






Article

Probabilistic Traffic Motion Labeling for Multi-Modal Vehicle Route Prediction

Alberto Flores Fernández ^{1,2,*} , Jonas Wurst ¹ , Eduardo Sánchez Morales ¹ , Michael Botsch ¹ ,
Christian Facchi ¹  and Andrés García Higuera ^{2,3} 

¹ Fakultät Elektro- und Informationstechnik, Technische Hochschule Ingolstadt, Esplanade 10, 85049 Ingolstadt, Germany; jonas.wurst@thi.de (J.W.); eduardo.sanchezmorales@thi.de (E.S.M.); michael.botsch@thi.de (M.B.); christian.facchi@thi.de (C.F.)

² Escuela Técnica Superior de Ingeniería Industrial, Universidad de Castilla-La Mancha, Calle Altagracia 50, 13001 Ciudad Real, Spain; andres.garcia@uclm.es or andres.garcia@europarl.europa.eu

³ European Parliamentary Research Service, Rue Wiertz 60, B-1047 Brussels, Belgium

* Correspondence: alberto.floresfernandez@thi.de or alberto.flores@alu.uclm.es

Abstract: The prediction of the motion of traffic participants is a crucial aspect for the research and development of Automated Driving Systems (ADSs). Recent approaches are based on multi-modal motion prediction, which requires the assignment of a probability score to each of the multiple predicted motion hypotheses. However, there is a lack of ground truth for this probability score in the existing datasets. This implies that current Machine Learning (ML) models evaluate the multiple predictions by comparing them with the single real trajectory labeled in the dataset. In this work, a novel data-based method named Probabilistic Traffic Motion Labeling (PROMOTING) is introduced in order to (a) generate probable future routes and (b) estimate their probabilities. PROMOTING is presented with the focus on urban intersections. The generation of probable future routes is (a) based on a real traffic dataset and consists of two steps: first, a clustering of intersections with similar road topology, and second, a clustering of similar routes that are driven in each cluster from the first step. The estimation of the route probabilities is (b) based on a frequentist approach that considers how traffic participants will move in the future given their motion history. PROMOTING is evaluated with the publicly available Lyft database. The results show that PROMOTING is an appropriate approach to estimate the probabilities of the future motion of traffic participants in urban intersections. In this regard, PROMOTING can be used as a labeling approach for the generation of a labeled dataset that provides a probability score for probable future routes. Such a labeled dataset currently does not exist and would be highly valuable for ML approaches with the task of multi-modal motion prediction. The code is made open source.

Keywords: PROMOTING; automated driving systems; autonomous vehicles; multi-modal; motion prediction; route prediction; machine learning; real traffic data



Citation: Flores Fernández, A.; Wurst, J.; Sánchez Morales, E.; Botsch, M.; Facchi, C.; García Higuera, A. Probabilistic Traffic Motion Labeling for Multi-Modal Vehicle Route Prediction. *Sensors* **2022**, *22*, 4498. <https://doi.org/10.3390/s22124498>

Academic Editor: Arturo de la Escalera Hueso

Received: 20 May 2022

Accepted: 11 June 2022

Published: 14 June 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Urban mobility and transportation are cornerstones of society. Due to the high socio-economic impact of road accidents, there is a motivation to continuously make improvements with regard to automotive safety. This motivation has derived from the development of the modern road infrastructure, which has brought major advances in terms of road safety and traffic-flow efficiency. Recent European Union (EU) road safety statistics [1] show, however, that these improvements stagnated in 2019. Specifically, they quantify a decrease in fatal accidents of 23% when compared to 2010 and of 2% when compared to 2018. For this reason, the EU has launched an ambitious initiative called “Vision Zero” [2], in which it establishes the goal of reducing fatalities caused by traffic accidents to near zero by 2050 and sets the target of halving the number of severe accidents by 2030. To this end, the EU initiative highlights the role that vehicle automation and connectivity play in increasing

safety. Given that the majority of accidents (94%) are caused by human error [3], the ADSs under development are mainly focused on improving safety by assisting drivers with the early recognition and avoidance of dangerous situations, while also considering other aspects such as emissions reduction, driving efficiency, and improved passenger comfort. The deployment of automated driving functions in traffic scenarios in open environments is being carried out progressively. The Society of Automotive Engineers defines six levels of automation from levels 0 to 5 [4], where level 5 corresponds to full and unsupervised autonomy. A level 5 automated vehicle demands a very high technological complexity, and, to date, the driving functions required for this level of automation do not have the necessary robustness for deployment in traffic scenarios in open environments. According to [5], the main aspects and systems related to ADSs can be summarized using ten categories: (1) connected systems, (2) end-to-end driving, (3) localisation, (4) perception, (5) assessment and motion prediction, (6) planning, (7) control and dynamic, (8) human machine interface, (9) dataset and software, and (10) implementation. In this work, multidisciplinary research is performed, covering mainly aspects from categories 5 and 9.

A recent line of research [6–12] focuses on multi-modal motion prediction. This is based on the consideration that traffic motion is multi-modal in nature, meaning that each traffic participant is not bound to follow a single trajectory in the future, but it can instead choose from a wide variety of possible trajectories. In this way, not just one, but multiple probable motion hypotheses are predicted for each traffic participant, allowing researchers to capture the different options a driver may take, such as turning left, making a U-turn or continuing straight ahead, among others. In the following, the term mode refers to a specific estimation of future motion within a finite set of possibilities, and the likelihood that a given mode will be selected is denoted as mode score or mode probability. One prominent approach to address multi-modal motion predictions makes use of ML methods based on the supervised learning paradigm. For this, a labeled dataset is necessary, i.e., the label associated with each sample is known. In case the dataset is generated from real traffic data, only a single real trajectory per traffic participant can be labeled, namely the one that has been driven. This shows the challenge of (a) predicting multiple motion hypothesis for each traffic participant, out of a single labeled one. In addition, the prediction of multiple motion hypotheses implies the assignment of a probability score to each one with respect to the total number of hypotheses. However, labeled datasets with probabilities for routes are not available, (b) resulting in a lack of ground truth for this probability scores.

These aspects ((a) and (b)) motivate the investigation of a method that addresses the following research questions: (1) how to extract the route (certain sections of the road) that represents each possible mode from real traffic datasets, (2) how to estimate the probability that a vehicle will drive a certain mode, and (3) how to generate an adequate multi-modal labeled dataset so that a ML model can learn from it the intrinsic multi-modal motion of traffic scenarios.

In this regard, this work introduces a novel data-based method named PROMOTING that allows the estimation of multiple routes for each traffic participant and provides a probability score for each of the possible future routes. In this way, PROMOTING can be used as a labeling approach for the generation of a labeled dataset that contains not only single trajectories as its ground truth, but also the multiple estimated routes. Given the fact that the early introduction of smart intersections will be of mixed traffic, i.e., automated and non-automated driving together, the modeling of the traffic flow at such scenarios will be significant. The smart intersection is a concept aimed at improving the safety and traffic flow of intersections. It is based on the use of sensors and communication systems that allow researchers to capture and analyze traffic to support ADSs functions. Thus, PROMOTING focuses on urban traffic scenarios, paying special attention to urban intersections.

Therefore, this work makes a contribution to the improvement of multi-modal motion predictions by introducing the PROMOTING method, highlighting the following. First, the method is able to extract multiple motion hypotheses for each traffic participant. Second, the method is able to estimate the probability that a vehicle will drive following a specific

motion hypothesis. Third, the method may be used for the generation of a labeled dataset that provides extra information that is useful for a multi-modal prediction task. Fourth, the method is evaluated using real-world traffic scenarios from a database, which allows us to obtain a realistic representation of the traffic's behavior in urban traffic scenarios.

The rest of the paper is structured as follows: in Section 2, related works are presented. In Section 3, the methodology of PROMOTING is detailed. In Section 4, the evaluation of PROMOTING is presented, and the associated results are shown and described. In Section 5, the main findings of the work are discussed. The paper is summarized in Section 6.

2. Related Works

According to [6], the motion prediction of traffic participants can be grouped into the following categories:

- (1) an engineering approach or physics-based methods,
- (2) planning-based methods, and
- (3) pattern-based methods.

Over the last few years, the research into motion prediction has shifted its focus from the physics-based generation of trajectories to the use of ML methods for the same purpose. The authors of [13] proposed the Attention mechanism that marked a shift in the way typical neuro-linguistic programming, time-series forecasting, and sequence-to-sequence problems are approached. Along with this Attention mechanism, Transformer Networks are also finding their way into motion prediction tasks. In [6], Multiple Attention Heads (MAH) are implemented together with a Long–Short Term Memory Encoder–Decoder architecture to predict multiple trajectories, thus addressing the multi-modality of the motion of traffic participants and considering cross-agent interaction modeling. A similar approach is taken in [14]. The difference between [6] and [14] is that the latter adds map-related information that is learned by the Attention mechanism, which assists in modeling the agent–map interaction and improves the system performance. In [15], an architecture based on an Encoder–Decoder structure is proposed, where both are based exclusively on MAH. This model achieves a better performance than the one proposed in [14]. Other recent approaches [9,16,17] build on the work of [13] and use Transformer Networks based on MAH. In [16], pedestrian trajectory prediction is investigated, where the behavior of the pedestrians is modeled without taking into account any kind of interaction with neither traffic participants nor with the map information. This approach is able to closely predict the motion of pedestrians, highlighting the suitability of using Transformer Networks for motion planning tasks. A similar method is presented in [17], where the orientation of the traffic participants is considered to be an additional feature to the input vector when compared to [16]. Furthermore, whereas in [16] only pedestrians are considered, in [17] the performance of the ML model is evaluated for different types of traffic scenarios and different types of road users. A more complex ML-architecture than [16,17] is used in [9], consisting on three stacked Transformer Networks: vehicle motion, vehicle–map interaction, and vehicle–vehicle interaction. The networks are trained sequentially for each epoch, where the vehicle–vehicle interaction network receives the output of the vehicle–map interaction network, and the vehicle–map interaction network receives the output of the vehicle motion network. In addition to receiving the output of the previous one, each network receives additional inputs, which allows each network to specialize in a particular task.

In order for a ML model to learn something as complex as urban traffic, a large amount of data captured from real-world driving scenes is necessary. To prevent over-fitting, the data should have a large variability; in this way, the ML model is able to capture as many as possible of the variations of relevant features.

In the case of urban intersections, for example, the behavior of the traffic participants varies depending on the time of day, working/non-working days, and construction sites, among others. All these situations influence the behavior of the traffic participants, and their consideration provides extra knowledge that must be taken into account by ADSs. On the

other hand, capturing real traffic data with these characteristics is a major challenge because of the required financial, computational, and time resources. One strategy to overcome this is to constrain the research and development of ADSs to bounded driving environments, such as smart urban corridors [18]. To this end, it is relevant to use appropriate databases for the training of the ML models.

Current research works [6–12] that focus on multi-modal motion prediction evaluate their performance either in terms of the Average Displacement Error (ADE), the Final Displacement Error (FDE), or the Root Mean Square Error (RMSE). That is, they consider a single labeled real trajectory and measure the Euclidean distances between the reference trajectory and each of the predicted ones. The best trajectory is then chosen based on one of the minimum ADE, the minimum FDE, or the minimum RMSE. The main problem with using these metrics both to reduce training losses and to evaluate the model during the inference phase is that it forces ML models to generate trajectories close to the reference trajectory. This may result in a subset of the predicted trajectories not being drivable, not following the road infrastructure, or colliding with other traffic participants. Furthermore, the prediction of multiple motion scenarios for each traffic participant entails assigning a probability score that indicates the likelihood of selecting a hypothesis within the set of multiple hypotheses; however, the existent datasets containing real traffic data as [19–25] do not provide this score, as there is only a single real trajectory labeled by each traffic participant.

In [26], the graphs of road topologies are used to identify similar examples through their isomorphism. This is required to shape the latent space for proper novelty detection. Moreover, in [27], the isomorphisms are used to identify similar traffic scenarios, also including the trajectories as paths inside the graphs. As before, this is used for shaping a latent space. However, in the present work, isomorphisms are used to identify similar intersections and routes in the intersections in order to identify similar modes.

Relevant work on the representation of motion hypotheses in traffic scenarios is presented in [28], with the introduction of the Predicted-Occupancy Grids (POGs). These represent the future traffic scenarios in the form of grid cells, where the confidence about the motion of dynamic agents is represented. This approach considers a spectrum of expected occupancy values beyond the simplistic binary approach, i.e., occupied or not occupied. This type of representation is used for the prediction of complex traffic scenarios in [29,30], where different types of machine learning based architectures for POGs estimation are presented. However, there are three notable differences between the work of [28] and the present work. In [28], the approach is based on expert knowledge (assumes physical models of vehicles and motion hypothesis), makes use of simulation data, and the method outputs POGs. In contrast, in the present work, a methodology based on a frequentist approach is proposed (recorded traffic data is analyzed without making a motion hypothesis), real-world traffic data is used, and the presented method (PROMOTING) outputs the modes, in the form of routes, and the mode probabilities.

With regard to all the above, the present research work addresses the shortcomings of multi-modal motion prediction research by proposing the novel PROMOTING method. This serves as the methodology for the generation of a labeled dataset that extracts information about the modes of traffic participants based on conditional prior information. The method is able to extract the number and route of the modes, as well as to estimate the probability that a traffic participant will drive a specific mode. To the best of the authors' knowledge, this is the first work seeking to estimate the modes with their probabilities in a probabilistic way from real-world data for the purpose of the labeling of multi-modal motion hypothesis.

3. Materials and Methods

In order to estimate the modes and the probabilities of each mode, PROMOTING requires (1) historical traffic data and (2) topological information of the road map. To cover these requirements, the publicly available Lyft database [25] is selected, so PROMOTING is evaluated in this work by making use of this database. This database contains traffic

motion information that is captured by a vehicle equipped with exteroceptive sensors. It contains a large amount of real-world trajectory data of dynamic participants, including urban intersections, and detailed map information covering the urban area where the traffic scenes were recorded. The methodology of PROMOTING is composed of five steps (see Figure 1), and each step is explained in a subsection of this section.

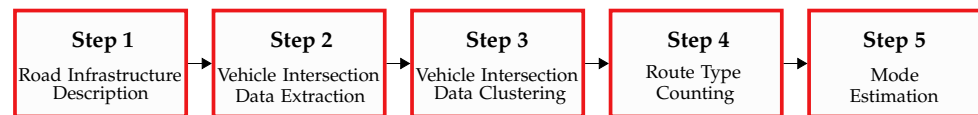


Figure 1. Sequence of operation of PROMOTING method.

3.1. Road Infrastructure Description

The first step of the PROMOTING method, see Figure 2, aims to describe static traffic information: the road infrastructure. This is described by the road map information contained in the Lyft database on the basis of the map description (see Section 3.1.1) and the intersection description (see Section 3.1.2).

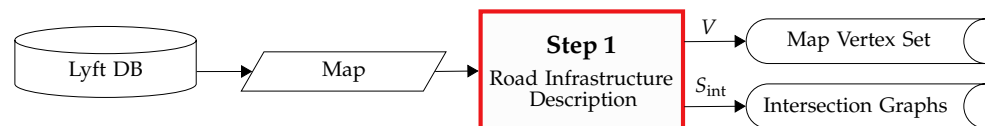


Figure 2. Overview of step 1 of the PROMOTING method: the road infrastructure description.

A visual representation of the road infrastructure generated from information from the Lyft database is shown in Figure 3.

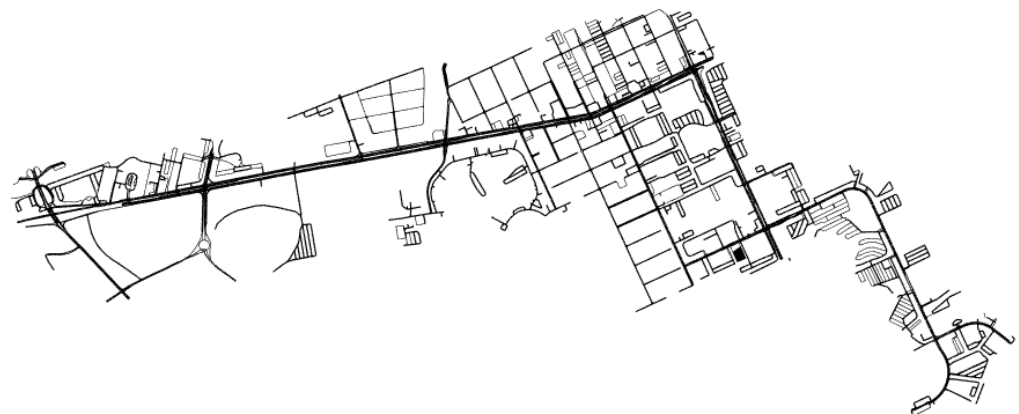


Figure 3. A visual representation of the road infrastructure generated from the Lyft database [25].

3.1.1. Map Description

The road map information contained in the Lyft database divides the road space into so called *ways*, which are road sections of finite length representing an individual lane in a given direction. In this work, each *way* is referred to as a *vertex*. Thus, the set of vertices v_i of the map V is defined as

$$V = \{v_1, v_2, \dots, v_i, \dots, v_{n_v}\}, \quad (1)$$

where n_v indicates the order of G , i.e., the number of vertices contained in the map. Each vertex $v_i \in V$ is characterized by a number of features that allow its geometric and connectivity definition, for example:

- **centreLine:** (x, y) coordinates in global coordinate frame of each vertex.

- **turnDirection:** indicates the type of change of direction of the vertex: “1” for straight, “2” for left turns, and “3” for right turns.
- **intersectionId:** unique identifier $\kappa_i \in \mathbb{N}$ of the intersection of which the vertex forms a part. Value “−1” if the vertex is not part of an intersection.
- **predecessors:** set $A_{i,pre}$ that contains the immediate previous vertices with respect to the driving direction, such that $A_{i,pre} \subseteq V$.
- **successors:** set $A_{i,suc}$ that contains the immediate following vertices with respect to the driving direction, such that $A_{i,suc} \subseteq V$.
- **leftNeighbours:** set $A_{i,left}$ that contains the immediate to the left vertices with respect to the driving direction, such that $A_{i,left} \subseteq V$.
- **rightNeighbours:** set $A_{i,right}$ that contains the immediate to the right vertices with respect to the driving direction, such that $A_{i,right} \subseteq V$.

Thus, the set $A_i \subseteq V$ that contains the adjacent vertices of the i th vertex is defined as a union of sets, so that

$$A_i := A_{i,pre} \cup A_{i,suc} \cup A_{i,left} \cup A_{i,right}. \quad (2)$$

The connection between the different vertices $v_i \in V$ provides valuable information for the vehicle motion prediction. In this paper, the connectivity information of the vertices is used to derive a graph-based model that represents the topology of the urban road network. The map topology G is then defined as a directed graph, so that

$$G = (V, E), \quad (3)$$

where E denotes the set of edges ϵ_k of the map, with

$$E = \{\epsilon_1, \epsilon_2, \dots, \epsilon_k, \dots, \epsilon_{n_e}\}, \quad (4)$$

where n_e indicates the size of G , i.e., the number of edges contained in the graph.

Each edge ϵ_k represents the connection between two adjacent vertices, so that

$$\epsilon_k = \{(v_i, v_j) \mid v_i, v_j \in V \wedge v_j \in A_i\}, i \neq j, \forall i, j \in \{1, 2, \dots, n_v\}. \quad (5)$$

The order of the vertex pair indicates the driving direction on the edge, where the first element is the “source vertex”, and the second one is the “target vertex”. For example, $\epsilon_k = (v_i, v_j) \in E$ indicates that the driving direction on the k th edge is from the i th vertex to the j th vertex.

3.1.2. Intersection Description

Similarly, the road topology of an intersection contained in the map, denoted as the i th intersection, is modeled as the graph $G_i = (V_i, E_i)$ with information from G so that $G_i \subseteq G$. That is, G_i is a sub-graph of G . Therefore, $V_i \subseteq V$ and $E_i \subseteq E$.

To model the graph of each intersection, it is necessary to identify which vertices belong to the same intersection and how are they connected to each other. In this sense, three types of vertex are differentiated for each intersection:

1. **Incoming vertex:** The vertex at the entrance of an intersection. These vertices are grouped in sets with the sub-index “in”.
2. **Crossing vertex:** The vertex on an intersection. These vertices are grouped in sets with the sub-index “x”.
3. **Outgoing vertex:** The vertex at the exit of an intersection. These vertices are grouped in sets with the sub-index “out”.

Thus, incoming vertices precede crossing vertices, and crossing vertices precede outgoing vertices. With this, the graph G_i of the i th intersection is generated as described in Algorithm 1 and a graphic depiction is shown in Figure 4.

Algorithm 1: Intersection graph generation**Input** : directed graph G of the map and the unique intersection identifier ι .**Output**: directed graph G_ι of the ι th intersection formed by the edges set E_ι and the vertex set V_ι with the incoming, crossing and outgoing vertices of the intersection.

- 1 $V_{\iota,x} := \{v_j \in V \mid \kappa_j = \iota\}$
- 2 $V_{\iota,in} := \{v_i \in V \mid (v_i, v_j) \in E \ \forall v_j \in V_{\iota,x}\}$
- 3 $V_{\iota,out} := \{v_k \in V \mid (v_j, v_k) \in E \ \forall v_j \in V_{\iota,x}\}$
- 4 $V_\iota = V_{\iota,in} \cup V_{\iota,x} \cup V_{\iota,out}$
- 5 $E_\iota := \{(v_i, v_j) \in E \mid v_i, v_j \in V_\iota \ \forall i \neq j\}$
- 6 $G_\iota = (V_\iota, E_\iota)$

Algorithm 1 can be used for as many intersections as required to generate the set of intersection graphs S_{int} , so that

$$S_{\text{int}} = \{G_1, G_2, \dots, G_{n_{\text{int}}}\}, \quad (6)$$

where n_{int} indicates the number of intersection graphs generated from the map.

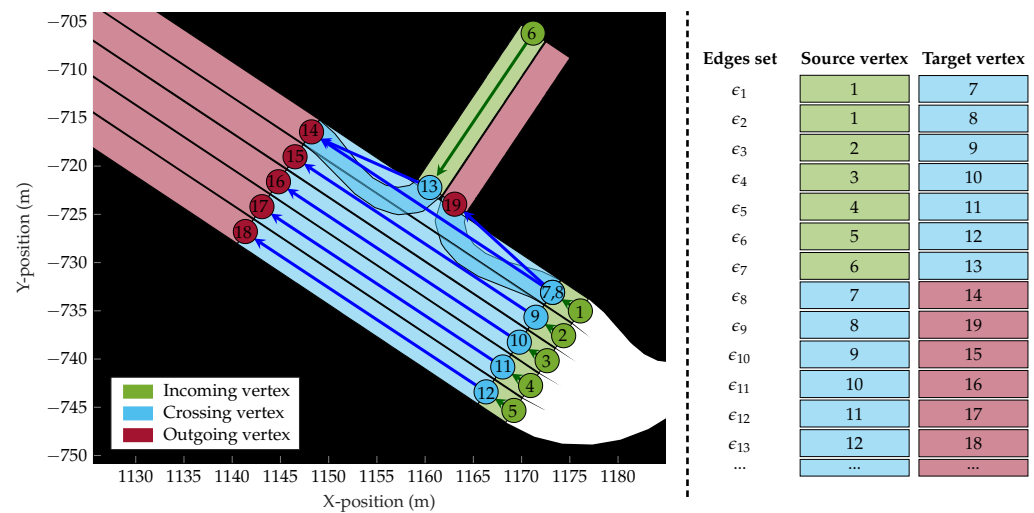


Figure 4. The graphical representation of an intersection. On the (left), the vertices are coloured polygons, where the circles represent the first point of the centreLine feature, and the arrows connecting the circles represent the edges. On the (right) are the edge matrix with the edge list and the corresponding vertices. In this case, $V_{\iota,in} = \{1, 2, 3, 4, 5, 6\}$, $V_{\iota,x} = \{7, 8, 9, 10, 11, 12, 13\}$, and $V_{\iota,out} = \{14, 15, 16, 17, 18, 19\}$.

3.2. Vehicle Intersection Data Extraction

Once the intersection graphs S_{int} and the map vertex set V are generated, the next step is the extraction of the list of the Vehicle Intersection Data (VID) X_{VID} . That is, the route information (sequence of vertices) of each vehicle that crosses an intersection and the graph of the crossed intersection. To accomplish this, the motion history of the vehicle is required, in addition to the intersection graphs and the vertex set obtained in the previous step, see Figure 5.

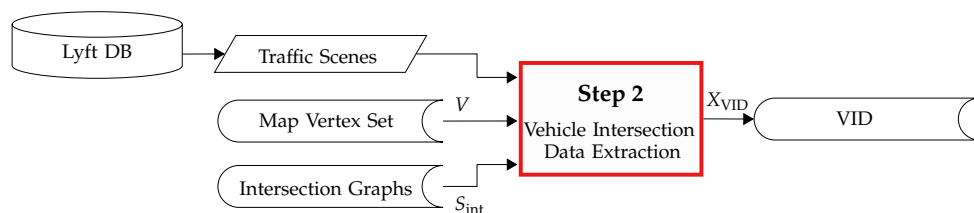


Figure 5. Overview of step 2 of the PROMOTING method: the route extraction.

A detailed description of the VID extraction process is depicted in Figure 6.

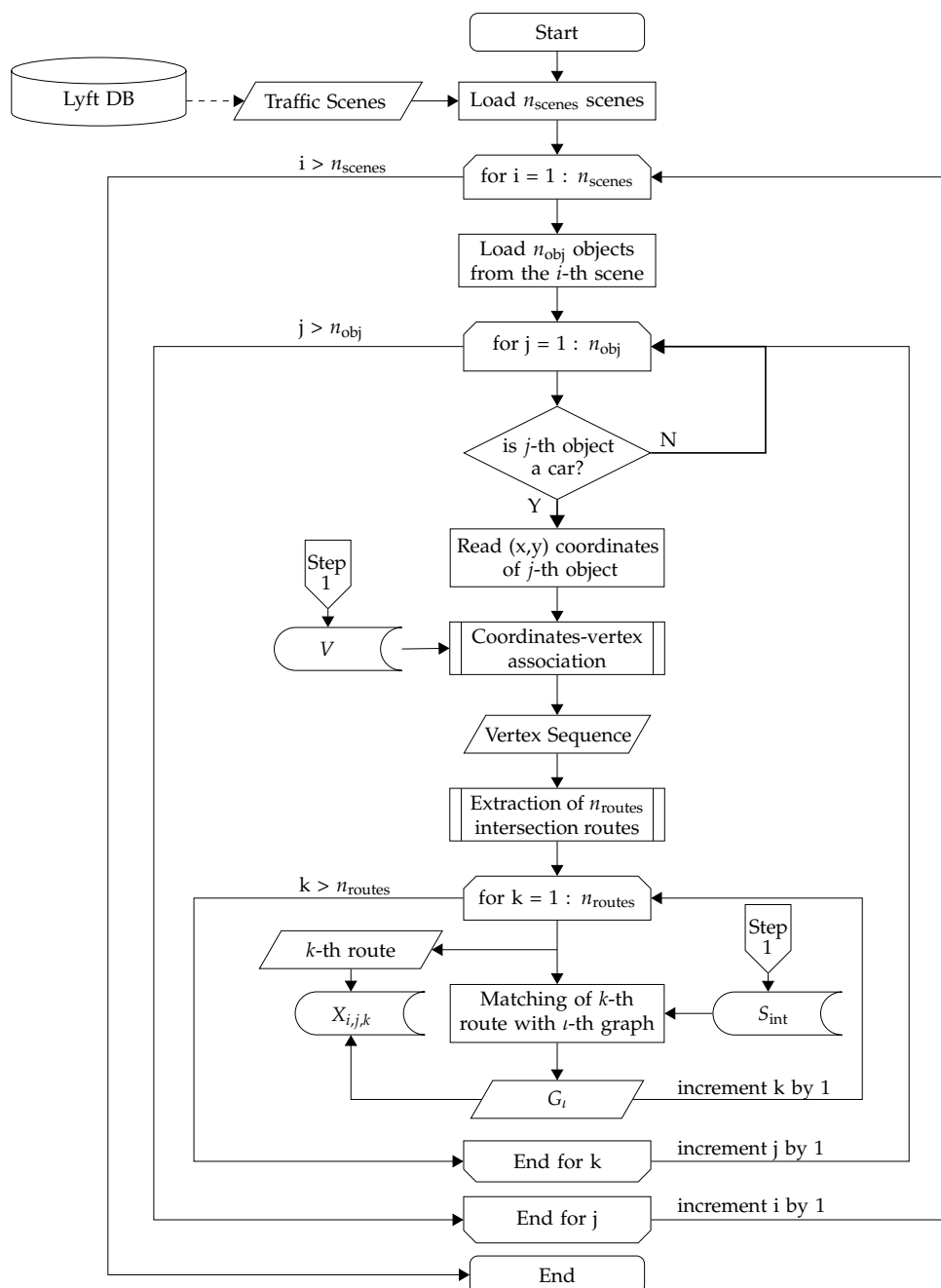


Figure 6. The process to extract the VID.

As depicted in Figure 6, the VID extraction starts by iterating over all n_{scenes} traffic scenes contained in the Lyft database. Each i th scene contains a record of the motion of

all n_{obj} registered objects. For each i th scene, the motion information of each j th object with “car” label is extracted. Next, for each j th vehicle, its (x,y) coordinates are read, and, together with V , the coordinates are associated with vertices so as to generate the vertex sequence $Q_{i,j}$, as detailed in Section 3.2.1. Later, $Q_{i,j}$ is used to extract n_{routes} routes that cross intersections, as detailed in Section 3.2.2. Then, for each R_k route that crosses an intersection with graph G_i , a new VID, denoted by $X_{i,j,k}$, is generated. Hence, the VID, $X_{i,j,k}$, for the k th route of j th vehicle in the i th scene is determined as

$$X_{i,j,k} = (R_k, G_i), \quad (7)$$

where the route R_k is represented by a sequence of vertices and is denoted as follows

$$R_k = [r_1, r_2, \dots, r_q, \dots], \quad r_q \in V_i \text{ and} \quad (8)$$

the intersection graph G_i is generated as indicated by Algorithm 1.

Thus, the VID list X_{VID} is defined as the list whose elements are the extracted $X_{i,j,k}$ and is denoted as follows

$$X_{VID} = (X_1, X_2, \dots, X_{n_{VID}}). \quad (9)$$

3.2.1. Coordinate–Vertex Association

The first step to extract the R_k route is to obtain the vertex sequence $Q_{i,j}$. For this, the (x,y) coordinates of the j th vehicle in the i th scene at each time instance are associated with vertices contained in V . This results in the vertex sequence $Q_{i,j}$, which represents the vertices that the vehicle has driven on. One should note that the association is not unique, meaning that a set of (x,y) coordinates may be associated with multiple vertices, and a vertex may be associated with multiple sets of (x,y) coordinates, which results in a multiple vertex associations. This occurs frequently when the (x,y) coordinates are located at intersections where different crossing vertices overlap. This means that $Q_{i,j}$ must be processed.

First, the invalid (empty) vertex associations are removed from the sequence. An invalid association can happen, for example, when the vehicle moves on “non-drivable” sections of the map. Second, duplicated vertex associations are unified. A duplicated association occurs when a vertex appears in $Q_{i,j}$ in two or more consecutive time instances. By unifying the duplicated vertex associations, only unique ones remain. Finally, $Q_{i,j}$ is filtered according to the intersection topology. This handles multiple vertex associations that can occur when various vertices overlap, see vertices 7, 8, and 9 in Figure 7. Filtering according to the intersection means that only the vertices included and connected in the intersection are kept.

3.2.2. Extraction of Intersection Routes

Once the vertex sequence $Q_{i,j}$ of the j th vehicle in the i th scene is extracted and processed, the next step is to extract the routes that cross intersections. It is possible for a single vehicle to contain more than one intersection route. An intersection route should fulfill the following two characteristics:

1. The route must contain at least one crossing vertex.
2. The route must contain either at least one incoming vertex or at least one outgoing vertex.

This approach allows us to differentiate four categories of intersection routes:

1. **Complete:** The route contains a full description of how the vehicle approaches, crosses, and leaves the intersection. The route starts with incoming vertices, follows crossing vertices, and ends with outgoing vertices. An example of a complete route is the vertex sequence [2, 9, 15], see Figure 4.

2. **Entering:** The route contains a description of how the vehicle approaches and crosses the intersection. The route starts with incoming vertices and ends with crossing vertices. An example of an entering is the vertex sequence [2, 9], see Figure 4.
3. **Leaving:** The route contains a description of how the vehicle crosses and leaves the intersection. The route starts with crossing vertices and ends with outgoing vertices. An example of a leaving route is the vertex sequence [9, 15], see Figure 4.
4. **Other:** Routes that do not belong to any of these three categories. One such route would be that of a vehicle that is standing still during the complete scene, thus remaining at a single vertex. These are omitted, as they do not provide information on how the vehicle approaches or leaves the intersection.

An example of this process is shown in Figure 7. There, the vertex sequence $Q_{i,j}$ of the j th vehicle in the i th scene is given by

$$Q_{i,j} = [4, 11, 17, 22, 25, 30]. \quad (10)$$

From $Q_{i,j}$, 11 and 22 are crossing vertices. Intersection IDs κ are taken from these vertices. So, let $\kappa_{11} = 8$ indicate that vertex 11 belongs to the 8th intersection and $\kappa_{22} = 9$ indicate that vertex 22 belongs to the 9th intersection. Then, the rest of the vertices of the $Q_{i,j}$ vertex sequence that belong to these given intersections are extracted. In this example, the 30th vertex is neglected, as it does not belong to any intersection of this vertex sequence. The remaining vertices are split in as many routes as unique intersection IDs. In this example, two routes are created: one for the 8th intersection and one for the 9th intersection. The elements of the vertex sequence are assigned to a route according to the intersection they belong to. In this example, as the 17th vertex belongs to both the 8th and the 9th intersection, it is assigned to both routes. Next, the type of each vertex of each route is assigned according to the intersection topology. This is the reason why the 17th vertex is assigned as being “outgoing” for the route that corresponds to the 8th intersection and “incoming” for the route that corresponds to the 9th intersection.

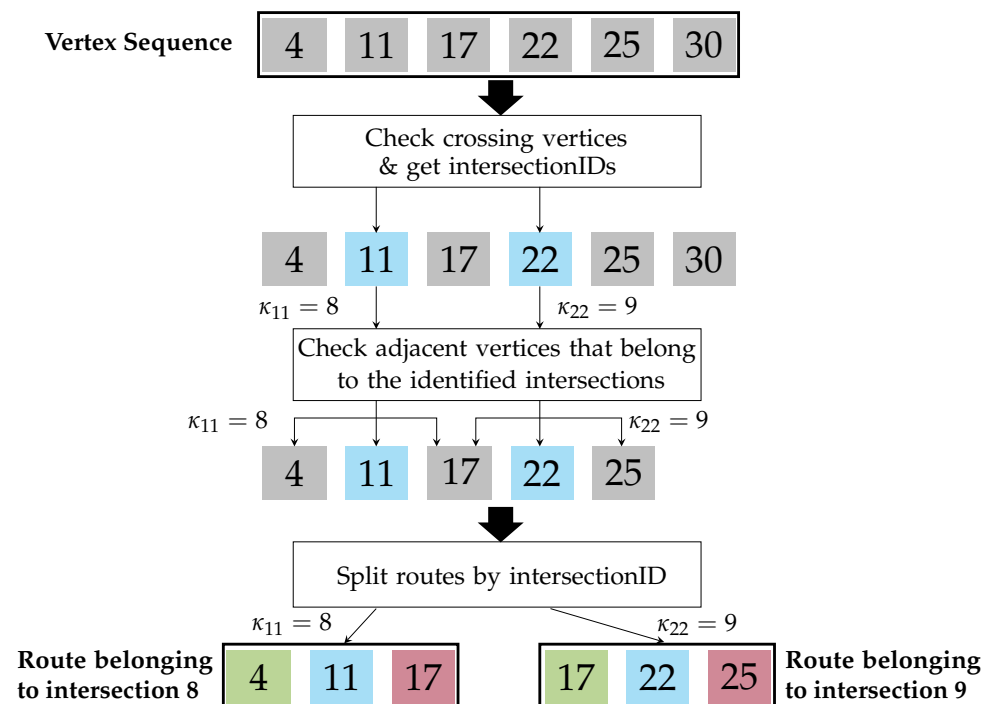


Figure 7. Graphical depiction of the method used to extract intersection routes. Crossing vertices are marked in blue, incoming vertices in green, and outgoing vertices in red.

3.3. Vehicle Intersection Data Clustering

Once the set of intersection graphs S_{int} and the map vertex set V are generated (per Section 3.1) and the VID list X_{VID} is obtained (as per Section 3.2), the next step is the VID clustering. This step aims to cluster the elements of the list of VIDs X_{VID} with respect to their graphs. Specifically, the graph isomorphism represents the similarity criterion. Then, the output of this step is n_{clusters} , where each cluster c is denoted by $X_{c,\text{VID}}$. A graphical depiction of this process is shown in Figure 8. This process consists of two steps: a pre-clustering of graphs (see Section 3.3.1) and an isomorphic clustering (see Section 3.3.2). These steps are detailed in what follows.



Figure 8. Overview of step 3 of the PROMOTING method: the VID clustering.

The list X_{VID} contains n_{VID} routes of vehicles crossing the intersections; these are classified as they are defined in Section 3.2.2. It should be noted that only complete routes have been selected, because they are the only type of routes that contain a full description of the intersection crossing from the entrance to the exit.

3.3.1. Pre-Clustering

The process of clustering based on isomorphism is computationally expensive. This is specially relevant for large databases, where graph-wise and vertex-wise associations are verified. A brute-force search for the $n_v!$ possible bijective functions that satisfy the definition of isomorphism between all extracted graphs is not practical.

For this reason, pre-clustering the graphs prior to the isomorphic clustering (Section 3.3.2) is proposed. This is performed by examining a series of preconditions that two graphs must possess in order to be isomorphic. The preconditions are evaluated in a hierarchical manner, allowing us to structure the database in the form of a tree. This database tree allows further analysis of the distribution of the data in terms of graph properties. Then, the first four hierarchical levels of the database tree are detailed in what follows. Alongside this, an example slice of such a database tree is shown in Figure 9.

- **Level 0:** The root node of the database tree is located at this level and is the highest hierarchical level from which all branches emerge. All VIDs are inside the root node.
- **Level 1:** The graphs are grouped by their order, i.e., the number of vertices contained in the graph. Hence, only VIDs with the same graph order are part of the same node. In Figure 9, A and B are two example nodes at that level, with graph orders 20 and 21, respectively.
- **Level 2:** The graphs are grouped by their size, i.e., the number of edges contained in the graph. VIDs with the same graph orders and sizes are part of the same node. In Figure 9, the node C group VIDs with graph order equal to 20 and graph size equal to 24.
- **Level 3:** The graphs are grouped by their matrix degree:

$$\Theta_{\text{seq}} = \begin{bmatrix} n_{0,\text{in}} & n_{1,\text{in}} & n_{2,\text{in}} & \cdots & n_{n_e,\text{in}} \\ n_{0,\text{out}} & n_{1,\text{out}} & n_{2,\text{out}} & \cdots & n_{n_e,\text{out}} \end{bmatrix}, \quad (11)$$

where the first row refers to the in-degree of the graph, and the second row refers to the out-degree of the graph, i.e., the number of incoming and outgoing edges to/from

the vertices, respectively. With this, $n_{2,in}$ indicates the number of vertices in the graph whose in-degree is equal to 2, and $n_{2,out}$ indicates the number of vertices in the graph whose out-degree is equal to 2. Therefore, at this level, only VIDs with the same graph order, the same graph size, and the same matrix degree are grouped. In Figure 9, the node E group VIDs with graph orders equal to 20, graph sizes equal to 24, and matrix degree Θ_1 .

The levels 0–3 describe the pre-clustering, which creates smaller groups according to their graph properties, such that computationally expensive isomorphism needs to be examined only with the nodes of level 3.

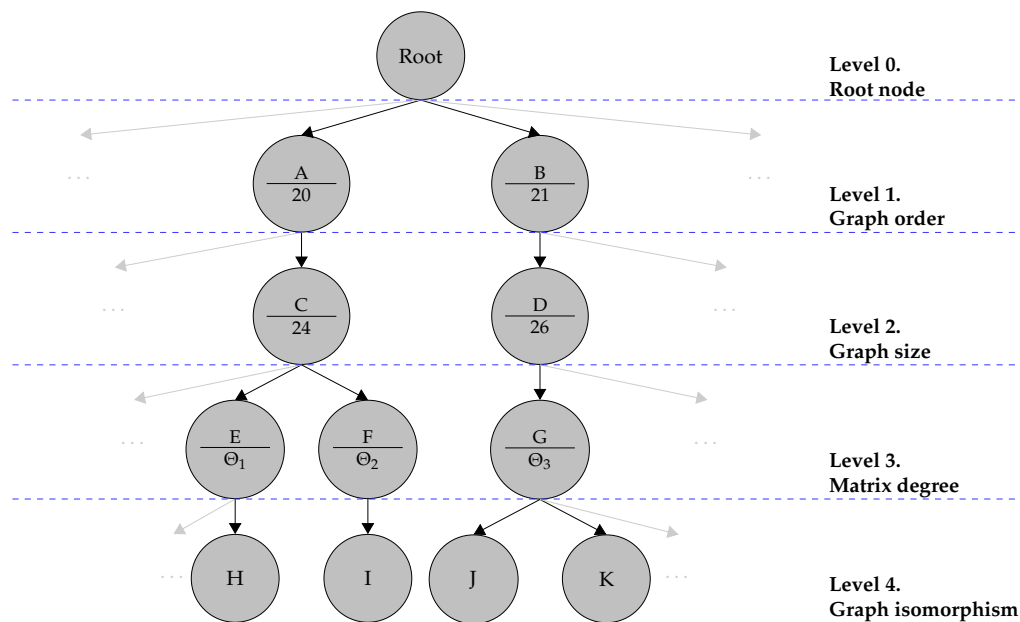


Figure 9. Slice of a database tree for the clustering of the VID based on isomorphic graphs. Each leaf node at level 4 groups the routes of vehicles crossing intersections whose graphs are isomorphic.

3.3.2. Isomorphic Clustering

Given the database tree from the pre-clustering, the aim is to identify VIDs with similar graphs. Only level 3 need to be taken into consideration, since isomorphism between the graphs is only possible within nodes of level 3.

Two graphs G_1 and G_2 are said to be isomorphic if

$$G_1 \cong G_2, \tag{12}$$

where Equation (12) holds true if a bijective function $f : V_{G_1} \rightarrow V_{G_2}$ exists, such that

$$\forall v_i, v_j \in V_{G_1}; (v_i, v_j) \in E_{G_1} \Leftrightarrow (f(v_i), f(v_j)) \in E_{G_2}. \tag{13}$$

This means that every vertex and edge of G_1 has a unique mapping to a vertex and edge of G_2 . All isomorphic graphs are then clustered in level 4 nodes. Nodes H, I, J, and K of Figure 9 are level 4 nodes.

3.4. Route-Type Counting

Once the set of intersection graphs S_{int} and the map vertex set V are generated (as per Section 3.1), the VID list X_{VID} is obtained (see Section 3.2), and the VIDs are clustered (see Section 3.3), the next step is the counting of route types. For this, each c th cluster of X_{VID} is analyzed in order to extract (1) the set of route types \hat{R}_c and (2) the counting list $\rho_{\hat{R}_c}$, whose elements indicate how often each route type appears in the cluster. A graphical depiction of this process is shown in Figure 10.

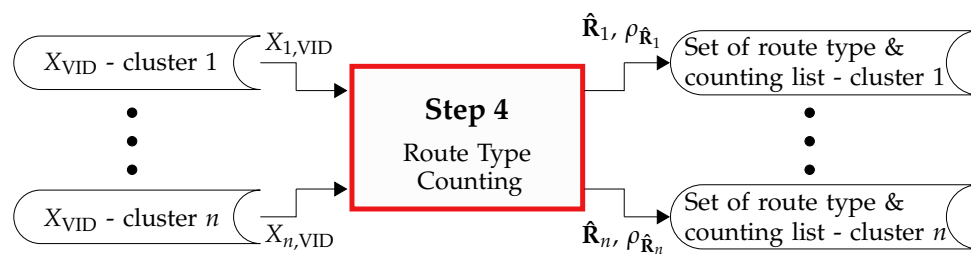


Figure 10. Overview of step 4 of the PROMOTING method: the route-type counting.

If one considers that the names of the vertices are unique, two intersections cannot be compared by the vertex name alone. Therefore, a common vertex representation per cluster is needed. This common representation is achieved in the form of a template graph that is created for each cluster. The graph of the first X_{VID} of each cluster is taken as the template of that node. Then, the bijective function (Equations (12) and (13)) is used to map the rest of the vertices of the routes within the cluster. A graphical depiction of this process is shown in Figure 11. There, the graph G_1 is mapped to G^* using the bijective function $f : V_{G_1} \rightarrow V_{G^*}$.

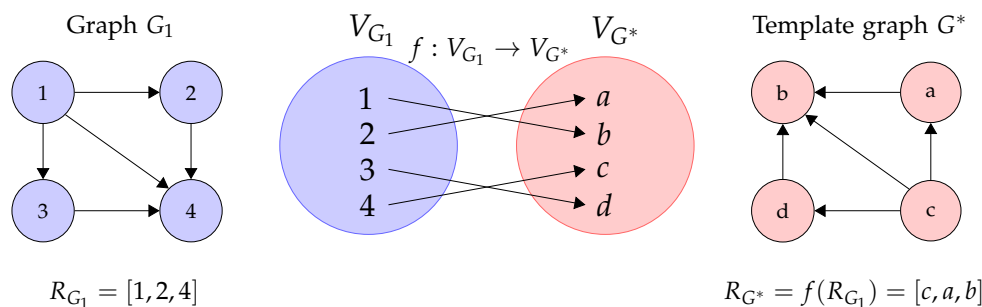


Figure 11. Graphical depiction of the route mapping. The graph G_1 is shown in purple. The graph G^* is shown in red.

Once the vertices of the routes within the cluster are mapped to those of the template graph, the route types are extracted. Each route type is a specific vertex sequence in the cluster. Then, the set of route types \hat{R}_c is generated for each c th cluster as follows:

$$\hat{R}_c = \{\hat{R}_{c,1}, \hat{R}_{c,2}, \dots, \hat{R}_{c,n}, \dots\}, \tag{14}$$

where the first sub-index of the elements of \hat{R}_c indicates the cluster to which the route type belongs, and the second sub-index is an identifier for the type of route within the cluster.

For each route type $\hat{R}_{c,n}$ identified, the frequency $\rho_{\hat{R}_{c,n}}$ is computed. This frequency represents how often the route type $\hat{R}_{c,n}$ appears in the cluster based on the dataset. This information is relevant for the estimation of the probability that a traffic participant will drive a given route. Then, the counting list of the route types $\rho_{\hat{R}_c}$ is generated for each c th cluster as follows

$$\rho_{\hat{R}_c} = (\rho_{\hat{R}_{c,1}}, \rho_{\hat{R}_{c,2}}, \dots, \rho_{\hat{R}_{c,n}}, \dots). \tag{15}$$

3.5. Mode Estimation

Once the set of intersection graphs S_{int} and the map vertex set V are generated (as per Section 3.1), the VID list X_{VID} is obtained (see Section 3.2), the VIDs are clustered (see Section 3.3), and the set of route types \hat{R}_c , and the counting list $\rho_{\hat{R}_c}$ are extracted (as per Section 3.4), the next step is to generate the modes and estimate the mode probability. That is, to create a set of routes that a traffic participant can drive for a given intersection type (cluster) and motion history, and to estimate the probability that a given mode will be

driven. Thus, for each c th cluster, this process extracts the mode data M_c . A graphical depiction of this process is shown in Figure 12.

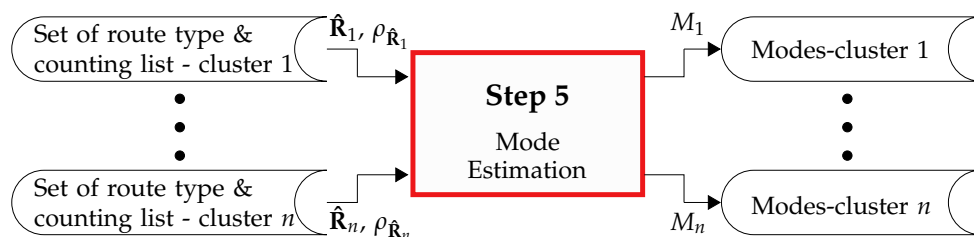


Figure 12. Overview of step 5 of the PROMOTING method: the mode estimation.

First, a set of sub-routes for each route type is generated in $\hat{\mathbf{R}}_c$ in the c -cluster. For example, for the first route type in the c -cluster $\hat{R}_{c,1}$, the set of sub-routes $S_{c,1}$ is generated, such that

$$S_{c,1} \subseteq \hat{R}_{c,1}. \tag{16}$$

Since a route is a vertex sequence, each sub-route $s \subseteq S_{c,1}$ is defined as a coherent sub-sequence of vertices of the corresponding route.

As an example of the creation of the set of sub-routes let the map topology correspond to Figure 4 and the intersection belong to the c th cluster. Given the set of route types $\hat{\mathbf{R}}_c$,

$$\hat{\mathbf{R}}_c = \{\hat{R}_{c,1}, \hat{R}_{c,2}, \hat{R}_{c,3}\}, \tag{17}$$

$$\hat{R}_{c,1} = [1, 7, 14], \tag{18}$$

$$\hat{R}_{c,2} = [1, 7, 9, 15], \tag{19}$$

$$\hat{R}_{c,3} = [2, 9, 15]; \tag{20}$$

the corresponding sets of sub-routes $S_{c,1}$, $S_{c,2}$, and $S_{c,3}$ for each route type $\hat{R}_{c,1}$, $\hat{R}_{c,2}$, and $\hat{R}_{c,3}$ can be generated, so that

$$S_{c,1} = \{[1], [7], [14], [1, 7], [7, 14], [1, 7, 14]\}, \tag{21}$$

$$S_{c,2} = \{[1], [7], [9], [15], [1, 7], [7, 9], [9, 15], [1, 7, 9], [7, 9, 15], [1, 7, 9, 15]\}, \tag{22}$$

$$S_{c,3} = \{[2], [9], [15], [2, 9], [9, 15], [2, 9, 15]\}. \tag{23}$$

Second, the set S_c^* that contains all unique sub-routes of the c th cluster is then defined as

$$S_c^* = S_{c,1} \cup S_{c,2} \cup S_{c,3}. \tag{24}$$

The mode data M_c of the c th cluster have as many elements, as the sub-routes s are driven in the cluster. This means that, for each sub-route $s \in S_c^*$, an element of M_c is computed. Each element of M_c contains (1) the set of modes $\mu_{c,s}$ and (2) the estimated probabilities $P(\mu_{m|c,s})$ of each mode $\mu_{m|c,s}$ and is computed as follows:

1. The set of modes $\mu_{c,s}$ used to forecast the possible modes that a vehicle can drive on (1) given the observation of the sub-route s , (2) where each mode ends with an

outgoing vertex, and (3) where each mode is part of S_c^* . For this, a set $\hat{S}_{c,s}$ is created, so that

$$\hat{S}_{c,s} = \{\hat{s}_{1|c,s}, \hat{s}_{2|c,s}, \dots, \hat{s}_{m|c,s}, \dots\}, \quad (25)$$

with

$$\hat{S}_{c,s} \subseteq S_c^* : s \in \{\hat{s}_{1|c,s}, \hat{s}_{2|c,s}, \dots, \hat{s}_{m|c,s}, \dots\}, \quad (\hat{s}_{m|c,s} \cap V_{c,\text{out}}) \neq \emptyset, \forall m, \quad (26)$$

where the set $V_{c,\text{out}}$ contains the outgoing vertices of the template graph of the c th cluster. Since the observed sub-route is not part of the modes, i.e., of the future motion, the observed sub-route s is extracted from each of the m th sub-sequences $\hat{s}_{m|c,s}$, generating the corresponding m th mode $\mu_{m|c,s}$. This allows the definition of the set of modes $\mu_{c,s}$ as follows

$$\mu_{c,s} = \{\mu_{1|c,s}, \mu_{2|c,s}, \dots, \mu_{m|c,s}, \dots\}, \quad (27)$$

where each element of $\mu_{c,s}$ represents a unique mode of completing the crossing of an intersection with the template graph of the c th cluster according to the recorded data and the observation s .

2. The conditional probability estimation $P(\mu_{m|c,s})$ of the m th mode $\mu_{m|c,s} \in \mu_{c,s}$ is estimated. This represents the probability that a traffic participant will drive on the m th mode given the c th cluster and the s th observed sub-route in this cluster. The conditional probability is given by

$$P(\mu_{m|c,s}) = \frac{\rho_{m|c,s}}{\rho_{c,s}}. \quad (28)$$

On the one hand, $\rho_{m|c,s}$ indicates how often a vehicle is traveling a route type in the c th cluster with the initial sequence-part defined by the s th observed sub-route, and the final sequence-part defined by the m th mode $\mu_{m|c,s}$. On the other hand, $\rho_{c,s}$ indicates how often a vehicle is traveling the s th observed sub-route in the c th cluster of the dataset. Then, $\rho_{c,s}$ is defined by

$$\rho_{c,s} = \sum_{q=1}^{|\hat{R}_c|} \rho_{\hat{R}_{c,q}} \cdot z_q, \quad (29)$$

where

$$z_q = \begin{cases} 1, & s \in S_{c,q} \\ 0, & \text{otherwise.} \end{cases} \quad (30)$$

The frequency $\rho_{\hat{R}_{c,q}}$ was introduced in Section 3.4 and indicates how often a vehicle is traveling the route type $\hat{R}_{c,q}$ in the c th cluster. The Boolean z_q allows us to select only those route types $\hat{R}_{c,q}$ in which the sub-route s is part of its sequence. Given the above, the sum of the probabilities of all modes is then given by

$$\sum_{m=1}^{|\mu_{c,s}|} P(\mu_{m|c,s}) = 1. \quad (31)$$

These two steps (Equations (25)–(30)) are applied for each observed sub-route s in the c th cluster in order to generate each element of the mode data M_c .

4. Evaluation and Results

In this section, the evaluation procedure and evaluation results are detailed. The proposed methodology is evaluated with respect to its ability to generate similar modes, mode probabilities, route types, graphs, and database trees, given similar datasets as inputs. For this, the Lyft database is used as data source, because it contains map information, as well as data about the motion of traffic participants. The data from the traffic participants are randomly divided into two independent datasets (D_1 and D_2), where D_1 is the small training dataset provided Lyft for the Kaggle Challenge <https://www.kaggle.com/c/lyft-motion-prediction-autonomous-vehicles> (accessed on 11 October 2021), and D_2 is the validation dataset provided by Lyft, while the map information remains the same for both datasets. An overview of the evaluation process is shown in Figure 13, and each step of the PROMOTING method is detailed in what follows.

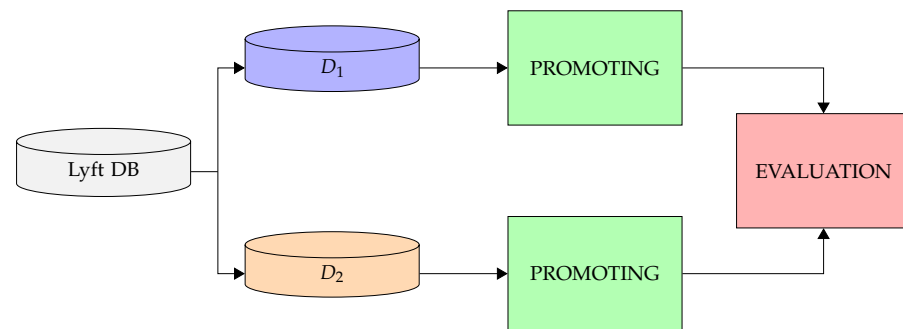


Figure 13. An overview of the evaluation process of the PROMOTING method.

The first step of the PROMOTING method (as per Section 3.1) describes the static traffic information (map vertex set V and intersection graphs S_{int}). Given that this information does not vary over time and is shared among datasets, the outputs of the first step for each given dataset are not compared. A summary of the road infrastructure description of the Lyft database is shown in Table 1.

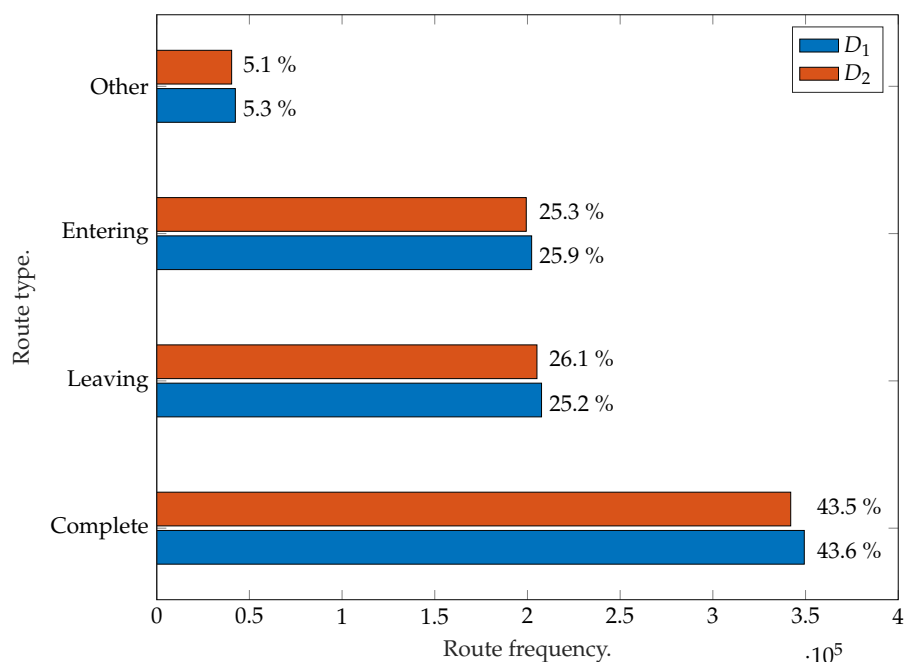
Table 1. Summary of the road infrastructure description of the Lyft database.

Feature	Name	Value
Graph order (number of map vertices)	n_v	8506
Graph size (number of map edges)	n_e	12,185
Number of intersections contained in the map	n_{int}	909

The second step of the PROMOTING method (as per Section 3.2) extracts the VID list X_{VID} . Given that each X_{VID} is generated from a unique set of traffic scenes, the VIDs from different datasets are inherently different. In this step, the routes contained in the VIDs from D_1 and D_2 cannot be compared, because the vertices that compose each route have different names and are not yet standardized to a template graph. However, the details of each VID (number of scenes, objects, vehicles, etc.) can be compared, which allows us to corroborate that both D_1 and D_2 are similar in size. This is important, because datasets of different sizes would imply different numbers of clusters, types of clusters, modes, and so on. Specifically, a total of ≈ 1.6 millions routes of vehicles crossing intersections are extracted from the Lyft database [25]. Approximately 50.5% of the routes belong to D_1 , while the remaining $\approx 49.5\%$ belong to D_2 . The route distribution according to Section 3.2 is shown in Figure 14, and a summary of the details of the VID of each dataset is shown in Table 2.

Table 2. Summary of the details of the VIDs generated from D_1 and D_2 .

Feature	Name	Value (D_1/D_2)
Number of traffic scenes	n_{scenes}	16,265/16,220
Number of traffic participants	$n_{\text{obj,total}}$	20,320,381/19,557,084
Number of vehicles	$n_{\text{veh,total}}$	4,710,949/4,621,107
Number of routes	n_{VID}	801,612/786,919
Number of “complete” routes	$n_{\text{VID,compl}}$	349,322/342,003
Number of “entering” routes	$n_{\text{VID,enter}}$	202,260/199,359
Number of “leaving” routes	$n_{\text{VID,leave}}$	207,595/205,122
Number of “other” routes	$n_{\text{VID,other}}$	42,435/40,435
Number of intersections crossed by “complete” routes	$n_{\text{int,veh}}$	250/250

**Figure 14.** Route distribution according to Section 3.2: complete, outgoing, entering, and other.

As can be inferred from Figure 14 and Table 2, both datasets, D_1 and D_2 , are similar in size, thus aiding in a fair evaluation of the method. Further, as mentioned in Section 3.2, only “complete” routes have been selected in the output of the second step of PROMOTING. The reason for this is that these routes are the only type that contain a full description of the intersection crossing from the entrance to the exit.

The third step of the PROMOTING method (as per Section 3.3) focuses on the clustering of the VIDs according to their graph isomorphism. The comparison metric is the structure of the database trees T_{D_1} and T_{D_2} that are generated when the datasets D_1 and D_2 are used as inputs. The node generation of both trees is analyzed, that is, how was the database tree was generated for each input dataset. If the trees are similar, it is an indication that the method is able to cluster similar routes, even when they come from different datasets. The common tree T_{com} is defined as one with a lineage such as the one that is present in both T_{D_1} and T_{D_2} , i.e., each node of T_{com} within each level of the tree has a counterpart in both T_{D_1} and T_{D_2} . T_{com} can be expressed as follows

$$T_{\text{com}} = T_{D_1} \cap T_{D_2}. \quad (32)$$

The comparison of the structure of the database tree of both T_{D_1} and T_{D_2} with T_{com} is shown in Table 3.

Table 3. Comparison of the database trees of both T_{D_1} and T_{D_2} with T_{com} .

Feature	Database Tree			
	T_{D_1}		T_{D_2}	
	$\cap T_{com}$	Total	$\cap T_{com}$	Total
Clusters ($n_{clusters}$)	168 (97.1%)	173	168 (97.1%)	173
Routes ($n_{VID,compl}$)	349,313 (99.99%)	349,322	341,997 (99.99%)	342,003

Given that both datasets D_1 and D_2 are similar in size, from the results shown in Table 3, it can be inferred that the method is able to comparably cluster the dynamic data from different datasets.

The fourth step of the PROMOTING method (as per Section 3.4) consists of the counting of route types within each cluster. Given a cluster a from T_{D_1} , its equivalent cluster b from T_{D_2} is the one with the similar template graph. The comparison metric is computed by the number of routes in cluster a that have an equivalence (same route type) in cluster b , normalized by the overall number of routes in cluster a . For this, let $n_c^{(D_1)}$ be the number of routes of the c th cluster of T_{D_1} and $n_{c,e}^{(D_1)}$ be the number of similar routes, given the c th cluster of T_{D_1} and its equivalent cluster in T_{D_2} . Then, the comparison metric is given by

$$\eta_{c,e} = \frac{n_{c,e}^{(D_1)}}{n_c^{(D_1)}}. \quad (33)$$

Then, the metric $\tilde{\eta}_{c,e}$ that represents the average of the ratio of equivalent routes between all common c th clusters from T_{D_1} and T_{D_2} is estimated as follows

$$\tilde{\eta}_{c,e} = \frac{1}{n_{clusters}} \sum_{c=1}^{n_{clusters}} \eta_{c,e}. \quad (34)$$

For this comparison, $\tilde{\eta}_{c,e} = 95.82\%$ was achieved. This indicates that common c th clusters from T_{D_1} and T_{D_2} contain mostly the same route types. This indicates that the method is able to cluster the routes of traffic participants from different datasets in a similar manner.

The fifth step of the PROMOTING method (as per Section 3.5) performs the mode estimation. Therefore, the comparison metric is based on the generated modes and their estimated probabilities. For this, let $P(\mu_{m|c,s}^{(D_1)})$ be the probability that a vehicle will drive the m th mode given the c th cluster and the s th observed sub-route, considering the dataset D_1 . Similarly, let $P(\mu_{m_e|c_e,s_e}^{(D_2)})$ be the probability that a vehicle will drive the m_e th mode given the c_e th cluster and the s_e th observed sub-route, considering the dataset D_2 . Here, the subscript \dots_e indicates that the corresponding equivalence is used, i.e., the m_e th mode is the equivalence of the m th mode. Therefore, only equivalent modes in equivalent clusters are considered.

Then, the relative difference $\eta_{m|c,s}$ between the probabilities of equivalent modes of both trees with respect to the probability $P(\mu_{m|c,s}^{(D_1)})$ is given by

$$\eta_{m|c,s} = \frac{|P(\mu_{m_e|c_e,s_e}^{(D_2)}) - P(\mu_{m|c,s}^{(D_1)})|}{P(\mu_{m|c,s}^{(D_1)})}. \quad (35)$$

Equation (35) is then estimated for all equivalent modes given all equivalent observations in all equivalent clusters. Then, the metric $\tilde{\eta}_{m|c,s}$ that represents the average of $\eta_{m|c,s}$

of all equivalent modes for all observations in all common clusters from T_{D_1} and T_{D_2} is computed as follows:

$$\tilde{\eta}_{m|c,s} = \frac{1}{n_{m,e}} \sum_{m=1}^{n_{m,e}} \eta_{m|c,s}, \quad (36)$$

where $n_{m,e}$ indicates the total number of equivalent modes between T_{D_1} and T_{D_2} . For the used datasets, $\tilde{\eta}_{m|c,s} = 0.39\%$. This shows that the mode probabilities, when estimated from two different datasets, are similar to each other. This indicates that the mode probability, when calculated using a large dataset, can estimate mode probabilities for similar datasets from same distributions. Even when PROMOTING uses different datasets, it is able to estimate the modes and the probability of each mode in a similar fashion for equivalent sub-route observations in equivalent intersections.

The main results of the evaluation of steps 4 and 5 of PROMOTING are summarized in Table 4.

Table 4. Main results of the evaluation of steps 4 and 5 of PROMOTING.

Feature	Name	Value
Average ratio of equivalent routes	$\tilde{\eta}_{c,e}$	95.82%
Average relative difference between equivalent modes	$\tilde{\eta}_{m c,s}$	0.39%

A representative graphical example of the extraction of modes and the estimation of the mode probabilities is shown in Figure 15.

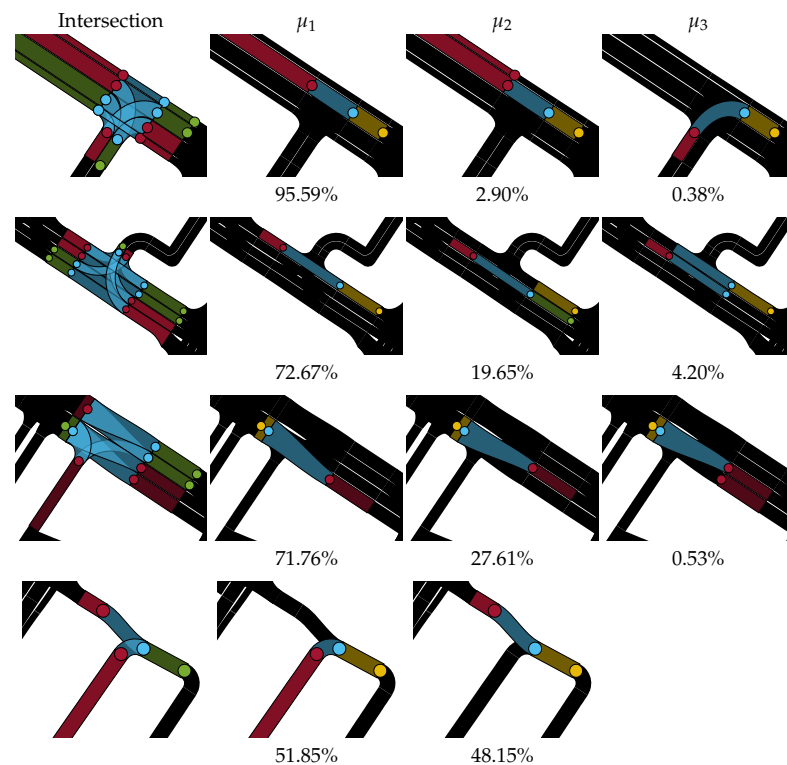


Figure 15. Example of the extraction of modes and estimation of the probability of each mode for four different types of intersections of the Lyft database. In the first column, the intersection is represented by the vertices that compose its graph. The second, third, and fourth columns represent the most probable modes (from highest to lowest probability), given the observed sub-route coloured in yellow and the history of the motion.

5. Discussion

A common challenge of multi-modal motion prediction, is to determine the “optimal” number of modes to predict, that is, how many trajectories per traffic participant should be predicted in order to comprehensively model a given traffic scene. This question has to take into consideration the amount of computational resources available, the time constraints, and the number of traffic participants, among others. Not only the number of trajectories is important, but also what they should look like. The PROMOTING method serves as a reference that shows both what the modes in a given intersection look like and what the probability is that a traffic participant will drive a specific mode. That is, the proposed method aids in the trajectory-prediction task. The method has the potential to be highly valuable for both the training and inference phases of ML methods for multi-modal motion prediction.

Along with the trajectory prediction that each traffic participant performs, the PROMOTING method could also prove to be useful at smart intersections with Vehicle-to-everything (V2X) capabilities. In that scenario, an automated vehicle could receive the information of the crossing (graphs, modes, etc.) from the infrastructure, so that the traffic participant could perform a better prediction of their own motion according to different parameters, such as efficiency or traffic load. This can be extended to all traffic participants, where each one knows where all the other traffic participants are and can predict the motion of the others with the help of the crossing information. This is relevant in the case of mixed traffic, where automated and human-driven vehicles coexist at the same intersection. Even when no V2X is present, the PROMOTING method could still be on board the EGO-vehicle, and, together with the information from exteroceptive sensors, the relationship between the surrounding traffic participants and their possible routes can be generated.

The PROMOTING method was evaluated in this work using the Lyft database. However, the method is not dependent on this database; instead, it can be used together with other map representations, as long as the required map properties are present, that is, the method is not limited to certain types of intersections but can instead generate the information from many different sources.

The method can be extended using real-time traffic information, as already provided by many navigation tools. The constant update of the traffic conditions (flow, weather, construction works, etc.) can provide an extra benefit for traffic analysis, as well as for the trajectory planning of traffic participants. This real-time traffic information does not necessarily have to come from navigation tools or infrastructure but could also be transmitted by other vehicles in the vicinity that have already crossed the intersection.

It should be noted that the mode probability estimation presented in this work does not take into account the interaction between traffic participants. In this paper, only the past sub-route, not the state of the other objects, are considered in the condition. This is a point for future research, with a special focus on the exchange of intentions between traffic participants via V2X. In addition, the investigation of abnormal behavior of traffic participants is also envisaged.

6. Conclusions

In this research work, a novel method named PROMOTING is proposed that is able to generate the modes (probable routes) of traffic participants, as well as estimate the probability that a traffic participant will drive a specific mode. This is done with the aim of supporting ADSs in their task of multi-modal motion prediction.

Mode generation is performed by clustering intersections based on the isomorphisms of their road topology. This allows us to cluster together equivalent intersections and, as a consequence, the equivalent routes of vehicles that crossed the isomorphic intersections. The probability of each mode is estimated based on the frequency with which each route is driven and a given observation (sub-route within the intersection).

The method is evaluated using the Lyft database. The results confirm that the method is able to cluster equivalent intersections and modes. The estimated probabilities of equivalent

modes are almost identical, which also corroborates that the method estimates similar probabilities for similar crossings given similar observations. Therefore, PROMOTING provides a methodology that makes it possible to generate a labeled dataset that allows researchers to estimate multiple routes for each traffic participant and provides a probability score for each of the estimated routes. This labeled dataset has the potential to be highly valuable for ML models aimed at the task of motion prediction.

The method could be improved with the inclusion of real-time traffic information that can be sent via V2X communication, including information about the road infrastructure, cellular networks, or other traffic participants. The method is not limited to the used dataset but could also be implemented for other map sources.

Interested readers are referred to the repository [31], where the code that implements the methodology proposed in PROMOTING is made publicly available.

Author Contributions: A.F.F. contributed to the investigation and conceptualization of the methodology, its implementation, and its evaluation. J.W. and E.S.M. contributed to the conceptualization of the methodology. M.B. and C.F. contributed to the conceptualization and supervision of the methodology, the funding acquisition, and the project administration. A.G.H. contributed to the supervision of the methodology. All authors contributed to the formal analysis, writing, and review and editing. A.G.H. is a policy analyst in the European Parliamentary Research Service (EPRS), the internal research service and think-tank of the European Parliament. He is writing in a personal capacity, and any views expressed do not represent an official position of the European Parliament. All authors have read and agreed to the published version of the manuscript.

Funding: This research and the APC were funded by the Federal Ministry of Transport and Digital Infrastructure of Germany (Bundesministerium für Digitales und Verkehr), project number 01MM20012J.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The code is public under [31].

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations and notation are used in this manuscript :

ADE	Average Displacement Error
ADSS	Automated Driving Systems
EPRS	European Parliamentary Research Service
EU	European Union
FDE	Final Displacement Error
MAH	Multiple Attention Heads
ML	Machine Learning
POGs	Predicted-Occupancy Grids
PROMOTING	Probabilistic Traffic Motion Labeling
RMSE	Root Mean Square Error
V2X	Vehicle-to-everything
VID	Vehicle Intersection Data
A_i	Set of the adjacent vertices of the i th vertex with respect to the driving direction
$A_{i,left}$	Set of the immediate vertices to the left of the i th vertex with respect to the driving direction
$A_{i,pre}$	Set of the immediate previous vertices of the i th vertex with respect to the driving direction
$A_{i,right}$	Set of the immediate vertices to the right of the i th vertex with respect to the driving direction
$A_{i,suc}$	Set of the immediate vertices following the i th vertex with respect to the driving direction

D_1	Training dataset provided by the Lyft database
D_2	Validation dataset provided by the Lyft database
E	Set of edges ϵ_k of the map
E_l	Set of edges of the l th intersection
G	Directed graph of the road topology of the map
G^*	Template graph
M_c	Mode data of the c th cluster
$P(\mu_{m c,s}^{(D_1)})$	Probability that a vehicle will drive the m th mode given the c th cluster and the s th observed sub-route for the dataset D_1
$P(\mu_{m_e c_e,s_e}^{(D_2)})$	Probability that a vehicle will drive the equivalent m_e th mode given the c_e th cluster and the s_e th observed sub-route for the dataset D_2
$Q_{i,j}$	Sequence of vertices that follows the j th vehicle in the i th traffic scene
\hat{R}_c	Set of route types in the c th cluster
$\hat{R}_{c,n}$	Sequence of vertices that represents the n th route type in the c -cluster
R_k	Sequence of vertices that represents the k -route of a vehicle that cross an intersection
$\hat{S}_{c,s}$	Set of sequence of vertices that allows to extract the modes given the c th cluster and the s th observed sub-route
$S_{c,n}$	Set of sub-routes of the c -cluster generated from $\hat{R}_{c,n}$
S_c^*	Set that contains unique sub-routes given all the n th sub-routes sets $S_{c,n}$
S_{int}	Set of intersection graphs generated from the map
T_{com}	Database tree whose lineage is present in both T_{D_1} and T_{D_2}
T_{D_1}	Database tree that distributes the VIDs extracted from the database D_1
T_{D_2}	Database tree that distributes the VIDs extracted from the database D_2
V	Set of vertices v_i of the map
V_l	Set of vertices of the l th intersection
$V_{l,\text{in}}$	Set of incoming vertices of the l th intersection
$V_{l,\text{out}}$	Set of outgoing vertices of the l th intersection
$V_{l,x}$	Set of crossing vertices of the l th intersection
$X_{i,j,k}$	VID extracted at the i th scene for the j th vehicle that drives the k th route.
X_{VID}	List of VIDs extracted from a given dataset
$X_{c,\text{VID}}$	List of VIDs grouped in the c th cluster given the list X_{VID}
$n_{1,\text{in}}$	Number of vertices of a graph whose in-degree is equal to 1
$n_{1,\text{out}}$	Number of vertices of a graph whose out-degree is equal to 1
$n_c^{(D_1)}$	Number of routes in the c th cluster of the tree T_{D_1}
$n_{c,e}^{(D_1)}$	Number of similar routes in the c th cluster of the tree T_{D_1} , given the c th cluster of T_{D_1} and its equivalent cluster in T_{D_2}
n_ϵ	Number of edges contained in the map
n_{clusters}	Number of VID clusters given a specific dataset
n_{int}	Number of intersection graphs contained in the map
$n_{\text{int,veh}}$	Number of intersections crossed by “complete” routes
n_{obj}	Number of traffic participants in a specific traffic scene
$n_{\text{obj,total}}$	Number of traffic participant in a corresponding dataset
n_{routes}	Number of routes that cross a intersection given a vehicle
n_{scenes}	Number of traffic scenes contained in a specific dataset
$n_{\text{veh,total}}$	Number of vehicles contained in a specific dataset
n_{VID}	Number of extracted routes from a specified dataset
$n_{\text{VID,compl}}$	Number of “complete” routes
$n_{\text{VID,enter}}$	Number of “entering” routes
$n_{\text{VID,leave}}$	Number of “leaving” routes
$n_{\text{VID,other}}$	Number of “other” routes
$n_{m,e}$	Number of equivalent modes between T_{D_1} and T_{D_2}
n_\vee	Number of vertices contained in the map
r_q	Element of the route R_k which represent a vertex of the map
s	Sub-route represented by a vertex or a sequence of vertices
$\hat{s}_{m c,s}$	Sequence of vertices used to extract the corresponding mode $\mu_{m c,s}$ given the c th cluster and the s th observed sub-route
z_q	Boolean that indicates if an observation is part of the sub-route set $S_{c,q}$ or not
ϵ_k	k th edge of the map

$\eta_{c,e}$	Ratio of equivalent routes between all common c th clusters from T_{D_1} and T_{D_2}
$\tilde{\eta}_{c,e}$	Average of the ratio $\eta_{c,e}$ of equivalent routes between all common c th clusters from T_{D_1} and T_{D_2}
$\eta_{m c,s}$	Relative difference between the probabilities of equivalent modes of the trees T_{D_1} and T_{D_2} with respect to the probability $P(\mu_{m c,s}^{(D_1)})$
$\tilde{\eta}_{m c,s}$	Average of $\eta_{m c,s}$ of all equivalent modes for all observations in all common clusters from T_{D_1} and T_{D_2}
Θ_{seq}	Matrix degree of a graph formed by all its in-degrees and out-degrees
κ_i	Unique identifier of the intersection of which the i th vertex is part of
$\mu_{c,s}$	Set of modes extracted given the c th cluster and the s th observed sub-route
$\mu_{m c,s}$	Mode m given the c th cluster and the s th observed sub-route
v_i	i th vertex of the map
$\rho_{c,s}$	Number of times the vehicles are traveling the s th observed sub-route in the c th cluster of a specific dataset
$\rho_{m c,s}$	Number of times the vehicles are traveling the s th observed sub-route in the c th cluster of a specific dataset, with the initial sequence-part defined by the s th observed sub-route and the final sequence-part defined by the m th mode $\mu_{m c,s}$
$\rho_{\hat{R}_c}$	List whose elements indicate how often each route type appears in the c th cluster
$\rho_{\hat{R}_{c,n}}$	Number of times that the route type $\hat{R}_{c,n}$ is registered

References

- European Commission. *Road Safety: Europe's Roads Are Getting Safer but Progress Remains Too Slow*; European Commission: Brussels, Belgium, 2020.
- European Commission. *EU Road Safety Policy Framework 2021–2030—Next Steps towards “Vision Zero”*; Publications Office of the European Union: Luxembourg, 2020. [CrossRef]
- Singh, S. *Critical Reasons for Crashes Investigated in the National Motor Vehicle Crash Causation Survey*; Technical Report, Traffic Safety Facts Crash Stats Report No. DOT HS 812 506; National Highway Traffic Safety Administration: Washington, DC, USA, 2018.
- On-Road Automated Driving (ORAD) Committee. *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*; SAE International: Warrendale, PA, USA, 2018. [CrossRef]
- Yurtsever, E.; Lambert, J.; Carballo, A.; Takeda, K. A Survey of Autonomous Driving: Common Practices and Emerging Technologies. *arXiv* **2019**, arXiv:1906.05113.
- Messaoud, K.; Yahiaoui, I.; Verroust-Blondet, A.; Nashashibi, F. Attention Based Vehicle Trajectory Prediction. *IEEE Trans. Intell. Veh.* **2021**, *6*, 175–185. [CrossRef]
- Cui, H.; Radosavljevic, V.; Chou, F.C.; Lin, T.H.; Nguyen, T.; Huang, T.K.; Schneider, J.; Djuric, N. Multimodal Trajectory Predictions for Autonomous Driving using Deep Convolutional Networks. *arXiv* **2019**, arXiv:cs.RO/1809.10732.
- Deo, N.; Trivedi, M.M. Multi-Modal Trajectory Prediction of Surrounding Vehicles with Maneuver based LSTMs. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018. [CrossRef]
- Liu, Y.; Zhang, J.; Fang, L.; Jiang, Q.; Zhou, B. Multimodal Motion Prediction with Stacked Transformers. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 19–25 June 2021; pp. 7573–7582. [CrossRef]
- Dong, B.; Liu, H.; Bai, Y.; Lin, J.; Xu, Z.; Xu, X.; Kong, Q. Multi-Modal Trajectory Prediction for Autonomous Driving with Semantic Map and Dynamic Graph Attention Network. *arXiv* **2021**, arXiv:abs/2103.16273.
- Deo, N.; Wolff, E.; Beijbom, O. Multimodal Trajectory Prediction Conditioned on Lane-Graph Traversals. In Proceedings of the 5th Annual Conference on Robot Learning, London, UK, 8–11 November 2021.
- Luo, C.; Sun, L.; Dabiri, D.; Yuille, A.L. Probabilistic Multi-Modal Trajectory Prediction with Lane Attention for Autonomous Vehicles. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 25–29 October 2020; pp. 2370–2376.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv* **2017**, arXiv:cs.CL/1706.03762.
- Messaoud, K.; Deo, N.; Trivedi, M.M.; Nashashibi, F. Multi-Head Attention with Joint Agent-Map Representation for Trajectory Prediction in Autonomous Driving. *arXiv* **2020**, arXiv:abs/2005.02545.
- Kim, H.; Kim, D.; Kim, G.; Cho, J.; Huh, K. Multi-Head Attention based Probabilistic Vehicle Trajectory Prediction. *arXiv* **2020**, arXiv:cs.CV/2004.03842.
- Giuliani, F.; Hasan, I.; Cristani, M.; Galasso, F. Transformer Networks for Trajectory Forecasting. *arXiv* **2020**, arXiv:cs.CV/2003.08111.
- Quintanar, A.; Fernandez-Llorca, D.; Parra, I.; Izquierdo, R.; Sotelo, M.A. Predicting Vehicles Trajectories in Urban Scenarios with Transformer Networks and Augmented Information. *arXiv* **2021**, arXiv:cs.CV/2106.00559.
- Neemann, A. Autonomous Shuttles: Riding Past the Traffic Jam. Available online: https://www.zf.com/mobile/en/technologies/domains/autonomous_driving/stories/20211124_autonomousshuttle.html (accessed on 19 April 2022).

19. Ettinger, S.; Cheng, S.; Caine, B.; Liu, C.; Zhao, H.; Pradhan, S.; Chai, Y.; Sapp, B.; Qi, C.R.; Zhou, Y.; et al. Large Scale Interactive Motion Forecasting for Autonomous Driving : The Waymo Open Motion Dataset. *arXiv* **2021**, arXiv:abs/2104.10133.
20. Huang, X.; Cheng, X.; Geng, Q.; Cao, B.; Zhou, D.; Wang, P.; Lin, Y.; Yang, R. The ApolloScape Dataset for Autonomous Driving. *arXiv* **2018**, arXiv:abs/1803.06184.
21. Maddern, W.; Pascoe, G.; Linegar, C.; Newman, P. 1 Year, 1000 km: The Oxford RobotCar Dataset. *Int. J. Robot. Res.* **2017**, *36*, 3–15. [[CrossRef](#)]
22. Caesar, H.; Bankiti, V.; Lang, A.H.; Vora, S.; Liong, V.E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; Beijbom, O. nuScenes: A multi-modal dataset for autonomous driving. *arXiv* **2019**, arXiv:abs/1903.11027.
23. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The KITTI dataset. *Int. J. Robot. Res.* **2013**, *32*, 1231–1237. [[CrossRef](#)]
24. Chang, M.; Lambert, J.; Sangkloy, P.; Singh, J.; Bak, S.; Hartnett, A.; Wang, D.; Carr, P.; Lucey, S.; Ramanan, D.; et al. Argoverse: 3D Tracking and Forecasting with Rich Maps. *arXiv* **2019**, arXiv:1911.02620.
25. Houston, J.; Zuidhof, G.; Bergamini, L.; Ye, Y.; Chen, L.; Jain, A.; Omari, S.; Igloukov, V.; Ondruska, P. One Thousand and One Hours: Self-driving Motion Prediction Dataset. *arXiv* **2020**, arXiv:cs.CV/2006.14480.
26. Wurst, J.; Balasubramanian, L.; Botsch, M.; Utschick, W. Novelty Detection and Analysis of Traffic Scenario Infrastructures in the Latent Space of a Vision Transformer-Based Triplet Autoencoder. In Proceedings of the 2021 IEEE Intelligent Vehicles Symposium (IV), Nagoya, Japan, 11–17 July 2021; pp. 1304–1311. [[CrossRef](#)]
27. Wurst, J.; Balasubramanian, L.; Botsch, M.; Utschick, W. Expert-LaSTS: Expert-Knowledge Guided Latent Space for Traffic Scenarios. In Proceedings of the 2022 IEEE Intelligent Vehicles Symposium (IV), Aachen, Germany, 5–9 June 2022.
28. Nadarajan, P.; Botsch, M. Probability estimation for Predicted-Occupancy Grids in vehicle safety applications based on machine learning. In Proceedings of the 2016 IEEE Intelligent Vehicles Symposium (IV), Gothenburg, Sweden, 19–22 June 2016; pp. 1285–1292. [[CrossRef](#)]
29. Nadarajan, P.; Botsch, M.; Sardina, S. Predicted-occupancy grids for vehicle safety applications based on autoencoders and the Random Forest algorithm. In Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017; pp. 1244–1251. [[CrossRef](#)]
30. Nadarajan, P.; Botsch, M.; Sardina, S. Machine Learning Architectures for the Estimation of Predicted Occupancy Grids in Road Traffic. *J. Adv. Inf. Technol.* **2018**, *9*, 1–9. [[CrossRef](#)]
31. Flores Fernández, A. PROMOTING. Available online: <https://github.com/albertofloresfernandez/PROMOTING> (accessed on 9 May 2022).