A Hybrid Reinforcement Learning Based Method for Generating Privacy-Preserving Trajectories in Low-Density Traffic Environments

Zhixiang Zhang, Wai-Choong Wong and Biplab Sikdar
Department of Electrical and Computer Engineering,
National University of Singapore,
Singapore 119077
Email: e0320869@u.nus.edu, wong_lawrence@nus.edu.sg, bsikdar@nus.edu.sg

***Abstract*—Intelligent Transportation Systems (ITS) optimize road network capacity, monitor traffic flow, and enhance overall road safety by analyzing real-time trajectory data. However, the utilization of such data raises privacy concerns, enabling potential attackers to gain insights into users' real-time activities and personal information. Furthermore, existing privacy preservation methods have multiple limitations, particularly in low-traffic density environments. To address these issues, this paper presents a novel approach for generating realistic trajectories that evade tracking. Existing trajectory generation mechanisms are coarse-grained and cannot adequately preserve the quality of location-based services while safeguarding individual privacy. To overcome this limitation, we first use differential privacy to determine a location near the actual destination and employ a path search algorithm to extract relevant road information. Subsequently, by leveraging our hybrid reinforcement learning model, we generate trajectories leading to this fictitious point. The comparison conducted on real-world maps with other trajectory generation methods reveals its superior ability to preserve spatio-temporal features. Finally, we propose two approaches that use the generated trajectories to protect privacy, ensuring both individual privacy protection and the utility of data.**

***Index Terms*—VANETs, reinforcement learning, differential privacy**

I. Introduction

With the development of networks and sensors, the Internet of Things (IoT) [1] has gained significant momentum in various industries. In particular, Vehicular Ad-hoc Networks (VANETs) have enabled Intelligent Transportation Systems (ITS) to enhance traffic efficiency and road traffic planning. In such systems, vehicles are equipped with GPS receivers, sensors, and communication devices, which can collect and transmit real-time traffic information.

Figure 1 shows the four main players in VANETs: on-board units (OBUs), roadside units (RSUs), trusted authority (TA), and service providers (SPs). OBUs are installed in vehicles to record travel trajectories and share real-time data with others for safety applications. OBUs generally broadcast safety messages [2] that contain their trajectory data to neighboring OBUs and RSUs at regular intervals. RSUs, positioned along the roadside, serve as access points that verify the source and authenticity of data before transmitting it to the TA.

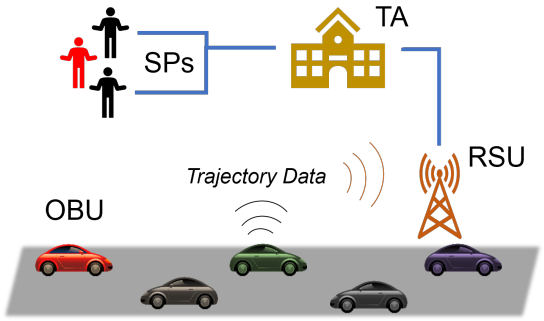


Fig. 1: The structure of VANETs.

Meanwhile, OBUs can also access traffic data through RSUs. The TA serves as the system manager, playing a pivotal role in ensuring the security of the ITS and acting as the backbone for establishing and maintaining trust across the network. Additionally, with powerful computing and communication ability, the TA integrates and analyses a large volume of real-time data which leads to the establishment of systems such as the Expressway Monitoring and Advisory System, Parking Guidance System, and Green Link Determining System [3]. These systems significantly improve traffic management and support informed decision-making. Furthermore, the TA may also enable the release of this data to other location-based SPs to enhance their services, such as offering real-time personalized route recommendations.

Privacy loss is a critical issue in VANETs. For instance, users need to disclose their current location to service providers to explore nearby parking options effectively. Similarly, for seamless navigation through intersections, users may have to broadcast real-time speed and location information to synchronize speeds with other vehicles and determine whether lane changes are necessary in advance. In both cases, without privacy preservation techniques, an adversary may listen on to the messages transmitted by a car and track the route followed by the car. In order to tackle this issue, extensive research has been conducted on exploring obfuscation and anonymity methods. Arif et al. [4] suggest adding noise along with real

ones by generating fake locations. Although obfuscation-based methods make it difficult to track vehicles by importing noise into real data, such fake information decreases the quality of the location-based services. The anonymity-based methods have received greater attention in the research community compared to obfuscation. References [5]–[7] propose the use of fictive identifiers, named pseudonyms, by vehicles when broadcasting safety messages. Those pseudonyms should be changed when vehicles enter mix-zones [8]. Furthermore, the authors of [9], [10] propose that vehicles can actively change their pseudonyms when they detect multiple vehicles around them. These methods assume that the attacker cannot distinguish between adjacent vehicles that change pseudonyms simultaneously within a small range. Nevertheless, they are not universally applicable to all scenarios. In low-density traffic environments, pseudonym-changing methods show unsatisfactory results due to the lack of sufficient neighbors to provide chances for pseudonym changes. The privacy level is close to 0 when the arrival rate of vehicles is less than 1 [11].

In order to address this challenge, this paper proposes the utilization of a trajectory generation model to produce realistic trajectories, thereby perplexing potential adversaries. To achieve this objective, we outline the following requirements for such privacy-preserving strategies. Firstly, it is essential to ensure that the generated trajectories keep the characteristics and utility of the overall dataset. Secondly, the model should be designed to generate appropriate trajectories tailored to different environmental conditions. Additionally, this method should balance privacy and service quality according to privacy requirements. Finally, in order to confuse attackers, the generated trajectories need to ensure sufficient accuracy and motion characteristics.

Existing trajectory generation methods include Markov Chain [12], Recurrent Neural Networks (RNNs) [13], [14], Long Short-Term Memory (LSTMs) [15], [16] and Generative Adversarial Networks (GANs) [17]. However, these models fail to fully meet the aforementioned requirements. The Markov Chains are trained and tested within the same scenario, thus requiring retraining when the map changes. Trajectories generated by RNNs, LSTMs, and GANs exhibit lower accuracy, and these methods are typically employed for generating large-scale trajectory datasets and are less suited for producing trajectories with specific start or endpoints. Therefore, this paper explores a different approach and proposes a trajectory generation model based on reinforcement learning and differential privacy. When protecting privacy, it automatically sets a false destination based on the preset differential privacy level by using Geo-indistinguishability. Then, with an agent trained by a hybrid reinforcement learning framework, it generates a plausible trajectory from the current location to this destination. Finally, users release this false trajectory with privacy preservation techniques. This method enables vehicles to independently protect their privacy in low-traffic density environments, even without the involvement of other vehicles. It also ensures that the original real data can only be accessed by local devices and the TA, thereby avoiding the risk of leakage and eavesdropping.

The paper makes the following key contributions:

- We develop a new trajectory generation framework that provides differential privacy and the ability to choose the destination of the trajectory through the privacy requirement.
- We develop a customized hybrid reinforcement learning model, that uses map information and is able to generate trajectories to any valid position with over 90% success rate in difficult and complex environments. Compared to existing methods, the trajectories generated by this model exhibit higher precision while preserving motion characteristics.
- We propose two privacy protection methods that use the generated trajectories: the substitution method and the pseudonym-changing method. The former allows for arbitrary destination changes as needed, while the latter enhances privacy protection coverage by 100% compared to traditional pseudonym modification methods.

The structure of the rest of this paper is outlined as follows. The next section discusses some related privacy protection strategies and trajectory generation methods. In Section III, we propose our novel reinforcement learning based trajectory generation framework. The validation tests and the comparison with other methods of this framework are presented in Section IV. Additionally, the methods for preserving privacy based on trajectory generation, as well as the experimental exploration of these approaches, are discussed in Section V. Finally, we conclude this paper and present possible future works in Section VI.

II. Related work

Various methods for protecting the privacy of vehicle trajectories have been proposed in literature. Encryption [18] effectively deals with external attacks but may not address internal eavesdropping. Privacy leaks can occur even from trusted entities and service providers. This paper discusses user-centric privacy protection methods and categorizes effective strategies into two types: obfuscation by adding noise and anonymity by changing pseudonyms.

A. Obfuscation Strategy

The addition of noise to the real dataset may be used to protect the privacy of users. Also, there are different obfuscation methods for preserving trajectory privacy.

(1) *Noisy Location*. Ardagna et al. [19] proposed that vehicles can add noise to their location data when broadcasting safety messages. This method allows vehicles to indicate a location several dozen meters away from their actual position. Additionally, Arif et al. [4] suggested that vehicles can generate dummy locations, making it difficult for adversaries to identify the real ones, as shown in Fig. 2. We categorize such simple noise addition methods as the "Noisy Location".

(2) *Differential Privacy (DP)* [20]. Differential privacy is a concept introduced to protect individual privacy and reduce the impact of noise addition on a dataset. This approach controls

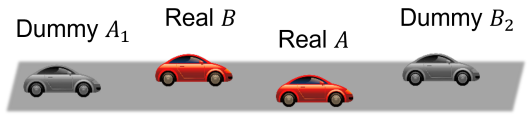


Fig. 2: Dummy locations of vehicles.

the type and magnitude of noise by managing a privacy budget. By doing so, it ensures that the probability distributions of query results derived from two neighboring datasets remain closely aligned. Feng et al. [21] explored the trade-off between the level of privacy and the quality of the location-based service.

B. Anonymity Strategy

In contrast to the obfuscation strategy, anonymity methods employ variable pseudonyms to conceal users' true identities, thereby avoiding location data distortion. During vehicular communication, maintaining a consistent identity makes users vulnerable to quick identification by attackers. Conversely, when the pseudonym changes, attackers need to make inferences based on the surrounding context, introducing greater uncertainty. Figure 3 shows vehicles B and C have changed their pseudonyms to D and E after Δt seconds. We have classified the conditions for triggering pseudonym changes into the following three categories.

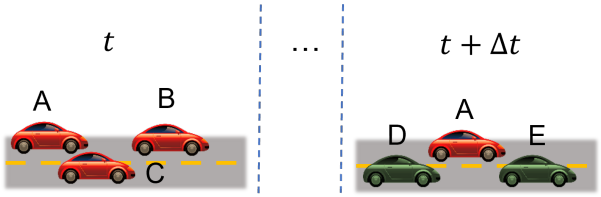


Fig. 3: Pseudonym-changing scenario.

(1) *Time-based strategy*: Time-based methods require vehicles to change pseudonyms according to a pre-set schedule. The radio silence technique is used before using new pseudonyms [22]. Based on this concept, Santos et al. [23] proposed that vehicles should change pseudonyms when there are malicious vehicles around. Wiedersheim et al. [24] suggested that the time duration of wireless silence should be randomly set, which can reduce the tracking success rate of attackers.

(2) *Zone-based strategy*: Different from time-based strategies, zone-based methods employ RSUs to predefine silence zones along the roads [8]. When vehicles enter these areas, they halt external communications and change their pseudonyms before leaving the zone. These zones are often strategically placed in high-density vehicle areas, such as intersections, parking lots, and traffic lights [25], [26]. Additionally, various studies [27], [28] have explored the optimal placement of these silence zones.

(3)*Context-based strategy*: In addition to the above two methods, context-based pseudonym modification strategies have also gained considerable attention due to their superior efficiency [9], [29]. Emara et al. [30] proposed that when a vehicle detects at least k vehicles around it, it will automatically create a movable silence zone with itself at the center and a radius of R meters. All vehicles within this zone will simultaneously modify their pseudonyms. Mdee et al. [31] provided the fundamentals of the encryption and authentication process, while Zhang et al. [32] enhanced this strategy by leveraging the inherent randomness during vehicle movement.

Regardless of the specific anonymous method used, all strategies rely on the presence of a sufficient number of other vehicles in the vicinity and collaboratively modify pseudonyms. This collective effort is essential to thwart attackers from discovering the correlation between old and new pseudonyms. Meanwhile, the obfuscation strategies rely on randomly generated dummy locations lacking inherent movement patterns. The dummy locations are sometimes positioned off legal roadways, making them vulnerable to easy detection by attackers and leading to a decline in privacy levels. Additionally, as these dummy locations are distributed around real positions, the absence of other vehicles nearby can still expose one's actual trajectory. Therefore, protecting user privacy in low-density traffic scenarios is an open and formidable challenge.

C. Trajectory generation

To address the low-density issues, a direct solution is to generate a set of sufficiently realistic trajectories. Most existing trajectory generation models, such as Markov chains [33], [34], are commonly employed for trajectory prediction. These models represent the map in a grid format, with each grid point corresponding to a state. By modeling the state transition function of the current map, they can generate possible trajectory predictions. In a related approach, Kong et al. [35] also divide the map into grids, but they introduce a traffic flux-based trajectory generation framework and utilize a simulation tool for trajectory generation.

Moreover, there are machine-learning-based trajectory generation methods [36]. These methods learn from existing trajectory datasets and achieve high accuracy predictions. Convolutional Neural Networks (CNNs) [37], [38] can not only extract road representation but also integrate trajectory patterns when getting high accuracy predictions. Additionally, Recurrent Neural Networks (RNNs) [13], [14] and Long Short-Term Memory networks (LSTMs) [15], [16] are commonly proposed as generators for trajectory data due to their ability to preserve mobility features.

In addition, GANs [39] as powerful generative models have also been utilized for generating trajectories. Kulkarni et al. [40] compared the performance of GANs and RNN models for generating traces. Jinmeng et al. [41] proposed an LSTM-TrajGAN, and replaced the backbone of the GAN with LSTM cells. Xingrui et al. [42] introduced a two-stage GAN (TSG) framework to improve the trajectory quality. They first generate the trajectory in grid form and then refine it inside each grid with map information. Moreover, Zhang

[43] et al. proposed the DP-TrajGAN, which initializes the privacy budget before generating the synthetic trajectory to preserve the trajectory privacy. Gang et al. [44] introduce a TrajSGAN which utilizes an attention-based generator to generate trajectories and employs a CNN-based discriminator in the GAN.

However, the above-mentioned methods have several drawbacks. Firstly, using discrete grid-based representations for continuous spatial states leads to a significant loss of accuracy in generated trajectories and also eliminates the motion characteristics of trajectories, making them easily identifiable by attackers. Secondly, when the scenario changes, these methods require the collection of new data, map modifications, and retraining, increasing the complexity of usage. Additionally, trajectory generation based on GAN models results in random distributions that do not align with the user's current location, thus rendering them unsuitable for protecting user privacy.

To address the issues present in traditional privacy protection methods and trajectory generation approaches, this paper proposes a novel solution. As illustrated in Figure 4, when a vehicle implements the proposed privacy protection strategy, it first determines a fictitious destination based on the current location and privacy settings. Subsequently, the A* algorithm is employed to plan and extract road information on the map. Then, utilizing an agent trained through reinforcement learning, high-quality trajectories are generated in continuous space. Finally, based on the generated trajectories, we introduce two distinct privacy protection strategies: trajectory substitution and pseudonym-changing. The details of these steps are provided in Section III and Subsection III.A, III.B, and III.C correspond to steps (2), (3), and (4) of Figure 4, respectively.

III. Trajectory Generation Framework

Assume that the vehicle with pseudonym i is driving in a low-density traffic environment and has access to the map of the area. It broadcasts safety messages to other members of VANETs in real time until it reaches its destination. The safety messages include time, location, and pseudonymous information. Unfortunately, a global attacker can easily obtain complete trajectory information about vehicle i, represented as $Tra_i = [p_0, p_1, p_2, ..., p_T]$, where $p_T = (x_T, y_T)$. This trajectory indicates that the vehicle arrives at the destination with coordinates (x_T, y_T) at time T. Furthermore, the broadcasting period of vehicles, using real-world taxi trajectory data as an example [45], has already reached 15 seconds, and it may become even shorter in the future.

A. The Privacy Settings of Destination

In order to prevent attackers from knowing the real position of vehicles, obfuscation methods add statistical noise to each position. Among these methods, those based on local differential privacy (LDP) have garnered significant attention. In the local setting, a random perturbation algorithm is deployed in each participant and the noise addition is independently performed to satisfy DP. It is formally defined as:

Definition 1 *(Local Differential Privacy)*: Given input pair $x, x' \in D$, a randomized algorithm $\mathcal{A}(\cdot)$ satisfies (ϵ, δ)-LDP, for all $O \subseteq Range(\mathcal{A})$, if and only if

$$\Pr\{\mathcal{A}(x) \in O\} \leq \exp(\epsilon) \times \Pr\{\mathcal{A}(x') \in O\} + \delta, \quad (1)$$

where $\epsilon \geq 0$ is called the privacy budget, and $0 \leq \delta \leq 1$. $Range(\mathcal{A})$ is the set of all possible outputs of algorithm $\mathcal{A}$. The randomized algorithm satisfies strict ϵ-LDP if $\delta = 0$. LDP ensures that given the output of the privacy algorithm, it is highly challenging to infer the specific input data point it corresponds to, providing plausible deniability for any two values. Specifically, the Laplace mechanism is most commonly used in location data to provide LDP with geo-indistinguishability [46].

Definition 2 *(Geo-Indistinguishability)*: A vehicle in location p_t enjoying ϵr-differential privacy is ϵ-geo-indistinguishable within radius r. It can be expressed as:

$$\frac{\Pr\{\mathcal{A}(p_t) \in O\}}{\Pr\{\mathcal{A}(p'_t) \in O\}} \leq e^{\epsilon d(p_t, p'_t)} \leq e^{\epsilon r}, \quad (2)$$

where $d(\cdot, \cdot)$ is the Euclidean metric. In essence, this definition ensures that the vehicle's geographical location information is safeguarded within the range $d(p_t, p'_t) \leq r$ with a privacy level of ϵr. To achieve ϵr-differential privacy, the vehicle reports a random point p'_t, which deviates from the actual location p_t by a multiplicative factor of at most $e^{-\epsilon d(p_t, p'_t)}$. To satisfy this requirement, the probability of generating a point should decrease exponentially as the distance from the true location p_t increases. The Laplace distribution with the following probability density function (PDF) precisely fulfills this condition:

$$\mathrm{D}_\epsilon(p) = \frac{\epsilon}{2} e^{-\epsilon |p_t - p|}. \quad (3)$$

Since the vehicles are positioned in a continuous two-dimensional plane, we employ polar coordinates to transform the noise into location data. The resulting PDF of the polar Laplacian noise is as follows:

$$\mathrm{D}_\epsilon(p_t)(p'_t) = \frac{\epsilon^2}{2\pi} e^{-\epsilon d(p_t, p'_t)}. \quad (4)$$

Because the location p can be replaced by (r, θ) in the polar coordinate system, the PDF becomes:

$$\mathrm{D}_\epsilon(r, \theta) = \frac{\epsilon^2}{2\pi} r e^{-\epsilon r}, \quad (5)$$

where $\frac{\epsilon^2}{2\pi}$ is the normalization factor. Since the two random variables, r and θ, are independent, we can generate random noise, r_ϵ and θ_ϵ, from the polar Laplacian noise distribution by:

$$r_\epsilon = -\frac{1}{\epsilon}(\mathcal{W}_{-1}(\frac{u-1}{e}) + 1), \ \theta \sim U[0, 2\pi], \quad (6)$$

where $\mathcal{W}_{-1}$ is the Lambert W function, and u is uniformly distributed between $[0, 1]$. Finally, the randomized algorithm gives the output:

$$\begin{cases} p'_x = p_x + r_\epsilon \cdot \cos(\theta) \\ p'_y = p_y + r_\epsilon \cdot \sin(\theta) \end{cases}. \quad (7)$$

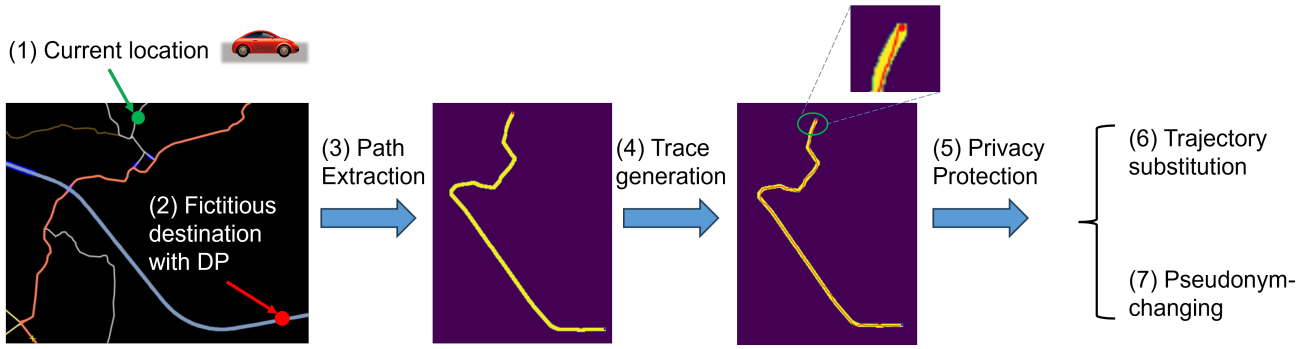


Fig. 4: The structure of our privacy protection method.

Adding noise to position data is a practical approach to protect individual location information. Nevertheless, in continuous trajectory data, attackers can execute denoising techniques using data from adjacent time points, thereby deducing precise positions. Moreover, the stochastic noise may cause the generated locations to deviate significantly from the valid road areas or exhibit considerable disparities of several hundred meters between consecutive time points, rendering them easily detectable by adversaries. Consequently, when addressing privacy protection for trajectory data, careful consideration of the overall trajectory's integrity is of paramount importance.

A viable solution involves the direct generation of a false trajectory that originates from the current location but deviates from the actual destination. This strategy confounds attackers and redirects them elsewhere. To ensure that the false destination appears plausible, Equation (7) can be adapted to facilitate the random selection of a point p' from the set of legitimate positions as:

$$S = \{p' | ||p' - p|| = r_\epsilon\}, \tag{8}$$

$$p' = S[i], i \sim U[1, |S|], \tag{9}$$

where S denotes the set of all valid positions that are at a distance of r_ϵ from the actual destination p, and p' is one of these positions. By adopting this methodology, the privacy of users is effectively protected, while also providing the advantage of adjusting privacy budgets, thereby mitigating any potential impact on the trajectory dataset.

B. Path Extraction from Road Map

Following the determination of the false destination, a path-planning algorithm is required to extract a road segment from the traffic map database that optimizes navigation from the current location to the specified endpoint. By integrating both the depth-first search and Dijkstra's algorithm concepts, the A* algorithm [47] demonstrates superior efficiency and optimality in finding the shortest path from a graph. It employs a heuristic function denoted as $h(k)$ that provides an estimate of the cost from any given node to the destination or target node. In our model, we calculate it as:

$$h(k) = d_1(k) + d_2(k), \tag{10}$$

Algorithm 1 Privacy Setting and Path Extraction

Input: Map $\mathbb{M}$; Current position p_0; Destination p_t; Privacy Budget ϵr

Output: The road map $\mathbb{M}_\epsilon$ to a fake destination.

Sample r_ϵ with ϵ-geo-indistinguishability:
$r_\epsilon = -\frac{1}{\epsilon}(\mathcal{W}_{-1}(\frac{u-1}{e}) + 1)$.
Randomly select a fake endpoint p'_t with a distance deviation of r_ϵ from point p_t.
Binarize the map and initialize the heuristic value matrix with:
$h(k) = d_1(k) + d_2(k)$.
closedList = empty set; openList = set containing p_0.
while p'_t not in closedList **do**
    currentNode = node in openList with the lowest h.
    Remove currentNode from openList.
    Add currentNode to closedList.
    for each neighborNode of currentNode **do**
        if neighborNode not in closedList **then**
            if neighborNode not in openList **then**
                Add neighborNode to openList
                Set currentNode as neighborNode's parent node
Backtrack the parent node of p'_t, and find the path.
Transfer the path into a map $\mathbb{M}_\epsilon$.
return $\mathbb{M}_\epsilon$.

where $d_1(k)$ is the distance from location k to the start and $d_2(k)$ is the distance to the destination.

Figure 5 presents an example of how to extract the path. First, we set the inaccessible locations on the map in the matrices d_1 and d_2 to infinity. Then, based on the positions of the starting point and the destination, we obtain complete matrices d_1 and d_2. After merging them, we obtain matrix h. When searching for the shortest path, we start from the starting point and iteratively choose the adjacent point with the minimum h value as the next step until reaching the destination. Finally, based on the recorded parent-child relationships during this search process, we extract the shortest path, as illustrated by the yellow portion in the diagram. Algorithm 1 shows the details of the privacy setting and path extraction process.

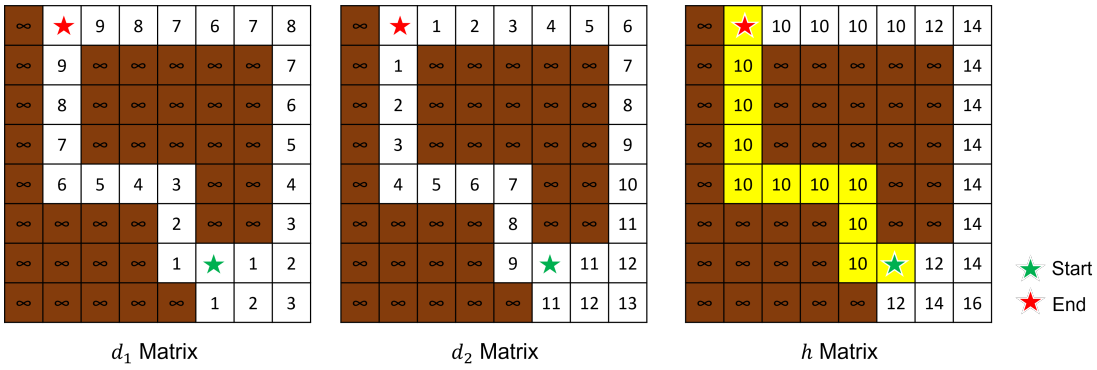


Fig. 5: An example of the shortest path extraction using the A* algorithm.

C. Reinforcement Learning for Trajectory Generation

Reinforcement Learning (RL) [48] is a dynamic learning paradigm designed to enable intelligent agents to make sequential decisions in an environment with the objective of maximizing cumulative rewards. It finds wide applications in domains such as robotics, autonomous control, and many other different fields [49]. The fundamental structure of an RL model is illustrated in Figure 6, where the agent interacts with the environment and has the mathematical framework of Markov Decision Processes (MDPs).

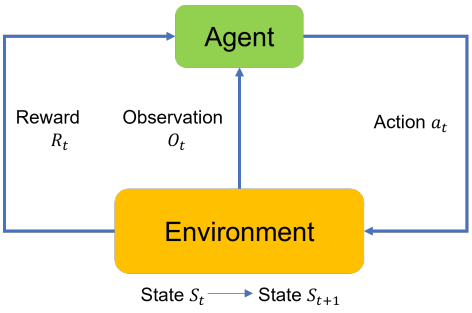


Fig. 6: The structure of reinforcement learning model.

The RL process begins with the agent observing the current state S_t of the environment. Based on this observation O_t, the agent selects an action a_t according to its current policy. The action is then executed in the environment, resulting in a transition to a new state S_{t+1}, and the agent receives a reward R_t based on the outcome of the action. The agent's goal is to learn an optimal policy that maximizes the expected cumulative reward over time.

There are two primary types of algorithms used to train RL agents: value-based and policy-based methods. Value-based RL algorithms [50] aim to learn the optimal value function, typically denoted as $Q(s, a)$, which represents the expected cumulative reward when taking action a in state s and following a specific policy thereafter. These algorithms iteratively update Q-values based on observed rewards and transitions during exploration. Value-based methods excel in discrete action spaces and environments where an accurate representation of the value function is feasible. On the other hand, policy-based RL algorithms directly learn the optimal policy, represented as $\pi(s)$ for each state s, which specifies the probability distribution over actions. These methods use gradient-based optimization to update policy parameters, seeking to maximize the expected cumulative reward. Policy-based approaches, such as Proximal Policy Optimization (PPO) [51] and trust region policy optimization (TRPO) [52] are suitable for continuous action spaces and complex environments.

In our application, we model the driver's behavior patterns using an agent. The agent generates two action variables every second, corresponding to the lateral and longitudinal accelerations. The lateral acceleration represents the steering wheel state, while the longitudinal acceleration represents the brake and throttle inputs, mimicking the real driving process. To mimic real driving behavior, we adopt a continuous action and state space, building upon the PPO framework, and designing and training the agent using the Actor-Critic architecture, which combines the advantages of value-based and policy-based methods. Figure 7 presents the outline of our proposed framework.

In the agent, we enable the actor model to produce average values of two Gaussian distributions with a predefined variance δ. The actual action for the current acceleration is sampled from the continuous space to ensure that the generated trajectory appears realistic. In the environment, we introduce two types of states, s_1^t and s_2^t. The first type comprises numerical states composed of variables such as velocity, acceleration, and others. The second type consists of road conditions displayed as two-dimensional images centered at the current position, with a radius of R to depict the surrounding environment.

States:

Prior to training vehicle control agents, selecting appropriate state variables is crucial. The vehicle's physical state comprises of position (p_x^t, p_y^t), velocity (v_x^t, v_y^t), and acceleration (a_x^t, a_y^t), with position as a scalar and velocity and acceleration as vectors. However, using position information as sensitive input during training can cause the model to overfit the specific training map, leading to potential failure when applied to different environments. To ensure model effectiveness across diverse scenarios, we exclude position information from the training states while retaining velocity and acceleration data.

Additionally, for different trajectory generation tasks, the

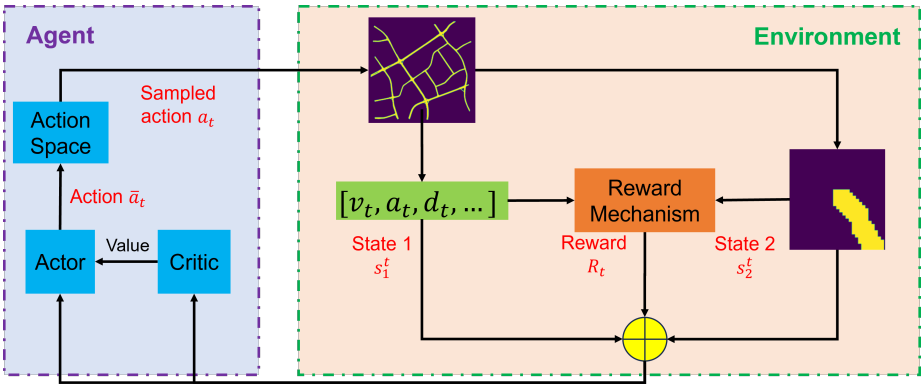


Fig. 7: The proposed hybrid reinforcement learning framework.

start and end points are not uniform. Due to the risk of model overfitting associated with scalar information, we include distance vectors from the current position to the start and end points as part of the state variables, which are calculated as:

$$\vec{d}_t = \begin{bmatrix} p_x^T - p_x^t \\ p_y^T - p_y^t \end{bmatrix}, \vec{g}_t = \begin{bmatrix} p_x^0 - p_x^t \\ p_y^0 - p_y^t \end{bmatrix}, \tag{11}$$

where $[p_x^0, p_y^0]$ is the coordinate of the start point, and $[p_x^T, p_y^T]$ denotes the fake destination.

In addition to the agent's internal state, the interaction between the agent and the environment is also crucial in determining the success of the task. Typically, vehicles travel in the middle of the road or lane to avoid crossing the road edges. Therefore, in s_1^t, we also observe the distance between the vehicle and road edges in eight directions. As illustrated in Figure 8, the vehicle radar search radius is r_s, and its distances to the road edge in eight directions are denoted as radar states, which are represented by $D^t = [l_0^t, l_1^t, ..., l_7^t]$. Wherein, the angle between each adjacent pair of directions is $45°$.

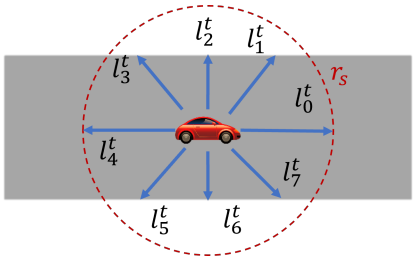


Fig. 8: The radar states of eight directions.

Furthermore, as we have the current velocity of the vehicle, it is easy to predict the location's coordinates for subsequent times and get the estimated radar state one second and two seconds later through the virtual environment. Therefore, we incorporate the simulation-predicted values of D^{t+1} and D^{t+2} for the next two seconds into the state S_1^t. This equips the agent with predictive capabilities, thereby enhancing learning efficiency. And State 1 becomes:

$$s_1^t = [v_x^t, v_y^t, a_x^t, a_y^t, \vec{d}_t, \vec{g}_t, D^t, D^{t+1}, D^{t+2}]. \tag{12}$$

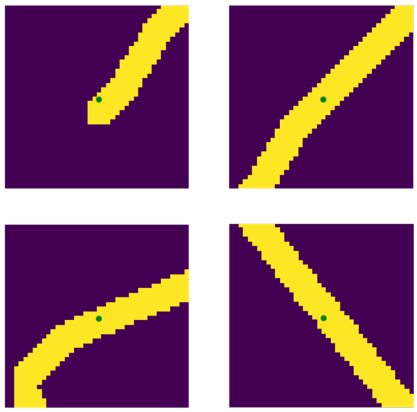


Fig. 9: The s_2^t in different time slices.

State 2, denoted by s_2^t, corresponds to an image depicting the neighboring view, with dimensions of $R_s \times R_s$. Figure 9 shows the s_2^t collected in different time slices during the trajectory generation process, and they are centered around the virtual agent's current location (p_x^t, p_y^t). This image encompasses road layout and length details, which are valuable for the agent's action determination. Finally, the overall state is $s^t = \{s_1^t, s_2^t\}$.

Reward Functions:

Throughout the training process, reward functions play a crucial role by steering the agent towards actions that facilitate the accomplishment of the ultimate task, while penalizing unproductive behaviors. Proper distribution of rewards even determines the success or failure of RL models. We have summarized the following essential guidelines from real-world driving scenarios:

1) Vehicle speed should comply with designated speed limits, avoiding both surpassing the maximum speed thresholds and excessively slow speeds that lead to extended travel durations.
2) Vehicles should actively avoid collisions with road edges, which can promptly lead to task failure.
3) The overarching objective for vehicles is to successfully reach their designated destinations.

To ensure that the trained agent meets the aforementioned

criteria, we have defined a target speed v_{tar}, a max speed v_{max}, and a minimum speed v_{min}, while simultaneously initializing the reward R_t as 0. The reward for the speed is:

$$R_t^1 = \begin{cases} -b_1, & v_t \leq v_{min} \\ -k_1 \dfrac{v_t - v_{max}}{v_{max}}, & v_{max} \leq v_t \\ 0, & \text{else} \end{cases}, \tag{13}$$

where k_1 and b_1 are scalars, which indicate the penalty intensity. As collisions are often caused by high velocity, we stipulate that the penalty intensity is positively correlated with the current velocity:

$$R_t^2 = \begin{cases} -b_2 - k_2 v_t^2, & \text{Collision} \\ 0, & \text{No collision} \end{cases}, \tag{14}$$

where b_2 is the basic penalty, and $k_2 v_t^2$ is the additional penalty.

Furthermore, the reward function also needs to encourage vehicles to travel in the middle of the road to enhance the likelihood of success. Therefore, we first find their relative position within the road based on the agent's radar state D^t and calculate the ratio of opposite-direction radar states as:

$$L_0 = \frac{l_0}{l_4}, L_1 = \frac{l_1}{l_5}, L_2 = \frac{l_2}{l_6}, L_3 = \frac{l_3}{l_7}. \tag{15}$$

Subsequently, if all ratios above are between 0.25 and 4, the positional state is determined to be in the middle of the road, otherwise, there is no reward. Thus, we compute the third reward as:

$$R_t^3 = \begin{cases} \dfrac{k_3 v_t}{v_{max}}, & 0.25 < All(L_0, L_1, L_2, L_3) < 4 \\ 0, & \text{else} \end{cases}, \tag{16}$$

where k_3 denotes the strengths of rewards. The reason for utilizing $\frac{v_t}{v_{max}}$ is to encourage the current velocity to approach the target velocity, thereby obtaining more rewards.

The most crucial role of the reward function is to encourage the agent to reach the destination. As shown in Figure 10(a), the vehicle travels from point P_A through point P_B to reach point P_C. During this process, the distance to the destination gradually decreases. It is reasonable at this point to use the reduced distance as the basis for the reward function. However, due to the irregular shape of the map, there might be a segment where the vehicle moves away from the destination, as depicted in Figure 10(b). In such a case, the direct distance from P_A to P_C might be shorter than the direct distance from P_B to P_C. As the vehicle moves from P_A to P_B, it might be moving away from point P_C. If the reward function is solely based on the reduced distance, it might cause the vehicle to stop at point P_A.

Hence, while evaluating rewards and penalties for each action of the agent, the validity of the direction of travel also needs to be considered. We define that, if at the current moment compared to the previous moment, the agent reduces

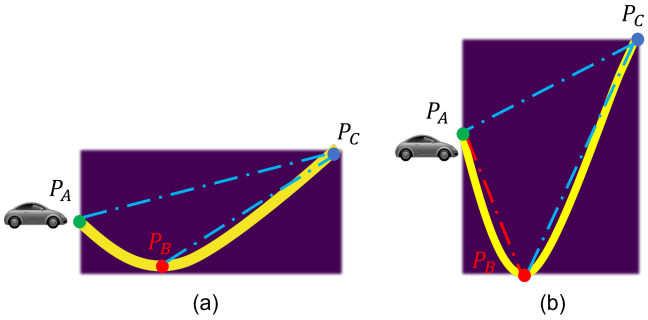


Fig. 10: The valid direction determination.

the distance to the destination or increases the distance to the starting point, then that direction is considered valid and receives a reward; otherwise, it incurs a penalty:

$$R_t^4 = \begin{cases} \dfrac{k_4 v_t}{v_{max}}, & ||\vec{d}_t| < ||\vec{d}_{t-1}|| \text{ or } ||\vec{g}_{t-1}|| < ||\vec{g}_t|| \\ -\dfrac{k_5 v_t}{v_{max}} - b_3, & \text{else} \end{cases}. \tag{17}$$

Lastly, the agent receives the greatest reward b_4 upon successfully reaching the destination:

$$R_t^5 = \begin{cases} b_4, & \text{reach destination} \\ 0, & \text{else} \end{cases}. \tag{18}$$

Let b_5 be the penalty due to the time consumed. The final reward that the agent obtains from the current state is:

$$R_t^c = R_t^1 + R_t^2 + R_t^3 + R_t^4 + R_t^5 - b_5. \tag{19}$$

In Equations (13) to (19), k_1 to k_5 represent scaling factors used in the computation of actual rewards and penalties for each corresponding behavior. Meanwhile, b_1 to b_5 are scalars representing the base rewards and penalties assigned to the corresponding behaviors. Notably, b_5 serves as the time penalty parameter, and in each reward calculation, it is subtracted. This implies that the agent should minimize the time taken to reach the destination if it seeks to maximize its overall reward. To determine the value of these scalars, we conducted multiple experiments and these are publicly available through the code link we have shared [1].

In addition, there is a point worth noting: the current actions and states actually impact all future moments. Rewards pertaining to future states should be considered as constituents of the present rewards. Consequently, to account for this aspect of influence, we introduce a discount factor β and calculate the final reward at the current moment as follows:

$$R_t = R_t^c + \sum_{i=1} \beta^i R_{t+i}. \tag{20}$$

Actor and Critic:

[1]https://github.com/zhixiangZHANG/Hybrid-Reinforcement-Learning-Based-Method-for-Generating-Privacy-Preserving-Trajectories

After the environment, state variables, and reward functions are established, consider a comprehensive state-action transition from $t = 1$ to $t = T$, to be denoted as $\tau = [s_1, a_1, s_2, a_2, ..., s_T, a_T]$, where $s_t = \{s_1^t, s_2^t\}$. The anticipated reward of the whole transition is expressed as:

$$\bar{R}_\gamma = \sum_\tau R(\tau) p_\gamma(\tau), \tag{21}$$

where $R(\tau)$ represents the reward associated with the sequence τ, and γ is the action policy represented by a neural network with parameters γ. In addition, $p_\gamma(\tau)$ is the probability of occurrence of the sequence τ under the current action policy. The application of the formula for conditional probability yields:

$$p_\gamma(\tau) = p(s_1) \prod_{t=1}^{T} p_\gamma(a_t|s_t) p(s_{t+1}|s_t, a_t). \tag{22}$$

In Equation (22), $p_\gamma(\tau)$ consists of two parts: $p(s_{t+1}|s_t, a_t)$ is the probability that the agent transitions from state s_t to state s_{t+1} by taking action a_t, which is determined by the environment, and $p_\gamma(a_t|s_t)$ is the probability that the agent takes action a_t based on policy γ. In a simple RL model with only an actor, the gradient of the reward function can be estimated through multiple samplings, as illustrated by the following expression:

$$\begin{aligned} \nabla \bar{R}_\gamma &= \sum_\tau R(\tau) \nabla p_\gamma(\tau) = \sum_\tau R(\tau) p_\gamma(\tau) \frac{\nabla p_\gamma(\tau)}{p_\gamma(\tau)} \\ &= E_{\tau \sim p_\gamma(\tau)}[R(\tau) \nabla log p_\gamma(\tau)] \\ &\approx \frac{1}{N} \sum_{n=1}^{N} \sum_{t=1}^{T_n} R(\tau^n) \nabla log p_\gamma(a_t^n|s_t^n) \end{aligned} \tag{23}$$

where N is the batch size, and T_n is the length of the trajectory for the n-th experiment.

As the rewards obtained by the agent are the cumulative results of multiple actions, the current action selection will impact the states at all future time steps. It implies that when there is a subtle change in our action policy, it could result in great fluctuations in the reward values. For instance, the agent's current acceleration behavior might lead to a collision with the roadside a few seconds later, causing a sharp drop in subsequent reward. However, such an outcome would not occur if no acceleration was applied. Consequently, the entire training process might become unstable due to the significant variance in reward values. To facilitate smoother convergence of the training process, we employ the agent architecture of A2C (Advantage Actor-Critic).

In the A2C network, the "Actor" component learns the policy, which is responsible for selecting actions given the current state. The "Critic" component learns the value function, providing estimates of the expected cumulative rewards from a given state. The key idea is using the advantage function to replace the original reward function in Equation (23):

$$\nabla \bar{R}_\gamma = E_{(s_t, a_t) \sim \gamma}[A^\gamma(s_t, a_t) \nabla log p_\gamma(a_t^n|s_t^n)], \tag{24}$$

where

$$A^\gamma(s_t, a_t) = R_t - V_t. \tag{25}$$

It is calculated as the difference between the estimated value and the actual rewards, which helps to indicate the quality of each action with respect to the current state.

Both the actor and the critic engage with the environment in distinct time intervals and take in two types of inputs: numerical variables and two-dimensional images. As a result, the agent's model is organized into two key components. One component centers on a Multilayer Perceptron (MLP) architecture, while the other relies on a Convolutional Neural Network (CNN) architecture. After undergoing separate processing, the latent variables derived from these inputs are combined and subsequently fed through fully connected layers to produce the ultimate output.

Due to the bounded nature of the actor's output, which represents the mean acceleration within a limited range, we apply the hyperbolic tangent (tanh) activation function in the final layer of the actor's architecture. Conversely, the critic's role entails assessing the evaluative value of the current state. Also, there are no predetermined upper or lower limits imposed on the critic's output. Consequently, the critic's output directly utilizes the linear result from the fully connected layer. The detailed architectural configurations of the actor and critic models introduced in our study can be observed in Figure 11 and Figure 12, respectively.

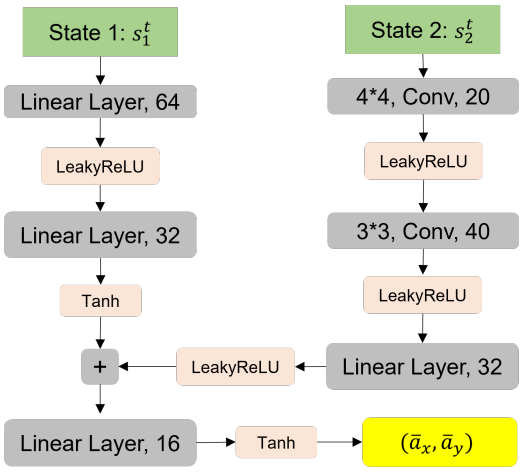


Fig. 11: The actor model.

Proximal Policy Optimization (PPO):

Apart from the advantage function, another approach for stable training involves limiting policy update steps. PPO centers on constraining update magnitudes and applying clipping during objective function updates. Additionally, PPO employs multiple mini-batches from sampled trajectories in each policy update to reduce variance, thus enhancing training stability. These strategies have propelled PPO's excellent performance across diverse reinforcement learning problems, making it a popular algorithm.

According to Importance Sampling, if $p(x)$ and $q(x)$ are two similar distributions, then:

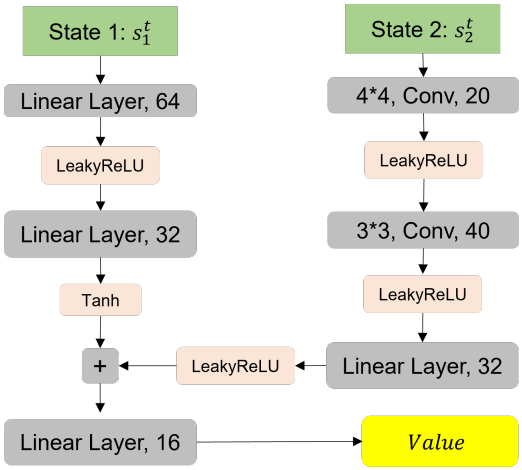


Fig. 12: The critic model.

$$E_{x\sim p}[u(x)] = E_{x\sim q}[u(x)\frac{p(x)}{q(x)}]. \tag{26}$$

Substituting it in Equation (24), we can get

$$\begin{aligned}\nabla \bar{R}_\gamma &= E_{(s_t,a_t)\sim\gamma}[A^\gamma(s_t,a_t)\nabla log p_\gamma(a_t^n|s_t^n)] \\ &= E_{(s_t,a_t)\sim\gamma'}[\frac{p_\gamma(s_t,a_t)}{p_{\gamma'}(s_t,a_t)}A^{\gamma'}\nabla log p_\gamma(a_t^n|s_t^n)] \\ &\approx E_{(s_t,a_t)\sim\gamma'}[\frac{\nabla p_\gamma(a_t|s_t)}{p_{\gamma'}(a_t|s_t)}A^{\gamma'}]\end{aligned} \tag{27}$$

where we denote γ' as the old policy and γ as the updated one. As γ cannot be very different from γ' in importance sampling, the 'clip' operation is used in the target function for PPO:

$$\begin{aligned}J_{PPO}^{\gamma'}(\gamma) = \sum_{(s_t,a_t)} \min\Big\{&\Big[\frac{p_\gamma(s_t,a_t)}{p_{\gamma'}(s_t,a_t)}A^{\gamma'}\Big], \\ &\Big[clip\Big(\frac{p_\gamma(s_t,a_t)}{p_{\gamma'}(s_t,a_t)}, 1-\alpha, 1+\alpha\Big)A^{\gamma'}\Big]\Big\},\end{aligned} \tag{28}$$

where the parameter α represents the upper and lower bounds of truncation.

Finally, we combine the objective functions of A2C and PPO to obtain the ultimate objective function as follows:

$$J_t^{\gamma'}(\gamma) = E_t[J_{PPO_t}^{\gamma'}(\gamma) - c_1 MSE(V_t, R_t) + c_2 H(\gamma_t)], \tag{29}$$

where, $J_{PPO_t}^{\gamma'}(\gamma)$ represents training the agent using the PPO method. The term $MSE(V_t, R_t)$ is intended for optimizing the critic. $H(\gamma_t)$ denotes the entropy of the agent, often utilized to enhance model training efficiency. The weights c_1 and c_2 are employed to balance these diverse optimization objectives and their significance throughout the training process.

IV. Trajectory Generation Experiments

In this section, we present the experiments to evaluate the proposed hybrid RL-based trajectory generation model.

A. The Training Experiment

The first experiment is conducted to investigate the impact of different reward components in the agent's training process. To establish a controlled environment for our experiments, a simplified road network was created based on the Manhattan mobility model, and it serves as the foundation for evaluating the trajectory generation capabilities of the proposed model. As shown in Figure 13, this road network has the size of 2 km $\times$ 2 km with 3 lanes on each edge.

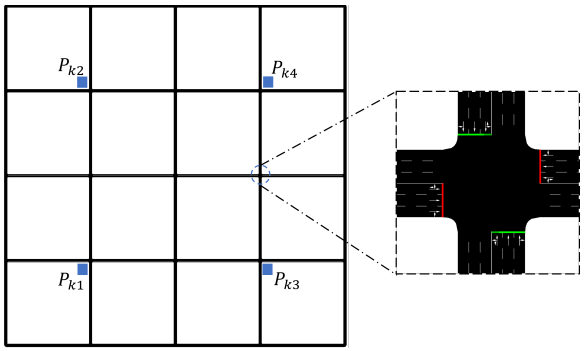


Fig. 13: The simplified Manhattan mobility model map.

To provide insights into the agent's ability throughout the training process, a comprehensive dataset is curated for both the training and testing phases. The training dataset consists of 15000 diverse scenarios, each characterized by randomly selected starting and ending points within the road network, while the testing set is composed of an additional 200 random scenarios.

Metrics

To assess the efficacy and the quality of the trajectory generation, we introduce three metrics for evaluation: **Success Rate**, **Average Speed**, and **Average Time-Spent**.

- **Success Rate:** The Success Rate, reflecting the proportion of successfully completed cases in the test set, serves as a key indicator of the model's capabilities; a value closer to 1 signifies a stronger generation capability of the agent.
- **Average Speed:** Although the spatial range of trajectories has been constrained through path extraction in Algorithm 1, the distance between each pair of adjacent points is also a metric for assessing the authenticity of a trajectory. This distance precisely corresponds to the velocity generated by the agent. So, we analyze the Average Speed during testing. In our experimental setup, $v_{max} = 18$ m/s, and this maximum speed serves as the baseline for comparison in Figures 14 (b) and 15 (b).
- **Average Time-Spent:** In addition, we should also consider the temporal features of trajectories. For example, a trajectory that moves in a circular motion at the target speed and eventually reaches the destination is not considered reasonable. Similarly, a normal trajectory does not exhibit forward movement one instant and then reverse movement the next instant. Furthermore, a normal driver does not engage in multiple collisions with the road edges. Directly assessing the frequency of these

behaviors in trajectories can be complex. However, these actions result in an increase in the time steps required for the virtual driver to reach the destination. Therefore, we choose the Average Time-Spent as another metric for evaluation.

Figures 14 and 15 illustrate the test performances of the agent when any one of R_1 to R_5 in Equation (19) is absent. Their x-labels indicate the number of scenarios that have been used for training. Figure 14 shows that the absence of R_2 or R_3 has minimal impact on the training outcomes. As training progresses, their results converge to nearly the same position as the training results from the complete combination (shown as the red line segment). However, in the case of R_1 being absent, there is a gap in the agent's success rate. Meanwhile, as depicted in Figure 14 (b), the absence of R_1 causes the agent to attempt virtual driving at speeds greater than the maximum velocity during the early stages of training. This is the reason behind the reduction in training success rate. Notably, R_1 corresponds to the limiting strategy we impose on velocity. Additionally, it is observed that without the constraints of R_1, the final converged velocity value is lower than our designated maximum velocity. This is attributed to the incorporation of the reward form v_t/v_{max} that is applied to each reward function.

From Figure 15 (a) and (c), it can be observed that when either of the rewards, R_4 or R_5, is missing, the probability of the agent successfully completing the test task within the given three hundred time steps is nearly zero. This highlights the distinct conditions associated with these two rewards, namely, direction determination and destination arrival determination, which play a decisive role in the agent's training outcome. This also implies that while setting the weights for rewards, R_4 and R_5 should hold a dominant position among all rewards. Otherwise, the gradient changes of R_1, R_2, and R_3 could overshadow R_4 and R_5, leading to training failures. Additionally, in Figure 15 (b), we can find that due to the presence of R4, the model exhibits better control over velocity during the early stages of training.

Based on the aforementioned result analysis, in conjunction with map features, our model, when applied to scenarios involving straight-line driving and turning at perpendicular angles, achieves a success rate of over 95% in single test instances after approximately three thousand training iterations. Ultimately, the success rate even reaches 100%. Furthermore, the imposed velocity constraints have been effectively managed. This experiment also reveals the importance of prioritizing reward allocation to R_1, R_4, and R_5 when designing the reward structure.

B. Real-world Scenario Experiment

To evaluate the compatibility of the proposed trajectory generation model across diverse complex scenarios, we utilize the road map within Portugal for both training and testing purposes. Comparative experiments are conducted using taxi trajectory data collected from 01/07/2013 to 30/06/2014 [45]. Additionally, we performed a comparative analysis of trajectory generation with other recent trajectory generation models, with a specific emphasis on data features.

As shown in Figure 16, the road map which is situated in a rural area is used for the test experiment. It is collected in the region of latitude: $[41.3777, 41.4239]$, longitude: $[-8.6837, -8.6044]$. We export only the road information from the OpenStreetMap platform.

Figure 17 shows a case of path extraction using Algorithm 1, where the white portions on the right side of the picture represent roads extracted. It can be observed that influenced by the geographical environment, the roads in this location exhibit winding paths with few linear roads. Therefore, the encountered challenges are also greater.

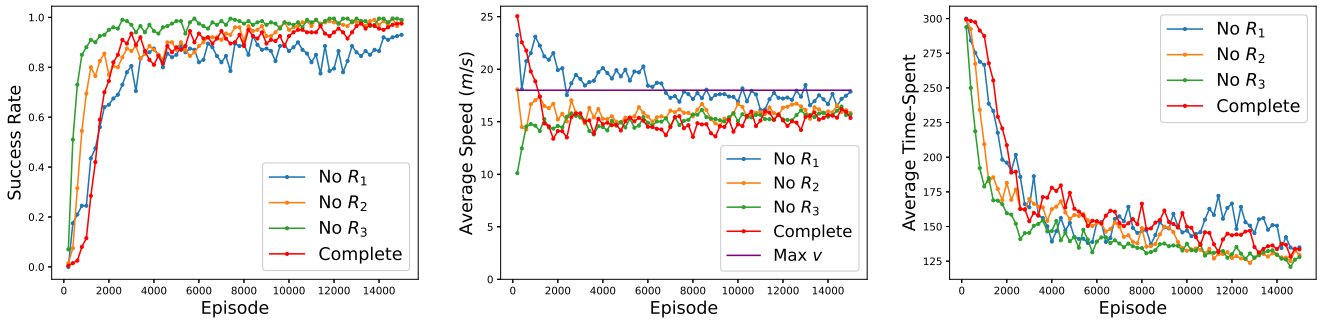
Figures 18 and 19 display the testing Success Rates and the Average Time-spent by the agent at various training stages. When compared with the results in Figure 14 (a) and (c), it can be observed that, although achieving the same success rate in complex road shapes demands a greater training cost, the agent trained through hybrid reinforcement learning can still proficiently accomplish trajectory generation tasks with over 90% success rate. In addition, due to the fewer straight roads, the intelligent agent requires a greater number of simulated time steps to complete driving tasks. Furthermore, Figure 20 indicates that our velocity control strategy has also achieved success.

To evaluate the quality of the generated outcomes, we employ the Jensen-Shannon Divergence (JSD) in this paper. JSD is commonly utilized for quantifying the degree of disparity between two distributions. The closer the probability distributions are, the smaller the value. Given the two probability distributions P and Q, it is depicted as:

$$\mathrm{JSD}(P \parallel Q) = \frac{1}{2}\mathbb{E}_P\left[\log \frac{P}{X}\right] + \frac{1}{2}\mathbb{E}_Q\left[\log \frac{Q}{X}\right] \tag{30}$$

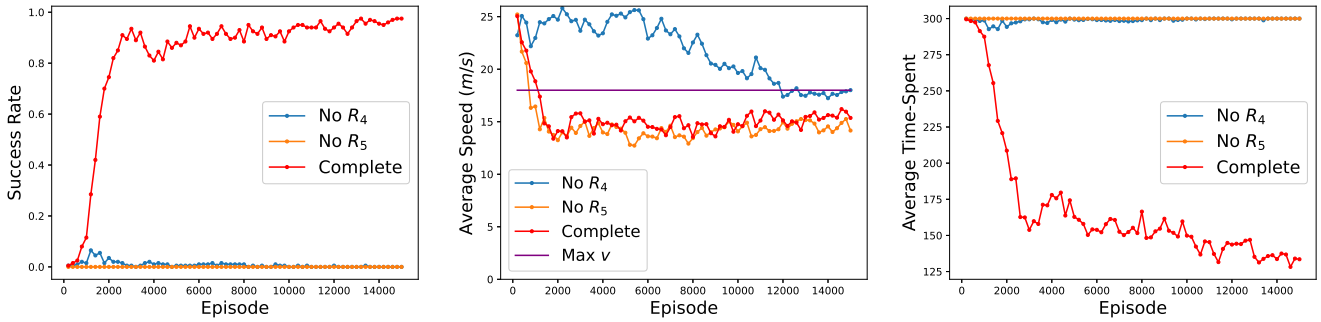
where $X = \frac{(P+Q)}{2}$.

When calculating the JSD, the map is divided into multiple grids of square cells with side length m. Subsequently, from the real trajectory dataset, we can get the frequency of visits to each grid cell. Finally, these frequencies are aggregated to form the probability distribution P, serving as the ground truth. The probability distribution Q is derived from trajectories generated using other methodologies. We compare our trajectory generation results with other baseline methods, including Trajectory Synthesis Intersection Partition (FTS-IP) [33], LSTM [13], and Two-stage GAN (TSG) [42]. When generating new trajectories, the FTS-IP method decomposes crossing trajectories and produces new trajectories by selecting proper intervals. The LSTM method generates new trajectories by iteratively taking the current output trajectory as input. The TSG method first generates a grid-based trajectory and then refines the trajectory using the GAN. These three approaches represent three different perspectives used in trajectory generation. For a fair comparison, in our experiments, we use the same real taxi trajectory data and set the grid side length to



(a) Success Rate (b) Average Speed (m/s) (c) Average Time-Spent (s)

Fig. 14: Validation results I.



(a) Success Rate (b) Average Speed (m/s) (c) Average Time-Spent (s)

Fig. 15: Validation results II.

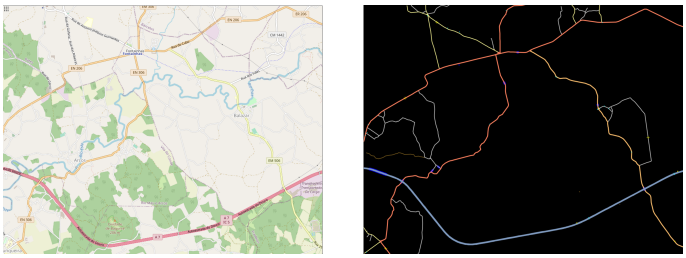


Fig. 16: The real-world scenario map.

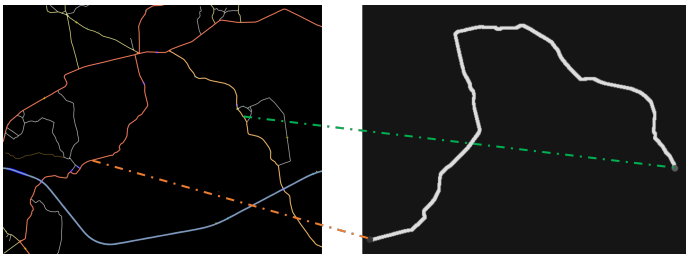


Fig. 17: An example of path extraction.

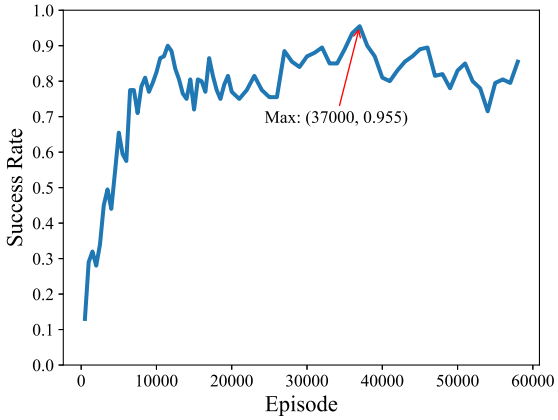


Fig. 18: Test of Success Rate in the real map.

0.01 in both latitude and longitude. The results are shown in Table I.

TABLE I: The JSD results comparison

Model	FTS-IP	LSTM	TSG	Ours
JSD	0.413	0.633	0.100	0.056

The results presented in Table I demonstrate that the trajectories generated using our approach closely resemble the positional distribution characteristics of actual trajectories. This is attributed to our method's ability to seamlessly align the generated trajectories with the shapes of different roadways. As shown in Figure 21, the green lines represent the trajectories we generated, with a sampling rate of up to 1 second. The white dots correspond to the trajectory data

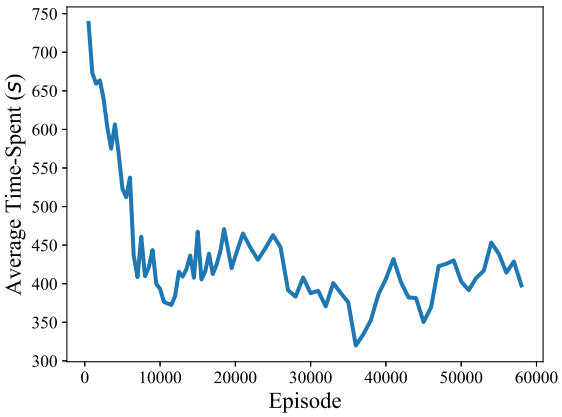


Fig. 19: Test for Average Time-spent in the real map.

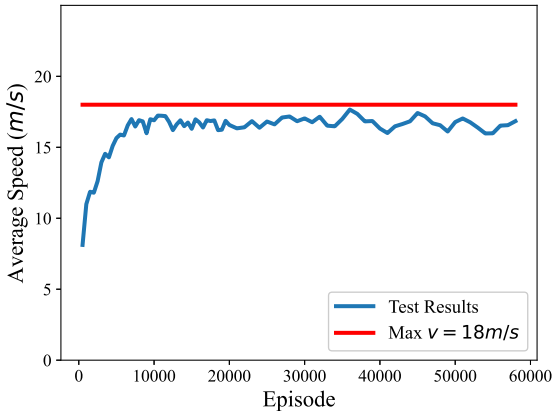


Fig. 20: Test for Average Speed in the real map.

collected from the real world, with a sampling interval of 15 seconds.

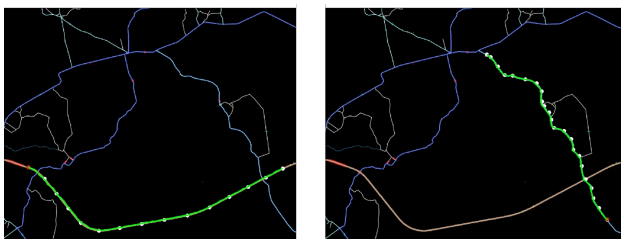


Fig. 21: Trajectory generation in real scenarios.

Moreover, in Figure 22, the yellow parts represent the road, and the red lines are the paths created. It shows that the intelligent agent can generate trajectories successfully even in different complex road layouts.

V. Privacy Protection with Generated Trajectories

In VANETs, user privacy faces multiple challenges. Existing research [53] categorizes attackers along three dimensions

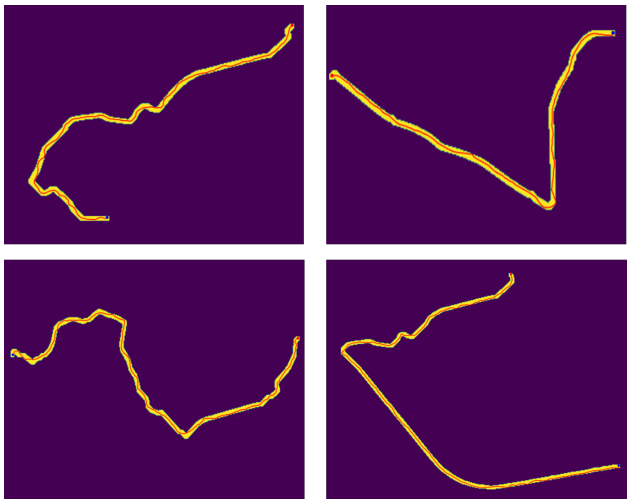


Fig. 22: The generated trajectories in different scenarios.

based on their capabilities and objectives: active/passive, internal/external, and global/local. Active attackers aim to disrupt communication and services through network attacks. Conversely, passive attackers engage in eavesdropping on communication or stealing data without affecting the normal operation of devices and services. The internal/external classification is determined by whether the attacker is considered part of the entire location-based service ecosystem; for example, a service provider is considered internal. Lastly, the global/local classification depends on the resources the attacker can control. A global attacker has access to all past and current data within the network, while a local attacker only possesses data related to limited vehicles, locations, or time periods. In this study, we assume a global passive attacker whose objective is to track target vehicles in real-time.

In Section II, we discussed existing privacy protection methods and their advantages and disadvantages. A recent study [11] has also shown that achieving k-anonymity becomes significantly challenging in environments with low traffic density, primarily attributed to the absence of cooperative entities. This difficulty poses a substantial challenge to privacy protection. To address this issue, in this section, we propose two methods to leverage the customizable trajectory generation capabilities introduced in Section III for privacy protection. The first is substituting fabricated trajectories for vehicles' authentic trajectories when disclosing their own trajectory information. The second is the joint publication of both generated and real trajectories, and further, using the fabricated trajectories in pseudonym-changing strategies to enhance privacy protection.

A. Trajectory substitution strategy

For the trajectory substitution strategy, we propose that each user independently generates synthetic trajectories using the intelligent agent based on a preset value ϵ when traveling halfway, simultaneously concealing their following true trajectories. Unlike traditional trajectory generation methods, our approach doesn't rely on the original real trajectory database; instead, it utilizes only map and maximum speed information

to customize trajectory start and end points, as well as speeds. This breakthrough allows the trajectory substitution strategy to operate without geographical constraints, making it widely applicable across various scenarios, even entirely new ones.

Essentially, as a method introducing noise interference, this strategy can be used without considering the traffic density, and the privacy level of this strategy can also be evaluated by analyzing the error between the fake trajectory destination and the real one. In our strategy, the setting of ϵ-geo-indistinguishability determines the discrepancy. Figures 23 and 24 present the distribution and the mean error under various values of parameter ϵ. These figures allow users to intuitively perceive the level of privacy protection based on distance.

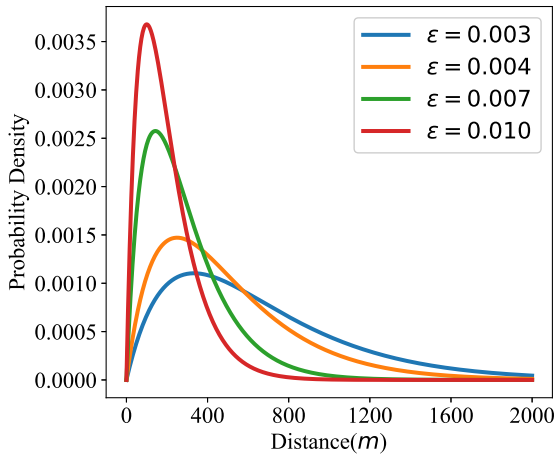


Fig. 23: The probability density function of disparity.

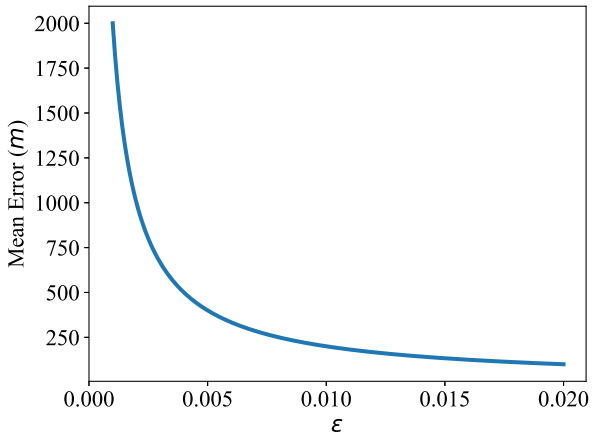


Fig. 24: The privacy level under different ϵ value.

1) Impact on Data Usability: In this section, we consider the impact of the privacy protection strategy on the usability of the data. For this evaluation, we consider a scenario in the Manhattan mobility model map depicted in Figure 13, where vehicles transmit their locations to a service provider to obtain the status of surrounding traffic lights. Due to the publication of synthetic trajectories, there is an impact on the quality of location-based service and the original database. To investigate whether reliable location services can be obtained using these trajectories, we consider the real-time location error between the actual trajectory and the synthetic trajectory to assess data usability. Figure 25 illustrates the results where 1000 vehicles were randomly selected to calculate their average error. Initially, the error increases over time, peaking within the $400-500$ meter range. This maximum error is approximately the distance between adjacent traffic lights which may be due to the influence of traffic lights causing vehicle stops at intersections. Subsequently, the error gradually decreases after reaching its maximum value. Specifically, at the same time point, when $\epsilon = 0.05$, the error is smaller compared to when $\epsilon = 0.01$. This reduction in error is attributed to higher values of ϵ, which lead to lower privacy levels, resulting in closer distance between the final positions of real and synthetic trajectories, thus reducing real-time error. Finally, in this scenario, users seeking real-time information about the next traffic light should query the status of all traffic lights within approximately a 500 meter radius centered on the position indicated by the synthetic trajectory. The results show that the usability of the data is not compromised by the privacy protection strategy.

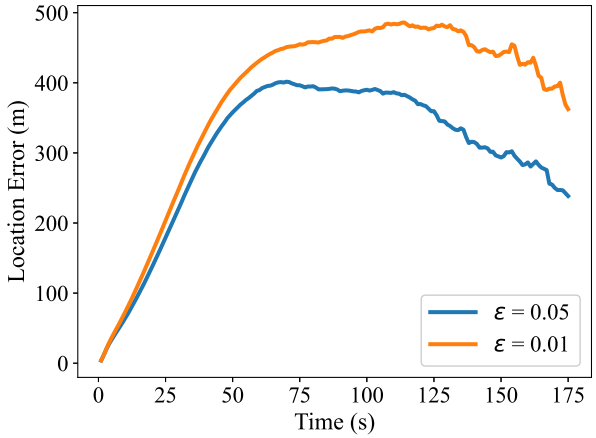


Fig. 25: The real-time location error to actual trajectories.

2) Impact on Data Statistical Characteristics: In addition to users utilizing location data to access services, some traffic monitoring systems also adjust current traffic guidance strategies based on the visitation rates of hotspot locations. Therefore, in this section, we use JSD to study the impact of fake trajectories on data statistical characteristics. In this evaluation, we also use the map depicted in Figure 13. When calculating JSD, the area, measuring 2 km $\times$ 2 km, is subdivided into numerous small grids, each sized 50 m $\times$ 50 m. The actual trajectory's probability distribution within these grids is taken as the reference distribution P in Equation (30), and the output semi-authentic and semi-synthetic trajectory distributions of the trajectory substitution strategy serve as Q.

Figure 26 shows this comparative outcome through an error bar plot. As depicted, with an increase in ϵ, the JSD gradually decreases, accompanied by a reduction in the standard deviation. This trend is attributed to the diminishing

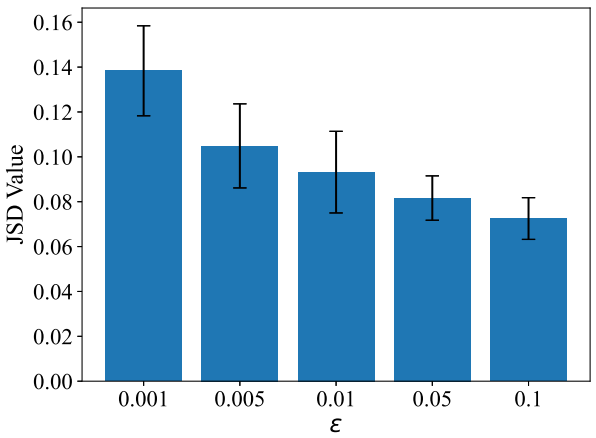


Fig. 26: Impact of different ϵ on real trajectory dataset.

privacy requirements (increase in ϵ), which in turn reduces the error between real and fake destinations, thereby leading the generated trajectories to progressively resemble the real trajectories.

Additionally, it can be observed that even when ϵ is set to 0.001, the JSD remains around 0.14. There are two reasons for this small value. Firstly, our approach ensures that trajectories are not generated outside the road network. Secondly, half of the generated trajectories are the same as the real data set. The lower JSD also denotes that our method can preserve certain fundamental characteristics of the database, such as analyzing the frequency of visits to specific locations and the density of vehicles in particular areas.

B. Pseudonym-changing strategy

In a recent comparison of pseudonym-changing approaches [11], it was observed that the privacy level remains low in a low traffic density environment. Different from the privacy assessment methods employed in trajectory substitution, researchers utilize entropy to evaluate the level of privacy protection afforded by pseudonym-changing strategies, which is calculated as:

$$MeanPrivacy = \frac{1}{N}\sum_{i=1}^{N}\log n_i \tag{31}$$

where N is the number of vehicles in the simulation period and n_i represents the size of the set of vehicles that have changed pseudonyms together with vehicle i. The larger the entropy, the higher the level of privacy.

In this section, we propose that users can publish in real-time the generated fictitious trajectories to protect themselves from being tracked. Moreover, these trajectories actively contribute to the pseudonym-changing strategy. To assess the effectiveness of this strategy in improving user privacy, experiments are conducted utilizing the scenario illustrated in Figure 13. Additionally, to acquire high-sampling-rate trajectory data from vehicles with diverse traffic densities, we employ the "Simulation of Urban Mobility" (SUMO) tool [54] to generate vehicle flows and simulate varying traffic densities by adjusting the arrival rates of vehicles.

When employing the previous context-based pseudonym-changing strategy [11] in low-density traffic scenarios, we set the radius of the mix-zone as 30 m. In order to maximize the privacy levels of all users, the triggering threshold is set as $k = 2$. This means that each vehicle starts changing pseudonyms when it detects the presence of at least two vehicles (including itself) within its mix-zone. In our privacy protection method, we set that vehicles generate fake trajectories halfway with $\epsilon = 0.05$. If fake trajectories enter the mix-zone, the other vehicles regard them as real vehicles, and start the pseudonym-changing process.

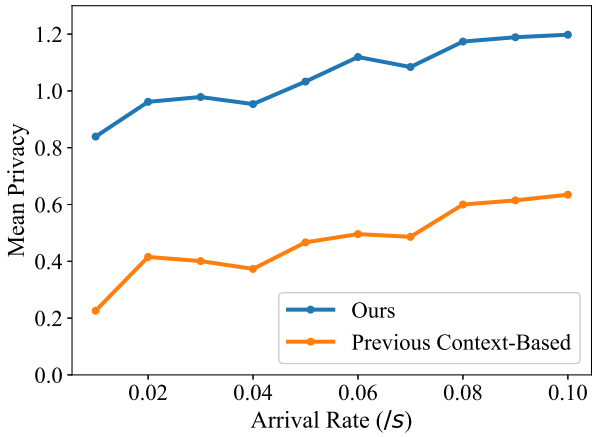


Fig. 27: The mean privacy with different arrival rates.

Figure 27 illustrates the privacy comparison between our method and the previous context-based approach under low arrival rate conditions within a 1000-second simulation period. It can be observed that as the arrival rate increases, both methods exhibit a gradual growth in the level of privacy protection. However, by taking the generated trajectories into our method, the privacy level of vehicles in low-density traffic environments has been significantly enhanced, reaching up to nearly fourfold in the case with an arrival rate of 0.01 /s.

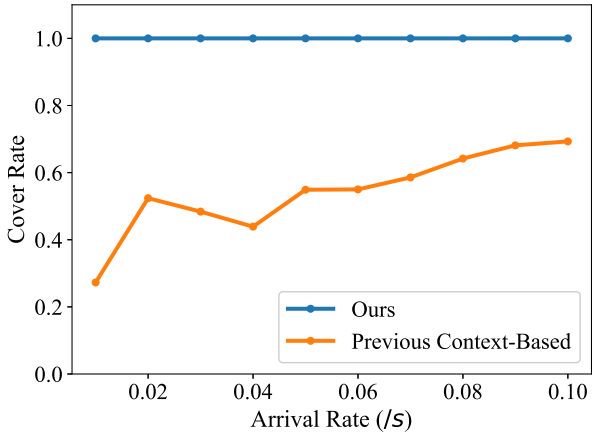


Fig. 28: The cover rate with different arrival rates.

Additionally, Figure 28 provides an analysis of the effective privacy protection cover rate for all vehicles. For this criterion, our method also performs better than the previous method. Furthermore, due to the generation of fabricated trajectories for each vehicle, our method achieves a coverage rate of 100%.

1) Impact on Data Usability: The pseudonym-changing strategy results in vehicles having multiple endpoints. In this part, we explore how these fictitious endpoints affect the quality of service when vehicles search for nearby parking facilities.

Consider the scenario illustrated in Figure 13. Suppose there are four parking lots located at coordinates $P_{k1}(500, 500)$, $P_{k2}(500, 1500)$, $P_{k3}(1500, 500)$, and $P_{k4}(1500, 1500)$. Initially, users submit their destinations to the service provider. Subsequently, the service provider responds with a list of parking lot information arranged by proximity, with the nearest parking lot listed first. For instance, if a vehicle's actual destination is at $(500, 600)$, it would receive a list $[P_{k1}, P_{k2}, P_{k3}, P_{k4}]$. However, due to the publication of fake trajectories, the fictitious destination might be positioned at $(500, 1200)$, resulting in the vehicle receiving the list $[P_{k2}, P_{k1}, P_{k4}, P_{k3}]$. To evaluate the quality of service experienced by users, we employ a custom Cumulative Gain (CG) metric to score the returned list as follows:

$$ServiceScore = \sum_{i=1}^{4} Sco_i \times (4 - i), \tag{32}$$

where we define Sco_i as the score assigned to the item in the actual rank i, with values set as $[8, 4, 2, 0]$. Using the aforementioned example, the real rank list $[P_{k1}, P_{k2}, P_{k3}, P_{k4}]$ corresponds to scores $[P_{k1} : 8, P_{k2} : 4, P_{k3} : 2, P_{k4} : 0]$. It can be calculated that the maximum $ServiceScore$ is $8 \times 3 + 4 \times 2 + 2 \times 1 + 0 \times 0 = 34$. After deploying the privacy protection method, the user receives the ranked list $[P_{k2}, P_{k1}, P_{k4}, P_{k3}]$, and the $ServiceScore$ becomes $4 \times 3 + 8 \times 2 + 0 \times 1 + 2 \times 0 = 28$. However, if the user receives the ranked list $[P_{k4}, P_{k3}, P_{k2}, P_{k1}]$, which is in reverse order compared to the actual rank, the $ServiceScore$ has the minimum value $0 \times 3 + 2 \times 2 + 4 \times 1 + 8 \times 0 = 8$.

This scoring framework assigns higher scores and weights to items ranked higher in the actual ranking, which aligns with our intuitive perception of service quality. Moreover, the scale ranging from a maximum of 34 points to a minimum of 8 points widens the gap between the best and worst scores, making it reasonable to utilize this scoring framework for evaluating parking recommendation systems. Figure 29 illustrates the impact of privacy setting ϵ on service quality.

Firstly, from the blue line in Figure 29, it can be observed that as ϵ increases, the distance between the fake destination and the real destination continuously decreases, leading to an improvement in service quality, as represented by the orange line. Secondly, it is noteworthy that reducing the distance error from 1200 m to approximately 400 m results in a significant enhancement in service quality. Considering that the nearest distance between parking lots is 500 m, when the destination error is less than this distance, it does not affect the top-ranked result in the search. These findings indicate that our privacy protection strategy not only enhances privacy levels in low-density traffic environments but also achieves a balance between privacy and service quality across various applications.

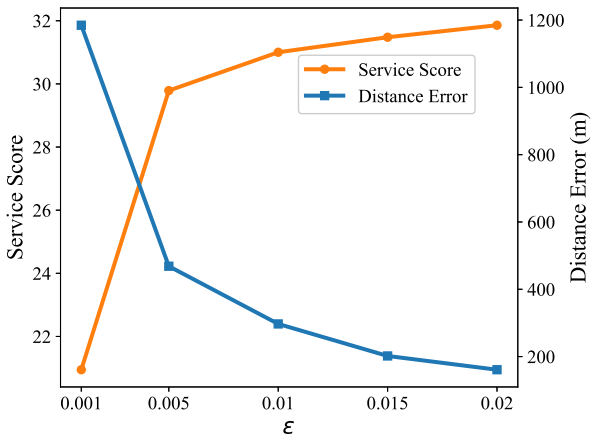


Fig. 29: The impact on service score.

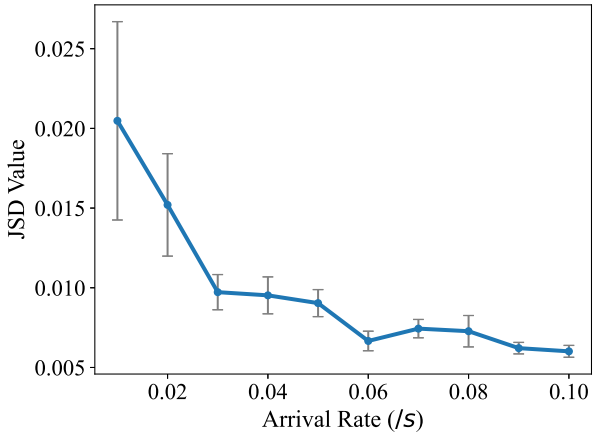


Fig. 30: The impact on data set with new trajectories added.

2) Impact on Data Statistical Characteristics: In addition, we use the same setting as Section VI. A. 1 to explore the influence on data statistical characteristics. Figure 30 shows the results. We can find that compared with Fig. 26, under the same $\epsilon = 0.05$ setting, the impact on the database caused by adding trajectories is smaller than that resulting from replacing the original trajectories. In addition, as more vehicles appear in the scenarios, the influence decreases. This is because the visited locations in the generated trajectories coincide with the locations of the new actual trajectories. In contrast, when the original data consists of only a few trajectories, the added fake trajectories may visit locations that are not present in the real trajectories.

Finally, as our privacy protection method involves two phases, namely virtual trajectory generation and real-time publishing, we assess the efficiency of each phase separately. The following data are the test results conducted on the

platform with an RTX 3080 (10G) GPU and an Intel(R) Xeon(R) Platinum 8255C CPU @ 2.50GHz, using Python 3.8: Generating 100 trajectories with an average length of 465s and an average speed of 16.33m/s takes approximately 113.3s. This implies that the GPU+CPU combination can perform over 400 pairs of environment updates and agent decisions in one second. This efficiency is attributed to our lightweight reinforcement learning model, resulting in a relatively light workload on the GPU. As for the trajectory publication phase, apart from adding the trajectory to be published, whose complexity is $\mathcal{O}(1)$, and no other settings need to be altered.

VI. Conclusion and Future Work

Due to the mechanisms of location broadcasting and sharing, safeguarding users' privacy in vehicular networks becomes a crucial challenge. Traditional user-centric privacy protection methods, such as pseudonym-changing, ensure data accuracy but prove inadequate in preserving privacy, especially in low-density traffic conditions. To address this challenge, this paper introduces a novel approach that utilizes virtual trajectories generated by a hybrid reinforcement learning agent to enhance privacy protection.

Initially, we apply ϵ-geo-indistinguishability and the A* algorithm to extract a path leading to a fictional endpoint from the map. Following this, we employ an agent trained by a hybrid reinforcement learning model for trajectory generation. This model not only achieves a success rate exceeding 90% in complex trajectory generation tasks but also enables the customization of trajectory speed to accommodate varied requirements in diverse settings. Leveraging generated trajectories, we propose two methods to protect privacy: trajectory substitution and pseudonym-changing strategy. Experimental results demonstrate that the implementation of these methods not only effectively addresses privacy concerns in low-density traffic environments but also manages the impact on the original database.

In future work, our method can be refined in the following aspects. Firstly, due to the multiple parameter combinations and reward functions, there remains the potential for further optimization within our hybrid reinforcement learning model. Secondly, this paper only discusses two methods that integrate generated trajectories for privacy protection. In reality, many other methods can be combined with them. Lastly, it's important to consider the potential impact of malicious attackers exploiting this mechanism to disseminate a large volume of fabricated trajectories. Such actions could also disrupt communication within VANETs and impact the utility of the database. Therefore, careful consideration is necessary on how to mitigate this issue effectively.

VII. Acknowledgment

This work was supported in part by the Asian Institute of Digital Finance (AIDF) under grant A-0003504-09-00.

References

[1] M. Z. Khan, O. H. Alhazmi, M. A. Javed, H. Ghandorh, and K. S. Aloufi, "Reliable internet of things: Challenges and future trends," *Electronics*, vol. 10, no. 19, p. 2377, Sep. 2021.

[2] X. Lei, X. Chen, and S. Rhee, "A hybrid access method for broadcasting of safety messages in ieee 802.11p vanets," *EURASIP Journal on Wireless Communications and Networking*, vol. 2021, pp. 71–90, Apr. 2021.

[3] M. Haque, H. Chin, and A. Debnath, "Sustainable, safe, smart—three key elements of singapore's evolving transport policies," *Transport Policy*, vol. 27, pp. 20–31, Feb. 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0967070X1200193X

[4] M. Arif, G. Wang, and T. Peng, "Track me if you can? query based dual location privacy in vanets for v2v and v2i," in *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, Aug. 2018, pp. 1091–1096.

[5] M. Babaghayou, N. Labraoui, A. Adamou, A. Ari, N. Lagraa, and M. A. Ferrag, "Pseudonym change-based privacy-preserving schemes in vehicular ad-hoc networks: A survey," *Journal of Information Security and Applications*, vol. 55, p. 102618, Oct. 2020.

[6] J. Qi and T. Gao, "A privacy-preserving authentication and pseudonym revocation scheme for vanets," *IEEE Access*, vol. 8, pp. 177 693–177 707, Sep. 2020.

[7] L. Benarous, B. Kadri, and S. Boudjit, "Alloyed pseudonym change strategy for location privacy in vanets," in *2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*, Jan. 2020, pp. 1–6.

[8] I. Memon, Q. Ali, A. Zubedi, and F. A. Mangi, "Dpmm: dynamic pseudonym-based multiple mix-zones generation for mobile traveler," *Multimedia Tools and Applications*, vol. 76, no. 22, pp. 24 359–24 388, Nov. 2017.

[9] A. Boualouache, S.-M. Senouci, and S. Moussaoui, "A survey on pseudonym changing strategies for vehicular ad-hoc networks," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 770–790, Nov. 2017.

[10] I. Ullah, M. A. Shah, A. Khan, C. Maple, A. Waheed, and G. Jeon, "A distributed mix-context-based method for location privacy in road networks," *Sustainability*, vol. 13, no. 22, p. 12513, Nov. 2021.

[11] Z. Zhang, T. Feng, W.-C. Wong, and B. Sikdar, "A geo-indistinguishable context-based mix strategy for trajectory protection in vanets," *IEEE Transactions on Vehicular Technology*, pp. 1–15, Jul. 2023, early access.

[12] M. Baratchi, N. Meratnia, P. Havinga, A. Skidmore, and B. Toxopeus, "A hierarchical hidden semi-markov model for modeling mobility data," in *Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing*, Aug. 2014, pp. 401–412.

[13] V. Kulkarni and B. Garbinato, "Generating synthetic mobility traffic using recurrent neural networks," in *Proceedings of the 1st Workshop on Artificial Intelligence and Deep Learning for Geographic Knowledge Discovery*, Nov. 2017, pp. 1–4.

[14] S. Yongfeng, W. Chen, C. Claramunt, and S. Yang, "A ship trajectory prediction framework based on a recurrent neural network," *Sensors*, vol. 20, p. 5133, Sep. 2020.

[15] R. Huang, C. Wei, B. Wang, J. Yang, X. Xu, S. Wu, and S. Huang, "Well performance prediction based on long short-term memory (lstm) neural network," *Journal of Petroleum Science and Engineering*, vol. 208, p. 109686, Oct. 2021.

[16] X. Song, Y. Liu, L. Xue, J. Wang, J. Zhang, J. Wang, L. Jiang, and Z. Cheng, "Time-series well performance prediction based on long short-term memory (lstm) neural network model," *Journal of Petroleum Science and Engineering*, vol. 186, p. 106682, Nov. 2019.

[17] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social gan: Socially acceptable trajectories with generative adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Mar. 2018, pp. 2255–2264.

[18] H. Zhong, S. Zhang, J. Cui, L. Wei, and L. Liu, "Broadcast encryption scheme for v2i communication in vanets," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 3, pp. 2749–2760, Mar. 2022.

[19] C. A. Ardagna, M. Cremonini, S. D. C. di Vimercati, and P. Samarati, "An obfuscation-based approach for protecting location privacy," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 1, pp. 13–27, Mar. 2011.

[20] C. Dwork, "Differential privacy," in *Automata, Languages and Programming: 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II 33*. Springer, Jul. 2006, pp. 1–12.
[21] T. Feng, W.-C. Wong, S. Sun, Y. Zhao, and Z. Zhang, "Location privacy preservation and location-based service quality tradeoff framework based on differential privacy," in *2019 16th Workshop on Positioning, Navigation and Communications (WPNC)*. IEEE, Oct. 2019, pp. 1–6.
[22] K. Emara, "Poster: Prext: Privacy extension for veins vanet simulator," in *2016 IEEE Vehicular Networking Conference (VNC)*, Dec. 2016, pp. 1–2.
[23] L. Santos Jaimes and E. Moreira, "Pseudonym change strategy based on the reputation of the neighbouring vehicles in vanets," *DYNA*, vol. 86, pp. 157–166, Oct. 2019.
[24] B. Wiedersheim, Z. Ma, F. Kargl, and P. Papadimitratos, "Privacy in inter-vehicular networks: Why simple pseudonym change is not enough," in *2010 Seventh international conference on wireless on-demand network systems and services (WONS)*. IEEE, Mar. 2010, pp. 176–183.
[25] A. Boualouache and S. Moussaoui, "Urban pseudonym changing strategy for location privacy in vanets," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 24, no. 1-2, pp. 49–64, Feb. 2017.
[26] B. Palanisamy and L. Liu, "Attack-resilient mix-zones over road networks: architecture and algorithms," *IEEE Transactions on Mobile Computing*, vol. 14, no. 3, pp. 495–508, Mar. 2015.
[27] Y. Sun, B. Zhang, B. Zhao, X. Su, and J. Su, "Mix-zones optimal deployment for protecting location privacy in vanet," *Peer-to-Peer Networking and Applications*, vol. 8, no. 6, pp. 1108–1121, Jun. 2014.
[28] N. Ravi, C. M. Krishna, and I. Koren, "Enhancing vehicular anonymity in its: A new scheme for mix zones and their placement," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 11, pp. 10 372–10 381, Aug. 2019.
[29] Z. Lu, G. Qu, and Z. Liu, "A survey on recent advances in vehicular network security, trust, and privacy," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 2, pp. 760–776, Apr. 2018.
[30] K. Emara, W. Woerndl, and J. Schlichter, "Context-based pseudonym changing scheme for vehicular adhoc networks," *arXiv preprint arXiv:1607.07656*, pp. 1–13, Jul. 2016.
[31] A. Mdee, M. T. R. Khan, J. Seo, and D. Kim, "Security compliant and cooperative pseudonyms swapping for location privacy preservation in vanets," *IEEE Transactions on Vehicular Technology*, vol. PP, pp. 1–15, Jan. 2023.
[32] Z. Zhang, T. Feng, B. Sikdar, and W.-C. Wong, "A flickering context-based mix strategy for privacy protection in vanets," in *ICC 2021 - IEEE International Conference on Communications*, Aug. 2021, pp. 1–6.
[33] J. Li, W. Chen, A. Liu, Z. Li, and L. Zhao, "Fts: a feature-preserving trajectory synthesis model," *GeoInformatica*, vol. 22, pp. 1–22, Jan. 2018.
[34] A. Asahara, K. Maruyama, A. Sato, and K. Seto, "Pedestrian-movement prediction based on mixed markov-chain model," in *Proceedings of the 19th ACM SIGSPATIAL international conference on advances in geographic information systems*, Nov. 2011, pp. 25–33.
[35] X. Kong, Q. Chen, M. Hou, A. Rahim, K. Ma, and F. Xia, "Rmgen: A tri-layer vehicular trajectory data generation model exploring urban region division and mobility pattern," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 9, pp. 9225–9238, 2022.
[36] X. Kong, Q. Chen, M. Hou, H. Wang, and F. Xia, "Mobility trajectory generation: a survey," *Artificial Intelligence Review*, vol. 56, no. Suppl 3, pp. 3057–3098, Sep. 2023.
[37] G. Shen, P. Li, Z. Chen, Y. Yang, and X. Kong, "Spatio-temporal interactive graph convolution network for vehicle trajectory prediction," *Internet of Things*, vol. 24, p. 100935, Sep. 2023.
[38] C. Chen, K. Li, S. G. Teo, X. Zou, K. Li, and Z. Zeng, "Citywide traffic flow prediction based on multiple gated spatio-temporal convolutional neural networks," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 14, no. 4, pp. 1–23, 2020.
[39] K. Ouyang, R. Shokri, D. Rosenblum, and W. Yang, "A non-parametric generative model for human trajectories," in *IJCAI*, vol. 18, Jul. 2018, pp. 3812–3817.
[40] V. Kulkarni, N. Tagasovska, T. Vatter, and B. Garbinato, "Generative models for simulating mobility trajectories," *arXiv preprint arXiv:1811.12801*, pp. 1–7, Nov. 2018.
[41] J. Rao, S. Gao, Y. Kang, and Q. Huang, "Lstm-trajgan: A deep learning approach to trajectory privacy protection," *arXiv preprint arXiv:2006.10521*, pp. 1–16, Jun. 2020.
[42] X. Wang, X. Liu, Z. Lu, and H. Yang, "Large scale gps trajectory generation using map based on two stage gan," *Journal of Data Science*, pp. 126–141, Feb. 2021.
[43] J. Zhang, Q. Huang, Y. Huang, Q. Ding, and P.-W. Tsai, "Dp-trajgan: A privacy-aware trajectory generation model with differential privacy," *Future Generation Computer Systems*, vol. 142, pp. 25–40, Dec. 2022.
[44] G. Xiong, Z. Li, M. Zhao, Y. Zhang, Q. Miao, Y. Lv, and F.-Y. Wang, "Trajsgan: A semantic-guiding adversarial network for urban trajectory generation," *IEEE Transactions on Computational Social Systems*, pp. 1–11, n.d. 2023.
[45] L. L. L. Starace, "Porto taxi trajectories," May 2020. [Online]. Available: https://figshare.com/articles/dataset/Porto_taxi_trajectories/12302165
[46] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: Differential privacy for location-based systems," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, Nov. 2013, pp. 901–914.
[47] A. Candra, M. A. Budiman, and K. Hartanto, "Dijkstra's and a-star in finding the shortest path: A tutorial," in *2020 International Conference on Data Science, Artificial Intelligence, and Business Analytics (DATABIA)*. IEEE, Jul. 2020, pp. 28–32.
[48] M. A. Wiering and M. Van Otterlo, "Reinforcement learning," *Adaptation, learning, and optimization*, vol. 12, no. 3, p. 729, Jan. 2012.
[49] M. Naeem, S. T. H. Rizvi, and A. Coronato, "A gentle introduction to reinforcement learning and its application in different fields," *IEEE access*, vol. 8, pp. 209 320–209 344, Jan. 2020.
[50] O. Nachum, M. Norouzi, K. Xu, and D. Schuurmans, "Bridging the gap between value and policy based reinforcement learning," in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., Feb. 2017.
[51] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, pp. 1–12, Aug. 2017.
[52] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International conference on machine learning*. PMLR, Feb. 2015, pp. 1889–1897.
[53] M. Raya and J.-P. Hubaux, "Securing vehicular ad hoc networks," *Journal of computer security*, vol. 15, no. 1, pp. 39–68, Jan. 2007.
[54] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using sumo," in *2018 21st international conference on intelligent transportation systems (ITSC)*. IEEE, Nov. 2018, pp. 2575–2582.

Zhixiang Zhang (Student Member, IEEE) received the B.Eng. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2018, and the M.Sc. degree from the National University of Singapore, Singapore, in 2019, where he is currently pursuing a Ph.D. degree with the Department of Electrical and Computer Engineering. His current research interests include Internet of Things, privacy preservation, and machine learning.

Wai-Choong (Lawrence) Wong (Life Senior Member, IEEE) received the B.Sc. and Ph.D. degrees in electronic and electrical engineering from Loughborough University, Loughborough, U.K., in 1976 and 1980, respectively. He was an Emeritus Professor with the Department of Electrical and Computer Engineering, National University of Singapore (NUS), Singapore. He was a member of Technical Staff with AT&T Bell Laboratories, Crawford Hill Lab, Holmdel, NJ, USA, from 1980 to 1983. From 2002 to 2006, he was the Executive Director of the Institute for Infocomm Research (I^2R), Agency for Science, Technology and Research (A*STAR), Singapore. In 1983, he joined NUS, where he has been serving in various leadership positions at the department, faculty, and university levels. His research interests include wireless and sensor networks and systems, ambient intelligent platforms, localization, and source matched transmission techniques, with over 300 publications and five patents in these areas.

Dr. Wong has received several awards, including the IEE Marconi Premium Award in 1989, the IEEE Millennium Award in 2000, the e-nnovator Awards (Open Category) in 2000, the Best Paper Award at the IEEE International Conference on Multimedia and Expo in 2006, and the Best Paper Award at the Second International Conference on Ambient Computing, Applications, Services, and Technology in 2012, and the SupercomputingAsia HPC Network Achievement Award in 2023. He is a member of IEEE-Eta Kappa Nu.

Biplab Sikdar received the B.Tech. degree in electronics and communication engineering from North Eastern Hill University, Shillong, India, in 1996, the M.Tech. degree in electrical engineering from the Indian Institute of Technology, Kanpur, India, in 1998, and the Ph.D. degree in electrical engineering from Rensselaer Polytechnic Institute, Troy, NY, USA, in 2001.

He is currently a Professor and Head of Department with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore. He was an Assistant and an Associate Professor at the Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute from 2001 to 2013. His research interests include IoT and cyber–physical system security, network security, and network performance evaluation. Dr. Sikdar has served as an Associate Editor for the IEEE Transactions on Communications, IEEE Transactions on Mobile Computing, IEEE Internet of Things Journal, and IEEE Open Journal of Vehicular Technology. He is a member of Eta Kappa Nu and Tau Beta Pi.