Efficient Sampling-Based Maximum Entropy Inverse Reinforcement Learning With Application to Autonomous Driving

Zheng Wu, Liting Sun, Wei Zhan, Chenyu Yang, and Masayoshi Tomizuka

***Abstract*—In the past decades, we have witnessed significant progress in the domain of autonomous driving. Advanced techniques based on optimization and reinforcement learning become increasingly powerful when solving the forward problem: given designed reward/cost functions, how we should optimize them and obtain driving policies that interact with the environment safely and efficiently. Such progress has raised another equally important question: *what should we optimize*? Instead of manually specifying the reward functions, it is desired that we can extract what human drivers try to optimize from real traffic data and assign that to autonomous vehicles to enable more naturalistic and transparent interaction between humans and intelligent agents. To address this issue, we present an efficient sampling-based maximum-entropy inverse reinforcement learning (IRL) algorithm in this letter. Different from existing IRL algorithms, by introducing an efficient continuous-domain trajectory sampler, the proposed algorithm can directly learn the reward functions in the continuous domain while considering the uncertainties in demonstrated trajectories from human drivers. We evaluate the proposed algorithm via real-world driving data, including both non-interactive and interactive scenarios. The experimental results show that the proposed algorithm achieves more accurate prediction performance with faster convergence speed and better generalization compared to other baseline IRL algorithms.**

***Index Terms*—Learning from demonstration, intelligent transportation systems, inverse reinforcement learning, autonomous driving, social human-robot interaction.**

I. Introduction

ALTHOUGH rapid progresses have been made in autonomous driving recently, many challenging problems remain open, particularly when interactions between autonomous vehicles (AVs) and humans are considered. Advanced techniques in reinforcement learning (RL) and optimization such as search-based methods [1]–[3], gradient-based approaches [4], [5] and nested optimization [6] have significantly boosted our capability in finding the optimal trajectories of autonomous vehicles that maximize the specified reward/cost functions. However, to enable more naturalistic and transparent interactions between the AVs and humans, another question is of equal importance: *what should we optimize*? A mis-specified reward function can cause severe consequences. The autonomous vehicles might be either too conservative or too aggressive, neither of which are desirable or safe in real traffic. More importantly, optimizing for reward functions that are misaligned with human expectations will make the behavior of AVs non-transparent to humans, which will eventually degrade the trust from human.

Thus, it is desired to extract what human drivers are optimizing from real traffic data. Inverse reinforcement learning (IRL) [7], [8] have been widely explored and utilized for acquiring reward functions from demonstrations, assuming that the demonstrations are (sub-)optimal solutions of the underlying reward functions. Many examples have proved the effectiveness of such IRL algorithms. For instance, [9] proposed an IRL algorithm based on expected feature matching and evaluated it on the control of helicopters. [10] proposed an online IRL algorithm to analyze complex human movement and control high-dimensional robot systems. [11] used RL and IRL to develop planning algorithm for autonomous cars in traffic and [12] designed a deep IOC algorithm based on neural networks and policy optimization, which enabled the PR2 robots to learn dish placement and pouring tasks. In [13] and [14], a courteous cost function and a hierarchical driving cost for autonomous vehicles were learned respectively via IRL to allow autonomous vehicles to generate more human-like behaviors while interacting with humans. [15] extensively studied the feature selection in IRL for autonomous driving.

Typically, acquiring humans' driving costs from real traffic data has to satisfy a set of requirements:

- The trajectories of vehicles are continuous in a high-dimensional and spatiotemporal space, thus the IRL algorithm needs to scale well in high-dimensional continuous space;
- The trajectories of the vehicles satisfy the vehicle kinematics and the IRL algorithms should take it into consideration while learning reward functions;
- Uncertainties exist in real traffic demonstrations. The demonstrations in naturalistic driving data are not

Manuscript received February 24, 2020; accepted June 7, 2020. Date of publication June 25, 2020; date of current version July 16, 2020. This letter was recommended for publication by Associate Editor A. Kim and Editor Y. Choi upon evaluation of the reviewers' comments. The work is conducted during his visit to the University of California, Berkeley. *(Zheng Wu and Liting Sun contributed equally to this work.) (Corresponding author: Liting Sun.)*

Zheng Wu, Liting Sun, Wei Zhan, and Masayoshi Tomizuka are with the Department of Mechanical Engineering, University of California, Berkeley, CA 94709 USA (e-mail: zheng_wu@berkeley.edu; litingsun@berkeley.edu; wzhan@berkeley.edu; tomizuka@me.berkeley.edu).

Chenyu Yang is with the Department of Computer Science, Shanghai Jiaotong University, Shanghai 200240, China (e-mail: yangcyself@sjtu.edu.cn).

Digital Object Identifier 10.1109/LRA.2020.3005126

necessarily optimal or near-optimal, and the IRL algorithms should be compatible with such uncertainties;
- The features adopted in the reward functions of the IRL algorithms should be highly interpretable and generalizable to improve the transparency and adaptability of the AVs' behaviors.

Unfortunately, it is not trivial to satisfy all the above requirements simultaneously. First of all, most existing IRL algorithms, for instance, apprenticeship learning [9], maximum-entropy IRL [16] and Bayesian IRL [17], are defined within the discrete Markov Decision Process (MDP) framework with relatively small-scale state and action spaces and suffer from the scaling problem in large-scale continuous-domain applications with long horizons. The continuous-domain IOC algorithm proposed in [18] effectively addressed such issue by considering only the local shapes of the reward functions via Laplace approximation. However, it is applicable to deterministic problems where all expert demonstrations are required to be optimal or sub-optimal, which is practically impossible to satisfy via naturalistic driving data, particularly when the features are not known in advance. Moreover, the Laplace approximation requires the calculation of both the gradients and the inverse of the Hessians of the reward functions at the expert demonstrations. When the demonstrations contains long-horizon trajectories, both variables are time-consuming to obtain. The deep IOC algorithm in [12] can also work in continuous domain. However, its features are not interpretable, and lead to poor generalization in different scenarios. In [19], a sampling-based IRL algorithm was also proposed to address this issue, but it only considered the uncertainties caused by the noises in control and can hardly capture the uncertainties induced by the sub-optimality of different driving maneuvers.

In terms of application domain, our work is closely related to [20] which also utilize inverse reinforcement learning to learn the reward functions from real driving trajectories. In the forward problem at each iteration, it directly solves the optimization problem and use the optimal trajectories to represent the expected feature counts. However, such optimization process might be quite time-consuming, especially when the driving horizon is long.

Thus, in this paper, we propose an efficient sampling-based maximum entropy IRL (SMIRL) algorithm that satisfies all the above mentioned requirements in autonomous driving. In the algorithm, we explicitly leverage our prior knowledge on efficiently generating feasible long-horizon trajectory samples which allow the autonomous vehicles to interact with the environment, including human drivers. More specifically, in terms of problem formulation, we adopt the principle of maximum entropy. In terms of efficient trajectory sampling for the estimation of the partition term with maximum entropy, we integrate our previous non-conservative and defensive motion planning algorithm with a sampling method to efficiently generate feasible and representative long-horizon trajectory samples. We compare the performance of the proposed SMIRL algorithm with the other two IRL algorithms, i.e., the ones in [18] and [20] on real human driving data extracted from the INTERACTION dataset. Three sets of evaluation metrics are employed, including both the deterministic metrics such as mean Euclidean distance (MED) and feature count deviation and the probabilistic metric such as the likelihood of the ground-truth trajectories. The experimental results showed that our proposed SMIRL can achieve more accurate prediction performance on the test set in both non-interactive and interactive driving scenarios.

II. The Method

A. Maximum-Entropy IRL

Let x and u denote, respectively, the states and actions of vehicles. The dynamics of the vehicle, $f(\cdot)$, can then be described as:

$$x_{k+1} = f(x_k, u_k). \tag{1}$$

A driving trajectory in spatial-temporal domain, denoted as ξ, contains a sequence of states and actions, i.e., $\xi = [x_0, u_0, x_1, u_1, \ldots, x_{N-1}, u_{N-1}]$ where N is the length of the planning horizon. Given a set of demonstrations $\Xi_D = \{\xi_i\}$ with $i = 1, 2, \ldots, M$, with the principle of maximum-entropy [16], the IRL problem aims to recover the underlying reward function from which the likelihood of the demonstrations can be maximized, assuming that the trajectories are exponentially more likely when they have higher cumulative rewards (Boltzman noisily-rational model [21]):

$$P(\xi, \theta) \propto e^{\beta R(\xi, \theta)} \tag{2}$$

where the parameter vector θ specifies the reward function R. β is a hyper-parameter that describes how close the demonstrations are to perfect optimizers. As $\beta \to \infty$, the demonstrations approach to perfect optimizers. Without loss of generality, we set $\beta = 1$ in this work.

Note that in this work, we assume that all the agents/demonstrations share the same dynamics defined in (1). We also assume the reward function underlying the given demonstration set is roughly consistent. Namely, we do not consider scenarios where human drivers change their reward functions along the demonstrations. We also do not specify the diversity of reward functions among different human drivers. Hence, the acquired reward function is essentially an averaged result defined on the demonstration set.

We adopt linear-structured reward function with a selected feature space $\mathbf{f}(\cdot)$ defined over the trajectories ξ, i.e.,

$$R(\xi, \theta) = \theta^T \mathbf{f}(\xi) \tag{3}$$

Hence, the probability (likelihood) of the demonstration set becomes

$$P(\Xi_D|\theta) = \prod_{i=1}^{M} \frac{e^{\beta R(\xi_i, \theta)}}{\int_{\tilde{\xi} \in \Phi_{\xi_i}} e^{\beta R(\tilde{\xi}, \theta)} d\tilde{\xi}} = \prod_{i=1}^{M} \frac{1}{Z_{\xi_i}} e^{\beta R(\xi_i, \theta)} \tag{4}$$

where Φ_{ξ_i} represents the space of all trajectories that share the same initial and goal conditions as in ξ_i. Our goal is to find the optimal θ^* which maximizes the averaged log-likelihood of the demonstrations, i.e.,

$$\theta^\star = \arg\max_\theta \frac{1}{M} \log P(\Xi_D|\theta) = \arg\max_\theta \frac{1}{M} \sum_{i=1}^{M} \log P(\xi_i|\theta). \tag{5}$$

From (4) and (5), we can see that the key step in solving the optimization problem in (5) is the calculation of the partition

factors Z_{ξ_i}. In sampling-based methods, Z_{ξ_i} for each demonstration is approximated via the sum over samples in the sample set $\{\tau_m^i\}, m = 1, 2, \ldots, K$:

$$Z_{\xi_i} \approx \sum_{m=1}^{K} e^{\beta R(\tau_m^i, \theta)}. \tag{6}$$

Thus, the objective function in (5) becomes:

$$\begin{aligned} L(\theta) &= \frac{1}{M}\sum_{i=1}^{M} \log P(\xi_i|\theta) \\ &= \frac{1}{M}\sum_{i=1}^{M} \log \frac{e^{\beta R(\xi_i,\theta)}}{\sum_{m=1}^{K} e^{\beta R(\tau_m^i,\theta)}} \\ &= \frac{1}{M}\sum_{i=1}^{M} \{\beta R(\xi_i, \theta) - \log \sum_{m=1}^{K} e^{\beta R(\tau_m^i,\theta)}\}. \end{aligned} \tag{7}$$

The derivative is thus given by:

$$\nabla_\theta L = \frac{\beta}{M}\sum_{i=1}^{M}(\mathbf{f}(\xi_i) - \widetilde{\mathbf{f}}(\xi_i)) \tag{8a}$$

$$\widetilde{\mathbf{f}}(\xi_i) = \sum_{m=1}^{K} \frac{e^{\beta R(\tau_m^i,\theta)}}{\sum_{m=1}^{K} e^{\beta R(\tau_m^i,\theta)}} \mathbf{f}(\tau_m^i) \tag{8b}$$

where $\widetilde{\mathbf{f}}(\xi_i)$ defines the expected feature counts over all samples given θ.

Note that an additional l_1 regularization over the parameter vector θ is introduced in the training process to compensate for possible errors induced via the selected set of features.

B. The Sampler

From (6), we can see that an efficient sampler is extremely important for solving the aforementioned optimization problem in (5). Many sampling-based planning algorithms have been widely explored by researchers [2], [3], [22]. In this section, we integrate the sampler from [2] with our previous work on non-conservative defensive motion planning [23] to efficiently generate samples to estimate Z in (4).

We represent maps via occupancy grids and feasible trajectory samples are generated using decoupled spatio-temporal approaches. As shown in Fig. 1, at each time step, the sampler includes three steps: 1) global path sampling via discrete elastic band (ED), 2) path smoothing via pure pursuit control, and 3) speed sampling via optimization and polynomial curves.

Step I-Path Sampling via Discrete ED: The objective of the first step is to generate collision-free paths. The key insight is that by constraining all samples to be safe, we have intrinsically considered the safety constraints of the optimal problem that the demonstrators try to solve. Moreover, it can also reduce the sample space to approximate Z, thus improving the efficiency of the inverse learning algorithm. To account for moving objects, it considers the sweep-volume of each object of interest within a temporal prediction horizon [2]. As shown in Step I in Fig. 1, a path τ consists of a sequence of elastic nodes (the blue shaded area), i.e., $\tau = \{\text{node}^i, \text{IN}^i, \text{OUT}^i\}, i = 1, 2, \cdots, T$. For each elastic node, we calculate the weighted sum of three types of forces: the contraction force (both left and right) from neighboring spatial nodes, the repulsive force from obstacles, and the attraction force to drive the vehicle towards desired lane centers. Moreover, collision check is conducted for the IN and OUT edges. Then a graph search method is utilized to find a set of collision-free paths that satisfy $\{\mathcal{T}_I\} = \{\tau : \text{collision(IN)} = 0, \text{Force(node)} \leq F_{\text{threshold}}\}$. The $F_{\text{threshold}}$ is a hyper-parameter describing the threshold of the sample set.

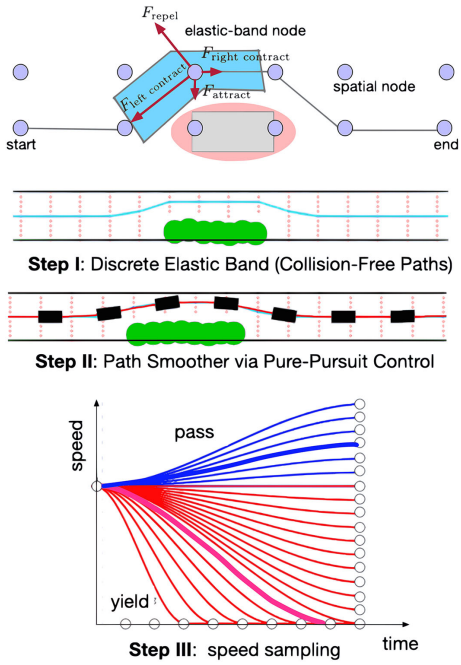


Fig. 1. The overview of the sampling process.

Step II-Path Smoothing: All the samples in Step I are piecewise linear but non-smooth paths, which is not feasible for the vehicle kinematics and thus not suitable to estimate Z. Hence, in Step II, a pure-pursuit tracking controller is employed to smoothen the paths in $\{\mathcal{T}_I\}$ and generate $\{\mathcal{T}_{II}\}$.

Step III-Speed Sampling: This step will generate speed samples for each path sample in Step I. First, a suggested speed profile is generated by finding the time-optimal speed plan under physical constraints (e.g., the acceleration/deceleration limits). Second, local speed curve sampling based on polynomials is performed to explore the neighborhood of the suggested speed profile. The key insight behind such a two-step speed sampling approach is to reduce the exploration space of the speed profile based on the prior knowledge on human drivers, namely they tend to pursue time-optimal speed plans. To account for discrete driving decisions in interactive scenarios, we will find one suggested speed profile under each decision and conduct separate local speed curve search around them as in [23]. For instance, as shown in Step III in Fig. 1, when the ego vehicle and another vehicle are driving simultaneously towards an intersection from crossing directions, we will generate suggested speed profile (thick curves in Fig. 1) and local speed samples under both the "yield" (red) and the "pass" (blue) decisions. Third-order polynomials are employed for speed curve samples. Hence, via Step III, a spatio-temporal trajectory set $\{\mathcal{T}_{III}\}$ is generated.

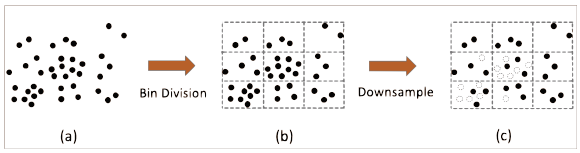
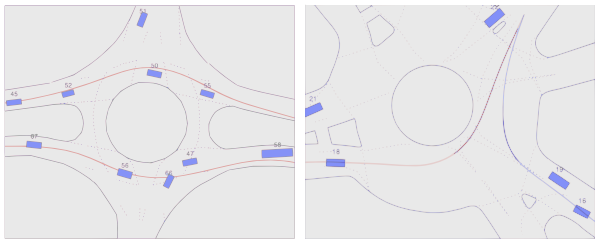


Fig. 2. Re-distribution of samples.



(a) The NID scenario with DR_USA_Roundabout_SR (b) The ID scenario with DR_USA_Roundabout_FT

Fig. 3. Two roundabout scenarios in INTERACTION dataset.

C. Re-Distribution of Samples

Note that in (4), the probability of a trajectory ξ is evaluated via the normalization term Z which is estimated via the samples in $\{\mathcal{T}_{III}\}$. However, the samples in $\{\mathcal{T}_{III}\}$ are not necessarily uniformly distributed in the selected feature space $\mathbf{f}(\xi)$, which will cause biased evaluation of probabilities, as pointed in [24]. To address this problem, we propose to use Euclidean distance in the feature space as a similarity metric for re-distributing the samples. As shown in Fig. 2, the re-distribution process takes two steps: 1) with an initial sample set $\{\mathcal{T}_{III}\}$ ([24](a)), we uniformly divide the feature space into discrete bins, and 2) re-sample within each bin to make sure that each bin has roughly equal number of samples.

D. Summary of the SMIRL Algorithm

The proposed SMIRL algorithm can thus be summarized in Algorithm 1.

Algorithm 1: The Proposed Sampling-based Maximum Entropy IRL for Driving.

Result: optimized reward function parameters $\theta^\star$
Input: The demonstration dataset $\mathcal{D}_M = \{\xi_i\}_{i=1:M}$, the convergence threshold ϵ and the learning rate α.

Initialize θ_0, $k = 0$ and compute expected expert feature count $\bar{\mathbf{f}}(\mathcal{D}_M) = \frac{1}{M}\sum_{i=1}^{M}\mathbf{f}(\xi_i)$;
Generate the sample set $\mathcal{D}_s^0 = \{\tau_m^i\}_{m=1:K,i=1:M}$ using the sampler in Section II-B;
Re-distribute the samples according to their similarities as discussed in Section II-C, and generate a new sample set $\mathcal{D}_s$;
Compute the initial expected feature count over all samples $\widetilde{\mathbf{f}}_0(\mathcal{D}_s) = \frac{1}{M}\sum_{i=1}^{M}\widetilde{\mathbf{f}}_0(\xi_i) = \frac{1}{M}\sum_{i=1}^{M}\frac{1}{K}\sum_{m=1}^{K}\frac{\exp R(\tau_m^i,\theta_0)}{\sum_{m=1}^{M}\exp R(\tau_m^i,\theta_0)}\mathbf{f}(\tau_m^i)$;
while $\|\bar{\mathbf{f}}(\mathcal{D}_M) - \widetilde{\mathbf{f}}_k(\mathcal{D}_s)\|_2 \geq \epsilon$ **do**
Update θ_k using gradient decent, i.e., $\theta_{k+1} = \theta_k + \nabla_{\theta_k}L = \theta_k + \alpha(\bar{\mathbf{f}}(\mathcal{D}_M) - \widetilde{\mathbf{f}}(\mathcal{D}_s))$;
Compute the expected feature count based on θ_{k+1} over all samples $\widetilde{\mathbf{f}}_{k+1}(\mathcal{D}_s) = \frac{1}{M}\sum_{i=1}^{M}\widetilde{\mathbf{f}}_{k+1}(\xi_i) = \frac{1}{M}\sum_{i=1}^{M}\frac{1}{K}\sum_{m=1}^{K}\frac{\exp R(\tau_m^i,\theta_{k+1})}{\sum_{m=1}^{M}\exp R(\tau_m^i,\theta_{k+1})}\mathbf{f}(\tau_m^i)$;
$k = k + 1$;
end
$\theta^* = \theta_k$;

III. Experiments on Driving Behavior

We apply the proposed SMIRL to learn the human driving behavior from real traffic data. Experiments on two types of driving scenarios are conducted: one focusing on independent driving behavior (i.e., non-interactive driving (NID)) and the other on interactive driving (ID) behavior at merging traffic. The NID scenario aims to recover humans' preference when they are driving freely, and the ID case aims to capture how human drivers interact with others in merging traffic.

A. Dataset

We select training data from the INTERACTION dataset [25], [26] with the NID trajectories from the subset DR_USA_Roundabout_SR and ID trajectories from the subset DR_USA_Roundabout_FT, as shown in Fig. 3.

In the NID scenario, we select 113 driving trajectories which travel across the roundabout horizontally, as demonstrated by the red lines in Fig. 3(a). 80 trajectories are used as training data and the remaining 33 as test data. In the ID scenario, we selected 233 pairs of interactive driving trajectories with two vehicles: an ego vehicle trying to merge into the roundabout and an interacting vehicle that is already in the roundabout (interactive trajectories at different locations of the roundabout are contained). An illustrative example is shown in Fig. 3(b) where the blue and red lines represent, respectively, the trajectories of the ego vehicle and the interacting vehicle. The solid segments of the red and blue lines in Fig. 3(b) indicate the time period when the two vehicles are interacting with each other, and we use them for learning. 150 pairs of trajectories are for training and the other 83 for testing. The sampling time of all trajectories is $\Delta t = 0.1\ s$.

B. Feature Selection

Recall that in (3), we assume that the reward function is a linear combination of a set of specified features. The goal of such a feature space is to explicitly capture the properties that humans care when they are driving. We categorize the features into two types: non-interactive features and interactive features.

Non-Interactive Features:

Speed: To describe human's incentives to drive fast and the influence of traffic rules, we define the speed feature as

$$f_v(\xi) = \frac{1}{N}\sum_{i=1}^{N}(v_i - v_{\text{desired}})^2 \tag{9}$$

where v_{desired} is the speed limit.

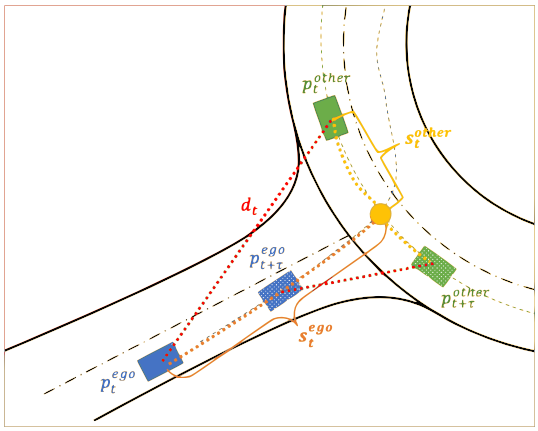


Fig. 4. The illustration of the proposed interactive features. The blue and green rectangles represent the ego and the other vehicle, respectively. The orange circle is the conflict point of the two vehicles on their routes. The spatial distance d at t is demonstrated via red dotted lines. The longitudinal distances to the collision point for both vehicles at t are shown by the yellow dot curves.

Longitudinal and lateral accelerations: Accelerations are related to not only the power consumption of the vehicle, but also to the comfort of the drivers. To learn human's preference on them, we include both of them as features:

$$f_{a\_lon}(\xi) = \frac{1}{N}\sum_{i=1}^{N} a_{lon}^2 \tag{10}$$

$$f_{a\_lat}(\xi) = \frac{1}{N}\sum_{i=1}^{N} a_{lat}^2 \tag{11}$$

Longitudinal jerk: We also have included longitudinal jerk as a feature since it can describe the comfort level of human's driving behavior. It is defined as:

$$f_{jerk}(\xi) = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{a_t - a_{t-1}}{\Delta t}\right)^2 \tag{12}$$

Interactive features: To capture the mutual influence between interactive drivers, we define two interactive features.

Future distance: The future distance d is defined as the minimum spatial distance of two interactive vehicles within a predicted horizon τ_{predict} assuming that they are maintaining their current speeds (in this paper, we use $\tau_{\text{predict}} = 1\ s$). As demonstrated in Fig. 4, between time t and $t+\tau$, the blue vehicle drives from p_t^{ego} to $p_{t+\tau}^{\text{ego}}$ and the green one drives from p_t^{other} to $p_{t+\tau}^{\text{other}}$. At time t, their spatial distance is demonstrated via d_t. Hence, The feature of future distance d is given by

$$f_{dist}(\xi) = \frac{1}{N}\sum_{i=1}^{N} e^{-\min_{\tau\in[0,\tau_{\text{predict}}]} d(t_i+\tau)}. \tag{13}$$

Future interaction distance: Different from $f_{dist}(\xi)$, the feature related to future interaction distance is defined as the minimum distance between their distances to the collision point, i.e.,

$$f_{\text{int\_dist}}(\xi) = \frac{1}{N}\sum_{i=1}^{N} e^{-\min_{\tau\in[0,\tau_{\text{predict}}]} |s^{\text{ego}}(t_i+\tau) - s^{\text{other}}(t_i+\tau)|} \tag{14}$$

where $s^{\text{ego}}(t_i+\tau)$ and $s^{\text{other}}(t_i+\tau)$ represent, respectively, the longitudinal distances of the blue and green vehicles to the shared collision point (the orange circle in Fig. 4) at time $t_i+\tau$. Again, we assume the vehicles maintain their speeds at t_i through the horizon τ_{predict}.

Note that the above features all have different physical meanings and units. Hence, to assure fair comparison of their contributions to the reward function, we normalize all of them to be within (0,1) before the learning process by dividing them by their own maximum values on the dataset.

C. Baseline Methods

To validate the effectiveness and efficiency of our proposed method, we compare our method with three other representative IRL algorithms: the continuous-domain IRL (CIOC) in [18], the optimization-approximated IRL (Opt-IRL) in [20], and the guided cost learning (GCL) algorithm in [12]. CIOC, Opt-IRL and our method are model-based, while GCL is model-free (i.e., deep learning based). All of them are based on the principle of maximum entropy, but differ in the estimation of Z.

- CIOC estimates Z in a continuous domain via Laplace approximation. Specifically, the reward at an arbitrary trajectory $\tilde{\xi}$ can be approximated by its second-order Taylor expansion at a demonstration trajectory $\hat{\xi}_D$, i.e.,

$$\begin{aligned} R(\theta,\tilde{\xi}) \approx R(\theta,\hat{\xi}_D) + (\tilde{\xi}-\hat{\xi}_D)^T\frac{\partial R}{\partial \xi_D} \\ +(\tilde{\xi}-\hat{\xi}_D)^T\frac{\partial^2 R}{\partial \xi_D^2}(\tilde{\xi}-\hat{\xi}_D). \end{aligned} \tag{15}$$

  This simplifies $Z = \int_{\tilde{\xi}} e^{\beta R(\theta,\tilde{\xi})} d\tilde{\xi}$ as a Gaussian integral. For more details, one can refer to [18].
- Opt-IRL estimates $Z = \int_{\tilde{\xi}} e^{\beta R(\theta,\tilde{\xi})} d\tilde{\xi}$ via the optimal trajectory ξ_{opt}. Namely at each training iteration, with the updated θ, an optimal trajectory ξ_{opt} can be obtained by minimizing the updated reward function, and $Z \approx e^{\beta R(\theta,\xi_{opt})}$ is utilized as an approximation.
- Different from model-based IRL, GCL parameterizes the reward function as well as the policy via two deep neural networks which are trained together. It uses samples of the policy network to estimate Z in each iteration. Note that the key difference between GCL and the model-based IRL is that GCL does not need manually crafted features, but automatically learns features via the neural networks.

D. Evaluation Metrics

We employ three metrics to evaluate the performance of different IRL algorithms: 1) feature deviation from the ground truth as in [9], 2) mean Euclidean distance to the ground truth as in [13] and [14], and 3) the likelihood of the ground truth. Definitions of the three metrics are given below.

1) Feature Deviation: Given a learned reward function, we can correspondingly generate a best predicted future trajectory for every sample in the test set. The feature deviation (FD) between the predicted trajectories and the ground-truth trajectories

TABLE I
A SUMMARY OF THE IRL ALGORITHMS IN THE NON-INTERACTIVE DRIVING SCENARIO

	a_lon	j_lon	v_des	a_lat	MED	Win Count	Log Likelihood
Ours	0.16±0.12	0.20±0.15	0.09±0.04	0.09± 0.03	0.21± 0.06	33	-238.98
Opt-IRL	0.19± 0.19	0.32± 0.19	0.13± 0.06	0.11± 0.03	0.29± 0.09	0	-398.93
CIOC	0.48± 0.42	0.23± 0.17	0.10± 0.07	0.06± 0.05	0.23± 0.09	0	-662.16
GCL	—	—	—	—	3.73± 1.95	0	-1377.65

is defined as follows:

$$\mathcal{E}_{FD} = \frac{1}{M}\sum_{i=1}^{M}\frac{1}{N_i}\frac{|\mathbf{f}(\xi_i^{gt}) - \mathbf{f}(\xi_i^{pred})|}{\mathbf{f}(\xi_i^{gt})}. \tag{16}$$

where M is the number of trajectories in the test set, and N_i is the length of the i-th trajectory.

2) Mean Euclidean Distance (MED): The mean Euclidean distance is defined as:

$$\mathcal{E}_{MED} = \frac{1}{M}\sum_{i=1}^{M}\frac{1}{N_i}||\xi_i^{gt} - \xi_i^{pred}||_2. \tag{17}$$

3) Probabilistic Metrics: We also evaluate the likelihood of the ground-truth trajectories given the learned reward functions through different IRL algorithms. Recalling (4), the likelihood of a ground-truth demonstration ξ in the test set is given by

$$P(\xi|\theta, \{\mathcal{T}\}) = \frac{\exp\left(R(\xi,\theta)\right)}{\exp\left(R(\xi,\theta)\right) + \sum_{i=1}^{M}\exp\left(R(\tau_i,\theta)\right)}, \tag{18}$$

where $\{\mathcal{T}\}$ is the set of samples generated via our proposed approach. In our proposed method, CIOC and Opt-IRL, $R(\xi,\theta) = \theta^T\mathbf{f}(\xi)$, and in GCL, it is given via a neural network. Among the four IRL algorithms, the one that generates the highest likelihood on the ground-truth trajectories wins.

E. Experiment Conditions

Among the four different IRL algorithms (i.e., ours vs the three baselines), the CIOC, Opt-IRL and ours are model-based and need manually selected features. GCL, on the other hand, learns to extract features. Hence, for first three approaches, we used the four mentioned features in the non-interactive case (*speed* (v_des), *longitudinal acceleration* (a_lon), *lateral acceleration* (a_lat) and *longitudinal jerk* (j_lon)) and six features in the interactive case (*speed* (v_des), *longitudinal acceleration* (a_lon), *lateral acceleration* (a_lat), *longitudinal jerk* (j_lon), *future distance* (fut_dis), and *future interaction distance* (fut_int_dis)). For the GCL algorithm, no manual features were utilized, and the inputs were directly trajectories.

Other hyper-parameters of the proposed algorithm is set as follows: $F_{\text{threshold}} = 1.0$ with weights on different forces as $w_{\text{contract}} = w_{\text{repel}} = w_{\text{attract}} = 1/3$, i.e., no preference was introduced during the path sampling process over the smoothness of the paths, deviations from the reference lane, and the distance to obstacles.

IV. RESULTS AND DISCUSSION

The performances of the four algorithms, i.e., ours, CIOC, Opt-IRL and GCL, are evaluated based on the metrics in Section III-D. We conducted two types of tests: one on test sets in the same environment (seen) as the training sets, and one on completely new test sets in a new merging environment (unseen).

A. Performance on Test Sets in Seen Environments

The results of the four IRL algorithms under the non-interactive and interactive driving scenarios are listed in Table I and Table II, respectively.

1) Feature Deviation: Feature deviation was evaluated among the three model-based approaches, i.e., ours, CIOC and Opt-IRL. We can see that compared to the other two algorithms, the proposed SMIRL algorithm achieved relatively smaller $\mathcal{E}_{FD}$ on most of the features in non-interactive scenario, except for the feature a_lat. In the interactive scenario, the proposed algorithm achieved smaller $\mathcal{E}_{FD}$ on all features. Moreover, the variations of $\mathcal{E}_{FD}$ are consistently smaller across different features and scenarios. This means that the proposed algorithm learned a reward function that can better capture the general driving preference in the dataset, and thus achieve more stable performance. The learned weights via our proposed approach are given in Table III. We can see that the contribution of the feature a_lat is relatively small in both scenarios, particularly in the non-interactive case. Therefore, the proposed algorithm generated a relatively large feature deviation on a_lat in the non-interactive case.

2) MED: MED was evaluated among all four IRL algorithms. In Table I and Table II, we can see that our method achieved smaller MEDs and variances compared to the CIOC, Opt-IRL and GCL in both driving scenarios. This indicates that the our algorithm can find a reward function that better explains human driving behaviors, i.e., generates more human-like trajectories. An interesting finding is that GCL, as a deep learning based method, did not necessarily achieve better performance than the model-based IRL algorithms. There might be multiple reasons leading to such an outcome. First, the training sets we utilized were too small for deep learning based models to converge well. Second, the data from real traffic contained noises which might deteriorate the performance of GCL, particularly when the amount of data was not sufficient. Such outcomes also further verified the data efficiency and better robustness of model-based IRL algorithms in the presence of data noises.

3) Probabilistic Metric: The results of the four methods in terms of the probabilistic metric are also shown in Table I and Table II. We can see that in both scenarios, the ground-truth trajectories in the test sets all have higher likelihood using the reward function retrieved by our approach compared to those by the other three algorithms, which demonstrates the effectiveness of our proposed approach.

The learned weights in Table III indicate that humans care more about longitudinal accelerations in both non-interactive and interactive scenarios. During interaction, human drivers

TABLE II
A SUMMARY OF THE IRL ALGORITHMS IN THE INTERACTIVE DRIVING SCENARIO

	a_lon	j_lon	a_lat	v_des	fut_dis	fut_int_dis	MED	Win Count	Log Likelihood
Ours	0.15± 0.24	0.54± 0.19	0.19± 0.24	0.034± 0.026	0.012± 0.0078	0.032± 0.045	0.066± 0.038	63	-515.97
Opt-IRL	0.69± 1.04	0.55± 0.40	0.20± 0.23	0.083± 0.11	0.021± 0.018	0.043± 0.066	0.14± 0.16	4	-802.01
CIOC	0.42± 0.77	0.69± 0.26	0.26± 0.23	0.064 ± 0.10	0.023± 0.012	0.045± 0.10	0.14± 0.14	9	-595.27
GCL	—	—	—	—	—	—	1.53± 1.16	0	-1196.75

TABLE III
THE LEARNED WEIGHTS OF REWARD FUNCTION USING OUR PROPOSED APPROACH (NID REFERS TO NON-INTERACTIVE DRIVING SCENARIO AND ID REFERS TO INTERACTIVE DRIVING SCENARIO)

	a_lon	j_lon	a_lat	v_des	fut_dis	fut_int_dis
NID	1	0.363	0.001	0.14	-	-
ID	1	0.007	0.002	0.102	0.007	0.022

TABLE IV
GENERALIZATION RESULTS OF DIFFERENT IRL ALGORITHMS UNDER THE MED METRIC. THE RESULTS ARE IN METERS

	Seen NID	Unseen NID	Seen ID	Unseen ID
Ours	0.21	0.74	0.066	0.072
Opt-IRL	0.29	0.89	0.14	0.17
CIOC	0.23	0.90	0.14	0.16
GCL	3.73	46.70	1.53	4.69

TABLE V
GENERALIZATION RESULTS OF DIFFERENT IRL ALGORITHMS UNDER THE PROBABILISTIC METRIC

	Seen NID	Unseen NID	Seen ID	Unseen ID
Ours	-238.98	-399.85	-515.97	-571.60
Opt-IRL	-398.93	-472.51	-802.01	-870.72
CIOC	-662.16	-1153.74	-595.27	-621.13
GCL	-1377.65	-3140.24	-1196.75	-2898.64

pay more attention to future interaction distance, and less on longitudinal jerks and lateral accelerations.

B. Performance on Test Sets in Unseen Environments

To validate the robustness and generalization ability of our proposed method, we also conducted a comparison among the four IRL algorithms on new test sets in an unseen environment in the training sets. More specifically, we trained all four algorithms using the roundabout merging in the training data and directly tested the trained models on other unseen merging scenarios with different road structures. Both interactive and non-interactive scenarios were tested. The results were given in Table IV and Table V. It is shown that all model-based IRL algorithms, including ours, can generalize better compared to GCL. Both the MED and likelihood did not degrade as significantly as the GCL algorithm.

C. Computation Complexity

We also compared the convergence speed of the four IRL algorithms. The time cost is summarized in Table VI. We can see that the convergence speed of our method is significantly faster than that of CIOC, Opt-IRL and GCL. Such a superior convergence speed benefits from two reasons. First, the sampling method in our proposed approach is efficient (around 1 minute to generate all samples for the entire training set). Second, the sampling process is *one-shot* in the algorithm through the training process. On the contrary, in each iteration, Opt-IRL needs to solve an optimization problem through gradient descent to find the optimal trajectory given the updated reward function. Such procedure can be extremely time-consuming, particularly when the planning time horizon is long. Thus, Opt-IRL might suffer from the scaling problem with long planning horizon and large training set. GCL also adopts a sampling-based method. However, it needs to re-generate all the samples in every training iteration, while our method only needs to generate all samples once. As for CIOC, the main computation load comes from the computation of gradient and hessian introduced via the Laplace approximation as shown in (15). Besides, we also find the stability of the training performance for CIOC is quite sensitive to data noise and the selection of feature sets. Numerical computation issues, particularly for the hessian calculation, could happen and influence the learning performance in the presence of large data noise and/or miss-specified feature sets. As a comparison, our proposed method is much less sensitive to either noise and feature selection, which is also a significant advantage.

TABLE VI
THE TIME COST OF THE THREE ALGORITHMS FOR BOTH NON-INTERACTIVE AND INTERACTIVE SCENARIOS. RESULTS ARE IN MINUTES

	Ours	CIOC	Opt-IRL	GCL
Non-interactive	6	60	1800	40
Interactive	5	90	1260	30

D. The Effect of Sample Re-Distribution

We investigated the effect of sample re-distribution step (in Section II-C) for the learning performance. The experiment results with and without the re-distribution step for the non-interactive and interactive scenarios are shown in Tables VII and VIII, respectively. We can see that with the procedure of sample re-distribution, the proposed SMIRL algorithm can learn a better reward function in terms of the three categories of metrics: feature count deviations, MED and probabilistic metric of the ground-truth demonstrations in the test set. Most significant improvements are the results on "Win Count": the re-distribution of samples can help better evaluate the probabilities of ground-truth demonstrations. Such results are consistent with the conclusion from [24], namely a uniformed distribution of samples via appropriate similarity function can help generate more accurate probabilistic predictions using the Boltzmann noisily-rational model in (2).

TABLE VII
EXPERIMENT RESULTS OF THE NON-INTERACTIVE SCENARIO WITH AND WITHOUT THE STEP OF SAMPLE RE-DISTRIBUTION

	a_lon	j_lon	v_des	a_lat	MED	Win Count	Log Likelihood
w/ sample re-distribution	0.16± 0.12	0.20± 0.15	0.09± 0.04	0.09± 0.03	0.21± 0.06	33	-238.982
w/o sample re-distribution	0.18± 0.10	0.30± 0.16	0.12± 0.05	0.11± 0.04	0.26± 0.08	0	-259.064

TABLE VIII
EXPERIMENT RESULTS OF THE INTERACTIVE SCENARIO WITH AND WITHOUT THE STEP OF SAMPLE RE-DISTRIBUTION

	a_lon	j_lon	a_lat	v_des	fut_dis	fut_int_dis	MED	Win Count	Log Likelihood
w/ sample re-distribution	0.14 ± 0.24	0.53 ± 0.18	0.19 ± 0.23	0.032 ± 0.026	0.012 ± 0.0074	0.027 ± 0.044	0.072 ± 0.043	76	-515.965
w/o sample re-distribution	0.23 ± 0.53	0.55 ± 0.18	0.19 ± 0.23	0.031 ± 0.028	0.012 ± 0.0062	0.027 ± 0.045	0.067 ± 0.041	0	-557.307

V. CONCLUSION

In this paper, we proposed a sampling-based maximum entropy inverse reinforcement learning (IRL) algorithm in continuous domain to efficiently learn human driving behaviors. By explicitly leveraging prior knowledge on vehicle kinematics and motion planning, an efficient sampler was designed to estimate the intractable partition term when retrieving the reward function. Such benefits were verified by experiments on both non-interactive and interactive driving scenarios using the INTERACTION dataset. Comparing to the other three popular IRL algorithms, the proposed algorithm achieved better results in terms of both deterministic metrics such as feature count deviation and mean Euclidean distance, and probabilistic metrics such as the likelihood of demonstrations in the test set. Moreover, the proposed IRL algorithm shows better generalization ability and converges significantly faster than the baseline methods.

In this work, the effectiveness of the proposed approach for learning driving cost functions was verified in an offline manner, i.e., the ground-truth trajectories and the optimal trajectories from the learned cost functions were compared. In the future, we would like to extend to online experimental verifications where real human drivers can interact with robot vehicles driven by the learned cost functions in simulators. Also, the proposed IRL algorithm in this work was designed specifically for mobile robot systems such as ground vehicles. Extension to general robotic systems with higher dimensions such as robot manipulators will also be considered. Furthermore, the Euclidean distance metric used in the re-sampling step is not necessarily the best metric, and we will explore better metrics in future works.

REFERENCES

[1] S. Ross, J. Pineau, S. Paquet, and B. Chaib-Draa, "Online planning algorithms for pomdps," *J. Artif. Intell. Res.*, vol. 32, pp. 663–704, 2008.

[2] T. Gu, J. Atwood, C. Dong, J. M. Dolan, and J.-W. Lee, "Tunable and stable real-time trajectory planning for urban autonomous driving," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 250–256.

[3] Z. Li, W. Zhan, L. Sun, C.-Y. Chan, and M. Tomizuka, "Adaptive sampling-based motion planning with a non- conservatively defensive strategy for autonomous driving," in *Proc. IEEE 21st IFAC World Congr.*, 2020, to be published.

[4] W. Li and E. Todorov, "Iterative linear quadratic regulator design for nonlinear biological movement systems," in *Proc. 1st Int. Conf. Informat. Control, Autom. Robot.*, 2004, pp. 222–229.

[5] J. Chen, W. Zhan, and M. Tomizuka, "Constrained iterative lqr for on-road autonomous driving motion planning," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst.*, 2017, pp. 1–7.

[6] D. Sadigh, N. Landolfi, S. S. Sastry, S. A. Seshia, and A. D. Dragan, "Planning for cars that coordinate with people: Leveraging effects on human actions for planning and active information gathering over human internal state," *Auton. Robots*, vol. 42, no. 7, pp. 1405–1426, 2018.

[7] R. E. Kalman, "When is a linear control system optimal?" *J. Basic Eng.*, Mar. 1964, vol. 86, no. 1, pp. 51–60.

[8] A. Y. Ng *et al.*, "Algorithms for inverse reinforcement learning," in *Proc. 17th Int. Conf. Mach. Learn.*, 2000, vol. 1, pp. 663–670.

[9] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proc. 21st Int. Conf. Mach. Learn.*, 2004, [Online]. Available: https://doi.org/10.1145/1015330.1015430

[10] K. Li, M. Rath, and J. W. Burdick, "Inverse reinforcement learning via function approximation for clinical motion analysis," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 610–617.

[11] C. You, J. Lu, D. Filev, and P. Tsiotras, "Advanced planning for autonomous vehicles using reinforcement learning and deep inverse reinforcement learning," *Robot. Auton. Syst.*, vol. 114, pp. 1–18, 2019.

[12] C. Finn, S. Levine, and P. Abbeel, "Guided cost learning: Deep inverse optimal control via policy optimization," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 49–58.

[13] L. Sun, W. Zhan, M. Tomizuka, and A. D. Dragan, "Courteous autonomous cars," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 663–670.

[14] L. Sun, W. Zhan, and M. Tomizuka, "Probabilistic prediction of interactive driving behavior via hierarchical inverse reinforcement learning," in *Proc. 21st Int. Conf. Intell. Transp. Syst.*, Nov. 2018, pp. 2111–2117.

[15] M. Naumann, L. Sun, W. Zhan, and M. Tomizuka, "On the suitability of cost functions for explaining and imitating human driving behavior," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 5481–5487.

[16] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning," in *Proc. 23rd AAAI Conf. Artif. Intell.*, Chicago, IL, USA, 2008, vol. 8, pp. 1433–1438.

[17] D. Ramachandran and E. Amir, "Bayesian inverse reinforcement learning," in *Proc. 20th Int. Joint Conf. Artif. Intell.*, 2007, vol. 7, pp. 2586–2591.

[18] S. Levine and V. Koltun, "Continuous inverse optimal control with locally optimal examples," in *Proc. 29th Int. Conf. Int. Conf. Mach. Learn.*, 2012, pp. 475–482.

[19] M. Kalakrishnan, P. Pastor, L. Righetti, and S. Schaal, "Learning objective functions for manipulation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013, pp. 1331–1336.

[20] M. Kuderer, S. Gulati, and W. Burgard, "Learning driving styles for autonomous vehicles from demonstration," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 2641–2646.

[21] O. Morgenstern and J. Von Neumann, *Theory of Games and Economic Behavior*. Princeton, NJ, USA: Princeton Univ. Press, 1953.

[22] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 846–894, Jun. 2011.

[23] W. Zhan, C. Liu, C.-Y. Chan, and M. Tomizuka, "A non-conservatively defensive strategy for urban autonomous driving," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst.*, 2016, pp. 459–464.

[24] A. Bobu, D. R. Scobee, J. F. Fisac, S. S. Sastry, and A. D. Dragan, "Less is more: Rethinking probabilistic models of human behavior," in *Proc. 2020 ACM/IEEE Int. Conf. Human-Robot Interact.*, Mar. 2020, pp. 429–437.

[25] W. Zhan *et al.*, "Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps," 2019, *arXiv:1910.03088*.

[26] W. Zhan, L. Sun, D. Wang, Y. Jin, and M. Tomizuka, "Constructing a highly interactive vehicle motion dataset," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 6415–6420.