

## ARTICLE TEMPLATE

# Generating lane-level road networks from high-precision trajectory data with lane-changing behavior analysis

Mengyue Yuan<sup>a,b</sup>, Peng Yue<sup>a,d,e</sup>, Can Yang<sup>a</sup>, Jian li<sup>b,c</sup>, Kai Yan<sup>b,c</sup>, Chuanwei Cai<sup>a</sup> and Chongshan Wan<sup>a</sup>

<sup>a</sup>School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, China;

<sup>b</sup>State Key Laboratory of Intelligent Vehicle Safty Technology, Chongqing, China;

<sup>c</sup>Chongqing Changan Automobile Co. Ltd, Chongqing, China; <sup>d</sup>Hubei LuoJia Laboratory, Wuhan, China; <sup>e</sup>Hubei Province Engineering Center for Intelligent Geoprocessing (HPECIG), Wuhan University, Wuhan, China

## ARTICLE HISTORY

Compiled November 1, 2023

## ABSTRACT

Recent advances in mobile mapping systems have facilitated the collection of high-precision trajectory data in centimeter positioning accuracy. It provides the potential to infer lane-level road networks, which are essential for autonomous driving navigation. This task is challenging due to the complicated lane merging and diverging structures as well as the lane-changing patterns in trajectory data. This paper presents a lane-level road network generation method from high-precision trajectory data with lane-changing behavior analysis. Trajectories are firstly partitioned by detecting intersections and changes in lane structure. Subsequently, in regions with consistent lane structure, a principal curve fitting algorithm is developed to extract lane centerlines. Erroneous lanes generated by lane-changing behavior are pruned based on a constructed lane intersection graph. In regions with merging and diverging lanes, a lane-group fitting algorithm is designed. This algorithm estimates lane locations by incorporating a Gaussian mixture model with lane width prior knowledge and then infers lane-level topological structures using trajectory flow information. The proposed method is evaluated on a real-world high-precision trajectory dataset. Comprehensive experiments demonstrate that it outperforms state-of-the-art methods in four metrics. Under complex scenarios, the method is capable of generating lane-level road networks with higher completeness and fewer fragments.

## KEYWORDS

Lane-level road network; high-precision trajectory data; lane-changing behavior

## 1. Introduction

The popularity of autonomous driving has promoted the study of high-definition maps. As a fundamental component of autonomous driving systems, high-definition maps provide accurate and rich semantic information, which contains critical elements (e.g., roads, lanes, traffic signals, and road markings) of the driving environment (Liu *et al.* 2020). The lane-level road network is a crucial component of high-definition maps and essential for autonomous driving navigation. Lane-level road network information

could be extracted from various sources, including high-resolution remote sensing imagery (Bastani *et al.* 2018), LiDAR data (Joshi and James 2015), and trajectory data (Arman and Tampère 2021). The limited availability of updated satellite imagery and the effect of weather conditions and obstructions on image quality impose limitations on image-based methods. While LiDAR data provides detailed 3D information on road surfaces, it can be costly due to expensive sensors. In contrast, trajectory data offers advantages in real-time data availability, high revisit frequencies, and cost-effectiveness. Advanced positioning techniques, such as Real-time kinematic, enable the collection of trajectories with centimeter accuracy, providing the potential to accurately infer lane-level road networks. The advancement and cost reduction of positioning devices have greatly enhanced the accessibility of high-precision trajectory data. A common resource of high-precision trajectory data is the mobile mapping system (Elhashash *et al.* 2022), which integrates mapping sensors on moving platforms like vehicles. It provides a reliable and cost-effective data resource for lane-level information.

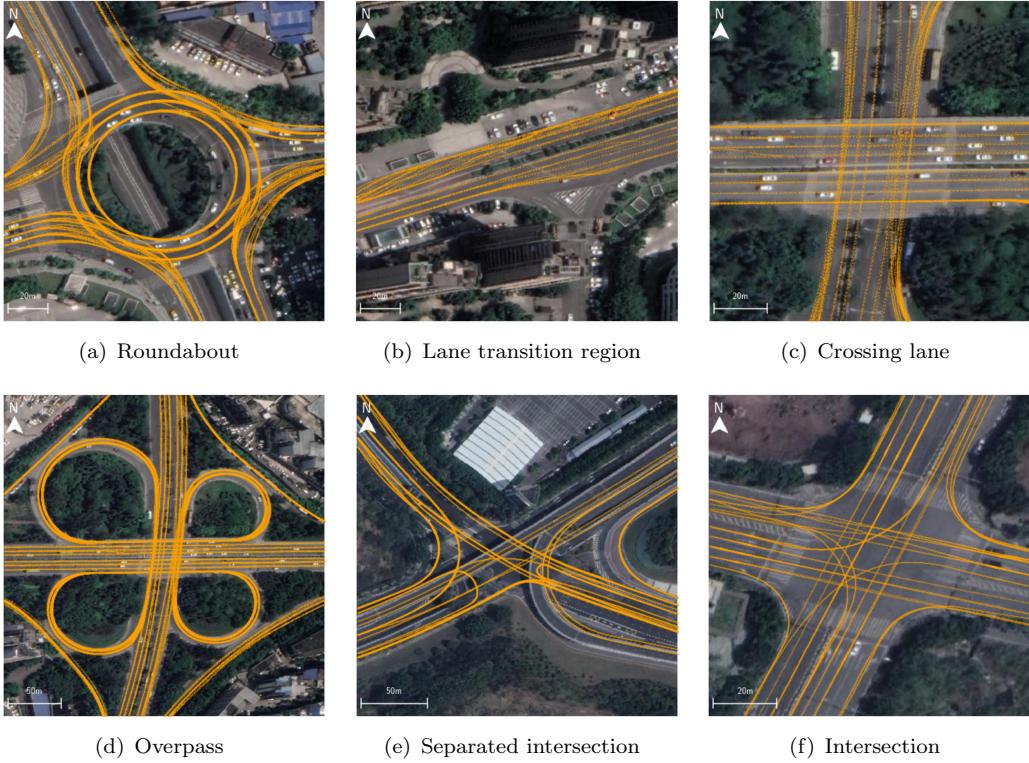
Generating lane-level road networks from high-precision trajectory data confronts two challenges. The first one is the complex topology structure at lane level, which is generally more complicated than that of road level due to the presence of merged and split lanes. Several scenarios are displayed in Figure 1. The complexity of lane structure comes from the intricate layout and connectivity. The challenge is partly caused by the conceptual model of lane-level road networks. Although there is no widely accepted standard for lane-level road networks, it is a common practice to represent each lane separately. The second challenge lies in uncertain driving patterns recorded in high-precision trajectory data. As shown in Figure 1(b), the lane-changing<sup>1</sup> behavior causes trajectories to connect parallel lanes. It could interfere with the lane topology inference in common road network generation procedures that rely on trajectory connection.

Most of the existing map inference methods are devoted to generating road-level networks without considering lane-level information, e.g., lane number, lane width, and lane geometry. Generating a lane-level road network could be a challenging task. Several complex circumstances are introduced by He *et al.* (2018), including parallel lanes, overpasses, underpasses, and crossroads. In addition, the uncertain lane-changing behavior in high-precision trajectory data further complicates the analysis. In traffic flow theory, lane-changing behavior is classified as discretionary and mandatory lane-changing behavior (Pan *et al.* 2016). Discretionary lane-changing behavior is performed by the driver's intention for better driving conditions, whereas mandatory lane-changing behavior is generally related to lane transition structure, such as lane merging or diverging, where trajectories must be adjusted to accommodate the changing lane structure. However, the impact of lane-changing behavior on lane-level road network generation remains under-studied.

To address these problems, this paper proposes a novel method for generating lane-level road networks from high-precision trajectory data. The proposed method partitions the trajectories based on detected intersections. Trajectory segmentation is conducted based on the identification of lane transition, which is defined as a specific road network structure where lanes merge or diverge. The trajectories are divided into two types of regions: lane transition regions, where lanes merge or diverge, and lane consistent regions, where lane structures remain consistent. Then, specialized lane generation strategies are designed for both types of regions. Compared to previous

---

<sup>1</sup>We use the term lane-changing to describe the driving behavior in trajectory data while lane transition refers to special road network structure where lanes merge or diverge.



**Figure 1.** Common scenarios of complex road network infrastructure and high-precision trajectory data.

studies, this paper specifically acknowledges and tackles the challenges arising from lane structure changes and the interference caused by lane-changing behavior. The main contributions of this paper are highlighted as follows:

- We develop a novel lane-level road network generation method. The method adopts a divide-and-conquer strategy for complex road network extraction and addresses the challenge posed by lane structure changes and interference of lane-changing behavior.
- Special designs are developed for high-precision trajectory data with lane-changing behavior analysis. A novel algorithm is proposed that constructs a lane intersection graph and searches maximum clique for pruning erroneous lanes caused by lane-changing behavior.
- A trajectory-based principal curve fitting algorithm is designed to accurately construct lane centerlines. For lane transition structures with merging and diverging lanes, a lane-group fitting algorithm is proposed to estimate lane locations and further construct lane topological structures.
- Comprehensive experiments are performed on a real-world trajectory dataset, and results compared with state-of-the-art (SOTA) algorithms demonstrate the superior performance of the proposed method for generating lane-level road networks in complex circumstances.

The remainder of this paper is organized as follows. Section 2 presents related work. Section 3 describes the proposed method of lane-level road network generation. Section 4 presents the experiments and evaluation of the proposed method. Finally, the conclusion is drawn in Section 5, and future works are discussed.

## 2. Related work

Inferring maps from trajectory data has been intensively studied in the past several decades. Map inference refers to the process of utilizing trajectory data to infer, reconstruct, and continuously update road networks. Early map inference works are surveyed and compared by Biagioni and Eriksson (2012a), Ahmed *et al.* (2014). A more recent survey can be found at Chao *et al.* (2020). In this section, we briefly divide existing map inference work from trajectory data into four categories, including graph-based methods, image-based methods, intersection-linking methods, and others.

Graph-based map inference methods maintain a graph and incrementally update it based on trajectories. An incremental map generation method is developed by Bruntrup *et al.* (2005). Cao and Krumm (2009) developed a trajectory merging method by iterating through the trajectories and inserting new edges into a graph. Chen *et al.* (2016) developed a similar graph generation process by ensuring the local smoothness of road segments. A map expansion algorithm is designed by He *et al.* (2018) where the search is performed by iterating over graph nodes, and the topology is updated by considering both flow and distribution information of trajectories. Yang *et al.* (2018c) studied the incremental update problem for lane-level road networks by combining map matching (Yang and Gidófalvi 2018) and change detection. This paper focuses specifically on generating road networks from scratch, which has not been fully addressed yet.

Intersection-linking methods rely on the accurate detection of intersection locations, which are then connected by trajectories (Karagiorgou and Pfoser 2012, Stanojevic *et al.* 2018, Wang *et al.* 2015). The work of Wang *et al.* (2015) split the trajectory dataset into the intersection and non-intersection regions, and graphs extracted from two types of regions are then merged. Trajectory clustering is commonly employed in map inference problems to group trajectories passing the same lane or route, such as k-means, mean-shift, and spatial linear clustering (Li *et al.* 2016).

Image-based methods generate density maps from trajectories, from which a skeleton of the road network is extracted. Due to noise embedded in trajectory data, kernel density estimation is commonly applied to smooth the density map (Davies *et al.* 2006, Biagioni and Eriksson 2012b). From the density map, an edge detector could be performed to extract road centerlines, which are generally fragmented. Shi *et al.* (2009) proposed morphological operations to tackle the fragmentation issue. As converting to density map losses direction information, Kuntzsch *et al.* (2016) applied map matching and developed a generative model to connect fragmented road networks.

Several other attempts have been made to tackle the map inference problem. Zheng *et al.* (2017) solved it as a topic extraction problem, which could be computationally expensive for large-scale network generation. Deep learning approaches have also been used for map inference (Ruan *et al.* 2020).

While most of the aforementioned studies focused on generating a complete road network from trajectory data, some studies explored the extraction of parts of road networks, such as intersections. A variety of intersection detection models have been developed, including shape descriptor matching (Fathi and Krumm 2010), clustering of turning points (Wang *et al.* 2017, Yang *et al.* 2018b) and hotspot analysis (Deng *et al.* 2018). Another trend emerges as inferring specific attributes of road segments such as lane number. Counting lane numbers from trajectory data is studied in Chen and Krumm (2010) where the Gaussian mixture model (GMM) was employed to model the distribution of trajectories on multiple lanes. Along this line, Tang *et al.* (2015) built a naïve Bayesian classifier to determine lane number based on trajectory features of

a road plane, such as strip width. In addition to lane number information, extracting lane geometry is extensively studied in more recent work. Zhang *et al.* (2016) designed a lane-level road network model and fit cubic Hermite spline with high-precision trajectories to extract lane geometric information. Yang *et al.* (2018b) focused on lane-level intersection map generation, where geometric matching is performed to match the exit and entry nodes of an intersection. He *et al.* (2018) developed a graph generation algorithm that is capable of building lane-level maps. Arman and Tampère (2021) developed a three-step method that first identifies intersection nodes, then extracts road centerline, and finally estimates lane number with GMM.

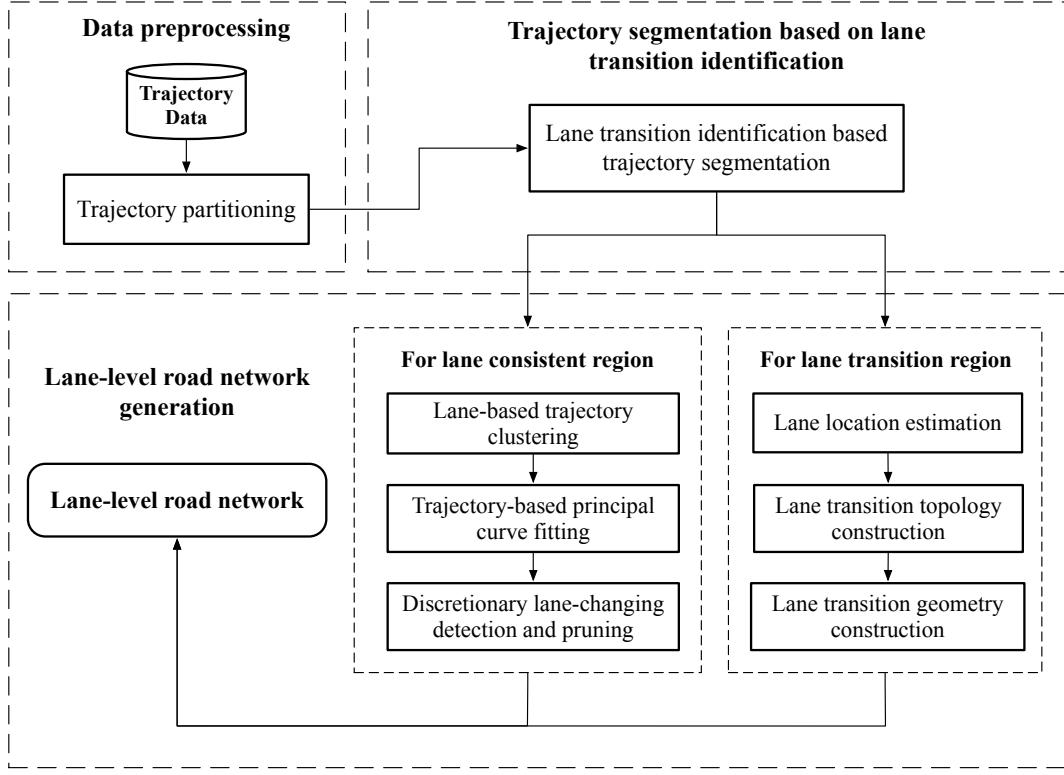
Lane-changing behavior, which is typically observed in high-precision trajectory data, has garnered considerable attention in the transportation research field. Analyzing and modeling lane-changing behavior are essential for various applications, such as lane-changing intention recognition (Balal *et al.* 2016, Bocklisch *et al.* 2017), prediction (Wei *et al.* 2022), and trajectory planning (Yang *et al.* 2018a). Lane-changing models can be categorized into rule-based models (Balal *et al.* 2016), which capture the physical relationships among driving vehicles, and data-driven models (Xie *et al.* 2019, Wei *et al.* 2022), which use extensive training data and machine learning. Despite its importance, the specific impact of lane-changing behavior on road network generation received limited attention in existing literature. Recent work by Arman and Tampère (2022) discussed the potential deviation caused by lane-changing behavior and proposed a trajectory correction method, but their method relied on supplementary information from loop detectors. Further investigation is required for lane-changing behavior in the context of lane-level road network generation.

In summary, preceding studies primarily concentrate on the road-level networks or specific lane attributes but lack comprehensive consideration of complicated topological structures in lane-level road networks, such as how lanes merge or split. The lane generation in complex scenarios has not been comprehensively investigated. In this paper, a novel lane-level road network generation method is developed to overcome the limitations in complex scenarios involving intricate road layouts and lane structure changes. The lane-changing behavior patterns in high-precision trajectory data are considered in the lane generation process. This paper provides a robust and practical method for accurately generating complicated road networks at lane level.

### 3. Methodology

Lane-level road networks are more complex than road-level networks, especially in urban areas with extensive and intricate structures. To manage this complexity, we adopt a divide-and-conquer strategy by partitioning the trajectories. Trajectory segmentation is conducted based on the identification of lane transitions where lanes merge or diverge. Next, specialized lane generation algorithms are designed based on the respective characteristics of lane consistent regions and lane transition regions. The complete workflow of lane-level road network generation is illustrated in Figure 2, which consists of the following steps:

- (1) Data preprocessing. This step localizes intersections for trajectory partitioning and groups trajectories into clusters representing different road segments.
- (2) Trajectory segmentation based on lane transition identification. This step identifies lane transition structures and segments trajectories into lane consistent regions and lane transition regions.



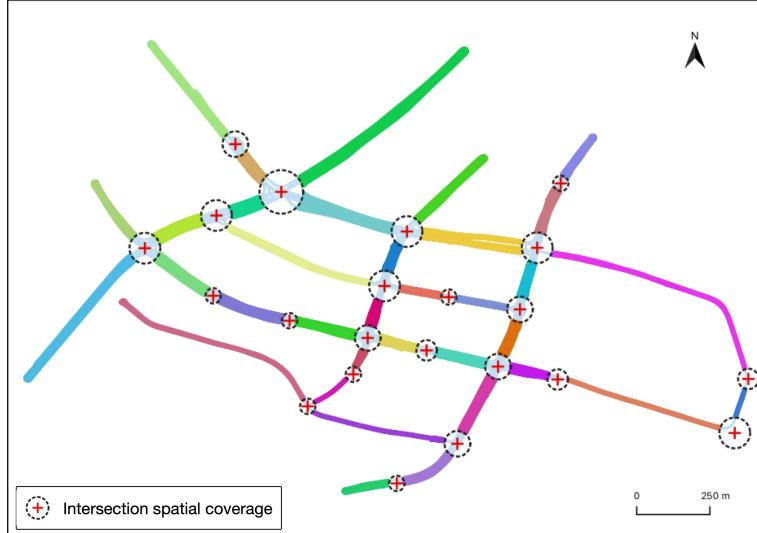
**Figure 2.** The framework of lane-level road network generation from trajectory data.

- (3) Lane generation in lane consistent regions. For lane consistent regions, lanes are identified by trajectory clustering, and a novel trajectory-based principal curve fitting algorithm is developed to extract lane centerlines. Erroneous lane centerlines caused by discretionary lane-changing behavior are detected and pruned.
- (4) Lane generation in lane transition regions. A lane-group fitting algorithm is developed to extract lane transition structures by estimating lane locations and constructing lane transition topology and geometry.

### 3.1. Preliminaries

The global positioning system (GPS) observation point of trajectory is represented by  $p_i = (t_i, x_i, y_i, a_i, v_i)$ , where  $t_i$  records the timestamp, the coordinates  $(x_i, y_i)$  denote the longitude and latitude,  $a_i$  denotes the azimuth of the heading direction expressed in degrees ( $0^\circ - 360^\circ$ ), and  $v_i$  is the current driving speed. A trajectory is represented as a sequence of  $n$  GPS observation points, denoted as  $tr = \langle p_1, \dots, p_n \rangle$ .

The lane-level road network is represented by  $G = (V, E)$ , where  $V$  is a set of vertices, and  $E$  is a set of directed edges representing lanes. In the graph representation of a lane-level network, each edge represents an individual lane, and multiple lanes can belong to a road segment. Each vertex in  $V$  is stored as a point representing the source or target node of the lane. Each  $l \in E$  is a directed lane, denoted by  $l = (id, n_s, n_t, geometry)$ . The lane starts from a source node  $n_s$  and ends at a target node  $n_t$ , and the geometry is represented as a polyline.



**Figure 3.** Illustration of trajectory partitioning. The red cross and dashed circle represent the spatial coverage of intersections, and sections with different colors represent road segments. Best viewed in color.

### 3.2. Data preprocessing

The data preprocessing step partitions raw long-distance trajectories to reduce the complexity of road network construction, avoiding the representation of compound road structures. In this approach, road intersections are considered as circle boundaries for separation, dividing the test area into regions representing road intersections and road segments.

First, road intersections, which represent regions where two or more roads meet or cross, allowing trajectories to turn in different directions, are detected for the purpose of partition. The spatial coverage of intersections is represented as a circle boundary. The detection of road intersections and inference of internal intersection structures have been investigated by Yang *et al.* (2018b). The radius of the intersection range is determined by spatial clustering of turning change points and finding the minimum circumscribed circle that encompasses the clustered points. The construction of the intersection's internal structures could be referenced in their work.

With the road intersection as separation, the raw trajectories are split into sub-trajectories. Each sub-trajectory of the road segment starts at an intersection boundary and ends at another adjacent boundary. To efficiently group the sub-trajectories that correspond to each road segment, agglomerative trajectory clustering is performed based on origin-destination (OD) pairs of the sub-trajectories. The clustering process involves the following steps. First, each sub-trajectory is initialized as an individual cluster, and then the distance similarity between each pair of clusters is calculated using the Euclidean distance between their OD pairs. Then, the two closest clusters are merged into a new cluster, and the distance similarity between the new cluster and the remaining clusters is recalculated. The closest clusters are iteratively merged until they exceed the set distance threshold of 100 meters, which is empirically defined to distinguish different road segments.

The output of preprocessing is the trajectory partition results, as shown in Figure 3. The preprocessing step ensures the resulting trajectory partitions consist of homogeneous road structures without complex structures with multiple turns. Then, trajectory segmentation is conducted on the road non-intersection trajectories.

### 3.3. Trajectory segmentation based on lane transition identification

Lane transition structures, where lanes merge or diverge, are critical components of the topological structure in lane-level road networks. In this section, we introduce a trajectory segmentation method that identifies the lane transition structures within road segments and divides the trajectories into lane consistent regions and lane transition regions.

To identify lane transition structures, the trajectory points are projected onto the Frenét coordinate system (Werling *et al.* 2010), which uses the longitudinal distance and lateral distance to describe the position of a trajectory from a reference path. First, a reference path is determined. In our study, the reference path is the center trajectory of the road segment. The center trajectory is determined by calculating centerline positions between the outermost two trajectories of the road segment and selecting the trajectory with the minimum Hausdorff distance to the centerline positions. The reference path is denoted by:

$$\tau_r = \langle (x_1, y_1, \vec{t}_1, \vec{n}_1), (x_2, y_2, \vec{t}_2, \vec{n}_2), \dots, (x_m, y_m, \vec{t}_m, \vec{n}_m) \rangle \quad (1)$$

where  $(x_i, y_i)$  is the position of  $i$ -th reference point in the global coordinate system,  $\vec{t}_i$  is the tangent vector at the reference point,  $\vec{n}_i$  is the normal vector, which is the unit vector perpendicular to the tangent vector.

Next, for each trajectory point  $(x_t, y_t)$ , the longitudinal distance  $s$  along the reference path and the lateral distance  $d$  from the reference path are calculated:

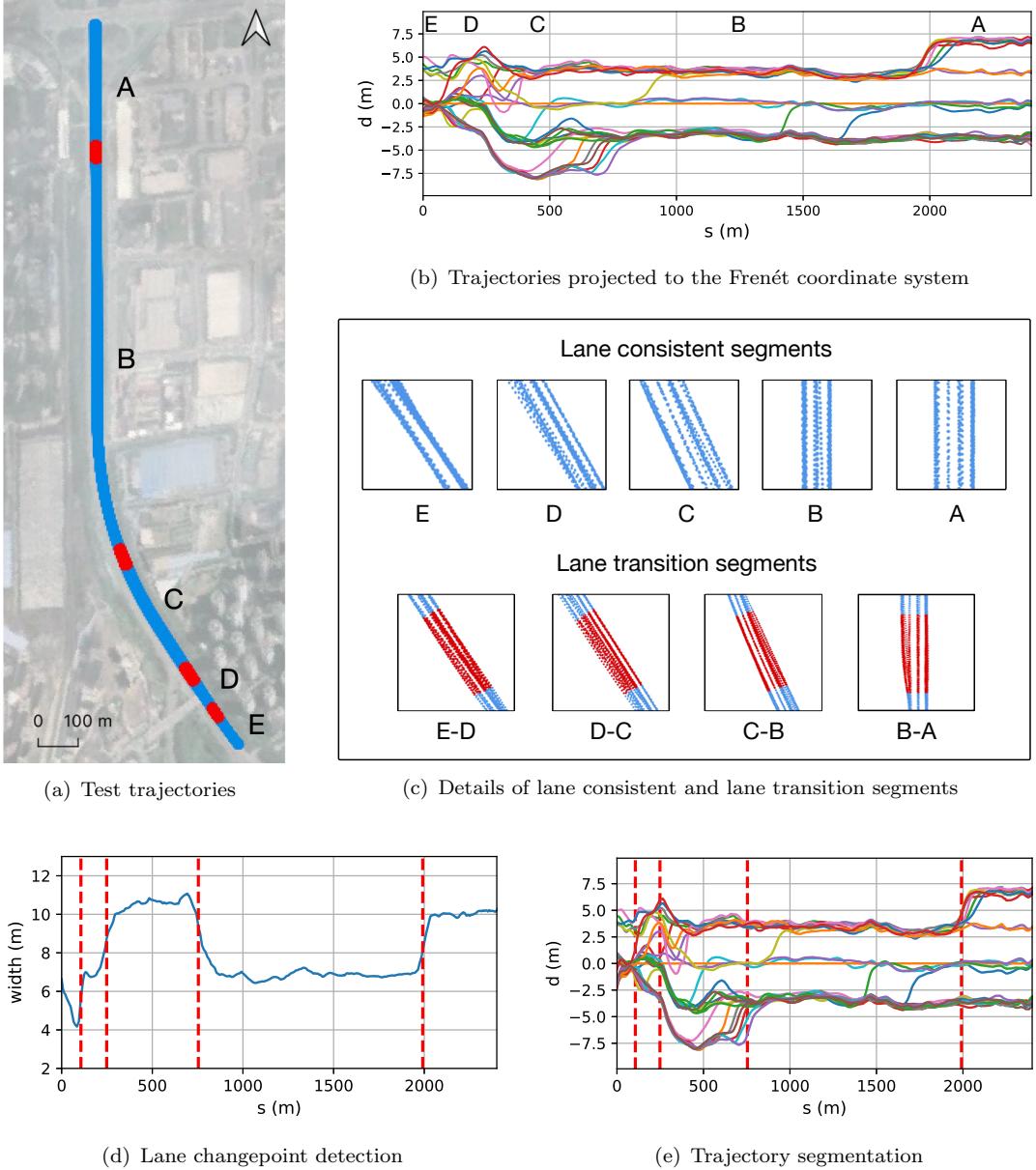
$$s = \sum_{i=1}^r \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}, \quad (2)$$

$$d = \pm (x_t - x_r, y_t - y_r) \cdot \vec{n}_r^T$$

where  $r$  is the index of the closest reference point  $(x_r, y_r)$  on the reference path for the trajectory point  $(x_t, y_t)$ . The longitudinal distance  $s$  is defined as the length along the reference path from the start of the reference path to the closest reference point. The lateral distance  $d$  is the signed projection distance of the trajectory point to the reference path. The sign of  $d$  depends on the side of the reference path on which the trajectory point is located. It is positive when the trajectory is to the right of the reference path and negative when it is to the left. Based on these, we obtain the trajectory representation on the Frenét coordinate system:

$$\tau_f = \langle (s_1, d_1), (s_2, d_2), \dots, (s_m, d_m) \rangle \quad (3)$$

After projecting the trajectories onto the Frenét coordinate system, the distribution of the lateral distances reveals the pattern of lane structure changes. Figure 4 gives an illustration of the segmentation process based on lane transition structures. Trajectories in a road segment, as depicted in 4(a), are projected from the global coordinate system onto the Frenét coordinate system. The projection result is presented in Figure 4(b). The lateral distance distribution remains stable in consistent lane regions, and changes significantly in lane transition regions. Figure 4(c) displays the corresponding details of the trajectory distribution in lane consistent and lane transition segments. Trajectory data reveals that a significant number of mandatory lane-changing behaviors indicate the formation of lane transition structures.



**Figure 4.** An example of trajectory segmentation based on lane transition identification.

To differentiate between lane-consistent regions and lane-transition regions, we consider the lane transition identification problem as a changepoint detection problem inspired by time series analysis. We employ the linearly penalized segmentation, a changepoint detection method proposed by Killick *et al.* (2012), which minimizes a cost function over possible changepoints to identify the optimal number and location of changepoints. The overall width distribution of the lateral distances serves as the input for the changepoint detection algorithm. The output is a series of  $K$  changepoints  $\tau_c = \langle (s_1, d_1), (s_2, d_2), \dots, (s_K, d_K) \rangle$ , as illustrated in Figure 4(d). Then, detected changepoints are mapped onto trajectories via the locations of reference positions  $s$ , as illustrated in Figure 4(e).

Considering that lane transitions occur gradually over a certain distance rather than at a single point, we define a buffer window around the changepoints to extract

trajectories within lane transition regions. The buffer window extends symmetrically on both sides of the changepoints along the longitudinal axis. To ensure adequate coverage for lane transition regions, we empirically set the window width to 80 meters. By extracting the trajectories located in lane-transition regions within buffer windows, trajectories are partitioned into lane consistent and lane transition regions.

### **3.4. Lane generation in lane consistent regions**

In lane consistent regions, the high-precision trajectory data exhibit clustering patterns that help lane identification. Nevertheless, the discretionary lane-changing behavior could cause inaccurate topology. Based on these, we propose a lane generation method for lane consistent regions. The method first generates lane centerlines through lane-based trajectory clustering and trajectory-based principal curve fitting. Then, erroneous lane centerlines caused by discretionary lane-changing behavior are pruned.

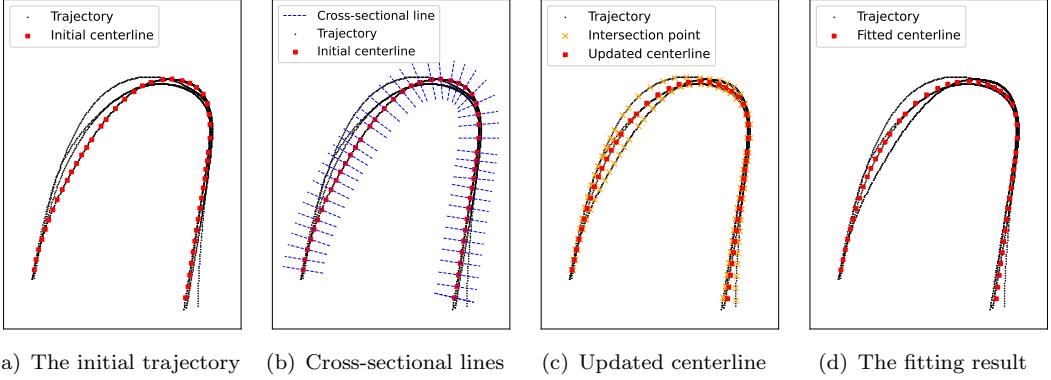
The initial step of lane extraction is to cluster the trajectories passing through specific lanes. In high-precision trajectory data, trajectories tend to exhibit clustering patterns on specific lanes. The lane clusters are formed by lane-based trajectory clustering, which groups trajectories based on their distance similarity. The Hausdorff distance is utilized to measure the distance similarity between two trajectories by computing the distance between each point of one trajectory and its nearest point of the other trajectory, which is defined as the maximum of these distances. The agglomerative clustering based on the Hausdorff distance of trajectories is employed to extract lane clusters. The clustering process terminates when the distance between two closest clusters exceeds a predefined distance threshold, which represents the maximum lane width. The lane clustering results are subsequently used for lane centerline extraction. In the lane clustering process, the discretionary lane-changing trajectories that cross over one lane into another lane, are treated as separate clusters. These trajectories are detected and pruned in later steps to further refine the lane extraction results.

In the following subsection, a trajectory-based principal curve fitting algorithm for lane centerline extraction is presented in 3.4.1, and the discretionary lane-changing detection method is introduced in 3.4.2.

#### *3.4.1. Trajectory-based principal curve fitting (TPCF) algorithm*

Extracting lane centerlines from trajectory clusters could be considered a curve-fitting problem. Previous methods for lane centerline extraction, such as spline curves (Zhang *et al.* 2016) and density estimation (Uduwaragoda *et al.* 2013), require manual parameterization of curve order or substantial data for estimation. In this paper, we propose a non-parametric and automatic algorithm for lane centerline extraction by extending the concept of the principal curve. The principal curve was first proposed by Hastie and Stuetzle (1989), which is defined as a smooth curve that best represents the intrinsic of multi-dimensional data sets. It is a non-parametric method extracting and iteratively updating the one-dimensional principal component based on nearby data points until it reaches a self-consistent state. The traditional principal curve fitting method uses point set data as input, whereas trajectory data is inherently sequential of line type and contains temporal information, which provides the potential to improve the fitting performance.

By extending the concept of the principal curve to trajectory data, we develop a trajectory-based principal curve fitting (TPCF) algorithm for lane centerline extraction. It incorporates the temporal characteristics of trajectories and utilizes an iterative



(a) The initial trajectory (b) Cross-sectional lines (c) Updated centerline (d) The fitting result

**Figure 5.** Schematic flow of the trajectory-based principal curve fitting algorithm. (a) A trajectory is randomly selected as the initial lane centerline. (b) Cross-sectional lines for the current lane centerline are constructed. (c) Geometric centers of intersection points between cross-sectional lines and trajectories are iteratively calculated to update the lane centerline. (d) The fitting result after iteration to convergence.

---

#### Algorithm 1: Trajectory-based Principal Curve Fitting algorithm

---

**Input:**  $T$ : cluster of trajectory set belonging to the same lane;  
 $\delta$ : length of sectional line;  
 $\epsilon$ : convergence threshold

**Output:** Lane geometric information

```

1 Initialize: Randomly select a trajectory  $\tau$  as the initial lane centerline
2 while  $\|\tau_p - \tau\|_2 > \epsilon$  do
3   // Create an empty list to store updated lane points
4   UpdatedCts = [ ];
5   for each point  $p$  in  $\tau$  do
6     // Construct the cross-sectional line for the current point  $p$ 
7      $L_s = \text{CrossSectionLine}(p, \delta);$ 
8     // Calculate the intersection points between  $L_s$  and trajectory set  $T$ 
9     Pts = FindIntersections( $L_s, T$ );
10    // Calculate the geometric center of intersection points
11    Center = ConditionalExpectation(Pts);
12    // Add the calculated center to the set of updated lane points
13    UpdatedCts.append(Center);
14  end
15   $\tau_p = \tau;$ 
16   $\tau = \text{UpdatedCts};$ 
17 end
18 Return  $\tau$ 
```

---

optimization approach to fit curves that represent lane centerlines. Given a cluster of trajectories  $T$  traversing the driving lane, the algorithm estimates the shape of the lane centerline. Figure 5 illustrates the schematic of the iterative optimization process of the TPCF algorithm. The algorithm consists of the following steps:

- (1) Initialization: A random trajectory is selected and sampled at a fixed interval distance to obtain the initial lane centerline (Figure 5(a)).
- (2) Create cross-sectional line  $L_s$  with the length of  $\delta$  for each sample point on the

- current lane centerline  $\tau_p$  (Figure 5(b)).
- (3) Calculate the intersection points of each cross-sectional line with input trajectories  $T$ . The geometric centers of the intersection points are calculated, and the sequence of new centers is updated as the fitted lane centerline  $\tau$  (Figure 5(c)).
  - (4) The iterative process of (2) and (3) continues until the convergence condition is satisfied  $\|\tau_p - \tau\|_2 < \epsilon$ . The distance between the centerline in current iteration  $\tau$  and the one in previous iteration  $\tau_p$  is calculated. Convergence is considered to be achieved when the distance falls below a predefined threshold  $\epsilon$ . Then, the final fitted lane centerline is obtained (Figure 5(d)).

The pseudocode of the TPCF algorithm is presented in Algorithm 1.

### 3.4.2. Discretionary lane-changing detection by maximum clique search

Discretionary lane-changing behavior occurs frequently in driving scenarios for various reasons, such as overtaking and cooperative driving. The discretionary lane-changing trajectories could lead to incorrect paths connecting parallel lanes and topology errors. To accommodate the additional complexity introduced by lane-changing behavior recorded in high-precision trajectory data, we present a novel algorithm that detects and prunes erroneous lane centerlines caused by discretionary lane-changing behavior, thus improving the extracted lanes.

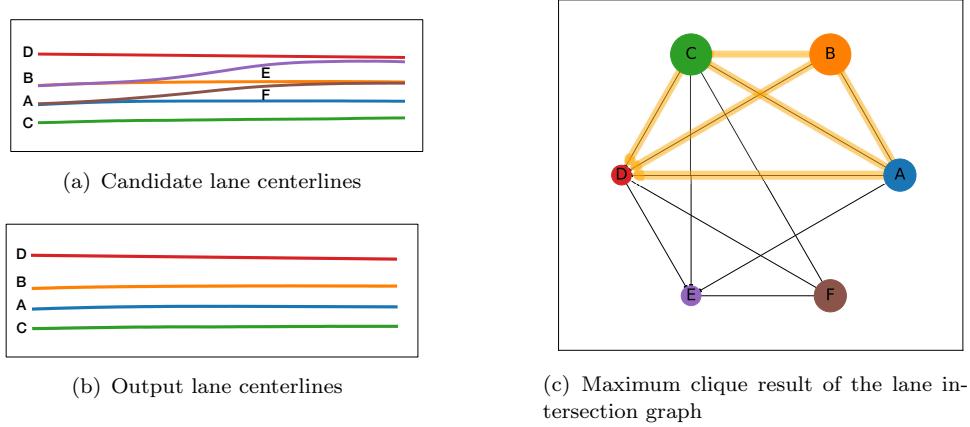
Inspired by graph-theoretic algorithms, we formulate the lane-changing detection problem as a maximum clique problem (MCP). The MCP is to find a maximum clique, which is the largest possible complete subgraph in a graph where each vertex is connected to all other vertices in the subset. Let  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  be an undirected graph with vertex set  $\mathcal{V} = \{1, \dots, N\}$  and edge set  $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ . A clique  $\mathcal{C}$  in  $\mathcal{G}$  is a subset of vertices that are all mutually connected to each other, i.e.  $\forall i, j \in \mathcal{C}, \{i, j\} \in \mathcal{E}$ . A maximum clique refers to the largest complete subgraph, whereas a maximum weight clique refers to the clique with the maximum total weight of vertices.

The task of discretionary lane-changing detection is formulated as a graph search problem. For the candidate lane centerlines as shown in Figure 6(a), there are some erroneous lane centerlines, caused by discretionary lane-changing behavior, need to be eliminated and obtain the correct lane centerlines depicted in Figure 6(b). A lane intersection graph is constructed as illustrated in Figure 6(c), where vertexes represent the candidate lane centerlines and edges represent the intersection relationship between lanes. An edge of the graph is constructed if two lanes do not intersect with each other. For each vertex  $i \in \mathcal{V}$ , there is a positive weight  $w_i$  that represents the number of trajectories that pass through this lane. For example, the maximum clique of the graph in Figure 6(c) is searched as  $\{A, B, C, D\}$ , which corresponds to the correct lane centerlines, and the erroneous lane centerlines  $\{E, F\}$  generated by lane-changing behavior are removed.

The MCP is mathematically formulated as an integer programming problem, with the objective of maximizing a weighted sum, as described below:

$$\begin{aligned} \max_{\mathbf{x}} \quad & \sum_{i=1}^n w_i x_i, \\ \text{s.t.} \quad & x_i + x_j \leq 1, \quad \forall \{i, j\} \notin \mathcal{E}, \\ & x_i \in \{0, 1\}, \quad i = 1, \dots, N. \end{aligned} \tag{4}$$

where  $x_i$  is a binary decision variable that takes the value 1 if vertex  $i$  is in the clique



**Figure 6.** Illustration of the discretionary lane-changing detection by maximum clique search.

and 0 otherwise. The MCP is a combinatorial optimization problem that has been extensively studied. To solve the MCP, we employ the subgraph exclusion method (Boppana and Halldórsson 2005), which is implemented using NetworkX, a Python package for network analysis (Hagberg *et al.* 2008).

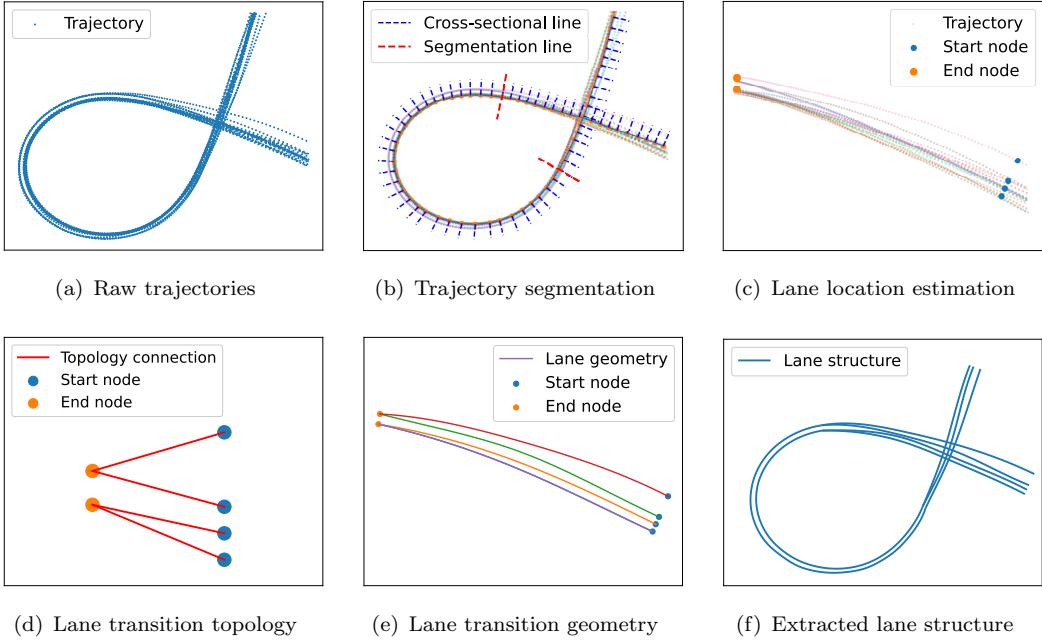
By formulating the problem as finding the maximum clique in the constructed lane intersection graph, the erroneous lane centerlines generated by discretionary lane-changing behavior are detected and removed, and lane centerlines with maximum mutual disjoint relations are obtained as the final extracted lane representation.

### 3.5. Lane generation in lane transition regions

Extracting lane transition structures is a critical and challenging problem in lane-level road network generation. Previous lane extraction algorithms are generally based on the assumption of consistent lane structures. However, lane transition structures in the real world often occur when lane structures change with lane merging or diverging.

To extract lane transition structures, a lane-group fitting algorithm is developed, which estimates lane locations and constructs topological and geometric connections. The algorithm considers groups of lanes as adjacent if they are spatially proximate but with varying lane counts. The process of the lane-group fitting algorithm is illustrated in Figure 7, using a curved ramp section with multiple lane transitions as an example. For the trajectories in Figure 7(a), the segmentation step is performed to segment these structures and form lane groups with similar lane structures, as shown in Figure 7(b). The extraction of lane transition structures consists of the following steps:

- (1) Lane location estimation with prior knowledge: This step introduces a Gaussian mixture model that integrates lane width prior knowledge to estimate the lane locations for the start and end nodes of lane transition structures.
- (2) Lane transition topology construction: This step constructs the lane transition topology based on trajectory flow information.
- (3) Lane transition geometry construction: The lane transition geometry is constructed through spatial projection in the Frenét coordinate system to ensure proper and smooth transition structure.



**Figure 7.** Schematic illustration of lane-group fitting algorithm. (a) Trajectories in the curved ramp section transition from four lanes to two lanes and then to three lanes. (b) A segmentation step is performed to obtain groups of lanes with similar lane structures. (c) For a lane group transitioning from four lanes to two lanes, the lane locations at the starting and ending points of lane transition are estimated. Then, the lane transition topology (d) and geometry (e) are constructed. (f) The final extracted lane transition structure.

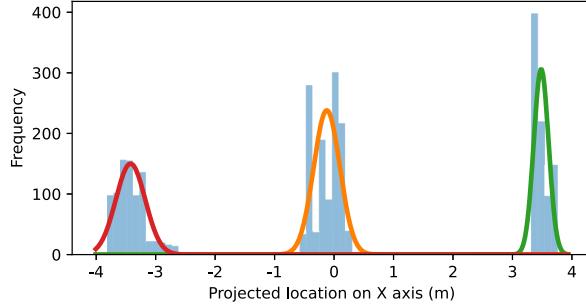
### 3.5.1. Lane location estimation with prior knowledge

In previous studies, the Gaussian mixture model (GMM) has been employed for lane location estimation and extended with various assumptions and parameter constraints (Chen and Krumm 2010, Tang *et al.* 2016, Arman and Tampère 2021). However, these constraints make the parameter estimations in GMM complicated and difficult to handle in complex scenarios like lane transition regions. In this paper, we simplify the complex parameter constraints in GMM and incorporate prior knowledge of variable lane widths to enhance the robustness of lane location estimation.

The GMM is employed to fit the distribution of trajectory points projected onto the road cross-section for lane location estimation, as illustrated in Figure 8. The GMM models the probability density function of the data as a weighted sum of Gaussian distributions, which has the following probability density:

$$p(x) = \sum_{j=1}^k w_j \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{(x - \mu_j)^2}{2\sigma_j^2}\right) \quad (5)$$

where  $k$  denotes the number of Gaussian components, which corresponds to the number of lanes. The parameters  $\mu_1, \dots, \mu_k$  are the means of trajectory projected points for each component, representing the lane locations. The parameters  $\sigma_1^2, \dots, \sigma_k^2$  are the variance for each component. The weights  $w_1, \dots, w_k$  for the component are the relative traffic volume in each lane with  $w_j > 0$  and  $\sum_{j=1}^k w_j = 1$ . The parameters in the GMM with a predefined component number are estimated using the expectation-maximization (EM) algorithm.



**Figure 8.** Gaussian mixture distribution.

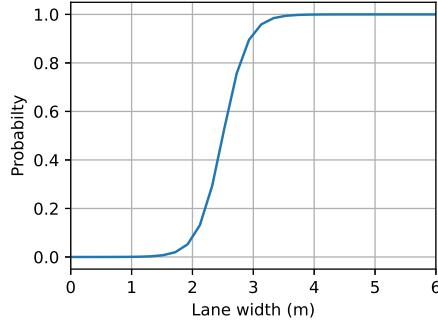
The number of components  $k$  in GMM is determined based on model selection criteria, such as the Bayesian information criterion (BIC) or the Akaike information criterion (AIC) (Bishop and Nasrabadi 2006). A model with a lower AIC or BIC is considered superior. However, we find through experimentation that AIC and BIC do not work well in selecting the number of lanes, as shown in Figure 10(a). As the number of mixture components increases, the model tends to overfit, leading to an inaccurate estimation of the lane number. Therefore, we integrate lane width prior knowledge into the GMM for lane number  $k$  selection. Specifically, we design a novel criterion that incorporates the prior probability distribution of lane width. The criterion  $R(\theta_k)$  is defined as following:

$$R(\theta_k) = -\log \left( \widehat{L}(\theta_k) \right) - \frac{1}{k-1} \sum_{i=1}^{k-1} \log \left( \frac{1}{1 + e^{-(\mu_{i+1} - \mu_i)}} \right) \quad (6)$$

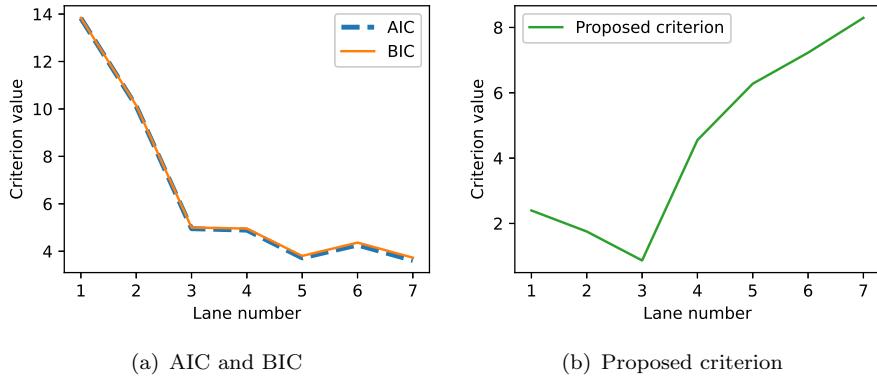
$$k^* = \arg \min_k (R(\theta_k)) \quad (7)$$

where  $\theta_k \triangleq \{w_i, \mu_i, \sigma_i^2\}_{i=1}^k$  is a set of parameters under different value of  $k$ . The maximum likelihood  $\widehat{L}(\theta_k)$  is used to evaluate the data fitting of a model. Further,  $(\mu_{i+1} - \mu_i)$  introduces the lane width into the criterion, calculated as the average distance between adjacent lane centerlines. The lane width is transformed into a probability score by using it as the input for a sigmoid function, which maps the values to a range between 0 and 1, as illustrated in Figure 9. An optimal value for  $k^*$  that minimizes the value of  $R(\theta_k)$  is determined. The criterion with lane width prior probability determines an optimal value of lane number.

In urban road design, the lane width often varies within a specific range rather than a fixed value, especially in lane transition scenarios. The lane widths typically range between 3 to 3.75 meters. In some cases, the distance between lane centerlines may be relatively large due to data inadequacy. Therefore, we design the lane width preference in the prior probability. The proposed criterion effectively incorporates prior knowledge of the variable lane width. As shown in Figure 10, the proposed criterion autonomously selects the lane number and performs better than AIC and BIC. After that, lane locations are determined by solving the parameters of the GMM.



**Figure 9.** Prior probability distribution of variable lane width.



**Figure 10.** Criterion for lane number selection in Gaussian mixture distribution depicted in Figure 8. Akaike information criterion (AIC), Bayesian information criterion (BIC), and proposed criterion.

### 3.5.2. Lane transition topology and geometry construction

This section introduces the construction of the topological and geometric connections between the start and end lane nodes of lane transition regions to ensure the proper connection and smooth lane transitions.

The topological relationship is determined based on the trajectory flow information, which refers to the movement patterns of vehicles through neighboring lane nodes. A lane transition probability matrix is created. Each element of the matrix, denoted as  $e_{ij}$ , represents the transition probability from the start node  $i$  to the end node  $j$ . This probability could be calculated as:

$$e_{ij} = \frac{N_{ij}}{\sum_{i=1}^{N_s} \sum_{j=1}^{N_e} N_{ij}} \quad (8)$$

where  $N_{ij}$  represent the number of trajectories passing from lane start node  $i$  to lane end node  $j$  within the lane transition region.  $N_s$  and  $N_e$  represent the total number of start and end nodes, respectively.

Then, the lane topology is established based on the lane transition matrix. Specifically, for each lane start node, we search for the lane end node with the highest transition probability in the lane transition matrix for bipartite matching. Discretionary lane-changing behaviors with low probabilities are disregarded, while mandatory lane-changing behaviors, occurring more frequently with higher probabilities, serve as the primary reference for establishing connections within the lane transition regions. For

the segment in Figure 7(c) as an example, the constructed lane transition topology is illustrated in Figure 7(d).

The geometric connections are constructed based on the topological relationships of the adjacent lane nodes. The complex distribution of trajectories in lane transition scenarios and the variations of road curvature pose increased difficulty. To deal with this, we construct lane geometric connections based on a spatial projection transformation method that ensures continuity in the trajectory movement process. Specifically, we convert the start and end lane nodes of lane transition regions from the global coordinate system into the Frenét coordinate system. Then, the lane start and end nodes with topological relationships are connected with a straight line segment, as shown in Figure 7(d). The linear connection in the Frenét coordinate system is reasonable because the lateral and longitudinal movements of the trajectory are always continuous, even in a lane transition process. After establishing the linear connections in the Frenét coordinate system, line segments between the start and end lane nodes are transformed back into the global coordinate system to obtain the final lane geometry, as displayed in Figure 7(e). This process ensures local smoothness of the lane transition geometry.

Lastly, the spatially adjacent segments are connected by matching and merging the endpoints of adjacent lanes. Specifically, we compute the distance between each lane node and the endpoints of the adjacent segments and match each lane node with the closest adjacent endpoint (Yang *et al.* 2018b). Lane nodes that match within tolerance are merged into the same node by averaging their coordinate. In this way, the connection of adjacent segments could be consistent in the final lane-level road network.

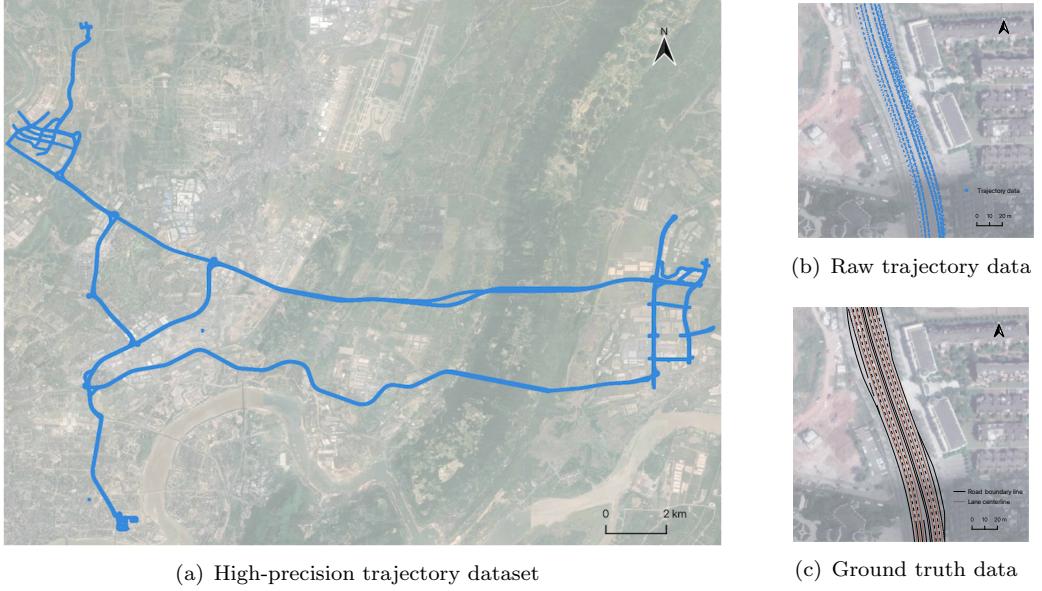
## 4. Experiments and evaluation

### 4.1. Data description and experimental setup

The proposed algorithms are evaluated on a real-world vehicle trajectory dataset collected by Changan Automobile Company in Chongqing, China, in August 2022. The dataset comprises trajectories collected by mobile mapping vehicles equipped with real-time kinematic systems, providing a total of 2,200 trajectories and 1.5 million observation points. Each observation point records the timestamp, longitude and latitude coordinates, azimuth of heading direction, and driving speed. With the data sampling frequency remaining at 10Hz and the positioning accuracy within 20 centimeters, the dataset ensures high precision. Figure 11(a) provides an overview of the dataset.

To evaluate the lane-level road networks extracted by our proposed method, we first obtain ground truth data for lane boundaries. This data is procured by processing point cloud data collected from real roads and using real-time kinematic technology for correction to meet the standards for autonomous driving. The ground truth for lane centerlines, which is the primary focus of our evaluation, is then manually annotated based on the lane boundary data. The detailed views of the raw trajectory data and the ground truth data are displayed in Figure 11(b) and 11(c), respectively.

The proposed algorithms are implemented in Python 3.8, and all the experiments are performed on a laptop computer running the macOS Monterey operating system with an Intel Corei7 processor with a 3.5 GHz CPU.



**Figure 11.** Experimental dataset.

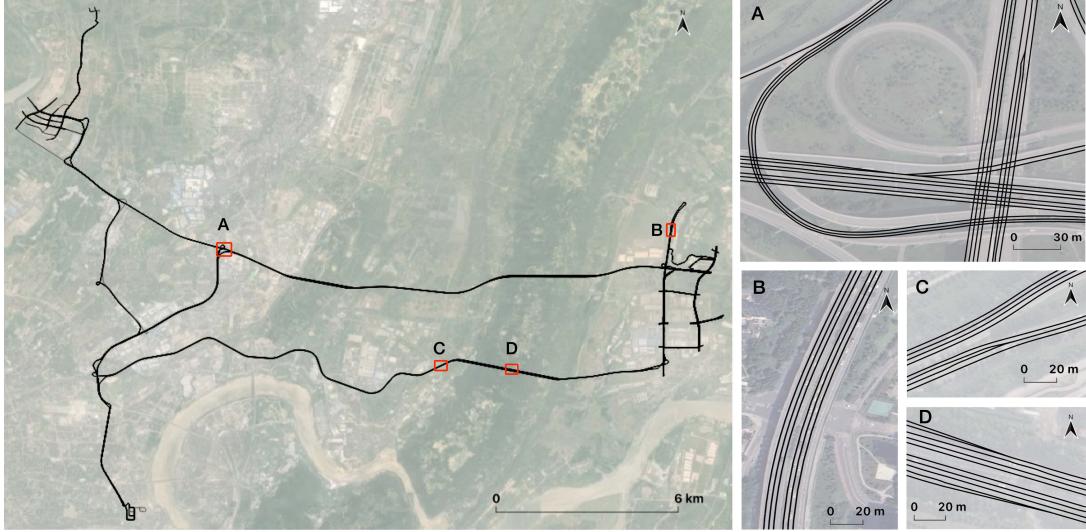
#### 4.2. Evaluation metrics

In the performance evaluation section, we utilize various evaluation metrics to compare the algorithms' performance. These metrics include lane location metrics and other commonly used evaluation metrics in map inference research, as described below.

- Lane location metric. This metric evaluates lane locations based on the Euclidean distance between each extracted lane location and the ground truth. Lane locations within 0.5 meters of the ground truth are designated as true positives. Lane location precision is calculated as the ratio of true positives to total extracted lane locations, whereas the recall is calculated as the ratio of true positives to total ground truth lane locations. The F1 score is also calculated for evaluation.
- TOPO. TOPO metric evaluates the topological similarity of the generated map and ground truth map using the holes and marbles method introduced by Biagioli and Eriksson (2012a). We place "holes" every 5 meters along the edges of the ground truth map within a 300-meter radius from a random initialization seed and drop "marbles" from the nearest corresponding point of the inferred map in the same manner. We then evaluate the maximum one-to-one matching within the 1-meter matching distance between the marbles and holes by:

$$precision = \frac{\text{matched marbles}}{\text{all marbles}}, \quad recall = \frac{\text{matched holes}}{\text{all holes}} \quad (9)$$

- Shortest-path (SP) metric. The SP metric reflects the ability of road networks to provide accurate route planning and navigation information (He *et al.* 2018). This is implemented by randomly sampling 1,000 origin-destination pairs and then calculating the shortest path distance between those on the inferred map, as compared to the ground truth map. The correct rate is calculated as the proportion of pairs with similar shortest paths within a 5% path distance tolerance.
- Junction metric. The Junction metric proposed by Bastani *et al.* (2018) evaluates the junction degree in the generated road network and the ground truth



**Figure 12.** Lane-level road network generation results of the entire study area, with close-up figures of four road sections to showcase lane details (A: road section with intersections, B: lane consistent segments, C and D: lane transition segments).

road network. The junction degree refers to the number of lanes connected to a junction in road networks, which reflects the local topology of road networks. We report the precision, recall, and F1 score for the Junction metric.

#### 4.3. Performance comparison

##### 4.3.1. Comparison with state-of-the-art (SOTA) methods

In this section, we present evaluations of the proposed method and comparisons with SOTA trajectory-based algorithms. We compare the proposed lane-level road network generation algorithm with the baselines briefly described below:

- Cao (Cao and Krumm 2009). This algorithm incrementally merges trajectories into a road network by examining each trajectory to determine whether it can be integrated with existing edges or if new edges are required to be created.
- Chen (Chen *et al.* 2016). This algorithm utilizes trajectory mean-shift sampling and graph-based clustering methods for extracting lane segments and infers links among segments for road network creation.
- Kharita (Stanojevic *et al.* 2018). This algorithm constructs an initial graph with k-means clustering for lane inference and uses a graph spanner to remove redundant edges.
- Arman (Arman and Tampère 2021). They develop a three-step approach that identifies nodes, extracts road centerline, and estimates lane offsets of a whole length of road segment with GMM.

Table 1 presents the performance results of the proposed method in comparison with SOTA algorithms in terms of lane location, TOPO, and Junction metrics. Our proposed method achieves outstanding results, reporting the highest F1 scores in lane location (0.891), TOPO (0.919), and Junction (0.764) metrics. The F1 scores of the proposed method are 7.9%, 8.6%, and 14.5% higher than the best comparative method

**Table 1.** Performance comparison of the proposed method and SOTA methods based on Precision (Prec.), Recall (Rec.), F1 score (F1) for lane location, TOPO, and Junction metrics.

Method	Lane location			TOPO			Junction		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
Cao (2009)	0.820	0.760	0.789	0.821	0.846	0.833	0.485	0.559	0.519
Chen (2016)	0.667	0.607	0.635	0.521	0.768	0.621	0.450	0.468	0.459
Kharita (2018)	0.816	0.808	0.812	0.550	<b>0.968</b>	0.702	0.480	0.523	0.501
Arman (2021)	0.720	0.461	0.562	0.712	0.799	0.752	0.597	0.644	0.619
Ours	<b>0.896</b>	<b>0.884</b>	<b>0.891</b>	<b>0.901</b>	0.939	<b>0.919</b>	<b>0.741</b>	<b>0.788</b>	<b>0.764</b>

The best result of each column is marked in bold.

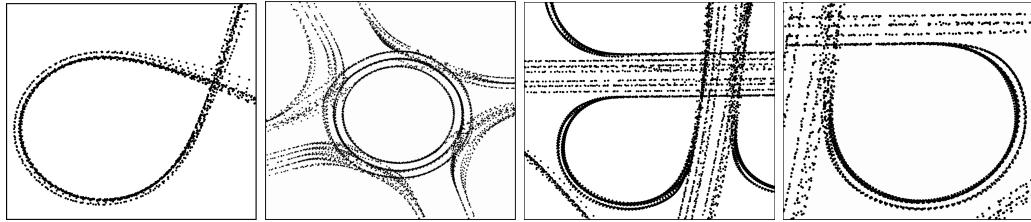
**Table 2.** Shortest-path (SP) metric performance comparison.

Method	Correct rate	Error rate <sup>a</sup>		
		Spurious path	No path	Others
Cao (2009)	0.663	0.115	0.219	0.003
Chen (2016)	0.479	0.153	0.290	0.078
Kharita (2018)	0.569	0.411	0.016	0.004
Arman (2021)	0.665	0.136	0.197	0.002
Ours	<b>0.845</b>	0.143	0.006	0.006

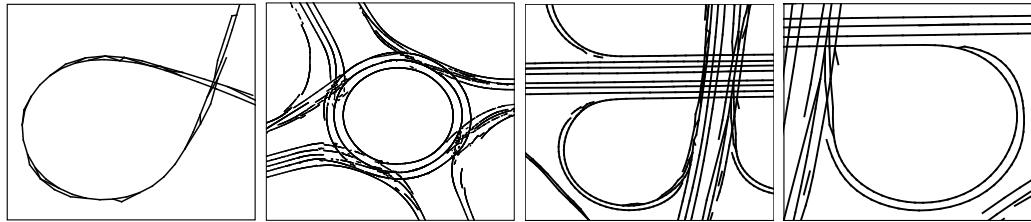
<sup>a</sup> Error reasons are categorized into spurious path, no path, and others.

in lane location, TOPO, and Junction metrics, respectively. In Table 2, we report the performance results of the SP metric, which quantifies road networks' ability to provide accurate shortest-path connectivity. The proposed method exhibits superior performance with the correct rate of 0.845, which is 18% higher than the best comparative algorithm. Cao and Chen's results exhibit lower correct rates due to no path errors caused by fragments. Kharita's result suffers from a high spurious path error, which generates many spurious paths connecting parallel lanes that do not exist. Arman's method generates lanes for the entire length of the road, which appear to be continuous but exhibit deviations from the actual lane positions of raw trajectories. In contrast, the proposed method minimizes spurious connections while preserving accurate and reliable path connectivity. The quantitative results indicate that the proposed method outperforms SOTA algorithms in lane-level road network generation.

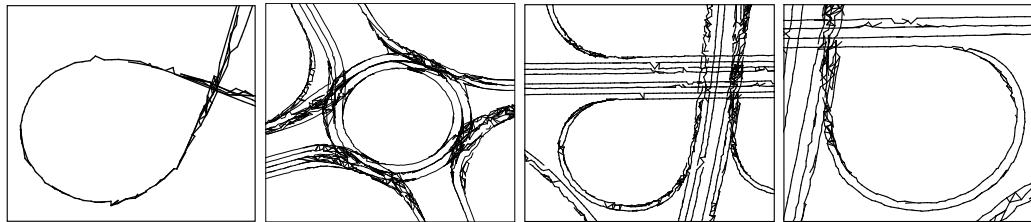
The lane-level road network generation results of the entire study area are presented in Figure 12. Figure 13 displays the visualization results generated by our proposed method and SOTA algorithms in different circumstances, including curved ramps, roundabouts, and overpasses. The results of the proposed method of various cases exhibit superior adaptability to complex road networks. As observed in Figure13(b), Cao's method suffers from disconnected lanes. This is potentially due to some trajectories becoming partially disconnected during the incremental merging process. Chen's method performs the worst in the evaluated metrics, as explained by Figure13(c). It produces numerous erroneous intertwined segments in regions where trajectories overlap or intersect. Kharita, although exhibiting high TOPO recall, often produces spurious fragments connecting parallel lanes, as displayed in Figure13(d). This is due to the interference caused by lane-changing trajectories crossing between parallel lanes. Arman's method tends to generate lanes with positional deviation and overestimate the number of lanes when lane structure changes, as shown in Figure13(e). Compared to other methods, our method is able to generate high-quality lane-level network structures under various scenarios, as displayed in Figure13(f). These generated lane structures are characterized by greater completeness and fewer fragments, which accounts



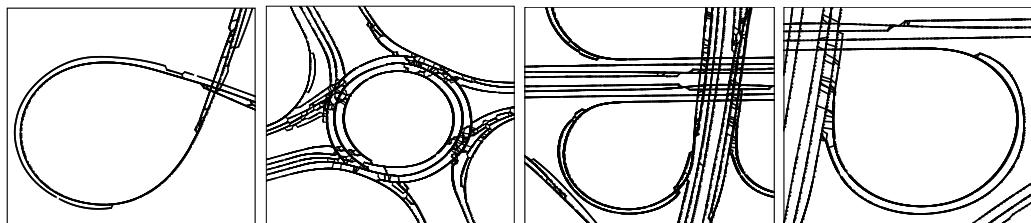
(a) Raw trajectories



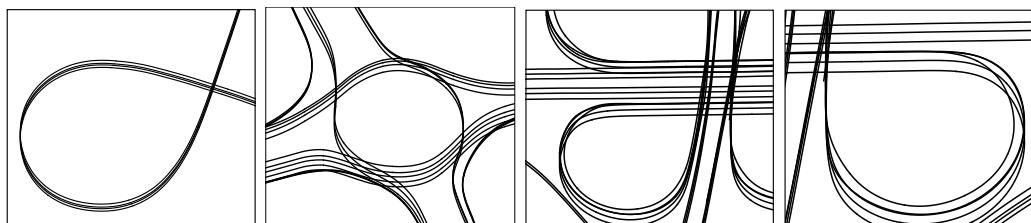
(b) Cao (2009)



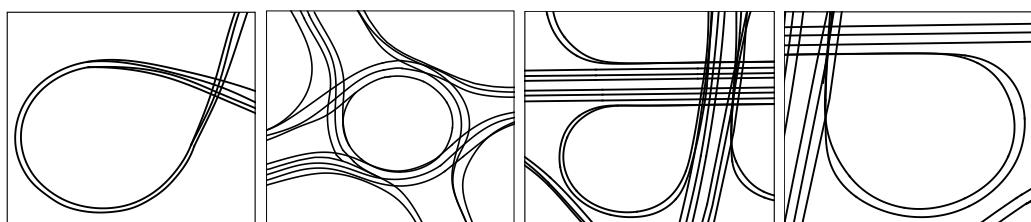
(c) Chen (2016)



(d) Kharita (2018)



(e) Arman (2021)



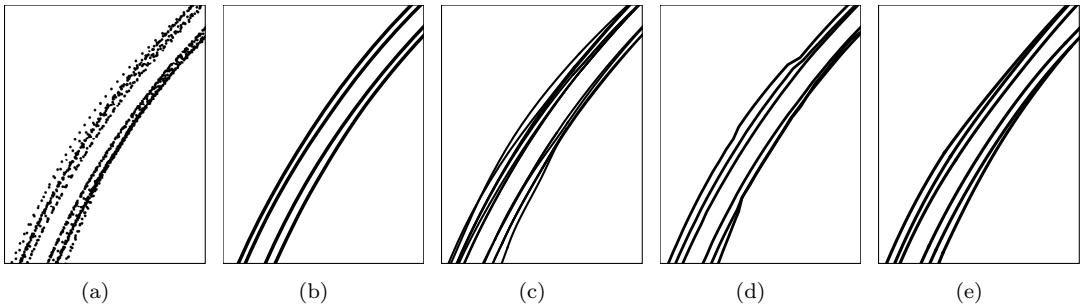
(f) Ours

**Figure 13.** Comparison of lane-level road networks generated by our method with SOTA methods.

**Table 3.** Performance comparison with variants of our proposed method based on lane location precision (Loc. Prec.), lane location recall (Loc. Rec.), lane location F1 score (Loc. F1), and F1 scores for TOPO, shortest-path (SP), and Junction metrics. The components of the proposed method are represented as trajectory segmentation (TS), trajectory-based principal curve fitting (TPCF), lane-changing detection (LCD), and lane-group fitting (LGF).

Variant	Metric					
	Loc. Prec.	Loc. Rec.	Loc. F1	TOPO	SP	Junction
TPCF + LCD	0.831	0.578	0.682	0.861	0.583	0.689
TS + TPCF + LGF	0.667	<b>0.916</b>	0.772	0.767	0.652	0.457
TS + TPCF + LCD	0.852	0.874	0.863	0.899	0.640	0.705
TS + TPCF + LCD + LGF	<b>0.896</b>	0.884	<b>0.891</b>	<b>0.919</b>	<b>0.845</b>	<b>0.764</b>

The best result of each column is marked in bold.



**Figure 14.** Visualization results of variant schemes. (a) Raw trajectories. (b) TPCF + LCD. (c) TS + TPCF + LGF. (d) TS + TPCF + LCD. (e) Complete method: TS + TPCF + LCD + LGF.

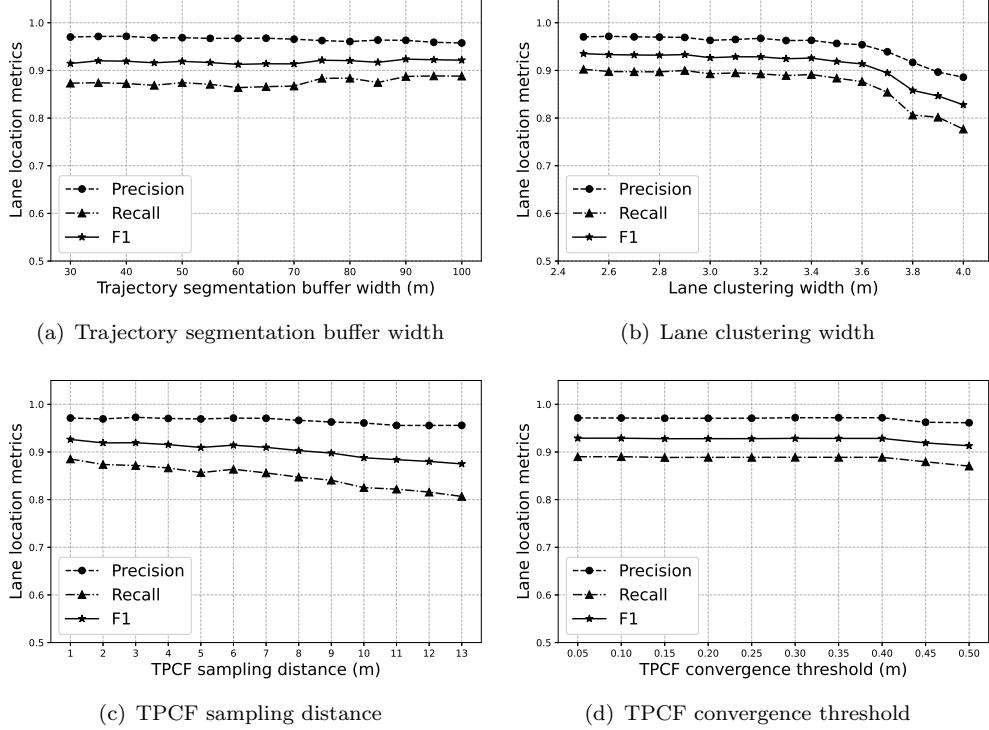
for our method’s superior performance in evaluation metrics.

#### 4.3.2. Comparison with variants of proposed method

To evaluate the effectiveness of the components incorporated in our proposed method, we conduct comparative experiments on several variants. These variants are designed to remove the main components of the proposed method:

- (1) Complete method: The complete method consists of all components, including lane transition identification-based trajectory segmentation (TS), trajectory-based principal curve fitting (TPCF), lane-changing detection (LCD) for lane consistent regions, and lane-group fitting (LGF) for lane transition regions.
- (2) TPCF + LCD: This variant removes the TS part and treats all road segments as lane consistent regions for processing due to the absence of lane transition identification.
- (3) TS + TPCF + LGF: This variant excludes the LCD part for lane consistent regions without dealing with discretionary lane-changing behavior and keeps the other components unchanged.
- (4) TS + TPCF + LCD: For this variant, we remove the LGF part for lane transition regions and instead directly connect the extracted lane locations without constructing lane topological and geometric connections for lane transition regions, while keeping other components unchanged.

The performance comparison results of our complete methodology and its variants are presented in Table 3. Figure 14 displays the visualization results of the generated lanes by variant schemes. The trajectory data in a road segment that transitions from two-lane to three-lane is displayed as a visualization example in Figure 14(a). For the



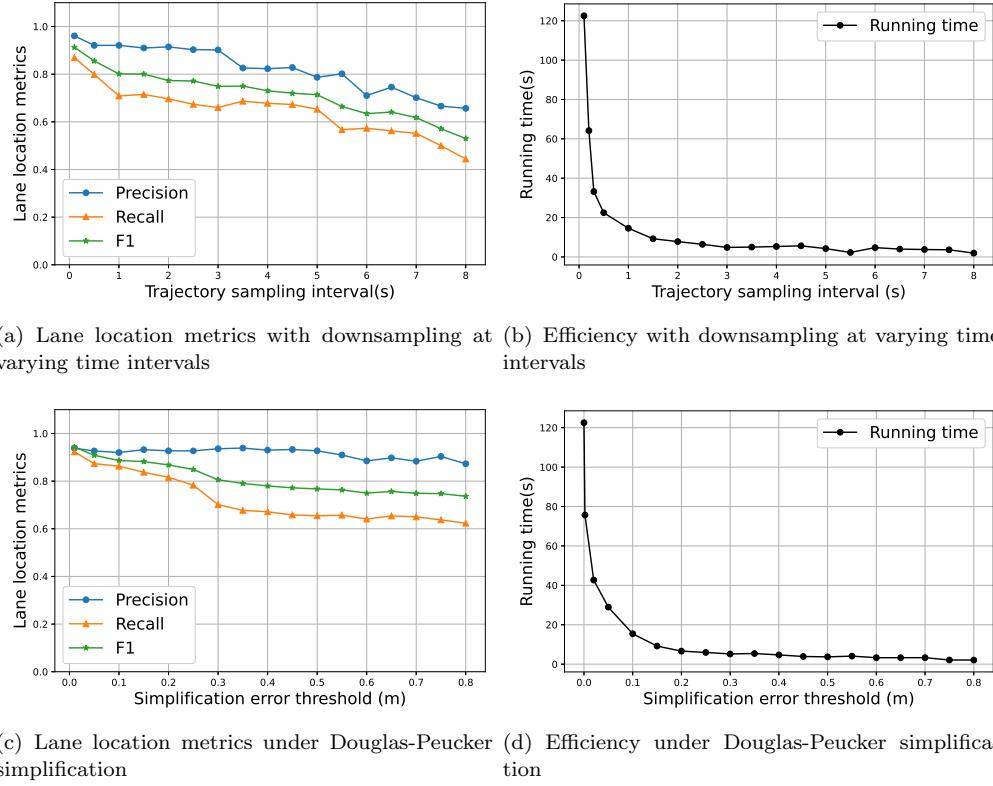
**Figure 15.** Sensitivity analysis of parameters.

variant of TPCF + LCD, which removes the TS part, the performance results are significantly reduced across all metrics. The removal of the TS part leads to the inability to distinguish changes in lane structure, resulting in the missing lanes, as displayed in Figure 14(b). This demonstrates the effectiveness of the trajectory segmentation step based on lane transition identification. The result of the TS + TPCF + LGF variant exhibits numerous redundant lane errors, as shown in Figure 14(c). Although this variant exhibited a high location recall rate, the absence of lane-changing behavior detection and pruning resulted in low precision. For the variant of TS + TPCF + LCD, the result indicates that the improper processing for lane transition regions could result in unrealistic lane transition structures, as presented in Figure 14(d). Although the lane transition regions constitute a relatively small portion of road networks and have limited impact on overall network evaluation, they do affect the local topological connectivity, as demonstrated by the decrease in SP and Junction metrics.

The highest performance results are achieved when all main components are combined, and the lane generation results are shown in Figure 14(e). The results of the comparison with variants reveal the effectiveness of the main components of the proposed method.

#### 4.3.3. Sensitivity analysis

To investigate the impact of parameter choices on the overall performance, we evaluate parameter sensitivity by varying the parameters. The buffer width parameter refers to the width of the buffer window used in the trajectory segmentation step. Figure 15(a) demonstrates that the buffer width is less sensitive within the range of 30-100 meters. When expanding it to 100 meters, there is a slight improvement in recall, indicating



**Figure 16.** Sensitivity analysis of the trajectory downsampling impacts on lane location metrics and efficiency.

better coverage of lane transition regions. The lane clustering width defines the maximum distance in the lane trajectory clustering. The performance results remain stable over a range of appropriate lane widths in Figure 15(b), but excessive lane width could deteriorate performance, as the real-world lane widths are typically no more than 3.75 meters. TPCF sampling interval distance is used in the selected initial trajectory for lane centerline fitting purposes. The sampling distance should not be excessively large to avoid losing lane geometry details and result in a performance decline, as shown in Figure 15(c). TPCF convergence threshold is used to determine the convergence of the TPCF algorithm. It can be observed that the convergence remains stable within an appropriate threshold in Figure 15(d).

The high-precision trajectory data collected by mobile mapping vehicles at a high sampling frequency of 10Hz is very dense, which could result in increased computational costs when dealing with large amounts of data. Through experiments, we evaluate the performance of different trajectory downsampling methods to conduct sensitivity analysis. Figure 16(a) and 16(b) respectively display the performance of lane location metrics and running efficiency as the trajectory sampling time interval uniformly increases. When the sampling time interval increases, the reduction in the number of trajectory sampling points results in a significant increase in algorithm efficiency. The downside, though, is that the details of the original trajectory are lost after excessive downsampling, leading to a decrease in accuracy. To improve the errors that arise in the trajectory downsampling process, we employ the Douglas-Peucker (DP) algorithm (Douglas and Peucker 1973) for trajectory downsampling. This downsampling method preserves the important shape features of a trajectory and reduces errors that may occur during downsampling. The performance and efficiency of the

**Table 4.** Sensitivity analysis of trajectory sample size based on lane location precision, recall, and F1 score. The scenarios with examples can be found as normal road segment (Figure 14), roundabout (Figure 13 second column), and overpass (Figure 13 third column).

Scenario	Metric	Percentage of full sample								
		100%	90%	80%	70%	60%	50%	40%	30%	20%
Normal road	Sample size	44	39	34	30	25	21	17	12	8
	Precision	0.96	0.96	0.96	0.97	0.96	0.96	0.95	0.95	0.95
	Recall	0.91	0.91	0.91	0.90	0.90	0.89	0.87	0.81	0.81
	F1 score	0.94	0.94	0.94	0.94	0.93	0.92	0.91	0.88	0.88
Roundabout	Sample size	114	101	90	79	67	56	45	33	22
	Precision	0.95	0.95	0.95	0.91	0.89	0.91	0.92	0.91	0.90
	Recall	0.91	0.91	0.89	0.81	0.80	0.79	0.77	0.69	0.60
	F1 score	0.92	0.92	0.92	0.86	0.85	0.85	0.84	0.76	0.72
Overpass	Sample size	145	130	116	101	87	72	58	43	29
	Precision	0.92	0.92	0.89	0.91	0.90	0.90	0.90	0.90	0.90
	Recall	0.95	0.95	0.94	0.93	0.85	0.86	0.85	0.82	0.67
	F1 score	0.94	0.94	0.93	0.92	0.93	0.88	0.88	0.86	0.77

proposed method after trajectory downsampling using the DP algorithm are presented in Figure 16(c) and 16(d). Following reasonable downsampling of the trajectory data, the results indicate that the proposed method achieves improved efficiency while maintaining performance. It should be noted that excessive downsampling could still result in the loss of lane details, and the sensitivity of recall is higher than that of precision.

To assess the required sample size of trajectory data for practical application, we randomly reduce the trajectory sample sizes with 10% steps and evaluate the lane location metrics of the extracted lanes. The sensitivity analysis results for trajectory sample size are reported for the normal road segment, roundabout, and overpass, which is presented in Table 4. The results demonstrate that the proposed method is robust to variations in trajectory sample size. With decreasing trajectory sample size, the lane location precision is minimally affected. When the trajectory sample size decreases to a level that is insufficient to fully cover the tested road network, the recall metric decreases. This is reasonable because the missing trajectories in some areas result in a loss of lane information. The generation of lane-level road networks requires sufficient coverage of trajectory data to ensure that all lanes within the test area have complete trajectory coverage. For normal road segments, a minimum of approximately 10 trajectories is needed. For more complex scenarios, such as roundabouts and overpasses, approximately 70-80 trajectory samples are required.

## 5. Conclusions and future work

In this paper, we propose a novel method for lane-level road network generation from high-precision trajectory data with lane-changing behavior analysis. The proposed method partitions the trajectories to reduce processing complexity. Lane transition regions, where lanes merge or diverge, are determined by analyzing the lateral distance distribution of trajectories. Subsequently, trajectory segmentation is carried out. Specialized lane generation strategies are developed for both lane consistent and lane transition regions. For lane consistent regions, a trajectory-based principal curve fitting algorithm is proposed to extract candidate lane centerlines, and erroneous lane centerlines caused by discretionary lane-changing are detected and pruned using a maximum clique search method. For lane transition regions associated with mandatory

lane-changing behavior, a lane-group fitting algorithm is proposed for extracting lane transition structures and constructing lane-level connections. The proposed method addresses the challenges posed by complicated lane-level road network structures and lane-changing behaviors recorded in high-precision trajectory data.

Comprehensive experiments are performed on a real-world high-precision trajectory dataset to demonstrate the superior performance of the proposed method in generating lane-level road networks. Compared with the existing algorithms, our proposed method achieves considerable performance in four evaluation metrics. The results indicate the effectiveness of our proposed method in extracting lane-level road networks under complex scenarios. The overall quality of the generated lane-level road networks is significantly improved with higher completeness and fewer fragments. The proposed method offers an innovative and practical solution for the automatic generation of lane-level road networks, which provide essential auxiliary information for downstream tasks in autonomous driving. The proposed method could also be applied to road network updates by integrating the generated lanes with the existing road networks.

This paper leaves several possible extensions for future research. Due to limitations in data collection, the proposed method has been tested on a limited set of scenarios. In the future, a comprehensive analysis across a wider variety of road scenarios could be conducted. One scenario that could be explored in future research is the detection and generation of tide-lane. Furthermore, additional information in trajectory data could be introduced. For instance, trajectory elevation information could be utilized to identify the trajectory crossing behavior in grand-separated road structures with similar planar layouts but different elevations.

### **Data and codes availability statement**

The data and codes that support the findings of this study are available at the link: <https://doi.org/10.6084/m9.figshare.23529336>. A subset of the data is shared for demonstration purposes.

### **Acknowledgement(s)**

We thank the editor and anonymous reviewers for their constructive comments.

### **Disclosure statement**

No potential conflict of interest was reported by the authors.

### **Funding**

The work was supported by the Chongqing Technology Innovation and Application Development Project [Grant No. CSTB2022TIAD-DEX0013]; funding from the State Key Laboratory of Intelligent Vehicle Safty Technology and Chongqing Changan Automobile Co. Ltd; and the Fundamental Research Funds for the Central Universities [Grant No. 2042022dx0001].

## Notes on contributors

**Mengyue Yuan** is an M.S. student in the School of Remote Sensing and Information Engineering at Wuhan University. Her research interest is geospatial data mining and transportation geography. She contributed to the idea, study design, methodology, implementation, and manuscript writing of this paper.

**Peng Yue** is a professor at Wuhan University. He serves as the deputy dean at the School of Remote Sensing and Information Engineering, the director at the Hubei Province Engineering Center for Intelligent Geoprocessing, and the director at the Institute of Geospatial Information and Location Based Services. He supervised this study and contributed to the study design, methodology, and manuscript writing.

**Can Yang** is a postdoctoral researcher in the School of Remote Sensing and Information Engineering at Wuhan University. His research interest is in trajectory pattern mining and recognition. He contributed to this paper's idea, study design, methodology, and manuscript writing.

**Jian Li** is an engineer at the State Key Laboratory of Intelligent Vehicle Safty Technology and Chongqing Changan Automobile Co. Ltd. He contributed to this paper's idea, study design, and methodology.

**Kai Yan** is an engineer at the State Key Laboratory of Intelligent Vehicle Safty Technology and Chongqing Changan Automobile Co. Ltd. He contributed to this paper's idea, study design, and methodology.

**Chuanwei Cai** is a Ph.D. student in the School of Remote Sensing and Information Engineering at Wuhan University. He contributed to this paper's study design and methodology.

**Chongshan Wan** is an M.S. student in the School of Remote Sensing and Information Engineering at Wuhan University. He contributed to this paper's implementation.

## References

- Ahmed, M., et al., 2014. A comparison and evaluation of map construction algorithms using vehicle tracking data. *GeoInformatica*, 19, 601–632.
- Arman, M.A. and Tampère, C.M.J., 2021. Lane-level routable digital map reconstruction for motorway networks using low-precision GPS data. *Transportation Research Part C: Emerging Technologies*, 129, 103234.
- Arman, M.A. and Tampère, C.M., 2022. Lane-level trajectory reconstruction based on data-fusion. *Transportation research part C: Emerging technologies*, 145, 103906.
- Balal, E., Cheu, R.L., and Sarkodie-Gyan, T., 2016. A binary decision model for discretionary lane changing move based on fuzzy inference system. *Transportation Research Part C: Emerging Technologies*, 67, 47–61.
- Bastani, F., et al., 2018. Roadtracer: Automatic extraction of road networks from aerial images. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4720–4728.
- Biagioni, J. and Eriksson, J., 2012a. Inferring road maps from global positioning system traces: Survey and comparative evaluation. *Transportation Research Record*, 2291 (1), 61–71.
- Biagioni, J. and Eriksson, J., 2012b. Map inference in the face of noise and disparity. In:

- Proceedings of the 20th International Conference on Advances in Geographic Information Systems.* 79–88.
- Bishop, C.M. and Nasrabadi, N.M., 2006. *Pattern recognition and machine learning*. vol. 4. Springer New York, NY.
- Bocklisch, F., et al., 2017. Adaptive fuzzy pattern classification for the online detection of driver lane change intention. *Neurocomputing*, 262, 148–158.
- Boppana, R. and Halldórsson, M.M., 2005. Approximating maximum independent sets by excluding subgraphs. In: *SWAT 90: 2nd Scandinavian Workshop on Algorithm Theory Bergen, Sweden, July 11–14, 1990 Proceedings*. Springer, 13–25.
- Bruntrup, R., et al., 2005. Incremental map generation with gps traces. In: *Proceedings. 2005 IEEE Intelligent Transportation Systems*. 574–579.
- Cao, L. and Krumm, J., 2009. From GPS traces to a routable road map. In: *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. vol. 2291, 3–12.
- Chao, P., et al., 2020. A survey and quantitative study on map inference algorithms from gps trajectories. *IEEE Transactions on Knowledge and Data Engineering*, 34 (1), 15–28.
- Chen, C., et al., 2016. City-scale map creation and updating using GPS collections. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1465–1474.
- Chen, Y. and Krumm, J., 2010. Probabilistic modeling of traffic lanes from GPS traces. In: *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*. 81–88.
- Davies, J.J., Beresford, A.R., and Hopper, A., 2006. Scalable, distributed, real-time map generation. *IEEE Pervasive Computing*, 5, 47–54.
- Deng, M., et al., 2018. Generating urban road intersection models from low-frequency GPS trajectory data. *International Journal of Geographical Information Science*, 32, 2337–2361.
- Douglas, D.H. and Peucker, T.K., 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: the international journal for geographic information and geovisualization*, 10 (2), 112–122.
- Elhashash, M., Albanwan, H., and Qin, R., 2022. A review of mobile mapping systems: From sensors to applications. *Sensors*, 22 (11), 4262.
- Fathi, A. and Krumm, J., 2010. Detecting road intersections from GPS traces. In: *Geographic Information Science: 6th International Conference, GIScience 2010 Proceedings*. 56–69.
- Hagberg, A., Swart, P., and S Chult, D., 2008. *Exploring network structure, dynamics, and function using NetworkX*. Los Alamos National Lab, Los Alamos, NM (United States).
- Hastie, T. and Stuetzle, W., 1989. Principal curves. *Journal of the American Statistical Association*, 84 (406), 502–516.
- He, S., et al., 2018. RoadRunner: improving the precision of road network inference from GPS trajectories. In: *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 3–12.
- Joshi, A. and James, M.R., 2015. Generation of accurate lane-level maps from coarse prior maps and lidar. *IEEE Intelligent Transportation Systems Magazine*, 7 (1), 19–29.
- Karagiorgou, S. and Pfoser, D., 2012. On vehicle tracking data-based road network generation. In: *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*. 89–98.
- Killick, R., Fearnhead, P., and Eckley, I.A., 2012. Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107 (500), 1590–1598.
- Kuntzsch, C., Sester, M., and Brenner, C., 2016. Generative models for road network reconstruction. *International Journal of Geographical Information Science*, 30, 1012–1039.
- Li, H., Kulik, L., and Ramamohanarao, K., 2016. Automatic generation and validation of road maps from GPS trajectory data sets. In: *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. 1523–1532.
- Liu, R., Wang, J., and Zhang, B., 2020. High definition map for automated driving: Overview

- and analysis. *The Journal of Navigation*, 73 (2), 324–341.
- Pan, T., et al., 2016. Modeling the impacts of mandatory and discretionary lane-changing maneuvers. *Transportation research part C: Emerging technologies*, 68, 403–424.
- Ruan, S., et al., 2020. Learning to generate maps from trajectories. In: *Proceedings of the AAAI conference on artificial intelligence*. vol. 34, 890–897.
- Shi, W., Shen, S., and Liu, Y., 2009. Automatic generation of road network map from massive GPS, vehicle trajectories. *2009 12th International IEEE Conference on Intelligent Transportation Systems*, 1–6.
- Stanojevic, R., et al., 2018. Robust road map inference through network alignment of trajectories. In: *Proceedings of the 2018 SIAM International Conference on Data Mining*. 135–143.
- Tang, L., et al., 2016. CLRIC: Collecting lane-based road information via crowdsourcing. *IEEE Transactions on Intelligent Transportation Systems*, 17 (9), 2552–2562.
- Tang, L., et al., 2015. Lane-level road information mining from vehicle GPS trajectories based on naive bayesian classification. *ISPRS International Journal of Geo-Information*, 4, 2660–2680.
- Uduwaragoda, E., Perera, A., and Dias, S., 2013. Generating lane level road data from vehicle trajectories using kernel density estimation. In: *16th International IEEE Conference on Intelligent Transportation Systems*. 384–391.
- Wang, J., et al., 2015. A novel approach for generating routable road maps from vehicle GPS traces. *International Journal of Geographical Information Science*, 29, 69–91.
- Wang, J., et al., 2017. Automatic intersection and traffic rule detection by mining motor-vehicle GPS trajectories. *Computers, Environment and Urban Systems*, 64, 19–29.
- Wei, C., et al., 2022. Fine-grained highway autonomous vehicle lane-changing trajectory prediction based on a heuristic attention-aided encoder-decoder model. *Transportation Research Part C: Emerging Technologies*, 140, 103706.
- Werling, M., et al., 2010. Optimal trajectory generation for dynamic street scenarios in a Frenét Frame. In: *2010 IEEE International Conference on Robotics and Automation*. IEEE, 987–993.
- Xie, D.F., et al., 2019. A data-driven lane-changing model based on deep learning. *Transportation research part C: Emerging technologies*, 106, 41–60.
- Yang, C. and Gidófalvi, G., 2018. Fast map matching, an algorithm integrating hidden markov model with precomputation. *International Journal of Geographical Information Science*, 32 (3), 547–570.
- Yang, D., et al., 2018a. A dynamic lane-changing trajectory planning model for automated vehicles. *Transportation Research Part C: Emerging Technologies*, 95, 228–247.
- Yang, X., et al., 2018b. Generating lane-based intersection maps from crowdsourcing big trace data. *Transportation Research Part C: Emerging Technologies*, 89, 168–187.
- Yang, X., et al., 2018c. Automatic change detection in lane-level road networks using gps trajectories. *International Journal of Geographical Information Science*, 32 (3), 601–621.
- Zhang, T., et al., 2016. A lane-level road network model with global continuity. *Transportation Research Part C: Emerging Technologies*, 71, 32–50.
- Zheng, R., et al., 2017. Topic model-based road network inference from massive trajectories. In: *2017 18th IEEE International Conference on Mobile Data Management*. 246–255.