2026 American Control Conference (ACC)
May 27-29, 2026. New Orleans, Louisiana, USA

Personalized MPC for Autonomous Vehicle Lateral Motion Control via Inverse Reinforcement Learning

Zhaodong Zhou, Mingyuan Tao, Jiayi Qiu, Peng Zhang, Meng Xu, Yong Sun and Jun Chen*, *Senior Member, IEEE*

***Abstract*— This paper presents a personalized lane change control framework that combines the maximum entropy inverse reinforcement learning (MaxEnt IRL) with model predictive control (MPC). Instead of manually tuning the MPC cost weight, the proposed method learns a cost function from expert driving demonstrations using interpretable trajectory features. The learned weights are incorporated into the MPC formulation to generate personalized lane change trajectories that mimic the expert driving style. Simulation results in CARLA show that the IRL-trained MPC controller can reproduce distinct driving styles and matches expert behaviors in heading, lateral motion, and steering. Furthermore, real world on-road testing on an ISUZU truck demonstrated that the proposed controller generalizes well to unseen initial conditions. Overall, the proposed approach reduces manual calibration efforts, eliminates the need for explicit path planning, improves controller interpretability, and enables personalized driver behavior replication.**

I. Introduction

Model predictive control (MPC) has emerged as one of the most effective methods for trajectory tracking and motion control in autonomous vehicles (AVs) [1]–[3]. By solving a finite-horizon optimal control problem at each sampling step, MPC can explicitly handle system dynamics and constraints while optimizing a cost function over future states and inputs [4]. This capability makes MPC particularly attractive for lane keeping, lane change, and obstacle avoidance tasks [5], [6]. However, a conventional MPC controller relies on a set of fixed cost weights to trade off objectives such as path tracking accuracy, control effort, and passenger comfort, which often requires extensive calibration across drivers and scenarios [7]. In addition to this, a well predefined reference path is important for MPC since the controller optimizes around a given trajectory [8]. Therefore, to obtain a personalized controller, both the cost weight and the reference path should ideally reflect the driver's preferred style.

However, manually tuning the cost weight and reference trajectory for every driver and scenario is challenging and time-consuming [9]. To overcome this challenge, we propose to leverage inverse reinforcement learning (IRL) to infer the underlying cost function from expert demonstrations [10]. Instead of directly optimizing a policy, IRL assumes the expert behavior is approximately optimal with respect to an unknown cost function and recovers this function by matching the features expectations between the expert demonstration and the trajectory generated under the candidate policy [11].

To address the ambiguity in reward learning, a variety of IRL algorithms have been proposed, such as Gaussian Process IRL [12], Bayesian IRL [13], and Maximum Entropy IRL (MaxEnt IRL) [14]. In this paper, MaxEnt IRL is adopted because it provides an unbiased solution by selecting the maximum-entropy distribution that matches the expert's feature expectations, and its convex objective enables efficient gradient based optimization. Recent studies have employed MaxEnt IRL to learn driver-specific reward functions from naturalistic traffic data, capturing diverse driving styles such as cautious or aggressive lane changes [15]. MaxEnt IRL has also been used for trajectory planning and providing interpretable rewards that improve safety and comfort without manual weight tuning [16]–[18].

Despite these advances, most prior research focused on offline reward learning or open-loop trajectory generation for path planning, with limited work on embedding IRL directly into a closed-loop MPC controller for personalized control. To fill this gap, we develop an IRL-based MPC framework and investigate its application to personalized lane change maneuvers. We collect expert demonstrations in the CARLA simulator [19] under two distinct driving styles—aggressive and conservative. The lane change portion of each trajectory is clipped and divided into short segments to match the prediction horizon of the MPC formulation. For each segment, four different features are computed, including lateral deviation, heading deviation, yaw rate effort, and steering effort. Using these expert features, we apply MaxEnt IRL to learn driver-specific feature weights, and substitute the learned weights into the MPC cost function to form a personalized controller. We evaluate the learned controllers in closed-loop CARLA simulations and further test the proposed framework in the real world by using the ISUZU truck, demonstrating effective sim-to-real transfer under different initial conditions.

The primary contributions of this paper are summarized as follows.

1) We design a set of interpretable features that capture the characteristics of the lane change maneuvers and serve as the basis for cost function learning.
2) We present a pipeline that integrates MaxEnt IRL with MPC, enabling automatic learning of driver-specific cost weight and generating the lane change trajectory

This work is supported in part by National Science Foundation through Award #2237317. *Jun Chen is the corresponding author*.

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

Mingyuan Tao, Jiayi Qiu, Peng Zhang, Meng Xu and Yong Sun are with Isuzu Technical Center of America, Inc, Plymouth, MI 48309, USA (email: {mingyuan.tao, jiayi.qiu, peng.zhang, meng.xu, yong.sun}@isuzu.com).

that replicates the expert style without a predefined reference path.

3) We evaluate the proposed framework in CARLA and on an ISUZU truck for real-world testing, showing that the IRL-based MPC controller reproduces expert behavior in closed loop and generalizes to real-world scenarios.

The remainder of the paper is organized as follows. Section II reviews the MaxEnt IRL. Section III describes the designed features and presents the proposed IRL-based personalized MPC framework.The evaluation results are discussed in Section IV, while Section V concludes the paper.

II. Maximum Entropy IRL

To learn a reward function that represents driver preferences, this work employs the *MaxEnt* IRL framework [14]. Consider N expert demonstrations $L = l_1, l_2, \ldots, l_N$, where each l_i denotes a trajectory consisting of a state and control sequence. A feature mapping $\mathbf{f}(l_i)$ is used to extract relevant behavioral attributes, and the mean feature vector across all demonstrations is then expressed as follows:

$$\bar{F} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{f}(l_i). \tag{1}$$

IRL seeks a reward function such that the expected feature statistics match $\bar{F}$. The trajectory reward is modeled as a linear combination of features:

$$J = W^T\mathbf{f}(l_i), \tag{2}$$

where W is a vector of unknown feature weights to be learned. The expected feature under the learned model has to satisfy,

$$\mathbb{E}_{p(l|W)}[\mathbf{f}] = \bar{F}. \tag{3}$$

where $p(l|W)$ is the trajectory distribution based on W.

To construct the trajectory distribution ($p(l|W)$), the maximum entropy principle is applied [14]. Among all distributions that match the feature expectation constraint (3), we select the one with maximum entropy, ensuring no additional assumptions are imposed beyond those supported by the data. This results in the following trajectory distribution:

$$P(l_i|W) = \frac{1}{Z(W)}e^{-J}, \tag{4}$$

where the partition function $Z(W) = \sum e^{-W^T\mathbf{f}(l_i)}$ serves to normalize the probability distribution over all feasible trajectories l. The reward parameters W are learned by maximizing the log-likelihood of the observed trajectories:

$$L(W) = \sum_{i=1}^{N}\log P(l_i|W). \tag{5}$$

Directly maximizing (5) to solve for the optimal weight vector W^* is analytically intractable due to the presence of the partition function $Z(W)$, which requires summing over all possible trajectories. To address this, we adopt the gradient based approach to iteratively update W. The gradient of the log-likelihood can be written as

$$\nabla L(W) = \mathbb{E}_{p(l|W)}[\mathbf{f}] - \bar{F}. \tag{6}$$

This gradient indicates the mismatch between the predicted feature and the expected feature from the expert demonstrations. When $\nabla L(W)$ becomes zero, the model reproduces the expert behavior and thus satisfies (3). The weights are iteratively updated in the gradient direction using

$$W = W + \eta\frac{\nabla L(W)}{|\nabla L(W)|}, \tag{7}$$

where η denotes the learning rate, which can be decreased over iterations to improve convergence stability.

A major computational challenge is evaluating $\mathbb{E}_{p(l|W)}[\mathbf{f}]$. Following [20], we adopt a maximum-likelihood approximation by replacing the expectation with the features of the most probable trajectory under the current model:

$$\mathbb{E}_{p(l|W)}[\mathbf{f}] \approx \mathbf{f}(\arg\max p(l|W)). \tag{8}$$

This approximation simply takes the features of the most likely trajectory under the current model as a stand-in for the full expectation. Such an approximation greatly reduces computation, avoids the need to sample many trajectories, while maintaining accurate weight learning.

III. Driver Lane Change Behavior Learning

This section presents the proposed approach for learning personalized lane change behaviors using the MaxEnt IRL framework, including feature design, MPC formulation, and the IRL-MPC training procedure.

A. Feature Design

The key component of the MaxEnt IRL framework (Section II), is the definition of interpretable features $\mathbf{f}$ that characterize a driver's lane change behavior. These features serve as the foundation to formulate the cost function that the IRL algorithm will learn. The goal is to select features that capture essential aspects of driving style and can be computed from both human demonstrations and MPC predictions. To this end, we define the following four driving features, each representing a specific aspect of the vehicle's motion during a lane change maneuver.

Lateral position deviation: This feature quantifies the vehicle lateral deviation from the centerline of the target lane (denoted as y_{ref}). It encourages accurate alignment with the centerline of the target lane over the whole lane change maneuver, and it is formulated as follows:

$$f_l = \sum_{t=1}^{n}(y_t - y_{ref})^2, \tag{9}$$

where y_t is the vehicle's lateral position at time step t, and y_{ref} is the constant lateral coordinate of the centerline of the target lane.

Heading angle deviation: This feature measures the difference between the vehicle's heading angle and the orientation of the target lane. The formulation is shown below:

$$f_h = \sum_{t=1}^{n} (\psi_t - \psi_{ref})^2, \tag{10}$$

where ψ_t is the vehicle's heading angle at time t, and ψ_{ref} is the heading angle of the target lane.

Yaw rate effort: This feature penalizes large yaw rate values, which correspond to rapid heading changes. The formulation is as follows.

$$f_y = \sum_{t=1}^{n} \dot{\psi}_t^2, \tag{11}$$

where $\dot{\psi}_t$ is the yaw rate of the vehicle at time t. A smaller f_y value indicates a more fluid and less abrupt transition, contributing to smoother and more comfortable lane change.

Steering effort: This feature penalizes high steering input magnitudes. Excessive or abrupt steering can lead to discomfort and instability. This feature is defined as follows.

$$f_s = \sum_{t=1}^{n} u_t^2, \tag{12}$$

where u_t denotes the steering command at time t. A smaller f_s value reflects lower control effort and greater comfort, thereby implying a smoother lane change trajectory.

These features form $F = [f_l, f_h, f_y, f_s]^T$ and define the parameterized reward function:

$$J(W) = W_1 f_l + W_2 f_h + W_3 f_y + W_4 f_s, \tag{13}$$

where $W = [W_1, W_2, W_3, W_4]^T$ are weights learned via IRL from expert demonstrations. Through the weighting of these features, the proposed IRL-MPC framework can replicate diverse driving styles.

B. Model Predictive Control Design

In this paper, the MPC is used to cooperate with the IRL learned cost function to control the vehicle to finish the personalized lane change maneuver. To achieve that, MPC needs to incorporate a vehicle prediction model, system constraints, and a set of weights that represent the driver's lane change style, which is obtained from IRL. MPC solves a finite horizon optimal control problem (OCP) to obtain the optimal state sequence and control sequence. The OCP is formulated as follows:

$$\min_{U} \quad J = W_1 f_l + W_2 f_h + W_3 f_y + W_4 f_s \tag{14a}$$

$$\text{s.t.} \quad s_t = \hat{s}_t \tag{14b}$$

$$\text{Vehicle dynamic model (15)}, \quad 1 \le n \le p \tag{14c}$$

$$u_{min} \le u_{t+n} \le u_{max}, \quad 0 \le n \le p-1 \tag{14d}$$

$$du_{min} \le u_{t+n} - u_{t+n-1} \le du_{max}, \quad 0 \le n \le p-1. \tag{14e}$$

The cost function (14a) is the same linear feature combination as (13), where the weights W are learned via IRL from expert demonstrations. The vehicle dynamic model (15) serve as the MPC prediction model (14c), and (14d), (14e) impose input and rate constraints. Furthermore, p is the prediction horizon for the MPC, s_t is the vehicle initial states for the MPC, $\hat{s}_t$ is the measured or estimated vehicle states at time instant t, and $U = \begin{bmatrix} u_t & u_{t+1} & \dots & u_{t+p-1} \end{bmatrix}^T$ is the control input sequence over the prediction horizon.

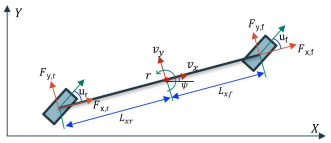

Fig. 1. Schematic of the vehicle dynamic model.

The vehicle dynamic model utilized in this paper is illustrated in Fig. 1. The state vector is defined as $s = \begin{bmatrix} x & y & v_y & \psi & r \end{bmatrix}^T$, where x and y are the vehicle position of the center of gravity (CG) on the global frame, v_y is the lateral velocity of the vehicle in the vehicle frame (origin at the CG, with the x-axis aligned with the heading), and ψ and r denote the heading angle and yaw rate, respectively, measured in the counter-clockwise direction. The governing differential equations of the dynamic vehicle model are then expressed as follows [21]:

$$x_{k+1} = x_k + (v_x \cos\psi_k - v_{y,k} \sin\psi_k) T_s \tag{15a}$$

$$y_{k+1} = y_k + (v_x \sin\psi_k + v_{y,k} \cos\psi_k) T_s \tag{15b}$$

$$v_{y,k+1} = v_y + \left(-v_x r_k + \frac{1}{m} \sum_{i=\{f,r\}} F_{y,i} \right) T_s \tag{15c}$$

$$\psi_{k+1} = \psi_k + r_k T_s \tag{15d}$$

$$r_{k+1} = r_k + \frac{1}{I} \left(L_{xf} F_{y,f} - L_{xr} F_{y,r} \right) T_s, \tag{15e}$$

where m denotes the vehicle mass, I is the vehicle yaw moment of inertia, and L_{xf} and L_{xr} are the distances from the CG to the front and rear axles, respectively. The lateral tire forces are given by $F_{y,i}$, where $i \in \{f, r\}$ corresponds to the front (f) and rear (r) tires, respectively. Moreover, v_x is the longitudinal velocity of the vehicle at the CG in the vehicle frame.

To compute the lateral tire force $F_{y,i}$ required in the vehicle dynamic model (15), a tire model is introduced. Denote $\bar{F}_{x,i}$ and $\bar{F}_{y,i}$ as the tire forces in the wheel frame, $F_{x,i}$ and $F_{y,i}$ as the tire forces in the vehicle frame. The transformation between the wheel frame and the vehicle frame is given by:

$$F_{x,i} = \bar{F}_{x,i} \cos u_i - \bar{F}_{y,i} \sin u_i \tag{16a}$$

$$F_{y,i} = \bar{F}_{x,i} \sin u_i + \bar{F}_{y,i} \cos u_i, \tag{16b}$$

Since this paper only considers front-wheel steering vehicles, u_r is always set to zero. Furthermore, a linear tire force model is used:

$$\bar{F}_{y,i} = C \alpha_i, \tag{17}$$

where C is the cornering stiffness of the wheels and α_i is the slip angle. The resulting lateral tire forces are then used in the vehicle dynamics.

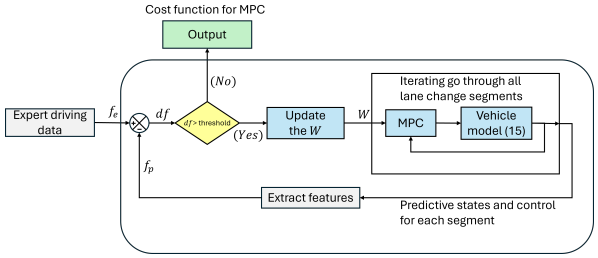

Fig. 2. The flow chart of IRL training.

C. Training

To ensure that the learned lane change strategy applies to different path orientations, a coordinate transformation is first applied. The global coordinate frame is redefined such that its origin is located at the start of the lane change and its x-axis is aligned with the target lane direction. All vehicle states s and reference values y_{ref} and ψ_{ref} are transformed into this new coordinate frame accordingly. Moreover, since in this paper only the lane change on a straight lane is considered, after the transformation, the ψ_{ref} is always equal to zero, i.e., the x-axis is always aligned with the lane direction.

After the transformation, the lane change trajectory is divided into 0.8 s segments $L = \{l_1, \ldots, l_N\}$ and resampled via linear interpolation to obtain (S_i) and (U_i) at each MPC sampling instant, from which features are computed for training.

Fig. 2 shows the flow chart of the training. The expert features are compared with the predicted features extracted from an MPC closed-loop model, and the weight vector W is iteratively updated until the feature difference d_f is below a predefined threshold. This closed-loop update ensures that the learned cost function gradually converges toward the expert's driving behavior. Algorithm 1 summarizes the details for the proposed driver lane change behavior training process. Line 1-2 extract the expert feature for each segment and get the average feature $(\bar{F})$ for IRL. Line 3 initializes the iteration counter (i), weight (W), and learning rate (η). Based on the current weight (W), solve the OCP (14) with the initial states $(S_i(1))$ for each segment to get the optimal states and control sequences in Line 6. Line 7 extracts the optimal features from the optimal states and control sequences. Line 9 calculates the expectation of the optimal features. Line 10-12 update the weight (W) based on the ∇L. The learning rate (η) is gradually reduced over iterations to ensure stable convergence in Lines 13-15.

IV. Results and Discussions

In this section, the proposed IRL with MPC framework is evaluated across various scenarios using CARLA simulation environment and the real world. In CARLA, the evaluation is conducted under two distinct driving styles, conservative and aggressive, at the same vehicle speed (60 km/h). The evaluation in the real world is tested on clear roads at low speed (20 km/h) for safety.

Algorithm 1 *MaxEnt* IRL Driver Behavior Learning Algorithm

Input: Expert lane change states: $\mathbb{S} = \{S_1,\ S_2,\ \ldots,\ S_N\}$, Expert lane change control sequence: $\mathbb{U} = \{U_1,\ U_2, \ldots,\ U_N\}$

Output: W

1: $F_i = \mathbf{f}(S_i, U_i)$;
2: $\bar{F} = \frac{1}{N}\sum_{i=1}^{N} F_i$;
3: Initialize $i \leftarrow 0$, $W \leftarrow$ all-ones vector, $\eta \leftarrow 0.1$;
4: **while** W not converge **do**
5: **for** all $S_i(1)$ in $\mathbb{S}$ **do**
6: $S_{opt,i}, U_{opt,i} \leftarrow$ OCP (14) with $s_t = S_i(1)$;
7: $F_{opt,i} \leftarrow \mathbf{f}(S_{opt,i}, U_{opt,i})$;
8: **end for**
9: $\mathbb{E}_{p(l|W)}[f] \leftarrow \frac{1}{N}\sum_{i=1}^{N} F_{opt,i}$;
10: $\nabla L \leftarrow \mathbb{E}_{p(l|W)}[f] - \bar{F}$;
11: $W \leftarrow W + \eta \frac{\nabla L}{\|\nabla L\|}$;
12: $i \leftarrow i + 1$;
13: **if** $i = 200$ **then**
14: $\eta \leftarrow \eta/2$;
15: $i \leftarrow 0$;
16: **end if**
17: **end while**

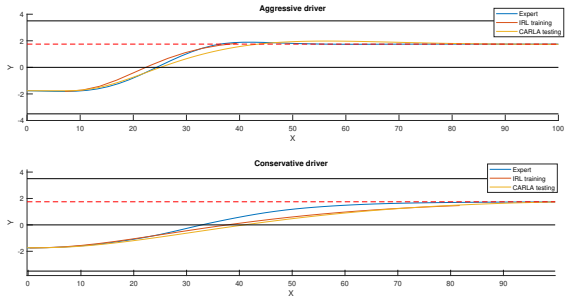

Fig. 3. Lane change trajectories for different drivers

A. CARLA Simulation Test

For the simulation experiments, the CARLA simulator [5], [19] was employed as the test platform. Expert trajectories are first generated by using CARLA's built-in autopilot to drive a sedan in both conservative and aggressive modes. These two driving styles produced distinct lane change trajectories, illustrated by the blue lines in Fig. 3. The vehicle starts from the first lane, which is from -3.6 to 0 m. The target lane is from 0 to 3.6 m The black lines represent the lane boundaries, and the red dashed line is the centerline of the target lane. Moreover, the genetic algorithm [22] is applied to identify the vehicle parameters used in the dynamic model of the MPC controller [5]. The identified parameters are summarized in Table I. The lane change segments are then extracted from the expert trajectories and used for training in the IRL framework. As described in Section III-C, the lane change trajectory is segmented into 0.8 s intervals. Given the MPC sampling time of 0.2 s, the prediction horizon p of

TABLE I

PARAMETERS FOR VEHICLE DYNAMIC MODEL USED IN CARLA.

m (kg)	1265	I (kg·m^2)	6481	L (m)	2.6
L_{xf} (m)	2.3	L_{xr} (m)	0.3	μ (-)	0.289
C (deg^{-1})	3.07				

TABLE II

IRL LEARNED WEIGHT FOR SIMULATION TEST.

Driver	f_l	f_h	f_y	f_s
Aggressive	29.061	10.402	1	30.182
Conservative	6.252	8.971	1	40.618

MPC is set to 4.

The training results for the lane change are presented by the red lines in Fig. 3, and the learned weights for different drivers are reported in the Table II. As shown in this table, the aggressive driver has a relatively larger weight of f_l and f_h, and a smaller weight of f_s, since they want to reach the target lane as soon as possible and can stand with a large steering effort. In contrast, the conservative driver has a smaller weight of f_l and f_h, and a larger weight of f_s, to get a flatter lane change maneuver. These learned weights are subsequently incorporated into OCP (14) to form the MPC lane change controller. The results of CARLA testing trajectories for different drivers are shown in the yellow line in Fig. 3. As shown in the Fig. 3, the training result (red), test result (yellow), and expert path (blue) are highly consistent, indicating that the IRL-personalized MPC lane change controller is able to mimic the expert's driving behavior.

The vehicle states under different driving styles are shown in Figs. 4 and 5, respectively. Each figure presents the expert trajectory (blue), the IRL training results (red), and the CARLA testing results (yellow) for heading angle, yaw rate, and steering angle over time. For the aggressive driver (Fig. 4), the IRL training result closely replicates the expert states, showing a sharp heading peak and larger, faster yaw-rate and steering responses. In contrast, for the conservative driver (Fig. 5), IRL learns a smoother trajectory with a more gradual heading change and lower peak yaw rate, leading to a flatter lane change and smaller steering variations. The CARLA testing further confirms the generalization ability of the IRL-personalized MPC controller. For both driver styles, the CARLA results (yellow lines) remain closely aligned with the IRL training trajectories (red lines), with only minor deviations due to the simulation noise and vehicle model mismatch.

The feature metrics of each trajectory are shown in the Table III. Similar to the calculation for the expert trajectories, the lane change segments of the CARLA testing trajectories are first extracted and further divided into 0.8 s intervals. The feature values f_l, f_h, f_y, and f_s are then computed for each interval and averaged over the entire maneuver to enable a consistent comparison across expert, IRL training, and CARLA testing results. For the aggressive driver, the expert trajectory gives $f_l = 0.1675$ and $f_h = 0.0826$.

TABLE III

FEATURES FOR EXPERT, IRL TRAINING, AND CARLA TESTING TRAJECTORIES UNDER DIFFERENT DRIVING STYLES.

Driver		f_l	f_h	f_y	f_s
Aggr.	Expert	0.1675	0.0826	0.1549	0.0072
	IRL training	0.1797	0.0883	0.1443	0.0060
	CARLA testing	0.1704	0.0366	0.0747	0.0038
Cons.	Expert	0.1642	0.0114	0.0082	0.0004
	IRL training	0.1662	0.0081	0.0048	0.0002
	CARLA testing	0.1978	0.0076	0.0031	0.0001

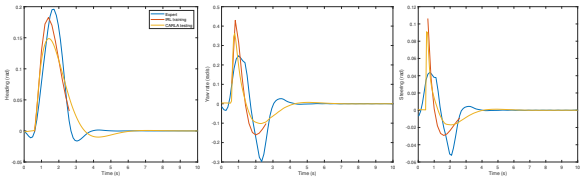

Fig. 4. Comparison of Expert, IRL training, and CARLA testing vehicle states for aggressive driver.

while after IRL training $f_l = 0.1797$ and in CARLA testing $f_l = 0.1704$. For the conservative driver, the expert features are $f_l = 0.1642$, $f_h = 0.0114$, and $f_y = 0.0082$, and in CARLA testing f_l rises slightly to 0.1978 while f_h and f_y decrease further. Overall, these results show that the learned controller preserves the style-dependent feature trends, and the remaining gaps are mainly due to the mismatch between the CARLA high fidelity dynamics and the simplified dynamic model used in MPC.

B. Truck On-road Test

To further test the feasibility of the proposed framework in real world implementation, experiments are conducted on an ISUZU NPR-HD truck, equipped with a drive-by-wire system and GPS for AV control. As illustrated in Fig. 6, during the on-road experiment, the driver executed a right lane change. The blue line represents the expert lane change path driven by a human. To investigate whether the proposed framework can generalize the learned lane change strategy to an unseen scenario, the driver initiated the lane change maneuver slightly off the centerline. Based on this expert demonstration, the IRL learned path is shown in a red line, and the corresponding learned weights are listed in Table IV. These weights are then substituted into the MPC to control the vehicle to do the lane change starting from the centerline. The resulting trajectory, shown as the yellow line in Fig.

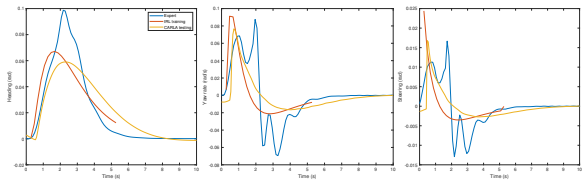

Fig. 5. Comparison of Expert, IRL training, and CARLA testing vehicle states for conservative driver.

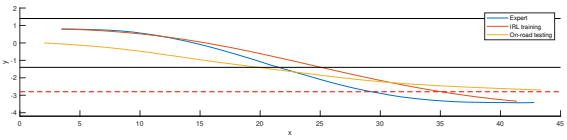

Fig. 6. Lane change trajectories for on-road testing

TABLE IV

IRL LEARNED WEIGHT FOR THE TRUCK DRIVER.

f_l	f_h	f_y	f_s
8.428	2.559	1	2.568

TABLE V

FEATURES FOR EXPERT, IRL TRAINING, AND ON-ROAD TESTING TRAJECTORIES.

	f_l	f_h	f_y	f_s
Expert	0.2092	0.0782	0.0179	0.0088
IRL training	0.2475	0.0864	0.0221	0.0124
on-road testing	0.1087	0.0294	0.0025	0.0077

6, demonstrates that although the vehicle began from an unfamiliar initial position, the IRL-based MPC successfully controlled the vehicle to complete the lane change along a comparable path, with the total distance of the maneuver being approximately 42 m.

The averaged feature values for the expert trajectory, IRL training, and the on-road test are listed in Table V. First, the feature values of IRL training are close to the expert values, confirming that the IRL successfully captured the driver's lane change style during the learning stage. In comparison, the feature values obtained from the on-road test are lower than those from both the expert and IRL training trajectories. This outcome is expected because the truck started closer to the lane centerline, requiring less heading correction, smaller yaw rate changes, and reduced steering effort. These differences do not indicate overfitting; rather, they highlight the generalization ability of the IRL-based MPC controller to adapt to unseen initial states while still executing smooth, human-like lane changes.

V. CONCLUSIONS

This paper presents a personalized lane change control framework that integrates maximum entropy inverse reinforcement learning (MaxEnt IRL) with model predictive control (MPC) to learn cost function weights from expert demonstrations. By designing interpretable features and embedding the learned reward into the MPC cost function, the resulting controller reproduces both aggressive and conservative driving styles in the CARLA, closely matching expert trajectories in heading, lateral position, and steering. Real world experiments on an ISUZU truck confirm adaptation to unseen initial conditions. Future work will extend the framework to more complex multi-vehicle traffic scenarios and explore online adaptation to capture evolving driver preferences.

REFERENCES

[1] D. Chu, H. Li, C. Zhao, and T. Zhou, "Trajectory tracking of autonomous vehicle based on model predictive control with pid feedback," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 2, pp. 2239–2250, 2022.

[2] 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.

[3] J. Chen and Z. Yi, "Comparison of event-triggered model predictive control for autonomous vehicle path tracking," in *IEEE Conf. Control Technology and Applications*, San Diego, CA, August 8–11, 2021.

[4] S. Yu, M. Hirche, Y. Huang, H. Chen, and F. Allgöwer, "Model predictive control for autonomous ground vehicles: a review," *Autonomous Intelligent Systems*, vol. 1, no. 1, p. 4, 2021.

[5] 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*, accepted for publication April 2023.

[6] L. D'Alfonso, F. Giannini, G. Franzè, G. Fedele, F. Pupo, and G. Fortino, "Autonomous vehicle platoons in urban road networks: A joint distributed reinforcement learning and model predictive control approach," *IEEE/CAA journal of automatica sinica*, vol. 11, no. 1, pp. 141–156, 2024.

[7] 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.

[8] C. Liu, S. Lee, S. Varnhagen, and H. E. Tseng, "Path planning for autonomous vehicles using model predictive control," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 174–179.

[9] J. Köhler, M. A. Müller, and F. Allgöwer, "Analysis and design of model predictive control frameworks for dynamic operation—an overview," *Annual Reviews in Control*, vol. 57, p. 100929, 2024.

[10] A. Y. Ng, S. Russell *et al.*, "Algorithms for inverse reinforcement learning." in *Icml*, vol. 1, no. 2, 2000, p. 2.

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

[12] S. Levine, Z. Popovic, and V. Koltun, "Nonlinear inverse reinforcement learning with gaussian processes," *Advances in neural information processing systems*, vol. 24, 2011.

[13] D. Ramachandran and E. Amir, "Bayesian inverse reinforcement learning." in *IJCAI*, vol. 7, 2007, pp. 2586–2591.

[14] 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.

[15] 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.

[16] 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.

[17] P. Zhang, S. Xie, X. Lu, Z. Zhong, and Q. Li, "Maximum entropy inverse reinforcement learning-based trajectory planning for autonomous driving," in *Third International Conference on High Performance Computing and Communication Engineering*, 2024, pp. 146–151.

[18] Z. Zhou and J. Chen, "Privacy-preserving personalized autonomous vehicle lane change using inverse reinforcement learning," *IEEE Transactions on Vehicular Technology*, accepted September 2025.

[19] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in *1st Conference on Robot Learning*, Mountain View, CA, 2017, pp. 1–16.

[20] 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.

[21] R. Rajamani, *Vehicle Dynamics and Control*, ser. Mechanical Engineering Series. Springer US, 2012.

[22] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: past, present, and future," *Multimedia tools and applications*, vol. 80, no. 5, pp. 8091–8126, 2021.