Privacy-Preserving Personalized Autonomous Vehicle Lane Change Using Inverse Reinforcement Learning

Zhaodong Zhou and Jun Chen, *Senior Member, IEEE*

***Abstract*—This paper presents an approach to model driver lane change behavior using maximum entropy inverse reinforcement learning (*MaxEnt* IRL). The proposed method aims to generate personalized lane change trajectories that reflect individual driving preferences while ensuring user privacy. To achieve this, driving data are collected from individual driver and used to train the model, while scale-based transformation is utilized to encrypt the data during cloud-based training. Bezier curves are employed to parameterize the lane change trajectories due to their ability to produce smooth, continuous paths. The *MaxEnt* IRL algorithm is then used to infer a reward function that represents each driver's preferences by learning optimal weights for a cost function that describes the lane change behavior. The proposed approach is tested over several real-world data to demonstrate its effectiveness in capturing personal driving styles under various conditions. The proposed trajectory encryption approach is compared to a benchmark differential privacy mechanism, and the results clearly show that the proposed method outperforms. Experimental results demonstrate that the proposed method can generate lane change paths that closely align with the behavior exhibited by individual driver, offering an approach for personalization in autonomous driving.**

***Index Terms*—Autonomous vehicles, inverse reinforcement learning, automated lane change systems, personalized motion control, privacy preservation.**

NOMENCLATURE

Parameters

$\bar{\zeta}$	Unmasked planned trajectory
$\hat{\zeta}$	Masked planned trajectory
$\check{\zeta}$	Masked expert trajectory
$\tilde{F}$	Encrypt feature vector for dataset
$\tilde{f}$	Encrypt feature vector for each trajectory
ζ	Expert trajectory
a	Scaling factor
D	Dataset
F	Feature vector for dataset
f	Feature vector for each trajectory

Variables

W	Weight

This work is supported in part by the Faculty Research Fellowship from Oakland University and in part by National Science Foundation through Awards #2237317 and #2432098. *Corresponding author: J. Chen.*

The authors are with the Department of Electrical and Computer Engineering, Oakland University, Rochester, MI 48309, USA (email: {zhaodongzhou,junchen}@oakland.edu).

The lane changing modeling have been presented in 2024 IEEE International Conference on Computing and Machine Intelligence [1], validated using simulated data. This paper enhances the feature design used in [1]. In addition, to address the user privacy during cloud-based training, this paper proposes a privacy-preserving training framework. Moreover, the experimental section has been significantly expanded by incorporating a driver-in-the-loop simulator and real-world tests using a golf cart platform.

I. INTRODUCTION

LANE change is a critical aspect of driving that can significantly impact traffic flow, safety, and driver comfort [2]–[5]. As autonomous driving technologies have made significant advancements in recent years, the ability to model and replicate human-like lane change behaviors has become increasingly important. Traditional approaches to lane change modeling often rely on predefined rules or paths, which may not adequately capture the preference and decision-making process of individual drivers [6], [7]. Moreover, with the development of sensor and communication technologies, a large amount of vehicle trajectory data has been collected. Consequently, many researchers have adopted data-driven models to study lane change. For example, various machine learning techniques, such as neural networks [8]–[11] and reinforcement learning [12], [13], have been used to capture the complexities of driving maneuvers. Despite their promising results, neural networks and reinforcement learning have notable drawbacks. First, neural networks can suffer from limited interpretability, making it challenging to understand the reason behind their predictions [14]. Second, reinforcement learning is highly sensitive to the design of reward functions [15], [16]. To overcome these limitations, this paper employs inverse reinforcement learning (IRL) to infer underlying reward functions from expert demonstrations, offering a more interpretable and robust framework for modeling complex driving behaviors.

IRL is a powerful approach for capturing the underlying reward functions that imply human driving behaviors [17]. Unlike traditional reinforcement learning, which optimizes a policy based on a predefined reward function, IRL works in reverse by inferring the reward structure directly from expert demonstrations [18]–[20]. In particular, the Maximum Entropy (*MaxEnt*) IRL has shown significant promise in learning complex behaviors while offering enhanced interpretability [21], [22]. *MaxEnt* IRL introduces a probabilistic framework that accounts for the uncertainty in human decision-making process. By maximizing the entropy of the policy, this approach ensures that among all policies that could explain the observed behavior, the one chosen is the most unbiased

and generalizable [23], [24]. This not only helps in capturing representative driving styles but also provides insights into the decision-making process, which is crucial for verifying and validating autonomous driving systems. After learning from expert demonstrations, the identified reward function can then be used to generate personalized trajectories that mirror individual preferences. For instance, in [25], the authors propose a personalized adaptive cruise control framework that employs IRL to extract individual driving preferences from expert demonstrations, i.e., longitudinal speed profiles. In [26], IRL is used to derive a cost function from continuous driving trajectories, capturing key features of overall driving behavior such as acceleration, deceleration, and lane changes to improve autonomous driving comfort and adaptability. Most of these studies focus primarily on longitudinal control. In contrast, [22] addressed lateral behavior by customizing automated lane change systems, where naturalistic driving data are clustered into three styles (aggressive, neutral, conservative) and reward functions are learned for each style. While this demonstrates style-level customization. However, our work focuses on driver-level personalization, where each individual driver's lane-change preferences are modeled separately. Moreover, all the aforementioned studies have not addressed the aspect of user privacy protection when utilizing IRL-based personalization techniques, a gap our research aims to bridge.

To realize personalized lane change path, our study leverages Bezier curves for trajectory planning. Bezier curves are widely used in path planning due to their ability to generate smooth and continuous trajectories. By parameterizing lane change paths with Bezier curves, we can effectively model both curvature and smoothness for ensuring driver comfort and safety [27], [28]. Defined by control points, Bezier curves offer significant flexibility, making them adaptable to various driving styles and conditions [29]. Note that for different drivers, the Bezier curves (and the corresponding control points) are different. Given user demonstrated path, the proposed IRL then optimize the control points to align with individual preference. Furthermore, Bezier curves allow for the incorporation of constraints, such as ensuring zero curvature at the start and end of a lane change. This ensures that the generated paths are physically feasible and align with real-world driving behaviors.

However, since personalized path planning relies on user-provided driving trajectory data, it raises critical concerns about data privacy [30]. More specifically, IRL uses driver demonstration data to extract features, which can potentially reveal user privacy. To address these privacy concerns, several privacy-preserving methods have emerged in recent years, such as differential privacy (DP) [31] and homomorphic encryption (HE) [32]. However, DP is highly sensitive to the amount of noise added. On one hand, excessive noise can affect model accuracy, while on the other hand insufficient noise fails to provide effective encryption [33], [34]. Meanwhile, HE imposes high computational demands, rendering it unsuitable for real time processing [35]. To address these limitations, transformation-based approaches, such as scale-based transformation, have been proposed as more efficient alternatives. By applying invertible scale-based transformation to trajectory data, users' information is effectively encrypted

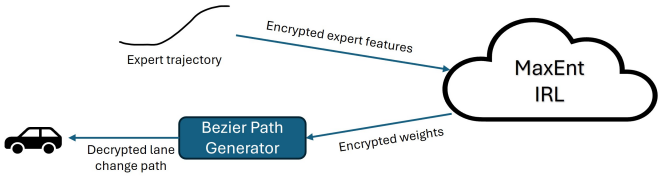

Fig. 1. Diagram of the proposed privacy-preserving personalized lane change framework.

[36]. It is worth noting that scale-based transformation, which has been explored in various cloud-based control contexts, provides a lightweight alternative to traditional cryptographic schemes by minimizing computational and maintaining real time performance [37]. To reduce encryption computation and its impact on training, this paper uses scale-based transformation to protect privacy, as shown in Fig. 1. Specifically, before uploading data to the cloud, user-provided lane change trajectories are locally scaled along the longitudinal axis, and features are then extracted from these transformed trajectories. These transformed (encrypted) features are used to train the MaxEnt IRL model on the cloud, ensuring that the cloud never accesses the original trajectory or raw behavior features. Benchmark comparison with DP-based data transformation is also conducted in this paper.

The primary contributions of this paper are summarized as follows.

1) We develop a personalized lane change framework that combines *MaxEnt* IRL with the Bezier curve. The proposed model is trained on expert driving data, with the output being a learned reward function, optimizing of which generates personalized lane change trajectory.
2) We develop a privacy-preserving framework using random scaling to encrypt user lane change features, and hence ensure user privacy during cloud-based training.
3) The framework is then tested on various human driving datasets, generated by both driver simulator based on CARLA [38], [39] and real-world experiment, to demonstrate its ability to capture personalized driving styles under different conditions. Experimental results show that the proposed method can generate lane change trajectories that closely align with the behaviors exhibited by different drivers. Furthermore, the privacy-preserving mechanism introduces negligible performance degradation, validating its effectiveness in protecting user privacy without sacrificing model accuracy. These findings highlight the potential of the proposed framework to advance personalized autonomous driving systems without concerns about user privacy.

The lane changing modeling have been presented in 2024 IEEE International Conference on Computing and Machine Intelligence [1], validated using simulated data. This paper enhances the feature design used in [1]. In addition, to address the user privacy during cloud-based training, this paper proposes a privacy-preserving training framework. Moreover,

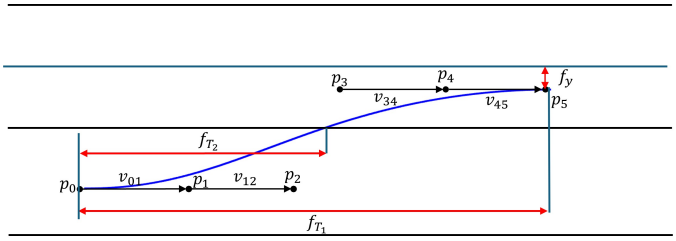

Fig. 2. Illustration of a 5th order Bezier curve for modeling lane change path.

the experimental section has been significantly expanded by incorporating a driver-in-the-loop simulator and real-world tests using a golf cart platform. The remainder of this paper is structured as follows. Section II describes Bezier curve and its application to model lane change path. Section III reviews the preliminary of *MaxEnt* IRL, while Section IV presents our main contribution on the privacy-preserving lane changing personalization algorithm. Section V outlines the experimental setup, presents the results, and discusses the main findings. Finally, Section VI concludes the paper with future work.

II. Lane Change Path Modeling

A Bezier curve is a parametric curve commonly used in path planning [6], [29], defined by control points that determine its shape and smoothness. The formula for an nth order Bezier curve is expressed as:

$$B(t) = \sum_{i=0}^{n} P_i b_{i,n}(t) = \sum_{i=0}^{n} P_i \binom{n}{i} (1-t)^{n-i} t^i. \quad (1)$$

Here, P_i denotes the ith control point, $b_{i,n} = \binom{n}{i}(1-t)^{n-i}t^i$ refers to the Bernstein polynomial associated with the curve, and t represents the parameter that varies from 0 to 1.

As shown in Fig. 2, for this study, the lane change trajectory is modeled using a 5th order Bezier curve, which is defined by six control points and whose equation can be simplified as:

$$B(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3 + c_4 t^4 + c_5 t^5.$$

Here each coefficient c_i is a function of control points P_i, as follows.

$$\begin{aligned}
c_0 &= P_0 \\
c_1 &= -5P_0 + 5P_1 \\
c_2 &= 10P_0 - 20P_1 + 10P_2 \\
c_3 &= -10P_0 + 30P_1 - 30P_2 + 10P_3 \\
c_4 &= 5P_0 - 20P_1 + 30P_2 - 20P_3 + 5P_4 \\
c_5 &= -P_0 + 5P_1 - 10P_2 + 10P_3 - 5P_4 + P_5.
\end{aligned}$$

The curvature κ at any point along the Bezier curve can be computed using the following expression:

$$\kappa(t) = \frac{\dot{B}(t) \times \ddot{B}(t)}{|\dot{B}(t)|^3}, \quad (2)$$

where $\dot{B}(t)$ and $\ddot{B}(t)$ represent the first and second order derivatives of $B(t)$ with respect to t, respectively, given by:

$$\dot{B}(t) = c_1 + 2c_2 t + 3c_3 t^2 + 4c_4 t^3 + 5c_5 t^4 \quad (3)$$

$$\ddot{B}(t) = 2c_2 + 6c_3 t + 12c_4 t^2 + 20c_5 t^3. \quad (4)$$

In this paper, a smooth lane change model is needed to learn driver behaviors, which requires the curvature at both ends of the path to be zero. Therefore, it is necessary that

$$\dot{B}(0) \times \ddot{B}(0) = 0 \quad (5)$$

$$\dot{B}(1) \times \ddot{B}(1) = 0. \quad (6)$$

By substituting $t = 0$ into (3), (4) and (5), we obtain the following:

$$5(P_1 - P_0) \times 20((P_2 - P_1) - (P_1 - P_0)) = 0. \quad (7)$$

Let v_{01} be the vector connecting P_0 to P_1, and v_{12} be the vector from P_1 to P_2, which are shown in Fig. 2. Then, $v_{01} = P_1 - P_0$ and $v_{12} = P_2 - P_1$, and (7) can then be simplified as:

$$5v_{01} \times 20(v_{12} - v_{01}) = 0. \quad (8)$$

For (8) to hold true, v_{01} and v_{12} must be parallel, implying that P_0, P_1, and P_2 must be colinear. This ensures that the curve starts in a straight line, guaranteeing zero curvature at the beginning of the lane change maneuver. Therefore, the constraint about zero curvature at the beginning of the curve is,

$$v_{01} \times v_{12} = 0. \quad (9)$$

Similarly, to ensure zero curvature at the end of the lane change ($t = 1$). Substituting $t = 1$ into (3), (4) and (6) yields:

$$v_{34} \times v_{45} = 0. \quad (10)$$

III. Maximum Entropy IRL

In this paper, the *MaxEnt* IRL, first introduced in [19], is used to learn the reward function from expert trajectories. With this method, each expert trajectory is considered as an actions, with the aim of inferring a reward function which can represent the driver's behavior. To achieve this, features that describe expert trajectories are extracted. These features are defined as functions that map specific aspects of a trajectory to real values, thereby creating a feature vector f that characterizes each trajectory. For each expert trajectory ζ_i, the corresponding feature vector is represented as $f(\zeta_i)$. To summarize the attributes of all expert trajectories, an average feature vector F representation is computed as follows:

$$F = \frac{1}{N} \sum_{i=1}^{N} f(\zeta_i), \quad (11)$$

which represents a generalized feature vector capturing the observed expert behaviors. The main objective of *MaxEnt* IRL is to ensure that the model generates feature expectations that align with those extracted from expert trajectories, i.e.,

$$\mathbb{E}_{p(\zeta|W)}[f] = F. \quad (12)$$

Instead of directly defining an optimal policy, *MaxEnt* IRL employs a reward function to quantify the drivers preference.

The reward function J is modeled as a weighted sum of features, represented by:

$$J = W^T f(\zeta_i), \tag{13}$$

where W is the weight vector to be learned and f is the feature vector defined earlier.

To determine the probabilities of different trajectories while introducing minimal bias and adequately representing expert trajectories, the maximum entropy principle is applied [19]. This yields the following form for the distribution over trajectories:

$$P(\zeta_i|W) = \frac{1}{Z(W)} e^{-J}, \tag{14}$$

where the partition function $Z(W) = \sum e^{-W^T f(\zeta_i)}$ serves to normalize the distribution, ensuring that the sum of the probabilities over all possible trajectories is equal to one. Note that distribution (14) in the MaxEnt IRL formulation is proposed by Ziebart et al. in [19]. As a normalized exponential family distribution, it assigns nonzero probability to all feasible trajectories. When multiple trajectories yield comparable rewards, each receives probability mass proportionally, naturally capturing multi-modal driving behaviors in this framework. Using the probability distribution (14) along with the set of expert trajectories $\tilde{\zeta}_i$, the weight vector W is inferred by maximizing the log-likelihood of the observed trajectories, i.e., $W = \arg\max\ L(W)$ where

$$L(W) = \sum_{i=1}^{N} \log P(\tilde{\zeta}_i|W). \tag{15}$$

However, analytical solution of this optimization problem is generally unavailable. Therefore, the gradient-based optimization is adopted, where the gradient of the log-likelihood can be expressed as:

$$\nabla L(W) = \mathbb{E}_{p(\zeta|W)}[f] - F. \tag{16}$$

Note that the log-likelihood $L(W)$ reaches its maximum when the gradient becomes zero, implying that the expected feature values match those derived from expert data. This condition aligns with the objective described in (12). The weight vector W can then be updated iteratively in the direction of the gradient as follows:

$$W = W + \eta \frac{\nabla L(W)}{|\nabla L(W)|}, \tag{17}$$

where η denotes the learning rate.

During the calculation of the gradient $\nabla L(W)$ in (16), evaluating the expected feature value $\mathbb{E}_{p(\zeta|W)}[f]$ over all possible trajectories is very complicated. Following literature [26], we therefore adopt the maximum likelihood approximation, in which the expectation is approximated using the feature values of the most likely trajectory under the current model:

$$\mathbb{E}_{p(\zeta|W)}[f] \approx f(\arg\max p(\zeta|W)). \tag{18}$$

This approximation allows for a more practical estimation of the gradient $\nabla L(W)$ during optimization, helping the model to adjust its weights efficiently and eventually reach a solution that can generate behavior similar to that of the provided expert data. As demonstrated in [26], such an approximation provides an efficient alternative to expensive sampling-based estimation, and is particularly suitable for modeling individual driving styles.

IV. Driver Lane Change Behavior Learning

This section presents the proposed approach for learning driver lane change behaviors using the MaxEnt IRL. We start with defining specific trajectory features for characterizing individual lane change trajectory, such as comfort, efficiency, and accuracy, followed by a comprehensive optimal control problem (OCP) formulation for determining the optimal control points of the Bezier curve. Finally, the complete privacy-preserving learning algorithm will be described towards the end of this section.

A. Feature Design

The learning algorithm presented in Section III is used to learn key features from expert lane change trajectories. In the following, the features f are defined, which are real-valued functions that characterize the specific aspects of lane change trajectories. These features are designed to effectively capture the preferences and decision-making criteria underlying lane change behaviors.

Comfort: Comfort is an important aspect of assessing driving quality, especially during lateral motion. To quantify comfort, a feature f_c is introduced that integrates the square of the curvature along the entire lane change path, indicating the sharpness of vehicle turns, defined as

$$f_c = \int_0^1 (\kappa(t))^2 dt, \tag{19}$$

where $\kappa(t)$ is defined in (2). Note that a smaller value of f_c signifies a smoother and more comfortable trajectory.

Traffic Efficiency: Traffic efficiency is a key feature that measures how effectively the vehicle completes a lane change in terms of both distance and timing. This feature helps understand how well the vehicle manages longitudinal movement during a lane change. The traffic efficiency is divided into two parts: the length of the lane change path (f_{T_1}) and the point at which the vehicle crosses the lane marking (f_{T_2}):

$$f_{T_1} = P_{5,x} - P_{0,x} \tag{20}$$

$$f_{T_2} = P_{cx} - P_{0x}. \tag{21}$$

In these equations, P_{ix} denotes the x-coordinate of the control point P_i, P_{cx} is the x-coordinate of the vehicle crossing the lane marking.

Lateral Offset: Lateral offset is another important feature for understanding lane change behavior, as it measures how well the vehicle ends up in the correct lateral position after a lane change. This feature provides insight into how precisely the vehicle aligns with the target lane. The lateral offset feature, represented by f_y, is defined as the square of the difference between the vehicle's final lateral position and the center of the target lane:

$$f_y = (P_{5,y} - y_c)^2, \tag{22}$$

where the y_c is the y-coordinate for the center of the target lane. This feature helps quantify the final positioning of the vehicle, contributing to an understanding of how well the vehicle completes the lane change.

These features (19)-(22) on comfort, longitudinal efficiency, and lateral accuracy collectively provide a comprehensive representation of lane change maneuvers. See Fig. 2 for examples of f_{T_1}, f_{T_2}, and f_y.

B. Complete Lane Change Model for IRL

A Bezier curve is fully determined once the coordinates of its control points are known. In the context of lane change modeling, the state space is represented by the coordinates of these control points. Therefore, in this study, the task of the lane change model is to solve an optimal control problem (OCP), where the goal is to find a set of control points that generate a Bezier curve minimizing a specific cost function. The initial control point P_0, which corresponds to the start of the lane change, is input of the this OCP. The decision variables of this OCP is denoted as $X = \left[P_{1,x}, P_{1,y}, P_{2,x}, P_{2,y}, P_{3_x}, P_{3_y}, P_{4,x}, P_{4,y}, P_{5,x}, P_{5,y}\right]^T$, where $P_{i.x}$ and $P_{i,y}$ are the coordinates of the control points P_i, respectively. The following optimization problem is set up to compute the desired Bezier curve:

$$\min_{X} \quad J(X) = W_1 f_c + W_2 f_{T_1} + W_3 f_{T_2} + W_4 f_y \tag{23a}$$

$$\text{s.t.} \quad B(t) = \sum_{i=0}^{5} P_i b_{i,5}(t) \tag{23b}$$

$$P_{i-1,x} < P_{i,x}, \quad i = 1, \ldots, 5 \tag{23c}$$

$$x_{\text{lb}} < P_{5,x} - P_{0,x} < x_{\text{ub}} \tag{23d}$$

$$y_{\text{lb}} < P_{5,y} < y_{\text{ub}} \tag{23e}$$

$$\text{Zero curvature constraints } (9) \text{ and } (10). \tag{23f}$$

In the objective function (23a), each term is defined in Section IV-A. The weights W_i, $i = 1, \ldots, 4$ correspond to each of these components, respectively, and are determined using the *MaxEnt* IRL approach which is described in Section III. Consequently, this cost function aligns with the reward function used in IRL. The constraint (23b) defines the Bezier curve as a 5th-order polynomial. Constraint (23c) ensures that the x-coordinates of the control points are strictly increasing, which maintains the correct direction for the lane change maneuver. Constraint (23d) limits the horizontal displacement, ensuring that the lane change occurs within an acceptable range. In this paper, the x_{lb} and x_{ub} are set to $15\ m$ and $50\ m$, respectively. Constraint (23e) restricts the y-coordinate of control point P_5 to ensure that the vehicle completes the lane change maneuver within the target lane, where y_{lb} and y_{ub} representing the lower and upper boundaries of the target lane, respectively, which are the $3.6\ m$ and $7.2\ m$ in this paper. Constraints (23f) enforce zero curvature at the beginning and end of the trajectory, as discussed in Section II.

Remark 1: Remark 1: This paper focuses on lane change when surrounding vehicles are further away from the ego vehicle. Under this setting, safety-related metrics such as collision risk are not applicable, and they are not included as features for IRL training nor incorporated in the optimization problem (23). Such metrics can be incorporated in future extensions that address multi-vehicle traffic environments, where safety-related constraints will also be added to (23)

C. Privacy-preserving IRL for Lane Change

In this section, the proposed privacy-preserving *MaxEnt* IRL algorithm is presented, which learns driver lane change behavior from encrypted expert features. As illustrated in Fig. 3, before sending the expert features to the IRL module (assumed to be executed in cloud), the real user lane change trajectory is masked by a random scaling factor a. More specifically, the trajectory is elongated in the longitudinal direction by a factor of a. This scaling alters key trajectory characteristics such as curvature and path length, thereby encrypting the true feature values that represent user preferences. The scaled trajectories are then used to extract encrypted feature vectors, which are uploaded to the cloud for MaxEnt IRL training. After processing through IRL, a set of masked weights is obtained, which is then passed to the local onboard Bezier lane change model that solves (23) based on the current vehicle position. The resulted lane change trajectory mimic the behavior based on the *encrypt* expert features, and hence possesses different path than the original expert trajectory. Therefore, this trajectory obtained by solving (23) using the masked weight then needs to be descaled back by the same factor a to return to its original property.

The proposed privacy-preserving mechanism is illustrated in Fig. 4, where $a = 1$ corresponds to the original user lane change trajectory. Specifically, the x-axis is scaled using the scaling factor a, which means that as a increases, the original path is elongated in the x-direction. As demonstrated in Fig. 4, with increasing scaling factors, trajectories extend correspondingly, impacting trajectory features such as comfort, and path length. Table I quantifies these changes in detail. For instance, the feature f_{T_1}, representing the length of the lane-changing path, increases with the scaling factor a. Similarly, the feature about the length cross the lane marking f_{T_2} also increases. Conversely, the feature f_c, associated with curvature, decreases because the elongation results in a flatter trajectory. It is worth noting that when the scaling factor $a = 1$, the encrypted trajectories ($\tilde{\zeta}$) are identical to the original expert trajectories (ζ), meaning the scaling does not alter any features of the original data.

Definition 1 (∞-Feature-Diversity): An encrypted feature vector $\tilde{f}$ is said to satisfy ∞-*Feature-Diversity* if, for any observed encrypted feature vector $\tilde{f}$, there exists an infinite set of trajectory and scale factor pairs $(a_i, \zeta_i)_{i=1}^{\infty}$, such that:

$$\tilde{f} = f(a_i, \zeta_i), \quad \forall i \tag{24}$$

In other words, infinitely many possible original trajectories, when scaled by their corresponding factors, produce the same encrypted feature vector. This implies that the true identity of the original driving behavior cannot be uniquely determined from $\tilde{f}$.

Theorem 1: The proposed longitudinal scaling-based feature encryption mechanism guarantees ∞-Feature-Diversity.

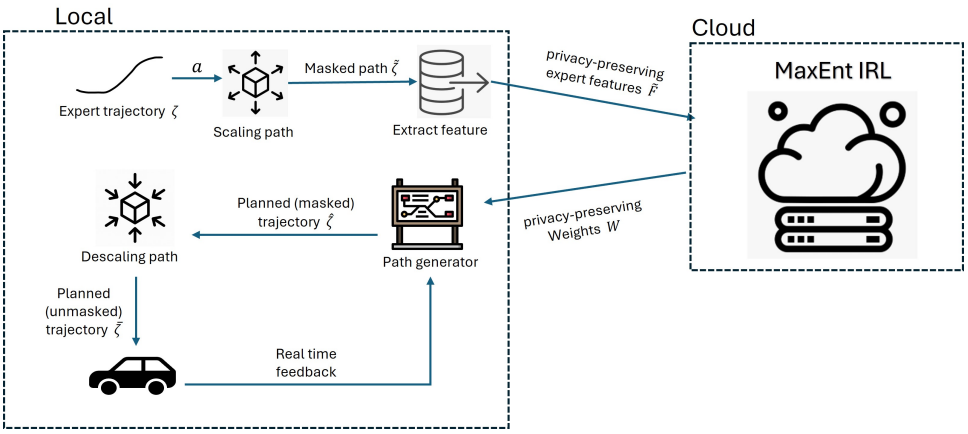

Fig. 3. Diagram of the proposed privacy-preserving *MaxEnt* IRL framework for personalizing lane change behavior.

There the cloud-based model cannot uniquely infer the original trajectory or behavior preference from the encrypted feature vector alone.

Proof: Given an encrypted trajectory $\tilde{\zeta} = (\tilde{\zeta}_x, \tilde{\zeta}_y)$, the encryption mechanism applies a positive scaling factor $a \in \mathbb{R}^+$ along the longitudinal (x) axis on the original trajectory $\zeta = (\zeta_x, \zeta_y)$, such that:

$$\tilde{\zeta}_x = a \cdot \zeta_x, \quad \tilde{\zeta}_y = \zeta_y. \tag{25}$$

To recover a candidate original path, for any arbitrary $a_i \in \mathbb{R}^+$, we define:

$$\zeta_{i,x} = \frac{1}{a_i} \cdot \tilde{\zeta}_{i,x}, \quad \zeta_{i,y} = \tilde{\zeta}_{i,y}. \tag{26}$$

which means that the same encrypted trajectory $\tilde{\zeta}$ can be generated by infinitely many trajectory–scaling factor pairs (a_i, ζ_i).

Furthermore, since the feature is deterministic and depends solely on the geometry of the trajectory, the same encrypted feature vector is obtained:

$$f(\tilde{\zeta}) = f(a_i \cdot \zeta_{i,x}, \zeta_{i,y}) = \tilde{f}, \quad \forall a_i \in \mathbb{R}^+. \tag{27}$$

Therefore, the encrypted feature vector $\tilde{f}$ can also be produced by infinitely many candidate pairs (a_i, ζ_i), each consistent with the same encrypted path $\tilde{\zeta}$.

This indicates that the encrypted feature $\tilde{f}$ does not uniquely determine the original trajectory and original behavior preference, and hence, the proposed encryption method satisfies ∞-Feature-Diversity. ■

Remark 2: According to Theorem 1, the proposed privacy protection mechanism ensures ∞-Feature-Diversity: an encrypted feature vector $\tilde{f}$ can correspond to infinitely many factor and trajectory pairs (a_i, ζ_i), all consistent with the same encrypted feature. Since the scaling factor a is retained locally and never shared with the cloud, it is computationally infeasible for the cloud or an attacker to infer the original trajectory ζ or the true behavioral features $f(\zeta)$. It is clear from Table I that, without the knowledge of a, one can never infer the true value of f_c, f_{T_1}, and f_{T_2}. Furthermore, if an attacker gains knowledge of the weight W and OCP (23), they can only reconstruct a trajectory aligned with $\tilde{f}$, but this does not reveal the unique original behavior.

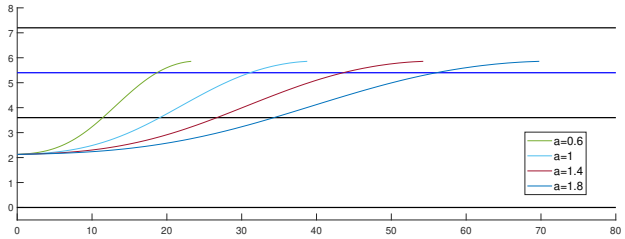

Fig. 4. Illustration of the proposed privacy-preserving mechanism. Note that $a = 1$ corresponds to the original expert trajectory without any privacy protection.

TABLE I
FEATURES WITH DIFFERENT SCALING FACTORS FOR THE EXAMPLE IN FIG. 4.

Scaling factor a	f_c	f_{T_1}	f_y	f_{T_2}
0.6	1.02×10^{-3}	23.26	0.0209	11.43
1.0	1.39×10^{-4}	38.76	0.0209	19.05
1.4	3.66×10^{-5}	54.27	0.0209	26.68
1.8	1.35×10^{-5}	69.77	0.0209	34.29

Remark 3: It is also worth noting that the lateral offset feature f_y is not encrypted. This feature measures the geometric distance between the vehicle and the center line of the target lane. Though each driver may prefer to deviate from the center line due to different personal preference, such deviation usually is minimum due to safety reasons. Therefore, its value may not carry much private information as they are very close to zero. Consequently, while f_y is treated as one of the features during IRL training, it is left unencrypted to avoid unnecessary processing without affecting privacy.

The complete privacy-preserving IRL algorithm is outlined in Algorithm 1. Specifically, the inputs are the encrypted feature vector and all the initial points in the dataset. Here, it is worth to mention that to reduce the influence of large numerical differences among features, since features with large

Algorithm 1 *MaxEnt* IRL Driver Behavior Learning Algorithm

Input: Encrypted expert feature vector: $\tilde{F}$, Beginning points in dataset: P_0

Output: W

1: Initialize $i \leftarrow 0$, $W \leftarrow$ all-ones vector, $\eta \leftarrow 0.1$;
2: **while** W not converge **do**
3: **for** all P_0 in D **do**
4: $\zeta_{opt,i} \leftarrow$ solving OCP (23) with aligned P_0;
5: $f_{opt,i} \leftarrow f(\zeta_{opt,i})$;
6: **end for**
7: $\mathbb{E}_{p(\zeta|W)}[f] \leftarrow \frac{1}{N}\sum_{i=1}^{N} f_{opt,i}$;
8: $\nabla L \leftarrow \mathbb{E}_{p(\zeta|W)}[f] - \tilde{F}$;
9: $W \leftarrow W + \eta \frac{\nabla L}{\|\nabla L\|}$;
10: $i \leftarrow i + 1$;
11: **if** $i = 200$ **then**
12: $\eta \leftarrow \eta/2$;
13: $i \leftarrow 0$;
14: **end if**
15: **end while**

values dominate the cost function, each feature is normalized to the range $[0, 1]$ based on masked expert feature distribution. Line 1 initializes the weight (W), iteration counter (i) and learning rate (η). In Lines 3-6, the OCP (23) with current weight is solved to get the optimal path (ζ_{opt}) and its normalized features (f_{opt}). Line 7 calculates the expected features $(\mathbb{E}_{p(\zeta|W)}[f])$ by averaging f_{opt}. The weight vector (W) is then updated via gradient ascent in Lines 8-10. The learning rate (η) is gradually reduced over iterations to ensure stable convergence in Lines 12-14.

V. Experiments and Results

This section presents the numerical evaluation of the privacy-preserving lane change framework. The results test the model's performance in various test scenarios, using both driver simulator and real-world experiments. This paper focuses on personalized lane change modeling, where each driver's preferences are learned individually.

A. Experiment Setup

In this study, we use a driver-in-the-loop driving simulator integrated with CARLA [38], as shown in Fig. 5, to collect realistic lane change driving data. The steering wheel, gear shift, and pedals are Logitech G923 TRUEFORCE Racing compatible for XBox, Playstation, and PC. Three monitors are aligned horizontally to provide a wide angle of view to mimic a realistic ride experience. An Alienware Aurora R15 gaming desktop is used to provide the necessary computing power for running CARLA. The driver simulator runs at around 20 frames per second, providing sufficient smoothness for human drivers. Coupled with Roadrunner [40], the driver simulator can be customized with urban, suburban, and rural driving scenarios, allowing realistic evaluations to be conducted. In this experiment, two driving datasets are collected at constant speeds of 10 m/s and 15 m/s, respectively, to capture a range

Fig. 5. The driver-in-the-loop simulator with CARLA used by Driver 1 & Driver 2.

of driving behaviors and lane change maneuvers under various conditions.

In addition to the driver simulator described above, real-world experiments are also conducted using a Polaris Gem e2 to create a more realistic dataset. As shown is Fig. 6, the golf cart is equipped with a high-precision GPS system to record the path during the lane change. Due to the small size of the test area (Fig. 7), the golf cart is driven at a low speed of 4.5 m/s.

A total of three datasets are then obtained: two using the driver-in-the-loop driving simulator (speeds of 10 m/s and 15 m/s), referred to as Driver 1 and Driver 2, and one from real-world driving tests using the Gem e2, referred to as Driver 3. For each dataset, 40 lane change trajectories are collected, with 35 paths randomly selected for training, while the remaining 5 paths are used for testing. A separate IRL model is trained for each driver (Driver 1–3) and data is not pooled across drivers. In this paper, the initial learning rate is set to $\eta = 0.1$, and the IRL is terminated after a maximum of 2000 iterations or earlier if convergence is reached.

B. Test Results without Privacy Protection

For the case without privacy protection ($a = 1$), the training results across different datasets are presented in Fig. 8, where the expert lane change trajectories (ζ) are depicted as gray lines and the unmasked trajectories $(\bar{\zeta})$ are shown as red lines. The results indicate a strong alignment between the unmasked trajectories and expert demonstrations, confirming the IRL model's capability to accurately replicate personalized driving behaviors. As shown in Table II, both absolute and relative errors between ζ and $\bar{\zeta}$ are low across all features. While f_{T_1} and f_{T_2} are highly accurate, the larger relative errors in f_c and f_y are due to the small magnitude of their feature values. Specifically, in Table II, the maximum absolute error in path length (f_{T_1}) is only 3.53 m, corresponding to a relative error below 10%. The maximum absolute error in lane mark crossing point (f_{T_2}) is 1.78 m, also under 10% relative error. Although the relative errors for curvature (f_c) and lateral offset (f_y) can reach up to 58%, their absolute deviations remain

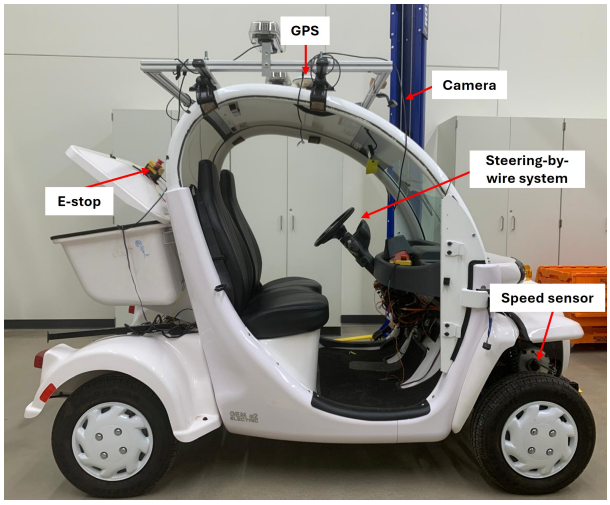

Fig. 6. The Polaris Gem e2, with highly automated driving systems, used by Driver 3.

TABLE II
ABSOLUTE ERRORS (e_a) AND RELATIVE ERRORS (e_r) OF FEATURES BETWEEN ζ AND $\bar{\zeta}$.

Dataset		f_c	f_{T_1}	f_y	f_{T_2}
Driver 1	e_a	$8.45 \times 10^{-}5$	3.17	$3.4 \times 10^{-}2$	0.50
	e_r (%)	39.93	8.63	29.11	2.64
Driver 2	e_a	$7.16 \times 10^{-}5$	3.53	$9.26 \times 10^{-}3$	1.78
	e_r (%)	55.96	9.25	9.63	9.26
Driver 3	e_a	$2.20 \times 10^{-}3$	0.53	$7.14 \times 10^{-}2$	0.51
	e_r (%)	57.89	2.81	59.21	5.33

within 10^{-5} to 10^{-3} and 10^{-3} to 10^{-2}, respectively, which is negligible in practice. These results confirm that, without any privacy transformation, our IRL model reproduces both the longitudinal and lateral characteristics of individual drivers' lane changes across all datasets.

C. Test Results with Privacy Protection

This section presents test results of the proposed privacy-preserving IRL personalization algorithm with different scale factors. Figs. 9, 10, and 11 show the results over test trajectories for different datasets. In these figures, The blue bars represent the average masked features of the five planned paths for each dataset. Since the masked planned path ($\hat{\zeta}$) are learned from encrypted expert features ($\tilde{f}$), the resulting masked features varies with a, consistent with our earlier discussion. Therefore, if an attacker gains access to the privacy-preserving weights (and so the masked prediction), he/she has no clue what the unmasked trajectory and unmasked features looks like. On the other hand, the goal of the proposed learning algorithm is to make the unmasked trajectories ($\bar{\zeta}$) closely matches the expert paths (ζ). Therefore, the average features of the unmasked trajectories under each condition are recorded as the orange bar in Figs. 9, 10 and 11. As can be seen, after scaling back, the features under every scaling factor maintain similar values. The solid horizon line in each subplot presents the average features for the training expert paths in each dataset. It can be observed that the average feature values for the unmasked paths ($\bar{\zeta}$) are similar to those of the corresponding training expert paths (ζ), indicating that the proposed privacy-preserving mechanism does not sacrifice learning accuracy.

Table III presents the absolute errors (e_a) and relative errors (e_r) between the average features of the test paths and the average features of the unmasked trajectories across three datasets (Driver 1, Driver 2, and Driver 3). The evaluation is conducted under varying scale factors from 0.4 to 2.0. It can be observed that the relative errors for features f_{T_1} and f_{T_2} consistently remain below 10% across all scale factors and datasets. For instance, in the case of Driver 1, the relative error for f_{T_1} varies from 7.51% to 8.63%, while for f_{T_2} it ranges from 1.90% to 2.64%. The corresponding absolute errors are also low, with f_{T_1} showing values varies from $2.76m$ and $3.17m$, and f_{T_2} ranging from $0.36m$ to $0.5m$. These small differences indicate that the IRL learned policy accurately reproduces the trajectory features observed in expert demonstrations.

In contrast, the features f_c and f_y show higher relative errors, reaching approximately 58 to 60% in certain instances, particularly in the Driver 2 and Driver 3 dataset. However, these features have much smaller absolute magnitudes. For example, the absolute error of f_c for Driver 2 remains as low as 7.14×10^{-5} at scale factor 2.0. Similarly, for Driver 3, the absolute error of f_y is approximately 7.25×10^{-2}. These results indicate that even though the relative percentage error appears high, the actual deviation is actually minimal.

Furthermore, the last column in Table III shows the standard deviation (σ) of the e_a and e_r across all scale factors. The small values of σ, consistent across all datasets and features, demonstrate that the encryption process does not degrade the learning performance of the IRL model. These findings confirm that the proposed framework is capable of accurately capturing the essential features of driver lane change behavior, while also exhibiting robustness to data scaling and privacy-preserving transformations.

Moreover, to evaluate the computation requirement for real-time deployment, the runtime of the proposed framework is measured. The simulation is conducted on a standard desktop computer equipped with an AMD Ryzen 9 3.5 GHz CPU and 32 GB RAM. The computation time required to generate one lane change trajectory is around 140 ms, which is efficient for real-time operation

D. Comparison with Differential Privacy

To further examine the performance of the proposed scaling-based privacy protection, the differential privacy (DP) method proposed in [41] is considered as the baseline. In order to apply DP in IRL, we add Gaussian noises to encrypt the features for every expert trajectories. To satisfy (ϵ, δ) differential privacy, the standard deviation is defined as follows:

$$\sigma = \frac{\Delta f \sqrt{2 \ln(1.25/\delta)}}{\epsilon}, \tag{28}$$

where Δf is the sensitivity of the feature, ϵ is the privacy budget, and δ is the probability of exceeding the privacy loss

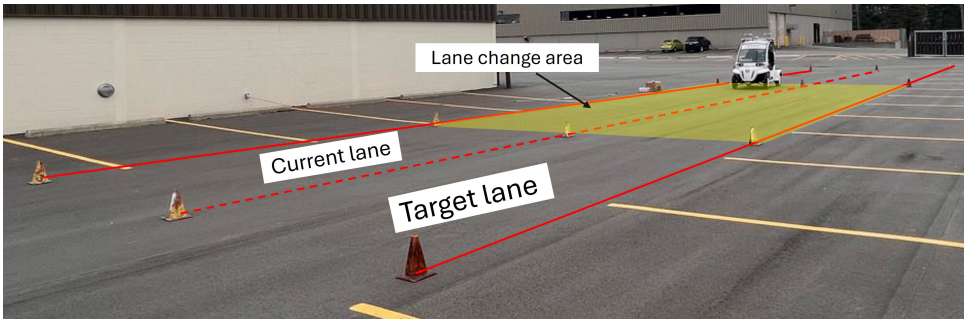

Fig. 7. The lane change test environment for Driver 3.

TABLE III
ABSOLUTE ERROR (e_a), RELATIVE ERROR (e_r) BETWEEN UNMASKED PATH FEATURE AND EXPERT FEATURE, AND THE STANDARD DEVIATION

Dataset	Feature \ Scale factor		0.4	0.6	0.8	1.0	1.2	1.4	1.6	1.8	2.0	σ
Driver 1	f_c	e_a	9.87e-5	9.60e-5	9.54e-5	8.45e-5	8.49e-5	8.50e-5	8.50e-5	8.51e-5	8.51e-5	5.61e-6
		e_r (%)	46.67	45.37	45.13	39.93	40.07	40.12	40.17	40.22	40.22	2.67
	f_{T_1}	e_a	2.76	2.96	3.05	3.17	3.16	3.15	3.14	3.13	3.13	0.13
		e_r (%)	7.51	8.06	8.17	8.63	8.61	8.58	8.55	8.52	8.52	0.35
	f_y	e_a	2.88e-2	3.16e-2	3.25e-2	3.39e-2	3.40e-2	3.40e-2	3.40e-2	3.40e-2	3.41e-2	1.7e-3
		e_r (%)	24.62	27.01	27.78	29.11	29.13	29.14	29.14	29.14	29.15	1.48
	f_{T_2}	e_a	0.36	0.46	0.47	0.50	0.49	0.48	0.48	0.48	0.47	3.89e-2
		e_r (%)	1.90	2.43	2.48	2.64	2.58	2.53	2.53	2.51	2.48	0.21
Driver 2	f_c	e_a	7.43e-5	7.43e-5	7.16e-5	7.16e-5	7.16e-5	7.15e-5	7.15e-5	7.14e-5	7.14e-5	1.16e-6
		e_r (%)	58.12	58.10	55.99	55.95	55.94	55.92	55.92	55.91	55.91	0.91
	f_{T_1}	e_a	3.80	3.77	3.53	3.53	3.53	3.53	3.53	3.53	3.53	0.11
		e_r (%)	9.96	9.88	9.25	9.25	9.25	9.25	9.25	9.25	9.25	0.28
	f_y	e_a	1.72e-3	4.87e-3	9.66e-3	9.26e-3	9.06e-3	8.86e-3	8.76e-3	8.76e-3	8.66e-3	2.5e-3
		e_r (%)	1.79	5.05	10.06	9.63	9.42	9.21	9.11	9.11	9.00	2.61
	f_{T_2}	e_a	1.95	1.91	1.78	1.78	1.78	1.77	1.77	1.77	1.77	6.56e-2
		e_r (%)	10.15	9.94	9.26	9.26	9.26	9.21	9.21	9.21	9.21	0.34
Driver 3	f_c	e_a	1.83e-3	2.14e-3	2.19e-3	2.20e-3	2.20e-3	2.21e-3	2.22e-3	2.21e-3	2.21e-3	1.18e-4
		e_r (%)	48.21	56.53	57.58	57.89	58.00	58.05	58.50	57.05	57.05	3
	f_{T_1}	e_a	0.16	0.42	0.51	0.53	0.54	0.54	0.67	0.55	0.55	0.13
		e_r (%)	0.87	2.23	2.71	2.81	2.87	2.87	3.56	2.92	2.92	0.70
	f_y	e_a	7.16e-2	7.56e-2	7.25e-2	7.14e-2	7.15e-2	7.11e-2	7.19e-2	7.09e-2	7.09e-2	1.38e-3
		e_r (%)	59.34	62.69	60.06	59.21	59.23	58.95	59.59	58.80	58.73	1.15
	f_{T_2}	e_a	0.92	0.60	0.54	0.51	0.50	0.49	0.49	0.48	0.48	0.13
		e_r (%)	9.14	5.91	5.33	5.07	4.94	4.85	4.86	4.80	4.79	1.32

threshold [42], [43]. In this paper, Δf represents the maximum feature difference between each expert trajectory. To ensure strong privacy protection, ϵ and δ are set to 0.5 and 1×10^{-5}, respectively. The corresponding IRL training results are shown in Fig. 12, where the gray lines represent expert trajectories and red lines represent the planned path under DP method. The red dashed lines indicate the planned trajectories using our proposed scale-based transformation method, which is reproduced from Fig. 8. It can be observed that the DP-based planned paths deviate significantly from the expert demonstrations, indicating that IRL fails to learn the personal reward function due to DP-based privacy-preserving transformation. Table IV compares the relative errors between the proposed privacy-preserving mechanism and DP, which clearly shows that the proposed method generally achieves lower relative errors across most datasets and features, indicating more stable performance. In contrast, DP is unstable and produces inconsistent results. In the case of Driver 2, the relative errors of f_c and f_c and f_y can exceed 500%.

Remark 4: Differential privacy (DP) is selected as the baseline since it is the most widely adopted and representative method in privacy-preserving machine learning. Other approaches such as federated learning (FL), partial feature masking (PFM), and homomorphic encryption (HE) operate under different assumptions than the one we consider in this paper. FL is designed for multi-client settings where raw data remain on local devices and only model updates are shared. PFM discards part of the feature set and may significantly reduce the information available for IRL, while HE enables secure computation but usually requires very high computational cost, which limits its use in real-time applications. In contrast, scaling-based encryption preserves all features while protecting their values with low complexity, making DP the most meaningful baseline for comparison.

Overall, the experimental results demonstrate that the proposed method not only generates lane change trajectories that are highly consistent with expert demonstrations but also maintains robust performance across different driving behavior

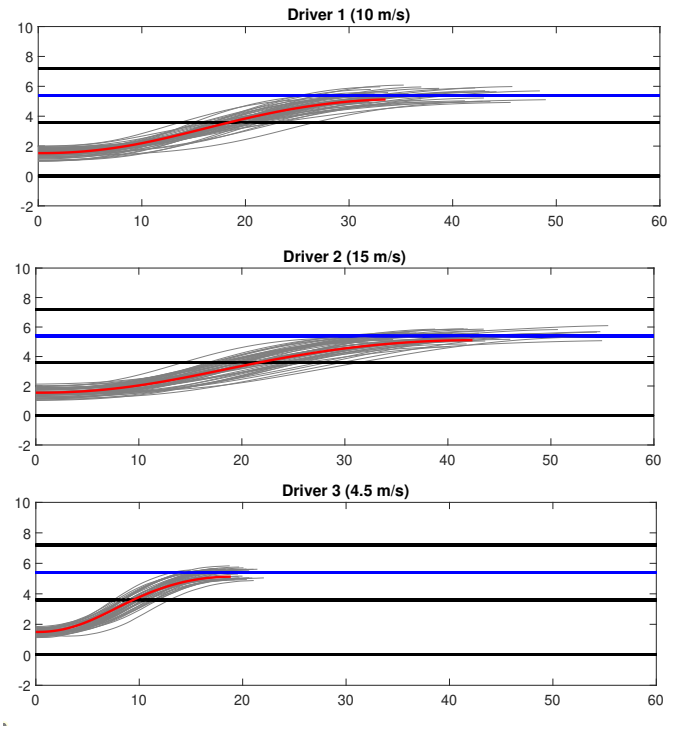

Fig. 8. Comparison of unmasked trajectories and expert trajectories across different datasets at $a = 1$. Black lines represent the lane markings and the blue lines indicate the middle of the target lane. Gray: expert lane change trajectories. Red: planned trajectories.

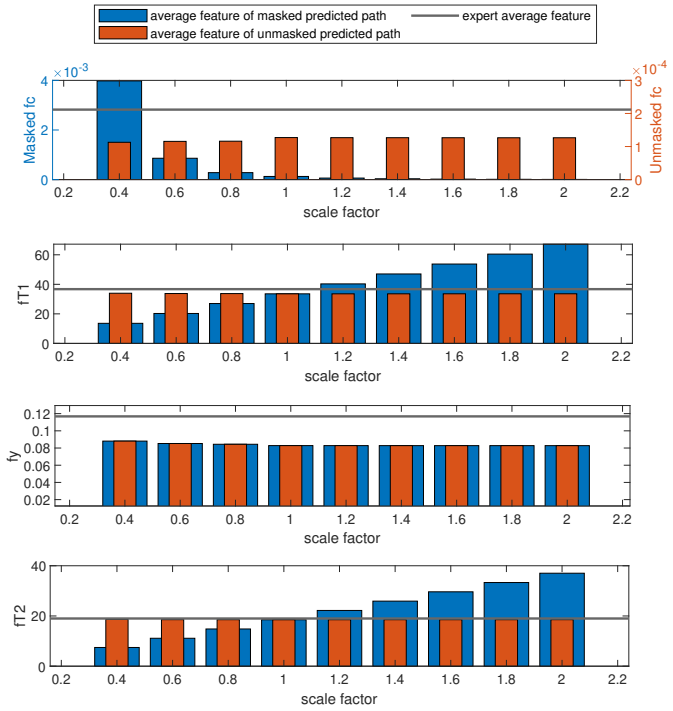

Fig. 9. Test results of Driver 1 (10 m/s). Blue: average features of $\hat{\zeta}$ (masked). Orange: average features of $\bar{\zeta}$ (unmasked).

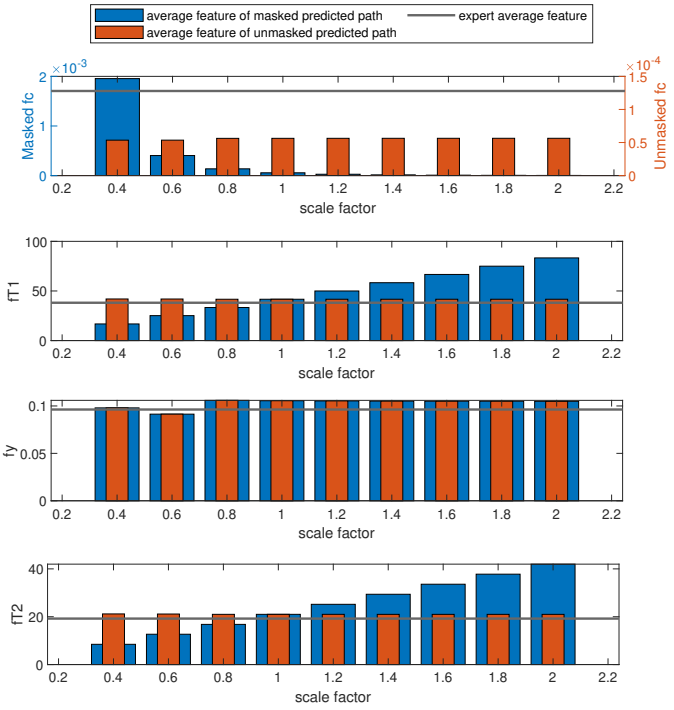

Fig. 10. Test results of Driver 2 (15 m/s). Blue: average features of $\hat{\zeta}$ (masked). Orange: average features of $\bar{\zeta}$ (unmasked).

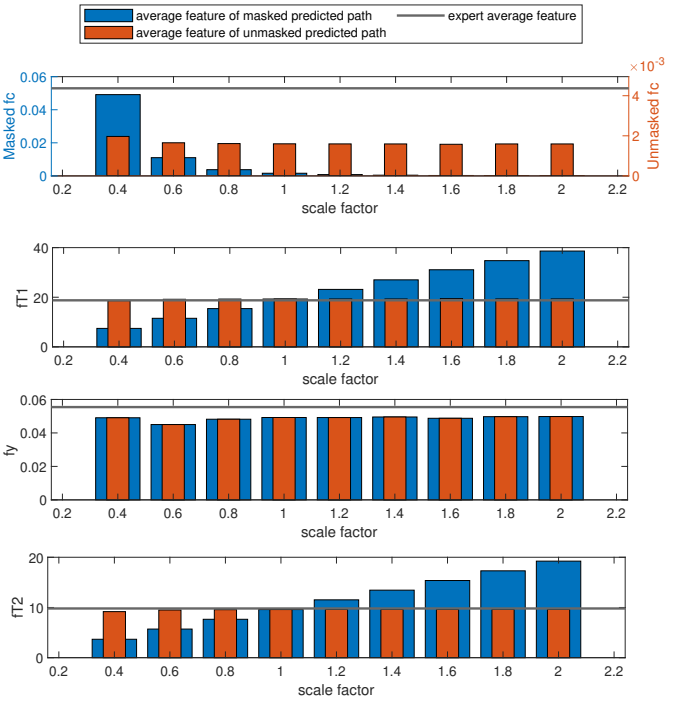

Fig. 11. Test results of Driver 3 (4.5 m/s). Blue: average features of $\hat{\zeta}$ (masked). Orange: average features of $\bar{\zeta}$ (unmasked).

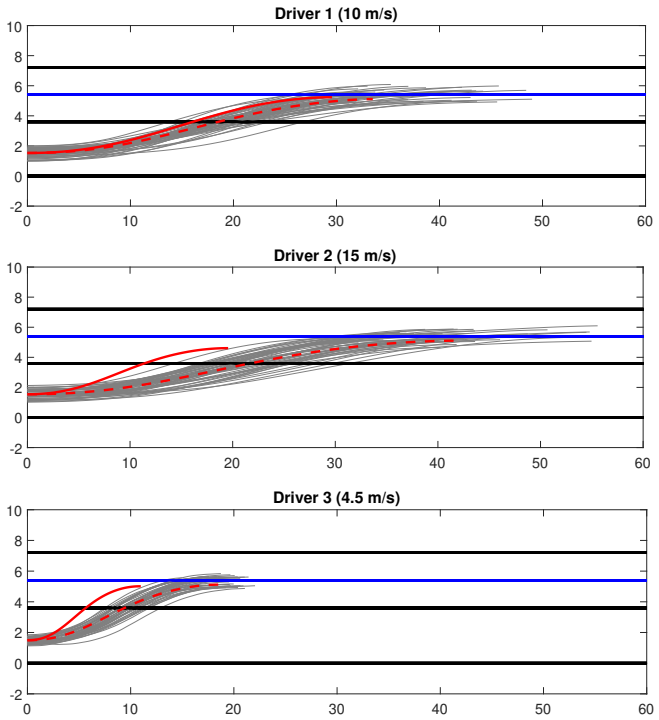

Fig. 12. IRL training results with different datasets using DP-based privacy protection (red solid line). Red dashed lines represent the training results using our proposed privacy protection mechanism. Black lines represent the lane markings and the blue lines indicate the middle of the target lane.

TABLE IV
RELATIVE ERRORS UNDER DIFFERENT PREVENTING-PRIVACY METHODS [%]

Dataset	Method	f_c	f_{T_1}	f_y	f_{T_2}
Driver 1	Proposed	41.98	**8.35**	**28.24**	**2.45**
	DP	**4.27**	19.75	76.19	16.39
Driver 2	Proposed	**56.42**	**9.39**	**8.03**	**9.41**
	DP	514.67	49.36	551.47	41.94
Driver 3	Proposed	**56.77**	**2.63**	59.62	**5.52**
	DP	195.48	40.22	**20.74**	43.24

and under scale-based transformation.

VI. CONCLUSIONS

This paper introduces a privacy-preserving IRL-based framework for learning and generating personalized lane change trajectories, leveraging Bezier curves to model vehicle paths and *MaxEnt* IRL to infer a reward function from expert trajectories. The use of Bezier curves provides flexible and smooth lane change path, making them well-suited for personalized driving applications. To address user privacy concerns during cloud-based training, the scale-based transformation is applied to the user provided lane change trajectories, ensuring that user-specific information is anonymized without degrading model performance. Experiments are conducted using both driver simulator and real world test platform, in which the proposed model successfully learns individualized lane change behaviors from different driving data. The results indicate that the planned trajectories remain similar with the original expert paths with and without privacy-preserving scaling, demonstrating the efficacy of the encryption method for protecting user data. In future work, further investigation of additional initial conditions that may influence lane change behavior will be a significant step forward. While Bezier curves are efficient, they may be less suitable for highly dynamic lane change scenarios which need to respond to surrounding vehicles or sudden events. Further extension will explore more adaptive trajectories such as splines to improve flexibility in such settings. Moreover, integration with model predictive control for lateral motion control [44]–[46] deserves further investigation.

REFERENCES

[1] Z. Zhou and J. Chen, "Modeling driver lane change behavior using inverse reinforcement learning," in *2024 IEEE 3rd International Conference on Computing and Machine Intelligence (ICMI), Mt Pleasant, MI, USA, April 13–14, 2024*, pp. 1–5.

[2] C. Ma and D. Li, "A review of vehicle lane change research," *Physica A: Statistical Mechanics and its Applications*, vol. 626, p. 129060, September 2023.

[3] W. Wang, T. Qie, C. Yang, W. Liu, C. Xiang, and K. Huang, "An intelligent lane-changing behavior prediction and decision-making strategy for an autonomous vehicle," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 3, pp. 2927–2937, March 2022.

[4] Z. Liu, Z. Wang, B. Yang, and K. Nakano, "Learning personalized discretionary lane-change initiation for fully autonomous driving based on reinforcement learning," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Totonto, ON, Canada, October 11-14, 2020*, pp. 457–463.

[5] Y. Ding, W. Zhuang, L. Wang, J. Liu, L. Guvenc, and Z. Li, "Safe and optimal lane-change path planning for automated driving," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 235, no. 4, pp. 1070–1083, May 2021.

[6] H. Li, Y. Luo, and J. Wu, "Collision-free path planning for intelligent vehicles based on bézier curve," *IEEE Access*, vol. 7, pp. 123 334–123 340, August 2019.

[7] R. Lattarulo, L. González, E. Martí, J. Matute, M. Marcano, and J. Pérez, "Urban motion planning framework based on n-bézier curves considering comfort and safety," *Journal of Advanced Transportation*, vol. 2018, July 2018.

[8] V. G. Lopez, F. L. Lewis, M. Liu, Y. Wan, S. Nageshrao, and D. Filev, "Game-theoretic lane-changing decision making and payoff learning for autonomous vehicles," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 4, pp. 3609–3620, April 2022.

[9] K. Gao, X. Li, B. Chen, L. Hu, J. Liu, R. Du, and Y. Li, "Dual transformer based prediction for lane change intentions and trajectories in mixed traffic environment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 6, pp. 6203–6216, June 2023.

[10] Y. Xia, Z. Qu, Z. Sun, and Z. Li, "A human-like model to understand surrounding vehicles' lane changing intentions for autonomous driving," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 5, pp. 4178–4189, May 2021.

[11] S. Yang, H. Zheng, J. Wang, and A. E. Kamel, "A personalized human-like lane-changing trajectory planning method for automated driving system," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 7, pp. 6399–6414, July 2021.

[12] X. He, H. Yang, Z. Hu, and C. Lv, "Robust lane change decision making for autonomous vehicles: An observation adversarial reinforcement learning approach," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 1, pp. 184–193, April 2022.

[13] G. Li, Y. Yang, S. Li, X. Qu, N. Lyu, and S. E. Li, "Decision making of autonomous vehicles in lane change scenarios: Deep reinforcement learning approaches with risk awareness," *Transportation Research Part C: Emerging Technologies*, vol. 134, p. 103452, January 2022.

[14] R. Chen, I. C. Paschalidis *et al.*, "Distributionally robust learning," *Foundations and Trends® in Optimization*, vol. 4, no. 1-2, pp. 1–243, 2020.

[15] S. Ibrahim, M. Mostafa, A. Jnadi, H. Salloum, and P. Osinenko, "Comprehensive overview of reward engineering and shaping in advancing reinforcement learning applications," *IEEE Access*, vol. 12, pp. 175 473–175 500, November 2024.

[16] A. Irshayyid, J. Chen, and G. Xiong, "A review on reinforcement learning-based highway autonomous vehicle control," *Green Energy and Intelligent Transportation*, vol. 3, no. 4, p. 100156, August 2024.

[17] T. Phan-Minh, F. Howington, T.-S. Chu, S. U. Lee, M. S. Tomov, N. Li, C. Dicle, S. Findler, F. Suarez-Ruiz, R. Beaudoin *et al.*, "Driving in real life with inverse reinforcement learning," *arXiv preprint arXiv:2206.03004*, 2022.

[18] Y. Ma and J. Wang, "Personalized driving behaviors and fuel economy over realistic commute traffic: Modeling, correlation, and prediction," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 7, pp. 7084–7094, April 2022.

[19] B. D. Ziebart, A. L. Maas, J. A. Bagnell, A. K. Dey *et al.*, "Maximum entropy inverse reinforcement learning." in *23rd AAAI Conference on Artificial Intelligence, Chicago, IL, USA, July 13-17, 2008*, vol. 8, pp. 1433–1438.

[20] S. Arora and P. Doshi, "A survey of inverse reinforcement learning: Challenges, methods and progress," *Artificial Intelligence*, vol. 297, p. 103500, August 2021.

[21] Z. Huang, J. Wu, and C. Lv, "Driving behavior modeling using naturalistic human driving data with inverse reinforcement learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 10 239–10 251, August 2022.

[22] J. Liu, L. N. Boyle, and A. G. Banerjee, "An inverse reinforcement learning approach for customizing automated lane change systems," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 9, pp. 9261–9271, September 2022.

[23] J. A. R. da Silva, V. Grassi, and D. F. Wolf, "Maximum entropy inverse reinforcement learning using monte carlo tree search for autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 9, pp. 11 552–11 562, September 2024.

[24] Z. Wu, F. Qu, L. Yang, and J. Gong, "Human-like decision making for autonomous vehicles at the intersection using inverse reinforcement learning," *Sensors*, vol. 22, no. 12, p. 4500, June 2022.

[25] M. F. Ozkan and Y. Ma, "Personalized adaptive cruise control and impacts on mixed traffic," in *2021 American Control Conference (ACC), New Orleans, LA, USA, May 25-28, 2021*, pp. 412–417.

[26] M. Kuderer, S. Gulati, and W. Burgard, "Learning driving styles for autonomous vehicles from demonstration," in *2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26-30 May, 2015*, pp. 2641–2646.

[27] B. Zhang and D. Zhu, "A new method on motion planning for mobile robots using jump point search and bezier curves," *International Journal of Advanced Robotic Systems*, vol. 18, no. 4, p. 17298814211019220, July 2021.

[28] B. S. Oldaç, E. E. Özdemir, and F. Kosova, "A path tracking and collision prevention control system for an electric vehicle with trajectories generated by bezier curves," in *2024 11th International Conference on Electrical and Electronics Engineering (ICEEE), Marmaris, Turkiye, April 22-24, 2024*, pp. 146–151.

[29] J. Chen, P. Zhao, T. Mei, and H. Liang, "Lane change path planning based on piecewise bezier curve for autonomous vehicle," in *Proceedings of 2013 IEEE International Conference on Vehicular Electronics and Safety, Dongguan, China, July 28-30, 2013*, pp. 17–22.

[30] P. Schafhalter, S. Kalra, L. Xu, J. E. Gonzalez, and I. Stoica, "Leveraging cloud computing to make autonomous vehicles safer," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Detroit, MI, USA, October 01-05, 2023*, pp. 5559–5566.

[31] N. Ponomareva, H. Hazimeh, A. Kurakin, Z. Xu, C. Denison, H. B. McMahan, S. Vassilvitskii, S. Chien, and A. G. Thakurta, "How to dp-fy ml: A practical guide to machine learning with differential privacy," *Journal of Artificial Intelligence Research*, vol. 77, pp. 1113–1201, July 2023.

[32] B. Pulido-Gaytan, A. Tchernykh, J. M. Cortés-Mendoza, M. Babenko, G. Radchenko, A. Avetisyan, and A. Y. Drozdov, "Privacy-preserving neural networks with homomorphic encryption: C hallenges and opportunities," *Peer-to-Peer Networking and Applications*, vol. 14, no. 3, pp. 1666–1691, March 2021.

[33] Z. Ji, Z. C. Lipton, and C. Elkan, "Differential privacy and machine learning: a survey and review," *arXiv preprint arXiv:1412.7584*, December 2014.

[34] K. Pan, Y.-S. Ong, M. Gong, H. Li, A. K. Qin, and Y. Gao, "Differential privacy in deep learning: A literature survey," *Neurocomputing*, vol. 589, p. 127663, July 2024.

[35] Y. Gong, X. Chang, J. Mišić, V. B. Mišić, J. Wang, and H. Zhu, "Practical solutions in fully homomorphic encryption: a survey analyzing existing acceleration methods," *Cybersecurity*, vol. 7, no. 1, p. 5, 2024.

[36] A. Sultangazin and P. Tabuada, "Symmetries and isomorphisms for privacy in control over the cloud," *IEEE Transactions on Automatic Control*, vol. 66, no. 2, pp. 538–549, February 2020.

[37] K. Zhang, K. Chen, Z. Li, J. Chen, and Y. Zheng, "Privacy-preserving data-enabled predictive leading cruise control in mixed traffic," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 5, pp. 3467–3482, May 2023.

[38] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning, Mountain View, CA, November 13–15, 2017*. PMLR, pp. 1–16.

[39] Z. Zhou, C. Rother, and J. Chen, "Event-triggered model predictive control for autonomous vehicle path tracking: Validation using CARLA simulator," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 6, pp. 3547–3555, June 2023.

[40] Mathworks, "Roadrunner: Design 3d scenes for automated driving simulation," accessed on November 5, 2024.

[41] C. Dwork, "Differential privacy," in *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006*, pp. 1–12.

[42] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security, Hofburg Palace, Vienna, Austria, October 24-28, 2016*, pp. 308–318.

[43] C. Dwork, A. Roth *et al.*, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, August 2014.

[44] C. Rother, Z. Zhou, and J. Chen, "Development of a four-wheel steering scale vehicle for research and education on autonomous vehicle motion control," *IEEE Robotics and Automation Letters*, vol. 8, no. 8, pp. 5015–5022, August 2023.

[45] F. Dang, D. Chen, J. Chen, and Z. Li, "Event-triggered model predictive control with deep reinforcement learning," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 459–468, January 2024.

[46] Z. Zhou, J. Chen, M. Tao, P. Zhang, and M. Xu, "Experimental validation of event-triggered model predictive control for autonomous vehicle path tracking," in *2023 IEEE International Conference on Electro Information Technology*, Romeoville, IL, May 18–20, 2023.

Zhaodong Zhou received his B.S. and M.S. degrees in Mechanical Engineering from Oakland University in 2019 and 2022, respectively. He is currently working towards a Ph.D. degree in Electrical and Computer Engineering from Oakland University.

His research interests include artificial intelligence and optimal control for automotive applications.

Jun Chen (S'11-M'14-SM'20) received his Bachelor's degree in Automation from Zhejiang University, Hangzhou China, in 2009, and Ph.D. in Electrical Engineering from Iowa State University, Ames IA, USA, in 2014. He was with Idaho National Laboratory from 2014 to 2016 and with General Motors from 2017 to 2020. Dr. Chen joined Oakland University in 2020, where he is currently an associate professor at the ECE department.

His research interests include advanced control and optimization, model predictive control, artificial intelligence, and stochastic hybrid systems, with applications in intelligent vehicles, robotics, and energy systems. Dr. Chen is a recipient of the NSF CAREER Award, the Best Paper Award from IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, the Best Paper Award from IEEE INTERNATIONAL CONFERENCE ON ELECTRO INFORMATION TECHNOLOGY, the Most Research Active Award and Outstanding Graduate Mentor Award from Oakland University, the Publication Achievement Award from Idaho National Laboratory, the Research Excellence Award from Iowa State University, and the Outstanding Student Award from Zhejiang University. He is currently a Senior Member of the IEEE.