left| \frac{g_k^i(\overrightarrow{\mu}_t) - y_i^t}{y_i^t} \right| \times 100\% \tag{27}$$

The $MAPE$ value approaching 0 signifies that the travel time of the reconstructed path is nearly identical to that of the observed path, thereby indicating superior reconstruction performance.

The spatial similarity between the reconstructed path and the original path was quantified using the Weighted Overlap Length Ratio ($WOLR$). The calculation formula was as follows:

$$WOLR = \frac{\sum\limits_{i=1}^{N} \frac{L_{common}(k_i,k_j) s_i}{L_{common}(k_i,k_j) + L_{different}(k_i,k_j)}}{\sum\limits_{i=1}^{N} s_i} \tag{28}$$

where s_i is the total length of path k_i, path k_j is the reconstructed path, L_{common} is the total length of road segments that are shared between path k_i and k_j, and $L_{different}$ is the total length of road segments that are unique to either path k_i or path k_j. The $WOLR$ is bounded between 0 and 1. A $WOLR$ value of 1 denotes that the two paths are identical in their spatial trajectories, whereas a $WOLR$ value of 0 indicates that the two paths have no overlapping segments.

3.3. Parameter Estimation Performance for Road Segment Travel Time

Complete vehicle trajectories are utilized to train the path choice model, comprising a training set of 35,762 paths and a testing set of 8940 paths. The model architecture is configured with eight attention heads $H = 8$, a feature vector dimensionality $d_k = 5$, three FNN layers, and 128 neurons per hidden layer. The Rectified Linear Unit (ReLU) is employed as the activation function. Training is conducted over 500 epochs using the Adam optimizer with a learning rate of 0.005 and a batch size of 32. This configured path choice model is denoted as $\mathcal{M}$ and serves as a foundational component supporting the GMM for accurate path reconstruction.

Since the proposed method targets relatively short time intervals, such as 30 min, vehicle trajectories on the road network during the half-hour period from 7:30 to 8:00 a.m. on 2 February 2024 are selected as an example for analysis. During this time period, the road network comprises 2790 vehicles with complete trajectories and 7659 vehicles with incomplete trajectories. For the 7659 vehicles with incomplete trajectory data, features such as OD, travel time, vehicle type, and the congestion index were extracted. By integrating these features into a set of candidate paths, the prior probabilities λ_k for the selection of each path are determined. These prior probabilities are incorporated into the GMM, and the EM algorithm is applied to estimate the travel time parameters for each road segment. The dataset containing 2790 vehicles with complete trajectories is utilized for validation.

To demonstrate the superiority of the proposed EM algorithm with multi-strategy initial value selection and L-BFGS-B, a comparison is conducted with the traditional EM algorithm [35,36] which randomly selected initial values and does not utilize the L-BFGS-B algorithm for optimizing the Q function. The results are presented in Figure 4. The traditional EM algorithm (represented by the blue curve) shows a relatively smooth decline in the early iterations, but the rate of decrease slows down in the later stages, requiring a total of 41 iterations to reach stability. In contrast, the improved EM algorithm (represented by the orange curve) demonstrates a rapid decline in the early iterations and converges within 17 iterations, indicating a significant acceleration in convergence. The quick descent and early stability of the improved algorithm highlight its superiority, while the slow convergence of the traditional algorithm underscores its limitations.

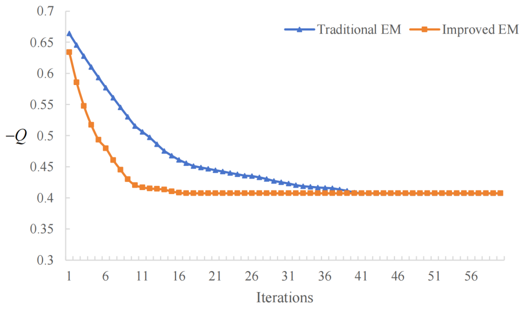

Figure 4. Comparison of EM algorithm performance.

The mean and standard deviation of the travel time for each road segment estimated by the GMM are validated using the dataset comprising 2790 complete vehicle trajectories. The result shows that $ratio_{95} = 0.965$, surpassing the 95% confidence threshold. This indicates that the mean and standard deviation of travel time for each road segment estimated by the GMM align well with the actual conditions.

3.4. Performance of Path Reconstruction

Intermediate trajectory points in the complete path data are excluded, retaining only the origin and destination points along with the travel duration. Using the path reconstruction model, each record's path is restored.

When the range of missing trajectory points is between [1, 3], all 1000 samples are accurately reconstructed. When the range of missing trajectory points is between [4, 6], all 1000 samples are also accurately reconstructed. For the range of missing trajectory points between [7, 9], 488 out of 500 samples are accurately reconstructed. When the range of missing trajectory points is between [10, 12], 192 out of 200 samples are accurately reconstructed. For cases where the range of missing trajectory points exceeds 12, 73 out of 90 samples are accurately reconstructed. Overall, the path reconstruction shows satisfactory performance.

The average travel time of the road segments along the reconstructed path are then summed to calculate the path's average travel time. This estimated travel time is compared to the observed value, and the $MAPE$ is subsequently calculated. The results are illustrated in Figure 5.

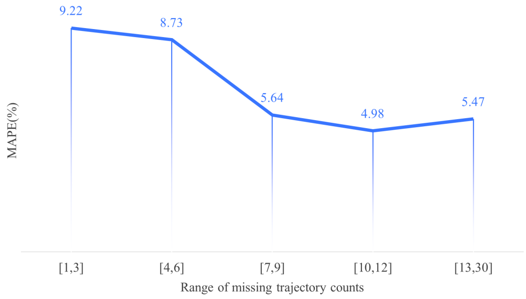

Figure 5. Path travel time deviation characteristics under different numbers of missing trajectory points.

When the number of missing trajectory points ranged from one to three, a sample size of 1000 yields a $MAPE$ of 9.22%. For missing point counts between four and six, the sample size remains at 1000, resulting in a $MAPE$ of 8.73%. When the number of missing points increased to a range from seven to nine, the sample size decreased to 500, with a corresponding $MAPE$ of 5.64%. In cases where missing points ranged from 10 to 12, the sample size was 200, and the $MAPE$ was 4.98%. Lastly, for missing point counts of 13 or more, the sample size was 90, and the $MAPE$ was 5.47%.

The $MAPE$ analysis revealed that as the number of missing trajectory points increases, the overall $MAPE$ initially decreases. However, when the number of missing points exceeds 12, a significant rise in $MAPE$ becomes evident. In cases with a small number of missing points, both the travel path length and the average travel time are relatively short. Consequently, minor fluctuations in the observed data under these conditions can lead to higher $MAPE$ values. Conversely, when the number of missing points is large (e.g., exceeding 12), the uncertainty in path selection becomes substantial, potentially resulting in reconstructed paths that do not accurately reflect the actual routes taken. This discrepancy leads to a considerable deviation between the travel time calculated based on average segment durations and the observed travel time, thereby increasing the $MAPE$.

3.5. Comparison of Methods

To demonstrate the superiority of the GMM, two comparison methods are selected as follows: Shortest Path (SP) and Bidirectional Long Short-Term Memory (Bi-LSTM) [13]. The SP is a method currently employed in highway toll management. It involves directly obtaining the shortest spatial distance between the start and end points in the road network and identifying the path that corresponds to this shortest distance. Bi-LSTM adds a bidirectional mechanism to the LSTM model. It inherits LSTM's ability to learn sequential data while enhancing its capacity to integrate bidirectional information from the sequence. The forward layer learns from front to back, and the backward layer learns from back to front. The information learned by both layers is then combined into a single output. In this problem, Bi-LSTM not only learns the node sequence but also incorporates the spatial geographic coordinates of each node to improve prediction performance.

We use $ARPM/ERPM$ as abbreviations for the number of paths accurately/erroneously reconstructed using the method proposed in this paper. We use $ARBL/ERBL$ as abbreviations for the number of paths accurately/erroneously reconstructed using the Bi-LSTM. To provide a detailed evaluation of each method's effectiveness in reconstructing actual paths, Tables 1–5 present the number of correctly reconstructed paths achieved by three different methods under varying numbers of missing trajectory points.

Since all observed paths in the dataset are the shortest paths where the number of missing trajectory points does not exceed three, the accuracy rate of using the shortest path reconstruction method is 100%. The accuracy rate of reconstruction using the proposed method is 100%. The reconstruction accuracy based on Bi-LSTM is 97.8%, with 22 paths inaccurately reconstructed.

When the number of missing points is in the range [4, 6], there are 993 shortest paths in the observed dataset. The proposed method accurately reconstructs 1000 of these paths, resulting in an accuracy rate of 100%. The performance of the Bi-LSTM method is 89.1% and the SP method's accuracy is 99.3%.

For missing point counts in the range [7, 9], the observed dataset includes 475 shortest paths. The proposed method accurately reconstructs 464 of these paths, yielding an accuracy rate of 97.7%. Among the 25 observed non-shortest paths, 24 are accurately reconstructed, resulting in an accuracy rate of 96%. The overall reconstruction accuracy for this range is 97.6%. The Bi-LSTM method's accuracy rate is 82% and the SP reconstruction accuracy is 95%.

When the number of missing points is in the range [10, 12], the observed dataset comprises 176 shortest paths. The proposed method accurately reconstructs 172 of these paths, achieving an accuracy rate of 97.7%. Among the 24 observed non-shortest paths, 20 are accurately reconstructed, resulting in an accuracy rate of 83.3%. The overall reconstruction accuracy for this range is 96.0%. The Bi-LSTM method's accuracy rate is 82.5% and the SP reconstruction accuracy is 88%.

For missing point counts of 13 or more, the observed dataset includes 71 shortest paths. The proposed method accurately reconstructs 59 of these paths, yielding an accuracy rate of 83.1%. Among the 19 observed non-shortest paths, 14 are accurately reconstructed, resulting in an accuracy rate of 73.7%. The overall reconstruction accuracy for this range is 81.1%. The SP reconstruction accuracy is 78.9% and the Bi-LSTM method's accuracy rate is 71.1%.

Table 1. Path reconstruction performance for missing trajectory points in the range [1, 3].

	Observed Shortest Paths Counts	Observed Non Shortest Paths Counts	Total
$ARPM$	1000	0	1000
$ERPM$	0	0	0
$ARBL$	978	0	978
$ERBL$	0	22	22

Table 2. Path reconstruction performance for missing trajectory points in the range [4, 6].

	Observed Shortest Paths Counts	Observed Non Shortest Paths Counts	Total
ARPM	993	7	1000
ERPM	0	0	0
ARBL	888	3	891
ERBL	105	4	109

Table 3. Path reconstruction performance for missing trajectory points in the range [7, 9].

	Observed Shortest Paths Counts	Observed Non Shortest Paths Counts	Total
ARPM	464	24	488
ERPM	11	1	12
ARBL	397	13	410
ERBL	78	12	90

Table 4. Path reconstruction performance for missing trajectory points in the range [10, 12].

	Observed Shortest Paths Counts	Observed Non Shortest Paths Counts	Total
ARPM	172	20	192
ERPM	4	4	8
ARBL	157	8	165
ERBL	19	16	35

Table 5. Path reconstruction performance for missing trajectory points exceeding 12.

	Observed Shortest Paths Counts	Observed Non Shortest Paths Counts	Total
ARPM	59	14	73
ERPM	12	5	17
ARBL	55	9	64
ERBL	16	10	26

The $WOLR$ of three methods under different numbers of missing trajectory points are shown in Figure 6. For missing trajectory counts between one and three, both the proposed method and SP achieved a perfect $WOLR$ of 1.0, while Bi-LSTM attained a $WOLR$ of 0.991. In the range from four to six missing points, the proposed method maintained $WOLR = 1.0$, SP achieved 0.994, and Bi-LSTM's $WOLR$ was 0.966. As the number of missing points increased to 7–9, the proposed method's $WOLR$ slightly decreased to 0.992, compared to SP's 0.966 and Bi-LSTM's 0.923. For missing points ranging from 10 to 12, the proposed method achieved 0.985, SP reached 0.925, and Bi-LSTM recorded 0.881. Finally, in cases with more than 12 missing points, the proposed method maintained a high $WOLR$ of 0.923, outperforming SP at 0.844 and Bi-LSTM at 0.828.

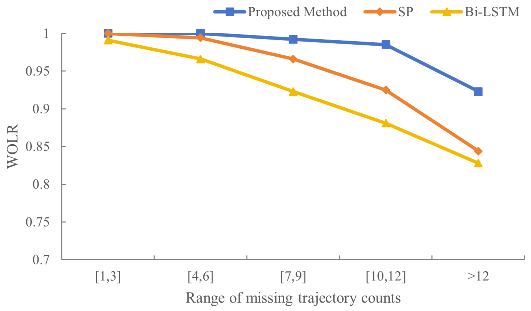

Figure 6. Comparison of $WOLR$.

3.6. Sensitivity Analysis

To further investigate the impact of the number of candidate paths on reconstruction accuracy, the number of candidate paths was varied from three to ten, in increments of one. A sensitivity analysis was conducted for our proposed method, specifically when the number of missing points surpassed thirteen. The results are presented in Figure 7. It illustrates that $WOLR$ peaks at $K = 4$ with 0.923 and gradually decreases as K increases, reaching 0.778 at $K = 10$. This trend suggests that an optimal number of candidate paths exists for maximizing reconstruction accuracy, beyond which the accuracy decreases.

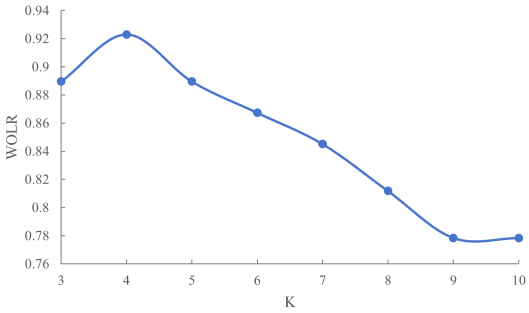

Figure 7. $WOLR$ varies with K.

4. Discussion

The results of this study demonstrate the effectiveness of the proposed method in accurately estimating road segment travel times and reconstructing vehicle paths, especially in scenarios with a significant number of missing trajectory points.

In terms of path reconstruction accuracy, the proposed method consistently outperforms both the SP and Bi-LSTM methods. When the number of missing trajectory points is three or fewer, both the SP method and the proposed method achieve perfect path reconstruction accuracy (100%). However, when the number of missing points exceeds three, the proposed method outperformed both the SP and Bi-LSTM methods in terms of path reconstruction accuracy. As the number of missing points increased, the accuracy of path restoration decreased for all methods. Nonetheless, the decrease in accuracy is more pronounced for the SP and Bi-LSTM methods compared to the proposed method. On the one hand, when the number of missing trajectory points is minimal, the disparities in path lengths among candidate routes are significant, thereby conferring a clear advantage to the SP method. Conversely, as the number of missing points increases, the distinctions between candidate paths diminish, and the lengths of multiple routes become comparable. In such cases, factors like path congestion exert a substantial influence on path selection, resulting in decreased accuracy for the SP method. Although the Bi-LSTM model is adept at capturing temporal dependencies and contextual information within paths, it does not fully leverage the structural information of the road network graph. This limitation leads to reduced efficiency and accuracy in path reconstruction tasks compared to graph-based algorithms.

The trend of the $WOLR$ metric aligns with the proportion of accurately reconstructed paths, and its values do not fall below the proportion of accurate reconstructions (Figure 7). This indicates that, although the paths are not entirely reconstructed accurately, certain segments within the paths are successfully reconstructed. When the number of missing trajectory points exceeds twelve, the $WOLR$ of path reconstruction significantly decreases. This decline is primarily attributed to the increased number of potential actual paths as the travel distance extends, which affects the reconstruction accuracy of all methods to varying degrees.The sensitivity analysis indicated that the optimal number of candidate paths K for maximizing path reconstruction accuracy is four (Figure 7). Increasing K beyond four led to a decrease in *WOLR*, likely due to the inclusion of non-viable paths that introduce noise and complicate the reconstruction process. This reduction is primarily due to the inclusion of an excessive number of non-viable paths, which causes the observed travel time to more closely resemble those of incorrect routes. Consequently, the presence of these interference paths complicates the reconstruction process, diminishing the overall effectiveness of the model. This highlights the necessity of selecting an appropriate number of candidate paths to balance model complexity and accuracy.

The study results indicate that although the proposed algorithm performs well in most scenarios, its accuracy decreases as the number of missing intermediate points increases. On the one hand, the precision of path reconstruction can be enhanced by deploying additional detection facilities on highways to obtain more comprehensive data and by employing segmented fitting methods. On the other hand, in cases with a significant number of missing trajectory points, integrating supplementary data sources, such as vehicle navigation data from map providers, can further improve the accuracy of vehicle path reconstruction.

5. Conclusions

This study presents a novel two-stage path reconstruction model that integrates GMM into a self-attention-enhanced neural network to accurately infer vehicle trajectories from sparse and incomplete data. In the first stage, the model estimates the mean and standard deviation of travel time for each road segment using an EM algorithm, effectively capturing the inherent uncertainties in vehicle behavior. In the second stage, these time estimates are combined with path choice probabilities through maximum likelihood estimation to reconstruct the most probable vehicle paths. Comprehensive experiments conducted on a large-scale vehicle trajectory dataset validate the robustness and effectiveness of the proposed model in handling missing trajectory data.

The contributions of this research are threefold. First, the proposed two-stage model enhances path reconstruction accuracy by integrating spatial and temporal factors, improving flexibility in handling incomplete trajectory data. Second, an optimized EM algorithm incorporating a novel initial value selection method and the L-BFGS-B optimization algorithm accelerates convergence, enhancing computational efficiency. Third, the integration of vehicle re-identification technology with traditional license plate recognition methods contributes to constructing a reliable vehicle trajectory dataset, mitigating issues such as unregistered or counterfeit license plates.

These advancements not only facilitate more precise vehicle path reconstructions but also hold implications for enhancing traffic management, urban planning, and smart transportation systems.

Author Contributions: Formulating the idea, W.Y. and J.Z.; methodology, W.Y., J.Z., and Y.Y.; theory, W.Y. and J.Z.; algorithm design, W.Y. and J.Z.; result analysis, W.Y. and J.Z.; writing, W.Y. and J.Z.; reviewing the research, W.Y. and J.Z.; supervision, J.Z.; project administration, J.Z.; funding acquisition, W.Y. and J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 62373096.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Due to privacy protection and commercial confidentiality, the data used in this study cannot be publicly shared. Researchers interested in accessing the data may contact the corresponding author to discuss potential collaborations or data access arrangements.

Conflicts of Interest: Author Yongbo Yu was employed by the company Jiangsu Communications Holding Digital Transportation Research Institute Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Li, A.; Lam, W.H.; Ma, W.; Wong, S.C.; Chow, A.H.; Tam, M.L. Real-time estimation of multi-class path travel times using multi-source traffic data. *Expert Syst. Appl.* **2024**, *237*, 121613. [CrossRef]
2. Cao, Q.; Ren, G.; Li, D.; Li, H.; Ma, J. Map matching for sparse automatic vehicle identification data. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 6495–6508. [CrossRef]
3. Xu, B.; Ji, X.; Cheng, Z. A comparison of three real-time shortest path models in dynamic interval graph. *Mathematics* **2025**, *13*, 134. [CrossRef]
4. Wei, H.; Zhang, S.; He, X. Shortest path algorithm in dynamic restricted area based on unidirectional road network model. *Sensors* **2020**, *21*, 203. [CrossRef]
5. Zhang, Z.; Liu, H.Q.; Rai, L.; Zhang, S.Y. Vehicle trajectory prediction method based on license plate information obtained from video-imaging detectors in urban road environment. *Sensors* **2020**, *20*, 1258. [CrossRef]
6. Li, R.M.; Liu, Z.Y.; Zhang, R.B. Studying the benefits of carpooling in an urban area using automatic vehicle identification data. *Transp. Res. Part C Emerg. Technol.* **2018**, *93*, 367–380. [CrossRef]
7. Feng, Y.; Sun, J.; Chen, P. Vehicle trajectory reconstruction using automatic vehicle identification and traffic count data. *J. Adv. Transp.* **2015**, *49*, 174–194. [CrossRef]
8. Yang, J.H.; Sun, J. Vehicle path reconstruction using automatic vehicle identification data: An integrated particle filter and path flow estimator. *Transp. Res. Part C Emerg. Technol.* **2015**, *58*, 107–126. [CrossRef]
9. Mo, B.C.; Li, R.M.; Dai, J.C. Estimating dynamic origin destination demand: A hybrid framework using license plate recognition data. *Comput.-Aided Civ. Infrastruct. Eng.* **2020**, *35*, 734–752. [CrossRef]
10. Qi, X.; Ji, Y.; Li, W.; Zhang, S. Vehicle trajectory reconstruction on urban traffic network using automatic license plate recognition data. *IEEE Access* **2021**, *9*, 49110–49120. [CrossRef]
11. Katariya, V.; Baharani, M.; Morris, N.; Shoghli, O.; Tabkhi, H. Deeptrack: Lightweight deep learning for vehicle trajectory prediction in highways. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 18927–18936. [CrossRef]
12. Wang, Y.; An, C.; Ou, J.; Lu, Z.; Xia, J. A general dynamic sequential learning framework for vehicle trajectory reconstruction using automatic vehicle location or identification data. *Phys. A Stat. Mech. Its Appl.* **2022**, *608*, 128243. [CrossRef]
13. Bian, J.; Chen, P. Vehicle path reconstruction using automatic vehicle identification data: A bi-directional long short-term memory-based approach. In Proceedings of the 2023 IEEE 26th International Conference on Intelligent Transportation Systems, Bilbao, Spain, 24–28 September 2023; pp. 5995–6000.
14. Ju, W.; Zhao, Y.; Qin, Y.; Yi, S.; Yuan, J.; Xiao, Z.; Zhang, M. Cool: A conjoint perspective on spatio-temporal graph neural network for traffic forecasting. *Inf. Fusion* **2024**, *107*, 102341. [CrossRef]
15. Chen, J.; Zheng, L.; Hu, Y.; Wang, W.; Zhang, H.; Hu, X. Traffic flow matrix-based graph neural network with attention mechanism for traffic flow prediction. *Inf. Fusion* **2024**, *104*, 102146. [CrossRef]
16. Deeva, I.; Bubnova, A.; Kalyuzhnaya, A.V. Advanced approach for distributions parameters learning in Bayesian networks with Gaussian mixture models and discriminative models. *Mathematics* **2023**, *11*, 343. [CrossRef]
17. Dempster, A.P.; Laird, N.M.; Rubin, D.B. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. Ser. B* **1977**, *39*, 1–22. [CrossRef]
18. Lin, S.; Zheng, Q.; Shang, L.; Xu, P.; Tang, M. Fitting penalized estimator for sparse covariance matrix with left-censored data by the EM algorithm. *Mathematics* **2025**, *13*, 423. [CrossRef]
19. Mahdavi, A.; Balakrishnan, N.; Jamalizadeh, A. EM algorithm for bounded generalized t mixture model with an application to image segmentation. *Comput. Appl. Math.* **2025**, *44*, 89. [CrossRef]
20. Xia, D.; Zheng, L.; Tang, Y.; Cai, X.; Chen, L.; Sun, D. Dynamic traffic prediction for urban road network with the interpretable model. *Phys. A Stat. Mech. Its Appl.* **2022**, *605*, 128051. [CrossRef]
21. Cui, H.; Liang, L.; Wang, J. Network traffic identification based on improved EM algorithm. *IEEE Access* **2024**, *12*, 26773–26786. [CrossRef]
22. Zhang, H.; Huang, L.; Wu, C.Q.; Li, Z. An effective convolutional neural network based on SMOTE and Gaussian mixture model for intrusion detection in imbalanced dataset. *Comput. Netw.* **2020**, *177*, 107315. [CrossRef]
23. Jin, S.; Wang, D.H.; Xu, C.; Ma, D.F. Short-term traffic safety forecasting using Gaussian mixture model and Kalman filter. *J. Zhejiang Univ. Sci. A* **2013**, *14*, 231–243. [CrossRef]
24. Zheng, O.; Abdel-Aty, M.; Yue, L.; Abdelraouf, A.; Wang, Z.; Mahmoud, N. CitySim: A drone-based vehicle trajectory dataset for safety-oriented research and digital twins. *Transp. Res. Rec.* **2024**, *2678*, 606–621. [CrossRef]

25. Ding, N.; Ma, H.; Gao, H.; Ma, Y.; Tan, G. Real-time anomaly detection based on long short-Term memory and Gaussian Mixture Model. *Comput. Electr. Eng.* **2019**, *79*, 106458. [CrossRef]
26. Zhan, X.; Ukkusuri, S.V.; Yang, C. A Bayesian mixture model for short-term average link travel time estimation using large-scale limited information trip-based data. *Autom. Constr.* **2016**, *72*, 237–246. [CrossRef]
27. Ding, Y.; Zhang, L.; Huang, C.; Ge, R. Two-stage travel itinerary recommendation optimization model considering stochastic traffic time. *Expert Syst. Appl.* **2024**, *237*, 121536. [CrossRef]
28. Ahmad, F.; Al-Fagih, L. Travel behaviour and game theory: A review of route choice modeling behaviour. *J. Choice Model.* **2024**, *50*, 100472. [CrossRef]
29. Qin, W.; Zhuang, Z.; Huang, Z.; Huang, H. A novel reinforcement learning-based hyper-heuristic for heterogeneous vehicle routing problem. *Comput. Ind. Eng.* **2021**, *156*, 107252. [CrossRef]
30. Kulyukin, V.A. On the computability of primitive recursive functions by feedforward artificial neural networks. *Mathematics* **2023**, *11*, 4309. [CrossRef]
31. Yin, W.; Min, Y.; Zhai, J. A vehicle comparison and re-identification system based on residual network. *Machines* **2022**, *10*, 799. [CrossRef]
32. Sun, W.; Zhang, G.; Zhang, X.; Zhang, X.; Ge, N. Fine-grained vehicle type classification using lightweight convolutional neural network with feature optimization and joint learning strategy. *Multimed. Tools Appl.* **2021**, *80*, 30803–30816. [CrossRef]
33. Yu, Z.; Yu, X.; Koudas, N.; Chen, Y.; Liu, Y. A distributed solution for efficient k shortest paths computation over dynamic road networks. *IEEE Trans. Knowl. Data Eng.* **2024**, *36*, 2759–2773. [CrossRef]
34. Qiao, B.; Wang, Y.; Yao, L.; Han, D.; Wu, G. Attention mechanism fusion neural network for typhoon path prediction. *Appl. Intell.* **2025**, *55*, 244. [CrossRef]
35. Stindl, T.; Chen, F. EM algorithm for the estimation of the RETAS model. *J. Comput. Graph. Stat.* **2024**, *33*, 341–351. [CrossRef]
36. Ramadan, M.S.; Bitmead, R.R. Maximum likelihood recursive state estimation using the expectation maximization algorithm. *Automatica* **2022**, *144*, 110482. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.