IEEE Open Journal of
Intelligent Transportation
Systems

Received 30 August 2022; revised 10 November 2022 and 17 December 2022; accepted 30 December 2022. Date of publication 3 January 2023; date of current version 23 January 2023.

Digital Object Identifier 10.1109/OJITS.2022.3233904

Estimating Link Flows in Road Networks With Synthetic Trajectory Data Generation: Inverse Reinforcement Learning Approach

MINER ZHONG[1], JIWON KIM[2], AND ZUDUO ZHENG[2]

[1]Department of Sustainable Mobility, Royal HaskongingDHV Group, 3800 BC Amersfoort, The Netherlands

[2]Department of Civil Engineering, University of Queensland, Brisbane, QLD 4072, Australia

CORRESPONDING AUTHOR: M. ZHONG (e-mail: miner.zhong@outlook.com)

This work was supported in part by the Australian Research Council under Grant DE190101020.

ABSTRACT While traffic volume data from loop detectors have been the common data source for link flow estimation, the detectors only cover a subset of links. These days, other data sources such as vehicle trajectory data collected from vehicle tracking sensors are also incorporated. However, trajectory data are often sparse in that the observed trajectories only represent a small subset of the whole population, where the exact route sampling rate is unknown and may vary over space and time. In this paper, we develop a method that leverage these two limited data sources to enhance link flow estimation. This study proposes a novel generative modelling framework, where we formulate a vehicle's link-to-link movements as a sequential decision-making problem using the Markov Decision Process framework. We propose an Inverse Reinforcement Learning-based method, based on which synthetic population vehicle trajectories can be generated to estimate link flows across the whole network. The proposed method ensures the generated population vehicle trajectories are consistent with the observed traffic volume and trajectory data. The proposed generative modelling framework is compared to two existing methods in a synthetic road network and validated in a real road network.

INDEX TERMS Link flow estimation, trajectory data, synthetic trajectory generation, inverse reinforcement learning.

I. INTRODUCTION

THE ACCURATE estimation of traffic flows (or volumes) on road links is critical in managing a road network and evaluating its performance. While loop detectors are installed to collect link flow data, the observation points are often limited to a subset of links and there are still a significant proportion of links that do not have direct observations. Unobserved link flows need to be estimated based on available data and this is referred to as the link flow estimation problem in the transportation literature [1], [2], [3], [4].

If historical traffic volume data collected on the target link are available, link flows may be estimated based on such data, using various data-driven methods [5], [6]. When there are no historical data (or not sufficient historical data), link flows on the target links may still be estimated using flow information from other links in the same road network. In the literature, the sensor location and flow observability models have been extensively studied [7]. These studies aim to find the smallest subset and/or optimal placement of sensors on a network that enables the accurate estimation of traffic flows on all links across the network [8], [9], [10]. However, it is of great interest to solve the link flow estimation problem with a fixed set of sensors that are already installed in the road network with a non-optimal layout. In this case, it may still be possible to solve the link flow estimation problem using available traffic volume data on surrounding locations. Many previous studies focused on applying interpolation algorithms for missing data imputation. Generally, these studies proposed probabilistic models to retrieve traffic flow features under assumptions on the statistical properties of the traffic data [11], [12]. It is common among these

The review of this article was arranged by Associate Editor Meng Li.

studies to assume the existence of spatial autocorrelation among traffic data [13]. However, these interpolation models might be vulnerable under complex traffic patterns and extreme outliers. Their performance also depends greatly on the missing ratio. Some recent studies also proposed network tomography models which explore the structure and characteristics of the road network [14], [15]. Overall, many previous missing data interpolation/imputation studies use only one data source (i.e., the observed traffic volume data).

An alternative approach is to incorporate other traffic data available in the same road network. Recent years have witnessed an increasing application of vehicle detection technologies on the road network. A vehicle trajectory is a time-ordered sequence of locations visited by the vehicle (in latitudes and longitudes). A collection of vehicle trajectories usually offers deep insights into vehicle propagation information. Since the entire travel paths of vehicles are captured, trajectory data have better spatial coverage than traffic volume data from loop detectors. It is thus desirable to combine these two data sources to improve link flow estimation, which has received great interest in recent years [16], [17], [18], [23], [24], [25]. However, considering the limited market penetration rate of vehicle detection technologies and the data collection errors, the observed vehicle trajectory data may not reflect the true population trajectory distribution. Therefore, link flows estimated from these trajectory data may deviate from those estimated from the traffic volume data. Thus, the key question becomes how to combine these two limited data sources to enable the estimation of traffic flows on all links.

Many previous studies that attempted to incorporate trajectory data in link flow estimation make strong assumptions that observed trajectory data have high market penetration rates, route sampling rates and/or local capture rates [16], [17], [18]. It is also common among some studies to make assumptions on traveller's route choice behaviours and/or the availability of prior information origin-destination (OD) trip matrices or route flows [3], [19], [20]. However, these assumptions are often violated in real-world situations. In this paper, we aim to address more realistic scenarios, where observed trajectory data are sparse, that is, they only represent a limited sample of the whole population. Specifically, we use the term *'route sampling rate'* to describe the level of sparsity that a given trajectory dataset has. For a given route between an OD pair in the studied network, the route sampling rate is defined as the ratio of probe vehicle route flow (i.e., the number of observed trajectories on this route) to the full route flow (i.e., the total number of population vehicle trajectories on this route). The lower the route sample rate, the sparser the trajectory data.

Problem Definition: This paper considers the problem of estimating traffic flows on all links in a network, where only a subset of links is observed and the traffic volume data from those observed links are not sufficient to estimate all other link flows. In this case, vehicle trajectory data are used to provide information relevant to unobserved link flows. We propose a method to combine such limited traffic volume data with sparse trajectory data to estimate all link flows with minimal assumptions and requirements on the available data. In particular, these two data sources are assumed to have the following characteristics:

- Traffic volume data capture the traffic flow of the whole population at observation points. However, the spatial coverage of the available volume data is limited because only a small subset of links has detectors installed.
- Trajectory data capture the vehicles' route preferences and movement patterns across the network. Each observed trajectory is translated into a time-ordered sequence of links (only the spatial aspect is considered). We assume that route sampling rates (hereafter referred to as sampling rates) are unknown and the distribution of such sampling rates across the network is not uniform. Some paths and links on the network may not be covered by the observed trajectories. While sparse, the trajectory data still provide information on the route distribution that does not deviate dramatically from the population route distribution.

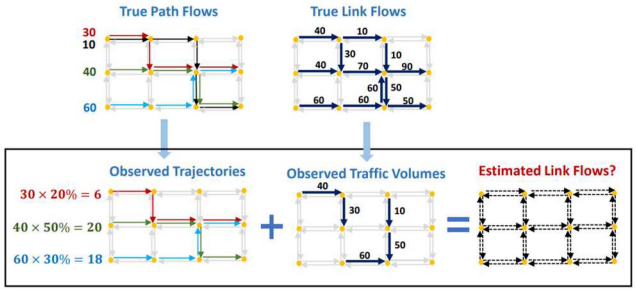

FIGURE 1. An illustrative example of the link flow estimation problem.

It is noted that we focus on estimating the 'aggregate' link flows over any given time interval for which data are provided. For example, our model produces link flows over an hour, a day, or a week if traffic volume and trajectory data are prepared to cover a specific 1-hr interval, the whole day, or the whole week, respectively.

Figure 1 shows an illustrative example of the proposed research problem. A road network is represented as a directed graph, where edges are road links and nodes are intersections. The top two figures show the true trajectory set and true traffic volumes that are unknown to a modeller. Consider the scenario where the modeller can observe a portion of trajectory data (with sampling rates of 0%, 20%, 50%, and 30% for black, red, green, and blue routes, respectively) and a subset of traffic volume data, as shown in the first two figures at the bottom. The modeller's goal is to estimate link flows across the whole network, as shown in the bottom-right figure.

This problem can be formulated as a route flow estimation problem, which aims to recover true vehicle trajectories

(that immediately leads to the knowledge of link flows) by using the observed link traffic volumes as constraints to satisfy conservation laws (conservation equations). The critical difficulty of this research problem, however, lies in its underdetermined nature because there are more 'unknowns' (route flows) than 'equations' (link observations). Instead of finding the optimal solution for route flows (true vehicle trajectories), this paper proposes a generative modelling approach, where the vehicle trajectory data are viewed as data samples from the true population trajectory distribution. The proposed generative modelling framework aims to learn a population trajectory distribution from the observed data and generate synthetic trajectories that mimic the true population trajectories, which can later be used for estimating unobserved link flows.

We formulate the trajectory data generation procedure as a Markov Decision Process (MDP), which we refer to as road-network MDP. The goal of our study is to first find a policy in the road-network MDP and then use this policy to generate synthetic vehicle trajectories for link flow estimation. Reinforcement Learning (RL) is a powerful way to learn policies for sequential decision-making tasks in MDPs [21]. Standard RL models aim to find a policy that maximizes the cumulative rewards of an agent's decisions. However, in our case, the reward function is not known, and the constraints (i.e., the generated trajectories should be consistent with the link traffic volume data) cannot be expressed as standard reward functions. Inverse Reinforcement Learning (IRL) aims at constructing a reward function given observed expert behaviours, which is suitable for solving the proposed problem. Ziebart et al. (2008a) developed a method called Maximum Entropy IRL (MaxEnt IRL), where the route choice decisions of drivers are modelled in an MDP [22]. The goal is to recover a reward function by viewing the available vehicle trajectory data as expert's demonstrations. The principle of maximum entropy is employed to resolve the ambiguity issue, which is, many different path distributions match the expert's demonstrations. Motivated by MaxEnt IRL [22], this paper proposes a method that can be implemented within the proposed generative modelling framework, namely Inverse Reinforcement Learning for link Flow estimation (IRL-F). IRL-F aims to find a policy in the road-network MDP that allows the generated vehicle trajectories to be consistent with the observed data. Once this policy is found, a simple technique is proposed to determine the optimal size of the synthetic trajectory set for solving the link flow estimation problem. The main contributions of this paper are as follows:

- The proposed framework makes it possible to estimate link flows across the network without relying on expensive infrastructures such as loop detectors covering every link in the road network. Unlike the existing methods, IRL-F does not require the optimal placement of sensors on the network or specific network structures. In addition, compared to existing interpolation methods, most common assumptions on traffic data statistical properties and/or spatial dependencies are relaxed.
- The proposed framework allows the incorporation of trajectory data in the link flow estimation problem, where the assumptions that the observed trajectories have known and uniform sampling rates and/or cover most link-to-link transitions in the road network are all relaxed. It makes no assumption about and the market penetration rates of the observed trajectory data and the traveller's route choice behaviours. This enables IRL-F to be applied to more realistic scenarios, which are challenging to the existing link flow estimation methods.
- The proposed framework provides a data-driven solution where the observed traffic data are used to recreate the underlying vehicle movement scenarios and generate synthetic trajectories. To the best of our knowledge, this is the first work to solve the link flow estimation problem using the concept of synthetic trajectory data generation.

II. LITERATURE REVIEW

A. LINK FLOW ESTIMATION WITH DATA FUSION

In addition to traffic volume data collected by sensors installed in the road network, a variety of other traffic data have been considered for link flow estimation problems. Zhang et al. (2020) proposed to solve the traffic flow estimation problem using both traffic volume data and crowdsourcing floating car data, which can be used to infer network-wide traffic speed information. Both data are used as input to a quadratic programming framework [23]. Ma et al. (2022) developed a route choice estimation framework considering both the probe vehicle trajectory and automated vehicle identification data as input, where route penetration rates are considered constraints. This framework is solved under the entropy maximization principle [24]. Brunauer et al. proposed to solve a local network propagation problem between observed links based on propagation rules indicated by the probe vehicle trajectories [2]. Such trajectories are assumed to cover most of the link-to-link transitions in the road network. Michau et al. proposed an estimation method for the link-based OD matrix using vehicle trajectory data, with sampling rates assumed to be a single numerical value for each OD pair in the road network [16]. Vogt et al. (2019) proposed an OD demand estimation model where the observed trajectory data are used to calculate the number of turns at each intersection in the road network [25]. Parry and Hazelton (2012) proposed a likelihood-based inference model based on traffic volume data and sporadic vehicle routing data, assuming the vehicle tracking probability is a fixed number across the network [18]. Lederman and Wynter (2011) proposed a two-phase solution framework. In the offline phase, the link-to-link splitting probabilities are determined according to traffic equilibrium principles. These probabilities are used in the online phase to propagate the observed link flows to unobserved links [3].

Another relevant study to the proposed link flow estimation problem is the link utilization method discussed in [4]. The authors proposed to first estimate the parameters for the recursive logit model using routing data collected from a set of fixed proximity sensors, which is then used to provide link utilization information on networks. This model shares some similarities with the generative model discussed in this paper, as both models aim to learn the sequential decision rules underlying the observed data. However, there are clear differences as follows: (1) The recursive logit model requires a utility function described using road characteristics chosen by domain experts. In contrast, the path probabilities in the proposed generative model are determined to satisfy the constraints imposed by the observed data. (2) Only one type of observed data is discussed in the recursive model, while the proposed generative method considers two types of observed data.

Overall, vehicle trajectory data are popular among previous studies when data fusion is considered for link flow estimation. However, they often require restrictive assumptions on sampling rates, prior traffic flow information and/or traveller's route choice behaviours, which we aim to relax in our proposed method.

B. GENERATIVE MODELS AND INVERSE REINFORCEMENT LEARNING

Generative models focus on finding the underlying distribution given some observed data samples and utilize this distribution to generate new data points. Recent years have seen many applications of such models in synthesizing mobility sequences [26] and generating human travel itineraries [27]. There are several recent studies that propose to use synthetic data generation approaches for traffic flow estimation problems. Dey et al. (2020) proposed a statistical method (the network tomography model) for OD flow estimation only using vehicle volumes observed by counting sensors [15]. Zhang et al. (2019) developed a traffic prediction model using Generative Adversarial Nets (GAN), in which historical traffic flow data are used as input for training such a model [28]. Chen et al. (2021) used a convolutional neural network (CNN) model to learn the traffic flow pattern from probe vehicle trajectories and automatic vehicle identification [29]. However, there is a lack of research in generative modelling approaches that consider both vehicle trajectory data and traffic volume data collected by fixed sensors on the road network while making minimal assumptions on the available data.

The applications of RL methods in the transportation literature depend on the availability of reward functions. Standard RL methods require reward functions to be known. A typical application of such methods is to solve traffic control problems [30]. For situations where it is challenging to manually determine reward functions, IRL has been applied. For example, Ziebart et al. proposed Maximum Entropy IRL, where the route choice decisions of drivers are modelled in an MDP [22]. The goal is to recover a reward function by viewing the available vehicle trajectory data as expert's demonstrations. Several extensions of MaxtEnt IRL have been proposed later with applications in learning route choice patterns from GPS trajectory data [31], [32]. However, few previous studies attempted to learn driving patterns from multiple data sources.

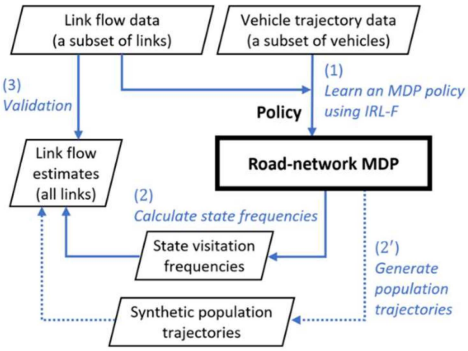

FIGURE 2. The proposed MDP-based generative modelling framework.

III. METHODOLOGY

A. THE MODELLING FRAMEWORK BASED ON MDP

A generative modelling framework is proposed to generate synthetic population trajectories, which are then used to estimate unobserved link flows in a road network. Figure 2 shows the overall concept of the proposed framework. Given link flow data from a subset of links and vehicle trajectory data from a subset of vehicles, the proposed IRL-F model learns a policy of a road-network MDP that mimics the true population trajectory distribution underlying the observed traffic data (process 1 in Figure 2). The most straightforward approach to estimating link flows using the learned policy is to generate synthetic vehicle trajectories for the whole population by sampling from the policy and counting the number of trajectories passing each link (process 2'). This approach is, however, computationally expensive due to the large number of trajectories that need to be generated until a proper population size is found. Instead, a simple approach is proposed to use state visitation frequencies measuring how often each link is visited by trajectories, which can be calculated as a by-product of the training process of the IRL-F (process 2). The link flows for all links can be estimated based on the state visitation frequencies from the road-network MDP and the link flows for the observed links can be validated against the actual volume data (process 3). The detailed methodologies for each process implemented in this generative modelling framework are discussed in the rest of this section.

Vehicles' sequential decision-making to perform link-to-link transitions in a road network can be modelled using a finite-horizon episodic MDP with absorbing states. In an episodic MDP, the agent-environment interaction breaks down into a series of separate episodes (episodic tasks), each of which consists of a finite sequence of time steps, rather than one long sequence of time steps (continuing

tasks). Vehicle trajectories are naturally expressed as episodic tasks, where each trajectory represents one episode in the MDP. This paper proposes to formulate the link flow estimation problem based on a road-network MDP, which can be described by a tuple $M = (S, A, \mu_0, P_T, r, \gamma, H)$.

- S is the set of states, which includes all links in the road network as well as an additional set of virtual links representing absorbing states corresponding to the end of an episode. First, possible destination locations in the network are identified by extracting a subset of links where the observed vehicle trajectories ended. Then, a virtual link is added to each of these identified links to allow an action to terminate trips on those possible destination locations. The subset S_v is defined as the subset of states that correspond to the links that have loop detectors installed (i.e., links that have available traffic volume count data).
- A is the set of actions, which are possible transitions from one link to the next link.
- μ_0 is the initial state distribution, which is a probability distribution over the set of links to start with. The initial state distribution is assumed to be equal to the distribution of initial links visited by the observed vehicle trajectories.
- $P_T : S \times S \times A \rightarrow [0,\ 1]$ is the state transition probability. $P_T(s_{t+1}|s_t, a_t)$ represents a probability of visiting s_{t+1} in the next time-step by choosing action a_t in state s_t at the current time-step t. The transition probability is known and deterministic in that choosing an action to move to a specific downstream link would indeed lead to that link with the probability of 1 and the probability of ending up in other links is zero.
- $r : S \times A \times S \rightarrow \mathbb{R}$ is the reward function, where $r_t = r(s_t,\ a_t,\ s_{t+1})$ is the reward associated with the transitioning to state s_{t+1} in the next time step by choosing action a_t in state s_t at the current time step t. Such a reward function is not known.
- $\gamma \in [0,\ 1]$ is the discount factor, which shows how much future reward should be discounted when the agent is making decisions. It is assumed to be 0.99.
- H is the horizon, which is the maximum number of steps in each episode. It is assumed to be equal to the maximum length of the observed vehicle trajectories.

A policy $\pi : S \rightarrow A$ is a function that maps a state to an action to take in that state, where $\pi(a|s)$ is the probability of selecting action a in state s under policy π. In the road-network MDP, this function represents the probability of a decision-making agent (vehicle) choosing the next link among a set of downstream links on the current link. For each episode, the agent starts from an initial state s_0. At each step $t = 0, 1, 2, \ldots, H$, the agent chooses an action a_t given the current state s_t based on the policy π, which results in transitioning to the next state s_{t+1} and receiving a reward r_t. The sequence of states and actions visited by the RL agent during an episode is normally called a *trajectory* in the RL literature, but we will refer to it as a *state-action path* $\tau = (s_0,\ a_0,\ s_1,\ a_1,\ \ldots,\ a_{H-1},\ s_H)$ to distinguish it from a vehicle trajectory in a road network. This state-action path can be translated into a spatial vehicle trajectory in the road network as the sequence of states visited by the agent represents the time-ordered sequence of road links travelled by a vehicle. In this paper, only the spatial aspect of vehicle trajectories (location sequences) will be considered, without the temporal aspect (travel time between locations).

Each episode is associated with a *return*, which is defined as the sum of the discounted rewards the agent received over the episode's state-action path. To learn the population trajectory distribution given the observed data samples (i.e., vehicle trajectory data and traffic volume data), we aim to find a policy in the road-network MDP that allows the agent to generate state-action paths that can be viewed as synthetic population vehicle trajectories which are consistent with the observed data samples. It is difficult to manually specify reward functions in the road-network MDP to achieve this goal. Instead, the observed traffic data offer information on the road-network MDP from a different aspect.

- The vehicle trajectory data consist of time-ordered sequences of links visited by the detected vehicles, which can be translated into sequences of states visited by the agent ordered by time-steps. By associating an action to each state, each vehicle trajectory can be translated into a state-action path in the road-network MDP. Let T_{obs} denote the set of state-action paths translated from the observed trajectory data. We can obtain a set of *state visitation* frequencies over the state set (S), $D_{obs} = \{D_s^{T_{obs}} | \forall s \in S\}$, where $D_s^{T_{obs}}$ is the visitation count on state s based on trajectory set T_{obs}, i.e., $D_s^{T_{obs}} = \sum_{\tau \in T_{obs}} \sum_{s' \in \tau} \mathbf{1}_s(s')$ and $\mathbf{1}_s(s')$ is the indicator function that returns 1 if $s' = s$ and 0 if $s' \neq s$. The observed trajectories only account for a subset of the population trajectories with non-uniform sampling rates (which are unknown). Thus, state visitation counts from the observed trajectories might have a different distribution from the state visitation counts calculated from the true population trajectories, the relationship between these two distributions is unknown.
- The traffic volume data are available each link that has a loop detector installed, which can be translated into the number of times each state is visited by the agent. We can obtain a set of state visitation counts over the detector links (S_v) from these volume data, $Q_{obs} = \{v_s | \forall s \in S_v\}$, where v_s represents the traffic volume on state s. Note that visitation counts on these states are equal to the counts derived from the true population trajectories because loop detectors installed on these links are assumed to be able to capture all vehicles passing the links. However, since traffic volumes are only observed on a subset of links (states), visitation counts on states outside S_v are not available.

Based on the above-mentioned conditions, the proposed research objectives can be expressed as finding a policy in the road-network MDP that generates state-action paths whose state visitation count distribution mimics the true state visitation count distribution implied by both types of observed data. These generated state-action paths represent synthetic population vehicle trajectories that can be used to estimate unobserved link flows by estimating the state visitation counts for the states that do not have detectors.

B. LINK FLOW ESTIMATION WITH IRL-F

In the road-network MDP, the agent needs to be trained to generate state-action paths that are consistent with the state visitation count distributions implied by the observed traffic data. MaxEnt IRL is adopted to train the agent to generate state-action paths that mimic a set of state-action paths demonstrated by an expert. However, the original MaxEnt IRL cannot be directly applied to solve the road-network MDP because it assumes that the expert's demonstrations are in the form of state-action paths, whereas in our case we need to represent the expert's demonstrations not only in terms of state-action paths (to account for trajectory data) but also in terms of state visit counts for a subset of states (to account for traffic volume data). As such, we propose a method called IRL-F that modifies MaxEnt IRL to solve the road-network MDP and further propose a simple link flow estimation method based on the policy found by IRL-F.

MaxEnt IRL: In this sub-section, we briefly introduce the MaxEnt IRL algorithm [22].

Environment definition: Given an MDP, MaxEnt IRL assumes that each state $s \in S$ is characterized by a feature vector $\mathbf{f}_s \in \mathbb{R}^k$, where k is the feature dimension. A state-action path τ is a sequence of all states and actions encountered by this agent, which is characterized by a path feature vector $\mathbf{f}_\tau \in \mathbb{R}^k$ that is defined as the sum of all state feature vectors in this path.

$$\mathbf{f}_\tau = \sum_{s \in \tau} \mathbf{f}_s \tag{1}$$

The agent makes sequential decisions based on some unknown reward values. It is assumed that visiting any state $s \in S$ incurs a state reward value that is linear to the state feature vector, parametrized by unknown reward weights $\theta \in \mathbb{R}^k$. The reward value $R_\theta(\tau)$ for a given path τ can be represented as the sum of state rewards along that path.

$$R_\theta(\tau) = \sum_{s \in \tau} \theta^T \mathbf{f}_s \tag{2}$$

MaxEnt IRL considers the distribution over the set of paths that this agent can take, aiming to find the path distribution that mimics the distribution indicated by the expert's demonstrations. For deterministic MDPs, the path distribution is parameterized by the reward weights θ. Let $P(\tau|\theta)$ denote the probability of taking path τ given reward parameter θ, which can be expressed as follows.

$$P(\tau|\theta) = \frac{e^{R_\theta(\tau)}}{\sum_{\tau'} e^{R_\theta(\tau')}} \tag{3}$$

IRL objective: The expert's behaviour is represented by a set of demonstrated paths (T_e). To train the agent to behave following such demonstrations, MaxEnt IRL aims to find the optimal reward weight (θ^*) that maximizes the likelihood of the expert's demonstrated paths under the maximum entropy distribution, which is expressed as follows.

$$\theta^* = \underset{\theta}{\operatorname{argmax}}\ L = \underset{\theta}{\operatorname{argmax}} \sum_{\tau \in T_e} \log P(\tau|\theta) \tag{4}$$

The optimal reward weight is obtained using a gradient descend method. To calculate the gradient, where M represents the number of demonstrated paths and D_s represents the expected state visitation frequency on state s.

$$\begin{aligned} \nabla_\theta L &= \frac{1}{M} \sum_{\tau \in T_e} \mathbf{f}_\tau - \sum_{s \in S} D_s \mathbf{f}_s \\ &= \mathbf{f}_{\text{expert}} - \mathbf{f}_{\text{policy}} \end{aligned} \tag{5}$$

The first part can be viewed as the expectation of path feature vectors over the expert's demonstrated paths, denoted by the expert's feature expectation $\mathbf{f}_{\text{expert}} \in \mathbb{R}^k$. The second part, denoted by the policy feature expectation $\mathbf{f}_{\text{policy}} \in \mathbb{R}^k$, can be viewed as the expectation of path feature vectors over a set of paths generated under the current policy, which is determined by the current reward weight θ. In this way, the gradient represents the difference between $\mathbf{f}_{\text{expert}}$ and $\mathbf{f}_{\text{policy}}$. The objective of MaxEnt IRL is thus to find the policy that matches the policy feature expectation with the expert's feature expectation.

Solution process: MaxEnt IRL can be solved using a gradient-based method. Generally, the reward weight θ can be randomly initialized, and then updated iteratively based on the gradient calculated using Eq. (5) Expert's feature expectation and policy feature expectation are needed to calculate such a gradient. The solution process will stop when the optimal value is found. The output of MaxEnt IRL is the optimal reward weight, based on which the reward function in the MDP can be recovered. Following the policy implied by this reward function, the agent generates the state-action paths that closely mimic the expert's demonstrations.

IRL-F: To solve the proposed link flow estimation problem, we propose a new method, IRL-F, which is adapted from the MaxEnt IRL method. Specifically, instead of observing the ground truth population trajectory set, we are given two types of observed data, each of which only reflects part of the true network flows. IRL-F is proposed to train the agent using these observed data so that the optimal policy generates a trajectory distribution that mimics the population trajectory distribution. To achieve this goal, we introduce

adaptations to the MaxEnt IRL method in the environment definition (i.e., new state feature definition) as well as the solution process (i.e., new gradient calculation process).

Environment definition: In the road-network MDP, each state $s \in S$ is characterized by a feature vector $\mathbf{f}_s \in \mathbb{R}^{k_1+k_2}$, which is a concatenation of two feature vectors $\mathbf{f}_s^{(1)}$ and $\mathbf{f}_s^{(2)}$. The objective is to match the policy feature expectation to the expert's feature expectation. Both observed data are expert demonstrations. Therefore, the first part of the state feature vector, $\mathbf{f}_s^{(1)}$, is designed to facilitate feature expectation matching regarding the observed trajectory data, while the second part of the state feature vector, $\mathbf{f}_s^{(2)}$, is designed to facilitate feature expectation matching regarding the observed traffic volume data.

$$\mathbf{f}_s = \left[\mathbf{f}_s^{(1)^T},\ \mathbf{f}_s^{(2)^T}\right]^T,\quad \mathbf{f}_s^{(1)} \in \mathbb{R}^{k_1},\ \mathbf{f}_s^{(2)} \in \mathbb{R}^{k_2} \tag{6}$$

In the road-network MDP, each link is represented by a state. We proposed a new state feature definition that is different from the definition in MaxEnt IRL. The original MaxEnt IRL method [22] was applied in the task of learning drivers' route choice behaviours from GPS trajectory data, where road segments are modelled as states and the state feature vector is defined in terms of four different road characteristics that describe each road segment: namely, road type, speed, lanes, and transitions. While using such general road features is useful in learning and interpreting the agent' route choice behaviour as a function of road characteristics, since our primary goal is to solve the link flow estimation problem rather than to learn accurate driver behaviours, using a unique feature associated with each link can better suit our purposes. For instance, with a unique link identification (link ID) as a feature, the feature expectation matching between the agent' and expert's trajectories can directly lead to the matching of the state visitation frequency for an individual link, which helps the agent replicate the link visitation patterns in observed trajectory and traffic volume data. As such, we propose the use of unique link IDs as feature vectors for $\mathbf{f}_s^{(1)}$ and $\mathbf{f}_s^{(2)}$ as follows:

- The state feature vector $\mathbf{f}_s^{(1)} \in \mathbb{R}^{k_1}$ is designed to convey information from the vehicle trajectory data (the distribution of state visitation counts over the state set, S). We define $\mathbf{f}_s^{(1)}$ as a k_1-dimensional binary vector, where k_1 represents the number of links in the road network, i.e., the number of states in S.A one-hot encoding is used to represent the k_1 different links in the network, where the i^{th} element in $\mathbf{f}_s^{(1)}$ is set to 1 and all other elements are set to 0 to represent the i^{th} link.
- The state feature vector, $\mathbf{f}_s^{(2)} \in \mathbb{R}^{k_2}$, is designed to convey information from the traffic volume data (the distribution of state visitation counts over the detector state subset, S_v). We define $\mathbf{f}_s^{(2)}$ as a k_2-dimensional binary vector, where k_2 represents the number of detector links in the road network, i.e., the number of states in S_v. For a detector link ($s \in S_v$), a one-hot encoding is used, where the i^{th} element in $\mathbf{f}_s^{(2)}$ is set to 1 and all other elements are set to 0 to represent the i^{th} detector link among the k_2 detectors. For a non-detector link ($s \in S \backslash S_v$), we simply assign a k_2-dimensional zero vector, where all elements in $\mathbf{f}_s^{(2)}$ are 0.

Based on the state feature vectors defined above, a state-action path τ is characterized by a path feature vector $\mathbf{f}_\tau \in \mathbb{R}^k$ ($k = k_1 + k_2$) by concatenating the following two path feature vectors:

$$\mathbf{f}_\tau = \left[\mathbf{f}_\tau^{(1)^T},\ \mathbf{f}_\tau^{(2)^T}\right]^T \tag{7}$$

where $\mathbf{f}_\tau^{(1)} = \sum_{s\in\tau} \mathbf{f}_s^{(1)}$ becomes a k_1-dimensional binary vector indicating which links τ (or its corresponding vehicle trajectory) passes through and $\mathbf{f}_\tau^{(2)} = \sum_{s\in\tau} \mathbf{f}_s^{(2)}$ becomes a k_2-dimensional binary vector indicating which detectors τ passes through. If we calculate these path feature vectors for all state-action paths in some demonstrated path set T_e, then total path feature vector $\sum_{\tau\in T_e} \mathbf{f}_\tau^{(1)}$ tells us how many times each link is visited by the trajectories in T_e (i.e., the i^{th} element in $\sum_{\tau\in T_e} \mathbf{f}_\tau^{(1)}$ corresponds to the number of vehicle trajectories that pass the i^{th} link). Similarly, total path feature vector $\sum_{\tau\in T_e} \mathbf{f}_\tau^{(2)}$ tells us how many times each detector is visited by the trajectories in T_e (i.e., the i^{th} element in $\sum_{\tau\in T_e} \mathbf{f}_\tau^{(2)}$ corresponds to the number of vehicle trajectories that pass the i^{th} detector). It is important to note that the latter leads us to obtain the following relationship:

$$\sum_{\tau\in T_e} \mathbf{f}_\tau^{(2)} = \sum_{s\in S_v} v_s \mathbf{f}_s^{(2)} \tag{8}$$

where v_s is the traffic volume on detector link s produced by the trajectories in T_e, indicating that the sum of path feature vectors can be expressed in terms of link traffic volume observations. This relationship plays a key role in allowing traffic volume data to be used as expert demonstrations to guide the learning of a policy in the proposed IRL-F.

In the road-network MDP, the path reward is calculated the same way as in MaxEnt IRL (see Eq. (2)), where the state reward value that is linear to the state feature vector, parametrized by unknown reward weights $\theta \in \mathbb{R}^{k_1+\ k_2}$. The path probabilities are defined the same way as in MaxEnt IRL (see Eq. (3)).

IRL objective: In the road-network MDP, let T_P denote the ground-truth population trajectory set and M denote the number of trajectories in T_P. To solve the link flow estimation problem, the goal is to find a policy (vehicles' link-to-link transition decisions) that produces the population trajectory distribution by finding the optimal reward weights θ^* that maximize the likelihood of population trajectories, as shown in Eq. (9). The gradient can be computed the same way as in MaxEnt IRL, as shown in Eq. (10), where the first part can be viewed as expert feature expectations and the second part can be viewed as policy feature expectation.

$$\theta^* = \underset{\theta}{\operatorname{argmax}}\ L = \underset{\theta}{\operatorname{argmax}} \sum_{\tau\in T_P} \log P(\tau) \tag{9}$$

$$\nabla_\theta L = \frac{1}{M}\sum_{\tau\in T_P}\mathbf{f}_\tau - \sum_{s\in S} D_s\mathbf{f}_s = \mathbf{f}_{\text{expert}} - \mathbf{f}_{\text{policy}} \tag{10}$$

However, T_P and M are unknown to traffic modellers and, thus, we cannot directly calculate $\mathbf{f}_{\text{expert}}$. Instead, only the observed traffic volume data from a subset of links and the observed trajectory data from a subset of vehicles are available. Both types of observed data provide information about the population trajectory distribution to some extent. Based on our newly proposed state feature definitions, we express $\mathbf{f}_{\text{expert}}$ as the concatenation of two feature expectations, $\mathbf{f}^{(1)}_{\text{expert}}$ and $\mathbf{f}^{(2)}_{\text{expert}}$, to describe the expert's desired behaviour by leveraging the available trajectory and traffic volume data, respectively. The gradient can then be rewritten as follows:

$$\begin{aligned}\nabla_\theta L &= \left[\frac{\sum_{\tau\in T_P}\mathbf{f}^{(1)T}_\tau}{M}, \frac{\sum_{\tau\in T_P}\mathbf{f}^{(2)T}_\tau}{M}\right]^T - \left[\sum_{s\in S} D_s\mathbf{f}^{(1)T}_s, \sum_{s\in S} D_s\mathbf{f}^{(2)T}_s\right]^T \\ &= \left[\mathbf{f}^{(1)T}_{\text{expert}}, \mathbf{f}^{(2)T}_{\text{expert}}\right]^T - \left[\mathbf{f}^{(1)T}_{\text{policy}}, \mathbf{f}^{(2)T}_{\text{policy}}\right]^T\end{aligned} \tag{11}$$

We propose to use the observed vehicle trajectory data to approximate $\mathbf{f}^{(1)}_{\text{expert}}$ and use the observed traffic volume data to approximate $\mathbf{f}^{(2)}_{\text{expert}}$. Similarly, the policy feature expectation $\mathbf{f}_{\text{policy}}$ is represented as the concatenation of $\mathbf{f}^{(1)}_{\text{policy}}$ and $\mathbf{f}^{(2)}_{\text{policy}}$. The goal of the proposed IRL-F is to match these two policy feature expectations with those two approximated expert feature expectations.

Solution process: The optimal reward weights (θ^*) in IRL-F can be found using gradient-based optimization methods. The challenge is how to determine the gradient using only the observed traffic data and, specifically, how to approximate expert's feature expectations using the observed vehicle trajectories and traffic volume data.

Let T_{obs} denote the set of observed vehicle trajectories. Since the population trajectory set is not available, we propose to approximate $\mathbf{f}^{(1)}_{\text{expert}}$ using the expected feature expectation over the observed trajectories in T_{obs} as follows:

$$\mathbf{f}^{(1)}_{\text{expert}} = \frac{\sum_{\tau\in T_P}\mathbf{f}^{(1)}_\tau}{M} \approx \frac{\sum_{\tau\in T_{obs}}\mathbf{f}^{(1)}_\tau}{|T_{obs}|} \tag{12}$$

If the observed vehicle trajectories are representative of the population (i.e., the sampling rate is the same across the network and all paths with non-zero true flow has at least one observed trajectories), this approximation gives the true feature expectation $\mathbf{f}^{(1)}_{\text{expert}}$. Otherwise, they may deviate from the ground truth. In this paper, it is assumed that the observed vehicle trajectory distribution does not deviate dramatically from the population trajectory distribution. Therefore, such approximated feature expectations can still guide the agent to find a realistic policy in the road-network MDP, which produces the state visitation count distribution that is close to the distribution from the population trajectory set.

The second feature expectation $\mathbf{f}^{(2)}_{\text{expert}}$ is designed to use the information from link traffic volume data. Based on feature definition for $\mathbf{f}^{(2)}_s$ and Eq. (8), we obtain the relationship $\sum_{\tau\in T_P}\mathbf{f}^{(2)}_\tau = \sum_{s\in S_v} v_s\mathbf{f}^{(2)}_s$, where v_s is the traffic volume observed on detector link s under the population trajectories in T_P. $\mathbf{f}^{(2)}_{\text{expert}}$ is expressed using this relationship and further approximate $\mathbf{f}^{(2)}_{\text{expert}}$ by using an estimated number of the population trajectories ($\widehat{M}$) as follows:

$$\mathbf{f}^{(2)}_{\text{expert}} = \frac{\sum_{\tau\in T_P}\mathbf{f}^{(2)}_\tau}{M} = \frac{\sum_{s\in S_v} v_s\mathbf{f}^{(2)}_s}{M} \approx \frac{\sum_{s\in S_v} v_s\mathbf{f}^{(2)}_s}{\widehat{M}} \tag{13}$$

It is noted that the traffic volume data from loop detectors (v_s) capture all the vehicles passing the detectors (i.e., reflect the population trajectories) and, therefore, we only need to find a substitute for M to obtain $\mathbf{f}^{(2)}_{\text{expert}}$. Different methods have been introduced to infer population traffic flows from the observed data that capture a proportion of the population traffic. In our study, to facilitate the implementation of the IRL-F model, the controlled Least Absolute Deviation (cLAD) method [33] is chosen to find an estimated size of population trajectories ($\widehat{M}$) based on the observed data. Let P denote a set of origin-destination pairs indexed with p and S_v be a set of observed links indexed with s. In the cLAD method, two terms 'local capture rate' and 'system capture rate' are defined, where the local capture rate is the ratio of the observed number of trajectories t_s to the observed traffic volume v_s for each detector link $s \in S_v$ and the system capture rate, denoted by r, is the median of all local capture rates among the observed links (i.e., $r = \text{median}\left\{\frac{t_1}{v_1}, \frac{t_2}{v_2}, \ldots, \frac{t_{|S_v|}}{v_{|S_v|}}\right\}$). Note that in this paper, it is assumed that the difference in local capture rate is small so that the local capture rates do not deviate much from the median. The cLAD method determines the optimal scaling factors as follows:

$$\underset{x_p}{\text{minimize}} \quad \sum_{s\in S_v}|e_s - v_s| + \gamma\sum_{p\in P}\left(x_p\right)^2 \tag{14}$$

$$s.t. \quad e_s = \sum_{p\in P} t_{s,p}\alpha_p \quad \forall s \in S_v \tag{15}$$

$$\alpha_p = \frac{1}{r} + x_p \quad \forall p \in P \tag{16}$$

where e_s and v_s are the estimated and observed traffic volume on detector link $s \in S_v$, respectively; α_p is the scaling factor for the observed trajectories between origin-destination pair $p \in P$; and $t_{s,\,p}$ is the number of observed trajectories between origin-destination pair p while crossing detector link s. Constraint (16) shows that the scaling factor (α_p) is determined based on the system capture rate r and a free parameter x_p, which is used to penalize large deviations of such factor from $1/r$. Constraint (15) shows that the estimated traffic volume (e_s) is determined by multiplying the number of observed trajectories on detector link s with the scaling factor. The objective function (14) then aims to

TABLE 1. Algorithm I: Solving IRL-F.

Input the road-network MDP $M = (S, A, \mu_0, P_T, \gamma, H)$; the set of state-action paths from the observed trajectory data T_{obs}; the set of state visitation counts from the observed traffic volume data Q_{obs}; number of iterations T; learning rate α; convergence tolerance ϵ
Initialize random reward weights, $\theta^1 \in \mathbb{R}^{k_1+k_2}$
Compute feature expectation $\mathbf{f}_{\text{expert}} = \left[\mathbf{f}_{\text{expert}}^{(1)\ T}, \mathbf{f}_{\text{expert}}^{(2)\ T}\right]^T$
For $t = 1$ **to** T:
  Find a policy π^t based on current reward weights θ^t
  Compute state visitation frequencies D_s^t based on π^t
  Compute feature expectation $\mathbf{f}_{\text{policy}}^t = \left[\mathbf{f}_{\text{policy},}^{(1),t\ T}, \mathbf{f}_{\text{policy},}^{(2),t\ T}\right]^T$
  Compute gradient $\nabla L(\theta^t)$
  If $\nabla L(\theta^t) < \epsilon$ then
    return θ^t
  end if
  $\theta^{t+1} \leftarrow \theta^t + \alpha \nabla L(\theta^t)$
End For
Return θ^t

find the optimal parameter x_p that minimizes the difference between the traffic volume estimated based on the scaled trajectories (e_s) and the ground-truth traffic volume (v_s) over all detector links $s \in S_v$. The second term of Eq. (14) is added to minimize the statistical variance of scaling factors, where a hyperparameter γ is used to control the statistical bias. Higher values of γ can be used prevent overfitting.

The result from the optimization model above is the optimal parameter ($\widetilde{x}_p$) that determines the trajectory scaling factors for each OD pair. Let t_p denote the number of observed trajectories between OD pair p. The estimated number of the population trajectories ($\widehat{M}$) can then be obtained as follows:

$$\widehat{M} = \sum_{p \in P} \left(\frac{1}{r} + \widetilde{x}_p\right) t_p \tag{17}$$

With this estimated $\widehat{M}$, the expert feature expectation $\mathbf{f}_{\text{expert}}^{(2)}$ can be approximated using Eq. (13). The policy feature expectation $\mathbf{f}_{\text{policy}}$ can be obtained by concatenating $\mathbf{f}_{\text{policy}}^{(1)}$ and $\mathbf{f}_{\text{policy}}^{(2)}$ defined as follows:

$$\mathbf{f}_{\text{policy}}^{(1)} = \sum_{s \in S} D_s \mathbf{f}_s^{(1)} \tag{18}$$

$$\mathbf{f}_{\text{policy}}^{(2)} = \sum_{s \in S} D_s \mathbf{f}_s^{(2)} \tag{19}$$

where the state visitation frequency, D_s, for a given policy is estimated the same way as in MaxEnt IRL.

Finally, Table 1 shows the algorithm to solve IRL-F to find the optimal reward weights θ that produce the agent's policy feature expectation that best matches the expert's feature expectation using a gradient-based method, where the gradient is computed as the difference between the expert's and policy feature expectations obtained from Eqs. (11)–(19).

Link Flow Estimation: The output of IRL-F is a reward function in the road-network MDP parameterized by the optimal reward weights. With this reward function, an optimal policy can be recovered using any RL method (e.g., dynamic programming method). Once the policy is found, state-action paths can be sampled on the road-network MDP. Each path generated by the agent can be viewed as a synthetic vehicle trajectory in the road network. A set of generated synthetic trajectories from this optimal policy reflect possible underlying population trajectories that produce the observed traffic patterns and, thus, can be used to obtain the traffic volume on each link to solve the link estimation problem in the road network.

With this ability to generate synthetic population trajectories, the goal of the proposed generative framework is to find the optimal set of synthetic vehicle trajectories that produce the best link flow estimates. The quality of a generated synthetic trajectory set can be evaluated by comparing the estimated link flows to the observed volume data for the detector links. To find the optimal size of the synthetic trajectory set, the most straightforward way is to keep generating trajectory sets with different scales until the one that minimises the difference between the estimated and observed volumes for the detector links is found. However, this method is computationally expensive due to the large number of candidate synthetic trajectory sets that need to be generated and evaluated.

In this paper, we propose an alternative method that utilises the state visitation frequencies, D_s, $\forall s \in S$, calculated during the process of solving IRL-F (see Algorithm I). Let $\widetilde{D}_s$ denote an estimate for D_s obtained from the IRL-F model. Since $\widetilde{D}_s$ represents the probability of the agent visiting each state s based on the policy learned by IRL-F, by assuming a uniform scaling factor $\beta \in \mathbb{R}$ across the states, we can translate $\widetilde{D}_s$ into the estimated traffic volume on each link s, denoted by $\widetilde{v}_s$, through $\widetilde{v}_s = \beta \widetilde{D}_s$. We use the following simple equation to calculate the optimal scaling factor, β^*:

$$\beta^* = \frac{1}{|S_v|} \sum_{s \in S_v} \frac{v_s}{\widetilde{D}_s} \tag{20}$$

which is the average scaling factor across the detector links obtained by first computing the ratio of the actual observed traffic volume (v_s) to the estimated state visit frequency ($\widetilde{D}_s$) for each detector link $s \in S_v$ and taking the mean over the detector link set (S_v). The traffic volume on an unobserved link ($\widetilde{v}_s$, $\forall s \in S \backslash S_v$) are estimated with this scaling factor.

$$\widetilde{v}_s = \beta^* \times \widetilde{D}_s, \qquad \forall s \in S \backslash S_v \tag{21}$$

IV. EXPERIMENTS

The proposed generative modelling framework has been applied to solve the link flow estimation problem in different test networks. The performance of the proposed model under each test scenario is measured and compared using the Weighted Absolute Percentage Error (WAPE) and Mean Absolute Percentage Error (MAPE). For the set of unobserved links, denoted by S_u, i.e., $S_u = \{s | s \in S \backslash S_v\}$, the value of WAPE and MAPE can be calculated as follows, where

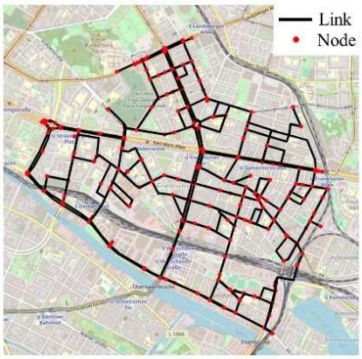

FIGURE 3. The Berlin-Friedrichshain network.

$\widetilde{v}_s$ is the estimated link flow on link s, v_s is its ground-truth link flow, and N is the number of states in set S_u.

$$WAPE = \frac{\sum_{s \in S_u} |\widetilde{v}_s - v_s|}{\sum_{s \in S_u} v_s} \tag{22}$$

$$MAPE = \frac{1}{N} \sum_{s \in S_u} \left| \frac{v_s - \widetilde{v}_s}{\widetilde{v}_s} \right| \tag{23}$$

Model Validation (Study network and data): To validate the proposed framework, we used the Berlin-Friedrichshain network, which is a road network data in Berlin available in a public repository [34], which include 224 nodes and 523 links, as shown in Figure 3. The dataset includes origin-destination trip matrices (hereafter referred to as OD demand) as well as the parameters of link performance functions. To obtain ground-truth link flows and trajectory data, we conducted User Equilibrium traffic assignment using the given OD demand and link performance functions using the Method of Successive Average. The traffic assignment results produced population trajectories covering a total of 1616 paths.

Traffic volume data input: Once the ground-truth link flows and population trajectories (path flows) are obtained, the set of links in the network is divided into two subsets: *observed links* (S_v) and *unobserved links* ($S \backslash S_v$). Among the total set of links, 157 links (about 30%) are selected as observed links, where sensors such as loop detectors are installed to collect traffic volume data. Note that the problem of selecting sensor locations is out of the research scope of this paper and, therefore, we consider one specific set of 157 links which, on its own, do not allow the full link flow estimation and therefore require additional sources of information such as trajectory data.

Trajectory data input and sampling rates: It is assumed that each of the 1616 paths has a sampling rate that is uniformly distributed within a specific range. In this experiment, we consider two sampling rate ranges: the range of 20 - 40% and the range of 10 - 30%, which are selected to reflect sparse trajectory datasets in real-world situations. The number of observed trajectories for each path is then calculated by multiplying the original path flow by the sampling rate drawn from a uniform distribution for that path. To create test scenarios where some paths in the road network are not covered by observed trajectories, 81 paths with non-zero true path flows (5% of the total path set) are selected to have zero observed trajectory. Additionally, some paths may have no observed trajectories if the computed number of trajectories after applying the sampling rate is less than 1. The observed trajectories sampled this way form the sample trajectory data input to our framework.

TABLE 2. Features describing general road characteristics.

Feature	Value
Road type	{freeway, arterial road, collector, local road}
Road length (m)	{0-100, 100-500, 500-1000, >1000}
Maximum travel speed (km/h)	{40, 50, 60, >60}

Feature definition: this paper proposes to use unique link IDs to define two state feature vectors, $\mathbf{f}_s^{(1)}$ and $\mathbf{f}_s^{(2)}$, associated with trajectory data and traffic volume data, respectively. For instance, consider a network with seven links. There are three observed trajectories: τ_1 traverses the 1st, 3rd, and 7th links; τ_2 traverses the 2nd, 3rd, and 5th links; τ_3 traverses the 2nd and 7th links. Then $\mathbf{f}_\tau^{(1)}$ becomes:

$$\mathbf{f}_\tau^{(1)} = [1, 2, 2, 0, 1, 0, 2]$$

Furthermore, in the network, loop detectors are installed in the 1st, 3rd and 5th links, then $\mathbf{f}_\tau^{(2)}$ becomes:

$$\mathbf{f}_\tau^{(2)} = [1, 2, 1]$$

However, in the original MaxEnt IRL, state feature vector is defined in terms of general characteristics describing each road segment. The MaxEnt IRL method was applied in the task of learning drivers' route choice behaviours from GPS trajectory data. While using such general road features is useful in learning and interpreting the agent' route choice behaviour as a function of road characteristics, it is not guaranteed that the feature expectation matching between the agent' and expert's trajectories will directly lead to the matching of the state visitation frequency for an individual link (e.g., there may exist two or more states share the same state feature vector). To evaluate the usefulness of the proposed feature definition over such a conventional feature definition, we test a scenario where $\mathbf{f}_s^{(1)}$ is defined in terms of general road characteristics such as *road type*, *road length*, and *maximum travel speed* for comparison. Table 2 shows the three features and their definitions considered in this study. All features are represented as categorical variables and the categories of each variable are defined based on the road segment data for the studied network obtained from the OpenStreetMap, which provides the link information based on real road networks. To construct $\mathbf{f}_s^{(1)}$, we express each feature as a binary vector using a one-hot encoding and concatenate the three binary vectors from the three features to form one state feature vector. For instance, $\mathbf{f}_s^{(1)}$ for a link

TABLE 3. Performance of the proposed model in test scenario set A.

Scenario	Descriptions	WAPE
A1	State Features: Unique Link IDs Trajectory Sampling Rate: [20%, 40%)	19.75%
A2	State Features: General Road Characteristics Trajectory Sampling Rate: [20%, 40%)	65.70%
A3	State Features: Unique Link IDs Trajectory Sampling Rate: [10%, 30%)	21.32%
A4	State Features: General Road Characteristics Trajectory Sampling Rate: [10%, 30%)	73.22%

with a road type of freeway, a road length of 300 m, and a maximum travel speed of 60 km/h is specified as:

$$\mathbf{f}_s^{(1)} = \left[\underbrace{1, 0, 0, 0}_{\text{type}}, \underbrace{0, 1, 0, 0}_{\text{length}}, \underbrace{0, 0, 1, 0}_{\text{max. speed}} \right]$$

For the example discussed above, the state feature vectors based on road characteristics are expressed as follows:

$$\begin{aligned}
\mathbf{f}_{s_1}^{(1)} &= [0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1] \\
\mathbf{f}_{s_2}^{(1)} &= [0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1] \\
\mathbf{f}_{s_3}^{(1)} &= [0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0] \\
\mathbf{f}_{s_4}^{(1)} &= [0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0] \\
\mathbf{f}_{s_5}^{(1)} &= [0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0] \\
\mathbf{f}_{s_6}^{(1)} &= [0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0] \\
\mathbf{f}_{s_7}^{(1)} &= [0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0]
\end{aligned}$$

With observed trajectory set $[\tau_1, \tau_2, \tau_3]$, $\mathbf{f}_\tau^{(1)}$ becomes:

$$\mathbf{f}_\tau^{(1)} = [0, 6, 2, 0, 5, 3, 0, 0, 2, 0, 3, 3]$$

It is noted that $\mathbf{f}_s^{(2)}$ is defined using unique link IDs for all test scenarios because $\mathbf{f}_s^{(2)}$ is specifically designed to capture traffic volume data from detector links, which is unique to our link flow estimation problem.

Test Scenarios: The goal of this validation study is to evaluate how closely the estimated link flows on the unobserved links agree with the ground-truth link flows. We consider the following four scenarios:

- Scenario-A1: the trajectory sampling rates range between 20% and 40%; state feature vector is defined using unique link IDs (proposed method).
- Scenario-A2: the trajectory sampling rates range between 20% and 40%; state feature vector is defined using the general road characteristics in Table 2.
- Scenario-A3: the trajectory sampling rates range between 10% and 30%; state feature vector is defined using unique link IDs (proposed method).
- Scenario-A4: the trajectory sampling rates range between 10% and 30%; state feature vector is defined using the general road characteristics in Table 2.

Results: The performance comparison is shown in Table 3, where the WAPE are much smaller in Scenarios A1 & A3 when compared to the results in Scenarios A2 & A4. In terms

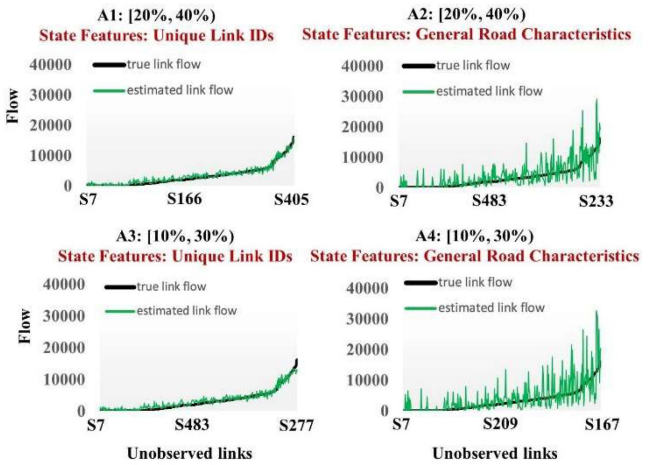

FIGURE 4. Link flow estimation results in the Berlin-Friedrichshain network.

of trajectory sampling rate, we observe that the decrease in the sampling rate from [20%, 40%) to [10%, 30%) tends to decrease the estimation accuracy. This indicates that the lower the sampling rate of the available trajectory data is, the less likely the data are to be representative of the population and, thus, the more challenging it is to recover the population link flow patterns from data. Figure 4 shows the graphical comparison of the estimated link flows and ground-truth link flows for the tested scenarios where the x-axis represents the IDs of the unobserved links sorted in ascending order by the ground-truth link flow values. The major finding based on the visual inspection is that the use of the unique link ID features (Scenarios A1 & A3) produces a much better agreement between the estimated and ground-truth link flows than the use of the general road characteristic features (Scenarios A2 & A4). This demonstrates the effectiveness of our proposed feature definition method in the context of the link flow estimation problem as it allows the feature expectation matching, which is the mechanism used in IRL-F, to directly learn the link flow patterns in trajectory data.

Model Comparison (Benchmark models): In this section, we compare the proposed generative modelling framework to two of the existing methods in the literature as benchmarks. The first method is proposed by Zhou and Mahmassani [35], referred to as ZM hereafter, which solves the OD demand estimation problem using traffic volume data and Automatic Vehicle Identification (AVI) data. The second method is proposed by Brunauer et al. [2], referred to as BHR hereafter, which solves link flow estimation using traffic volume data and probe vehicle trajectories.

Study network and parameter settings: The comparison analysis is conducted using the Nguyen-Dupuis network, which consists of 13 nodes and 38 links, as shown in Figure 5. We chose the Nguyen-Dupuis network for the comparison analysis, instead of the Berlin-Friedrichshain network, because the coverage of the ground-truth path flows in the Berlin-Friedrichshain network is relatively small thereby making it unsuitable for implementing the BHR

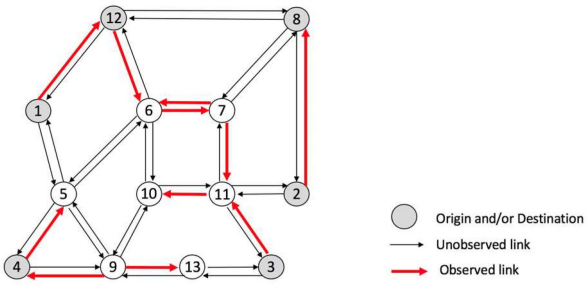

FIGURE 5. The Nguyen-Dupuis network.

TABLE 4. Performance comparison (MAPE) scenario set B.

	IRL-F	ZM model (P1)	ZM model (P2)	ZM model (P3)
5%-15%	19.82%	9.50%	16.72%	18.54%
5%-25%	24.83%	9.49%	16.46%	19.00%
5%-35%	30.95%	9.46%	16.73%	18.95%
15%-25%	6.31%	9.43%	16.46%	18.58%
15%-35%	15.14%	9.58%	16.72%	19.05%
15%-45%	14.30%	9.52%	16.46%	18.45%
25%-35%	3.76%	9.47%	16.73%	18.78%
25%-45%	9.11%	9.46%	16.46%	18.92%
25%-55%	12.79%	9.48%	16.73%	18.54%

model, which requires trajectory data with high spatial coverage. For the Nguyen-Dupuis network, we used the network descriptions and OD and flow data provided in [36]. Among 38 links, 11 links are selected as observed links with detectors (S_v) as shown in Figure 5, which is 29% of the links in the network. The ground-truth link flows from these observed links are used as traffic volume data input to the link estimation problem. For the trajectory data input, we apply the same sampling procedure used in the validation study above to sample observed trajectories from the ground-truth path flows. Among all paths with non-zero ground-truth path flows, we select 5% of the paths and assume that they have zero observed trajectory, in order to create situations where not all paths are covered by the observed trajectory data. For the rest of the paths, considering that nowadays there are multiple vehicle-tracking technologies that can achieve different trajectory sampling rates, it is assumed that the sampling rates fall between 5% and 50%. To simulate different scenarios where the observed trajectories have different levels of representativeness of the population trajectory set (i.e., the observed trajectory distributions have different levels of deviations from the true trajectory distribution), nine sampling rates intervals are designed: [5%, 15%), [5%, 25%), [5%, 35%), [15%, 25%), [15%, 35%), [15%, 45%), [25%, 35%), [25%, 45%), [25%, 55%). For the feature definition for the IRL-F, we use our proposed method based on the unique link IDs.

Test scenarios and performance measure: The ZM model assumes that the traffic simulator successfully captures the true traffic flow patterns so the observed traffic data (e.g., link count data and AVI data) are not too much deviated from the estimated traffic flows generated by the simulator. The BHR model assumes that the observed trajectories cover most of the link-to-link transitions in the study network. To allow fair comparisons, we design two sets of scenarios for each benchmark model:

- The first scenario set provides the conditions required by a given benchmark model (i.e., scenario set B for comparing to ZM model; scenario set C for comparing to BHR model).
- The second scenario set lifts these requirements to test the model performances in a more flexible and realistic environment (i.e., scenario set D for comparing to ZM model; scenario set E for comparing to BHR model).

Comparison with the ZM (Zhou-Mahmassani) model: Two sets of test scenarios are designed to compare our proposed model to the ZM model: Scenario set B, which assumes the traffic simulator used in the ZM model captures the true flow patterns, and scenario set C, which relaxes this assumption. Note that, besides the traffic volume data and the vehicle trajectory data, the ZM model further requires historical observations on OD trip matrices (hereafter referred to as prior knowledge of the OD demand). To investigate the performance of the models concerning different levels of prior knowledge, three different prior OD demand settings are designed as follows:

- P1: the prior demand for each OD pair equals its true demand value with small disturbances by adding an error sampled from a normal distribution with zero mean and a standard deviation of 20% of true demand value.
- P2: the prior demand for each OD pair equals its true demand value with moderate disturbances by adding an error sampled from a normal distribution zero mean and a standard deviation of 40% of true demand value.
- P3: the same prior demand is used for all OD pairs, where the common prior demand value is obtained by taking the average of true demands over all OD pairs.

Among the three settings, *P1* reflects the most accurate prior knowledge of the OD demand, whereas *P3* reflects the least accurate prior knowledge.

Scenario set B: It is assumed that the traffic simulator used to implement the ZM model conducts traffic assignment under static user equilibrium conditions. Meanwhile, the ground-truth traffic observations on the Nguyen-Dupuis network are obtained by conducting traffic assignment under the same user equilibrium conditions. By doing this, the traffic simulator can be considered to capture the true traffic flow pattern as the simulated traffic flow patterns would be consistent with the ground-truth traffic flow patterns.

Figure 6 and Table 4 show the comparison results between the proposed generative modelling framework and the ZM model tested under the nine trajectory sampling rates, in WAPE and MAPE respectively. When comparing across different scenarios, the estimation errors are smaller when

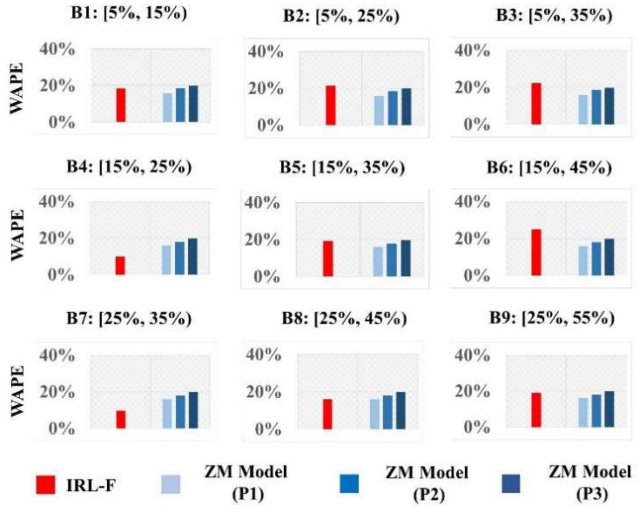

FIGURE 6. Performance comparison with ZM model for scenario set B.

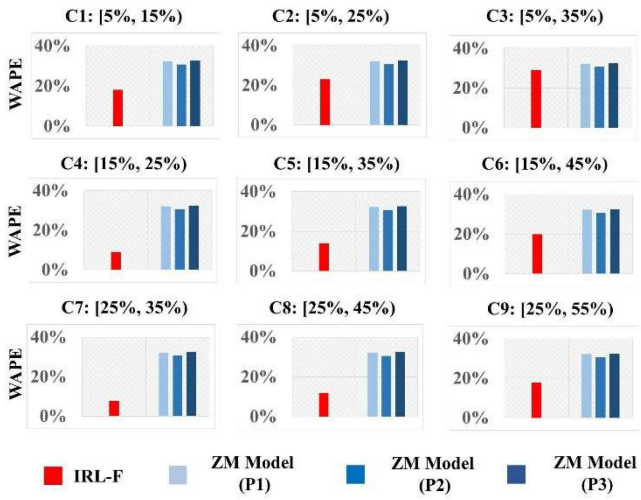

FIGURE 7. Performance comparison with ZM model for scenario set C.

trajectory sampling rates are higher (e.g., the errors are smaller in Scenario B7 than in Scenario B1) or have a smaller range (e.g., the errors are smaller in Scenario B7 than in Scenario B9), meaning that the observed data provides better distribution information about the population trajectories when sampling rates are higher and less heterogeneous. The ZM model shows a similar level of error across different trajectory sampling rates, indicating that it is less sensitive to the trajectory sampling rate. This is because the multi-objective optimisation technique used in the ZM model, which adjusts the objective function weights associated with different input sources, gives less weight to the trajectory data while giving more weight to other data sources such as link counts and prior OD demand in this particular experiment. Overall, in scenario set B where the conditions required by the ZM model is met, the ZM model achieves better performance when the prior knowledge on OD demand is assumed to be very accurate (i.e., demand setting P1) or when the trajectory sampling rate is low (i.e., minimum 5%). However, such advantage does not exist in all other scenarios, where IRL-F can achieve similar or better performance when compared to ZM model.

Scenario set C: The assumption that a traffic simulator can capture the true traffic flow pattern is relaxed. To create this test environment, we use two different path cost functions to perform traffic assignment in generating the ground-truth traffic flow and in implementing the ZM model, respectively, so that the simulator used in implementing the ZM model cannot accurately replicate the ground-truth traffic flow pattern. Specifically, we assume that the true traffic flows on the Nguyen-Dupuis network are from the traffic assignment results based on a path cost function, while a simulator used to implement the ZM model performs traffic assignment based on another slightly different path cost function. Using these settings, the performance of our model and the ZM model are again compared under the nine trajectory sampling rates (Scenarios C1–C9) and the resulting WAPE and MAPE

TABLE 5. Performance comparison (MAPE) scenario set C.

	IRL-F	ZM model (P1)	ZM model (P2)	ZM model (P3)
5%-15%	22.88%	69.13%	73.92%	83.38%
5%-25%	24.76%	72.36%	72.83%	83.46%
5%-35%	27.99%	69.86%	76.92%	80.54%
15%-25%	14.47%	68.27%	75.77%	81.97%
15%-35%	15.29%	70.94%	74.02%	83.12%
15%-45%	20.57%	70.70%	73.54%	82.85%
25%-35%	11.79%	68.67%	73.81%	84.91%
25%-45%	11.58%	69.98%	72.92%	81.39%
25%-55%	13.10%	71.73%	75.34%	87.17%

are shown in Figure 7 and Table 5 respectively. The ZM model produces much higher estimation errors compared to the results in scenario set B regardless of the changes of values in sampling rates. Our models outperform the ZM model in all scenarios with different trajectory sampling rates and prior OD demands. Specifically, the proposed method shows great advantage in scenario C4, C7 and C8, while in scenario C3 the proposed method performs similarly to the ZM model. It is rather difficult to achieve accurate estimates in link flows when the sampling rates of the observed trajectories have higher heterogeneity. Note that it is common that certain behavioural assumptions used in traffic assignment and simulation models may not fully reflect real-world behaviours. This highlights the advantage of the proposed data-driven approach over traditional simulation-based approaches in inferring the population travel patterns by effectively leveraging the information embedded in the available data without relying on prior knowledge or behavioural assumptions.

Comparison with the BHR (Brunauer-Henneberger-Rehrl) model: Two sets of test scenarios are designed to compare our proposed model to the BHR model. Scenario set D satisfies the trajectory data coverage assumed by the BHR model, while scenario set E relaxes this assumption.

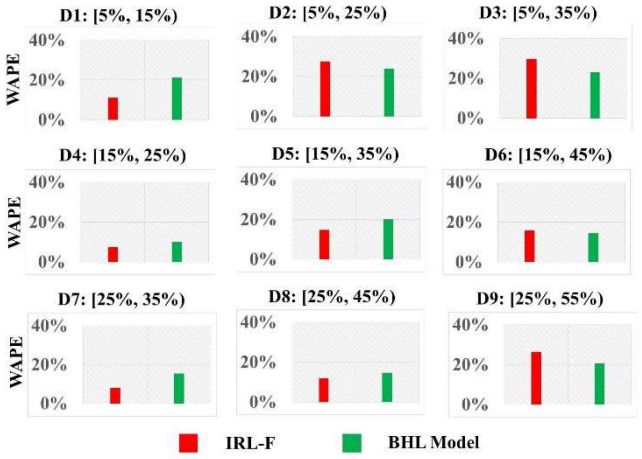

FIGURE 8. Performance comparison with BHR model for scenario set D.

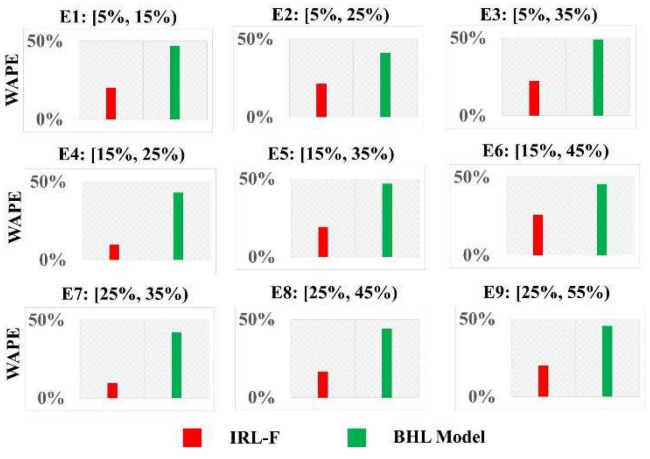

FIGURE 9. Performance comparison with BHR model for scenario set E.

Scenario set D: It is assumed that the observed vehicle trajectories cover most link-to-link transitions in the road network so that the propagation rules defined in the BHR model are valid. Traffic assignment is first conducted under user equilibrium conditions to generate the ground-truth path set. Initially, the set of paths assigned non-zero flows does not reach a high enough coverage of the link-to-link transitions, as suggested by BHR method. To ensure fair comparisons, some unused paths are selected to be assigned with non-zero path flows. The adjusted observed trajectories set covers about 80% of the link transitions on the network. The link flow estimation is performed under the nine different trajectory sampling rates. The comparison between our model and the BHR model is shown in WAPE values in Figure 8 and MAPE values in Table 6. The performance of IRL-F varies with the trajectory sampling rate, especially with the width of its range. The proposed method shows similar performance when compared to other scenario sets – with much lower error values in scenario D4, D7 and D8. By contrast, the BHR model shows fewer variations across different trajectory sampling rates. Overall, in scenario set D where the conditions required by the BHR model is met, IRL-F shows similar performance with the BHR model. Specifically, the differences in MAPE are less than 3%. when the trajectory sampling rate is low (i.e., minimum 5%). However, with higher trajectory sampling rate, IRL-F performs better than the BHR model, with a difference in MAPE up to 11% when the trajectory sampling rate is between 25% and 45%.

Scenario set E: We now relax the assumption on a high coverage of observed trajectory data to evaluate the models in more realistic conditions as the penetration rates of available vehicle trajectory data are still quite low in many cities. To create such test environments, we only use the paths obtained by solving the traffic assignment problem on the Nguyen-Dupuis network as the ground-truth path set without further adjustment applied in scenario set D. This path set covers about 60% of the link-to-link transitions on the network, which is lower than the level of coverage used for scenario set D (i.e., 80%). The performance comparison under the nine trajectory sampling rates (Scenarios E1–E9) is shown in Figure 9 in WAPE and Table 6 in MAPE.

TABLE 6. Performance comparison (MAPE) scenario set D and set E.

	Scenario Set D		Scenario Set E	
	IRL-F	BHR model	IRL-F	BHR model
5%-15%	18.23%	16.98%	22.88%	46.93%
5%-25%	23.12%	24.96%	24.76%	50.34%
5%-35%	25.37%	22.83%	27.99%	40.73%
15%-25%	11.82%	17.01%	14.47%	35.90%
15%-35%	15.40%	19.48%	15.29%	35.89%
15%-45%	18.09%	23.28%	20.57%	41.00%
25%-35%	4.74%	14.87%	11.79%	33.03%
25%-45%	7.25%	18.92%	11.58%	34.41%
25%-55%	8.28%	18.02%	13.10%	41.99%

The estimation errors have significantly increased in the BHR model compared to scenario set D, suggesting that the propagation rules used in this method depend heavily on the coverage of observed trajectories and fail to estimate the true link flows when many of the link-to-link transitions have no observed trajectories. On the other hand, our models produce similar performance to scenario set D, indicating a high level of robustness of the proposed generative approaches against the sparsity and low coverage of real-world trajectory data. Note that it is not guaranteed that higher sampling rates in the observed trajectories will necessarily lead to better estimation results. For example, the error value in Scenario-E6 is larger than that in Scenario-E3. The main reason is that a higher sampling rate does not guarantee that observed vehicle trajectory distribution is more similar to the population trajectory distribution, although it is true in most cases. Given two test scenarios with the same set of observed traffic volume data, the scenario where the observed vehicle trajectory distribution deviates less from the

population trajectory distribution is more likely to perform better than the other test scenario. In sum, in scenario set E where the requirements by the BHR model is lifted, IRL-F achieves better performance with difference in MAPE up to 28% when the trajectory sampling rate is between 25% and 55%. This highlights the advantage of the proposed method over the existing spatial imputation method when modellers only have access to sparse observed trajectory data. Missing link flows can be estimated with the proposed method without strong assumptions on the coverage and sampling rate of the trajectory data.

V. CONCLUSION

This paper proposes a novel data-driven approach to solve the link flow estimation problem with limited traffic volume data and sparse vehicle trajectory data. The main idea is to learn vehicle movement patterns from the available data and generate synthetic population vehicle trajectories to estimate link flows on unobserved links. We develop a formal mathematical framework based on MDP and RL-based solution approach, namely IRL-F. Our generative modelling framework was validated using a real road network in Berlin, producing reasonable estimation results on test scenarios with different data availability settings. When compared to the two benchmark methods from the literature, the proposed method shows a considerable advantage under more realistic scenarios, where behavioural assumptions about drivers are not met or the network coverage of the trajectory data are low. This highlights our contribution to the link flow estimation literature: the proposed IRL-F method can be used to specifically deal with challenging scenarios where the modellers only have limited traffic volume data and sparse vehicle trajectory data, but no prior knowledge about the travellers' demand or route choice behaviours.

While this study focuses on the link flow estimation problem, the application of our framework is not limited to this problem. Our framework can be viewed as an attempt to develop a more general synthetic trajectory generator using inverse reinforcement learning methods, which is capable of generating realistic trajectories that would have caused the observed traffic counts and sample trajectory data. In the future, more efforts can be made to improve the accuracy of generated trajectory dataset. Such a trajectory generator would have many applications in broader urban mobility studies including data augmentation, privacy protection, and mobility prediction.

In future work, we plan to consider a more sophisticated method to determine the scaling factor to scale up generated trajectories to obtain the population trajectories. The computational efficiency is also an important issue, which we plan to improve by adopting more efficient RL algorithms or advanced deep RL/IRL architectures. For applications beyond the link flow estimation problem, the model evaluation should consider more diverse metrics to assess the quality of individual-generated trajectories and their representativeness of the true population trajectories.

REFERENCES

[1] A. Abadi, T. Rajabioun, and P. A. Ioannou, "Traffic flow prediction for road transportation networks with limited traffic data," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 653–662, Apr. 2015.

[2] R. Brunauer, S. Henneberger, and K. Rehrl, "Network-wide link flow estimation through probe vehicle data supported count propagation," in *Proc. IEEE Conf. Intell. Transp. Syst.*, 2017, pp. 1–8.

[3] R. Lederman and L. Wynter, "Real-time traffic estimation using data expansion," *Transp. Res. B, Methodol.*, vol. 45, no. 7, pp. 1062–1079, 2011.

[4] T. P. Van Oijen, W. Daamen, and S. P. Hoogendoorn, "Estimation of a recursive link-based logit model and link flows in a sensor equipped network," *Transp. Res. B, Methodol.*, vol. 140, pp. 262–281, Oct. 2020.

[5] M. S. Ahmed and A. R. Cook, "Analysis of freeway traffic time-series data by using Box-Jenkins techniques," *Transp. Res. Rec.*, no. 722,ă pp. 1–9, 1979.

[6] R. Panda and X. V. Nguyen, "Large scale real-time traffic flow prediction using SCATS volume data," M.S. thesis, Dept. Comput. Inf. Syst., Univ. Melbourne, Melbourne, VIC, Australia, 2016.

[7] P. Rubin and M. Gentili, "An exact method for locating counting sensors in flow observability problems," *Transp. Res. C, Emerg. Technol.*, vol. 123, Feb. 2021, Art. no. 102855.

[8] M. Ng, "Synergistic sensor location for link flow inference without path enumeration: A node-based approach," *Transp. Res. B, Methodol.*, vol. 46, no. 6, pp. 781–788, 2012.

[9] S. R. Hu, S. Peeta, and C. H. Chu, "Identification of vehicle sensor locations for link-based network traffic applications," *Transp. Res. B, Methodol.*, vol. 43, nos. 8–9, pp. 873–894, 2009.

[10] D. R. Morrison and S. E. Martonosi, "Characteristics of optimal solutions to the sensor location problem," *Ann. Oper. Res.*, vol. 226, no. 1, pp. 463–478, 2015.

[11] L. Qu, L. Li, Y. Zhang, and J. Hu, "PPCA-based missing data imputation for traffic flow volume: A systematical approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 3, pp. 512–522, Sep. 2009.

[12] B. Ran, H. Tan, J. Feng, Y. Liu, and W. Wang, "Traffic speed data imputation method based on tensor completion," *Comput. Intell. Neurosci.*, vol. 2015, 2015, Art. no. 364089.

[13] X. Ma, S. Luan, C. Ding, H. Liu, and Y. Wang, "Spatial interpolation of missing annual average daily traffic data using copula-based model," *IEEE Intell. Transp. Syst. Mag.*, vol. 11, no. 3, pp. 158–170, Jun. 2019.

[14] Y. Vardi, "Network tomography: Estimating source-destination traffic intensities from link data," *J. Amer. Stat. Assoc.*, vol. 91, no. 433, pp. 365–377, 1996.

[15] S. Dey, S. Winter, and M. Tomko, "Origin–destination flow estimation from link count data only," *Sensors*, vol. 20, no. 18, p. 5226, 2020.

[16] G. Michau et al., "A primal-dual algorithm for link dependent origin destination matrix estimation," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 3, no. 1, pp. 104–113, Mar. 2017, doi: 10.1109/TSIPN.2016.2623094.

[17] S. Carrese, E. Cipriani, L. Mannini, and M. Nigro, "Dynamic demand estimation and prediction for traffic urban networks adopting new data sources," *Transp. Res. C, Emerg. Technol.*, vol. 81, pp. 83–98, Aug. 2017, doi: 10.1016/j.trc.2017.05.013.

[18] K. Parry and M. L. Hazelton, "Estimation of origin-destination matrices from link counts and sporadic routing data," *Transp. Res. B, Methodol.*, vol. 46, no. 1, pp. 175–188, 2012, doi: 10.1016/j.trb.2011.09.009.

[19] W. Ma and Z. Qian, "Estimating multi-year 24/7 origin-destination demand using high-granular multi-source traffic data," *Transp. Res. C, Emerg. Technol.*, vol. 96, pp. 96–121, Nov. 2018, doi: 10.1016/J.TRC.2018.09.002.

[20] M. Gallo and G. De Luca, "Spatial extension of road traffic sensor data with artificial neural networks," *Sensors*, vol. 18, no. 8, p. 2640, 2018.

[21] R. S. Sutton and A. G. Barto, *Reinforcement Learning—An Introduction*. Cambridge, MA, USA: MIT Press, 2018.
[22] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning," in *Proc. 23rd AAAI Conf. Artif. Intell.*, Jul. 2008, pp. 1433–1438,
[23] Z. Zhang, M. Li, X. Lin, and Y. Wang, "Network-wide traffic flow estimation with insufficient volume detection and crowdsourcing data," *Transp. Res. C, Emerg. Technol.*, vol. 121, Dec. 2020, Art. no. 102870.
[24] W. Ma, J. Yuan, K. An, and C. Yu, "Route flow estimation based on the fusion of probe vehicle trajectory and automated vehicle identification data," *Transp. Res. C, Emerg. Technol.*, vol. 144, Nov. 2022, Art. no. 103907.
[25] S. Vogt, W. Fourati, T. Schendzielorz, and B. Friedrich, "Estimation of origin-destination matrices by fusing detector data and floating car data," *Transp. Res. Procedia*, vol. 37, pp. 473–480, Jan. 2019, doi: 10.1016/j.trpro.2018.12.216.
[26] V. Bindschaedler and R. Shokri, "Synthesizing plausible privacy-preserving location traces," in *Proc. IEEE Symp. Security Privacy*, 2016, pp. 546–563, doi: 10.1109/SP.2016.39.
[27] M. Yin, M. Sheehan, S. Feygin, J.-F. Paiement, and A. Pozdnoukhov, "A generative model of urban activities from cellular data," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 6, pp. 1682–1696, Jun. 2018, doi: 10.1109/TITS.2017.2695438.
[28] Y. Zhang, S. Wang, B. Chen, J. Cao, and Z. Huang, "TrafficGAN: Network-scale deep traffic prediction with generative adversarial nets," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 1, pp. 219–230, Jan. 2021.
[29] C. Chen, Y. Cao, K. Tang, and K. Li, "Dynamic path flow estimation using automatic vehicle identification and probe vehicle trajectory data: A 3D convolutional neural network model," *J. Adv. Transp.*, vol. 2021, Feb. 2021, Art. no. 8877138.
[30] C.-J. Hoel, K. Wolff, and L. Laine, "Automated speed and lane change decision making using deep reinforcement learning," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, 2018, pp. 2148–2155.
[31] J. Choi and K.-E. Kim, "Hierarchical Bayesian inverse reinforcement learning," *IEEE Trans. Cybern.*, vol. 45, no. 4, pp. 793–805, Apr. 2015.
[32] M. Pan et al., "Dissecting the learning curve of taxi drivers: A data-driven approach," in *Proc. SIAM Int. Conf. Data Min.*, 2019, pp. 783–791.
[33] S. Miller, Z. Vander Laan, and N. Marković, "Scaling GPS trajectories to match point traffic counts: A convex programming approach and Utah case study," *Transp. Res. E, Logist. Transp. Rev.*, vol. 143, Nov. 2020, Art. no. 102105.
[34] "Transportation networks for research core team." Transportation Networks for Research. Accessed: Dec. 1, 2020. [Online]. Available: https://github.com/bstabler/TransportationNetworks
[35] X. Zhou and H. S. Mahmassani, "Dynamic origin-destination demand estimation using automatic vehicle identification data," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 1, pp. 105–114, Mar. 2006, doi: 10.1109/TITS.2006.869629.
[36] E. Castillo, J. M. Menéndez, and P. Jiménez, "Trip matrix and path flow reconstruction and estimation based on plate scanning and link observations," *Transp. Res. B, Methodol.*, vol. 42, no. 5, pp. 455–481, 2008, doi: 10.1016/j.trb.2007.09.004.